20th Annual Conference of Doctoral Students

WDS'11

"WEEK OF DOCTORAL STUDENTS 2011"

CHARLES UNIVERSITY

FACULTY OF MATHEMATICS AND PHYSICS

PRAGUE, CZECH REPUBLIC

May 31, 2011
to
June 3, 2011

Part I

Mathematics and Computer Sciences

J. Šafránková and J. Pavlů (editors)

ISBN 978-80-7378-184-2

Contents

Preface 5

Part I — Mathematics and Computer Sciences

J. Bártek, On the Uniqueness of Solution to Homogeneous SPDEs, **m-3** 7

D. Stibůrek, Testing the Parametric Form of the Volatility in Continuous Time Diffusion Models, **m-4** 13

V. Sečkárová, Tools for Decision Making under Uncertainty, **m-4** 19

I. Kasanický and K. Eben, Ensemble Kalman Filter, **m-4** 25

J. Dvořák, On Moment Estimation Methods for Spatial Cox Processes, **m-4** 31

O. Šedivý, Dependencies in Stochastic Geometry—A Simulation Study, **m-4** 37

P. Kříž, How to Construct Borel Measurable PLIFs?, **m-4** 43

K. Starinská, Change Detection in Autoregressive Time Series with Martingale Difference White Noise, **m-5** 49

M. Hadrava, Interaction of Incompressible Flow with an Elastic Wall, **m-6** 55

A. Kosík, Numerical Simulation of Interaction of Fluid Flow and an Elastic Body, **m-6** 61

M. Novelinková, Comparison of Clenshaw-Curtis and Gauss Quadrature, **m-6** 67

J. Doležal, The Story of a Right Wavelet Conoid, **m-8** 72

D. Lessner, Graph Theory at Czech Grammar Schools, **m-8** 78

M. Štěpánová, The Change of Paradigm in the Matrix Theory, **m-8** 85

L. Vízek, Josef Úlehla and His Calculus Textbook, **m-8** 91

T. Balyo, Decomposing Boolean Formulas into Connected Components, **i-1** 95

Š. Gurský, Minimization of Matched Formulas, **i-1** 101

M. Babka, A Worst Case Aware Version of Universal Hashing, **i-1** 106

M. Kukačka, Neocognitron: A Survey of a Classical Hybrid Neural Network Model, **i-1** 112

T. Plch, Towards Believable Intelligent Virtual Agents with StateFull Hierarchical Reactive Planning, **i-1** 119

P. Černo, Learning Automata and Grammars, **i-1** 125

J. Kozák, Rules in Database Systems, **i-2** 131

N. Green, Dependency Parsing, **i-3** 137

L. Ramasamy, TamilTB: An Effort Towards Building a Dependency Treebank for Tamil, **i-3** 143

G. L. Nguy and M. Ševčíková, Unstated Subject Identification in Czech, **i-3** 149

M. Novák, Utilization of Anaphora in Machine Translation, **i-3** 155

B. Jawaid, Machine Translation with Significant Word Reordering and Rich Target-Side Morphology, **i-3** 161

K. Veselovská, Sentence-Level Polarity Detection in a Computer Corpus, **i-3** 167

A. Vernerová, Nominal Valency in Lexicons, **i-3** 171

PREFACE

The three already traditional volumes of the WDS Proceedings you are holding in the hands are composed of the contributions which have been presented during the **20th Annual Conference of Doctoral Students** that was held in Prague, at Charles University, Faculty of Mathematics and Physics from May 31 to June 3, 2011. In this year, 130 student manuscripts were submitted to publishing and 118 were accepted after the review process.

Proceedings are divided into three volumes. **Part I** presents the contributions on Mathematics (15) and Computer Sciences (14, together 29 papers published in this part). The contributions from the field of Physics of Plasmas and Ionized Media are the main topics of **Part II** (49 papers published in this part). **Part III** consists of contributions from all other symposia on Physics (36 papers published in this part). In this year, 4 contributions were selected as the best representatives of Physics and submitted into the Acta Universitatis Carolinae, Mathematica et Physica (abbr. AUC) journal. Those are the papers of the following students: P. Kaspar (f-1), M. Hejduk (f-2), K. Kalousova (f-7), and B. Bittova (f-13).

All manuscripts were reviewed by two referees and, maybe, you would ask *why*. We believe that it helps students to learn how to prepare the written presentation of their results in a foreign language and how to respond the referee's notices. They should take into account their comments and prepare a new submission including letters to both referees where they explain their reaction. This is a standard way of publication in scientific journals and we are offering an exercise that would help them in future. Since one of referees is a doctoral student, he/she can learn the preparation of a review. Moreover, the subject of the paper under review is often rather different from his/her field of research and thus the referees should learn new problems and another style of writing. They can see that a good and comprehensive review is sometimes a very difficult task. Even now, one can see a different approach of student referees—some of them are very active and bring a lot of comments and suggestions, whereas others carry out a formal review only. The second referee was the experienced senior researcher and, in this year, 18% reviewers from this group was from abroad. We would like to express many thanks to them for their understanding and great help. Their effort significantly contributed to traditionally a good quality of the Proceedings and thus, you can find a lot of references to previous volumes in scientific journals.

Editors thank to following referees:
Mathematics and Computer Sciences — J. Antoch, T. Balyo, R. Bartak, E. Bejcek, V. Benes, O. Bojar, J. Bulin, K. Dedecius, N. Dmitrenko (Russia), P. Dostal, M. Dvorak, S. Gergelitsova, P. Gregor, Z. Hlavka, O. Honzl, M. Hyksova, L. Chrpa, A. Karger, A. Karlova, I. Kasanicky, A. Kazda, V. Kettnerova, P. Knobloch, A. Komarek, M. Kopecky, A. Koubkova, K. Kozel, P. Kucera, P. Lachout, J. Lamac, D. Marecek, J. Misutka, S. Nagy, K. Najzar, R. Neruda, M. Novak, Z. Pawlas, M. Pilat, M. Popel, Z. Praskova, J. Prokopova, D. Prusa, Z. Reitermanov, I. Sebestova, M. Sevcikova, J. Snuparkova, J. Stanek, J. Stepanek, M. Stepanova, J. Strakova, P. Stranak, M. Suda, P. Surynek, P. Svacek, O. Sykora, O. Tichy, Z. Uresova, P. Valesova, J. Vesely, L. Vizek, V. Vlcek, Z. Zabokrtsky, M. Zemlicka, S. Zikanova, M. Zikmundova.
Physics of Plasmas and Ionized Media — M. Aftanas, G. Bano (Slovakia), V. Barinova (Russia), M. Beranek, P. Bilkova, J. Blecki (Poland), T. Burian, K. N. Crabtree (USA), M. Danko (Slovakia), A. G. Demekhov (Russia), P. Dohnal, K. Dryahina, I. Duran, J. Enzl, M. Farnik, C. Foissac (France), M. Fuciman, J. Glosik, O. Goncharov, G. Granko, P. Hacek, J. Havlicek, M. Hejduk, G. Horvath, V. Hrachova, Z. Hrbackova, M. Hron, V. Hruby, W.-C. Hsieh (Taiwan), J. Chalupsky, M. Chichina, J. Chum, T. Ibehej, K. Jelinek, P. Jusko, A. Kanka, J. Kluson, J. Kocisek, A. Kolpakova, M. Komm, I. Korolov, T. Kotrik, D. Kouba, A. Koval (USA), F. Krcma, H. Kreckel (USA), V. Krupar, P. Kudrna, P. Kuzel, M. Laca, Y. Q. Liu (UK), D. Loffhagen (Germany), E. Macusova, M. Masat, S. Matejcik, M. Matejka, Z. Mosna, D. Mulin, D. Naydenkova, F. Nemec, J. M. Oliveros (USA), S. Opanasiuk, R. Panek, R. Paprok, M. Parrot (France), J. Pavlu, I. Pickova, P. Pira, D. Pisa, R. Plasil, V. Poterya, L. Prech, S. G. Pylypenko (Ukraine), A. Pysanenko, E. Romashets (Russia), S. Roucka, P. Rubovic, J. Safrankova, A. Samsonov (Austria), O. Santolik, J. Seidl, T. Sindelarova, E. Spetlikova, Z. Sternovsky (USA), V. Stranak, M. Svec, O. Tkachenko, E. Tognoni (Italy), S. Trippel (Germany), J. Urbar, D. Vacarcel (Portugal), J. Varju, J. Vaverka, A. Velyhan, S. Vidojevic (France), M. Vysinka, T. J. Wasowicz (Poland), G. W. Wei (USA),

R. Wester (Austria), J. Wild, J. Zabka, A. Zahoranova (Slovakia), J. Zajac, G. Zastenker (Russia), X. Zhang (China), I. Zymak.

Physics — L. Benda, L. Benes, Y. Birol (Turkey), J. Borovicka, M. Broz, A. Carrassi (Belgium), J. Cechal, J. Cermak, P. Cermak, P. Cizek, S. Danis, J. Durech, J. Endres, D. Gaskova, M. Grmela (Canada), N. Guerlebeck, J. Haas, J. Hanus, S. Haviar, D. Heyrovsky, P. Hlidek, V. Holubec, V. Holy, G. Choblet (France), Z. Janour, P. Javorsky, M. Jerab, P. Kaplicky, V. Karas, M. Karlik, J. Karlovska, M. Kekule, O. Kopacek, V. Kopp (Germany), P. Koten, V. Koudelkova, R. Kral, N. Kucerka, O. Kylian, I. Lagzi (Hungary), J. Langer, S. Lizunova (Ukraine), P. Lukac, J. Malek, P. Matejka, J. Miksovky, P. Minarik, G. S. Mityurich (Belarus), K. Netocny, U. Netzelmann (Germany), J. Olsina, L. Ondic, A. S. Ovchinnikov (Russia), A. V. Ovsjannikov (Ukraine), M. Pisarcik, P. Pisoft, P. Pokorny, A. D. Popolo (Italy), M. Pracharova, P. Pravec, J. Preclikova, J. Prchal, J. Prokleska, V. Prusa, P. Pustejovska, Z. Sabatka, O. Semerak, M. Setvak, M. Setvin, L. Schmiedt, V. Sima, T. Skala (Italy), A. Smirnov (Russia), O. Soucek, J. Stastka, R. Stork, J. Strasky, V. Synyuk (Ukraine), I. Telezhynsky (Germany), J. Tichy, K. Tuma, V. Tyrpekl, M. Vaclavu, V. Vales, M. Veis, V. Vojacek, I. Vovk (Switzerland), V. Zak.

Finally, we would like to express our thanks to all speakers and other participants for helping us to keep a fruitful and friendly atmosphere during the whole meeting and, last but not least, to our colleagues who assisted us greatly both prior to the conference and after it.

Pleasant reading

Jana Šafránková and Jiří Pavlů

editors

On the Uniqueness of Solution to Homogeneous SPDEs

J. Bártek

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. We study homogeneous stochastic partial differential equations driven by a fractional Brownian motion with Hurst index $H > 1/2$. We state that there is a one-to-one correspondece between solutions to homogeneous stochastic equation and its deterministic counterpart. In particular, we show that if the deterministic equation has a unique solution, so does the stochastic one.

Introduction

If we deal with stochastic differential equations we are often not able to express the solution to our equation by a closed form formula and thus we have to study its properties indirectly by studying properties of coefficients in the equation. Therefore, it is desirable to find a way to obtain such an explicit formula. In the case of equations with fractional Brownian motion, there are some articles on this topic, for example [*Duncan, Maslowski, Pasik-Duncan*, 2005] and [*Tindel, Tudor, Viens*, 2003], which deal with the linear case and possibly space-dependent fractional Brownian motion. When dealing with so-called homogeneous equations an explicit formula has been derived for stochastic equations driven by Wiener process in [*Lototsky*, 2007].

This work continues the study of homogeneous equations with fractional Brownian motion presented in [*Bártek*, 2010]. It is divided into two parts. In the first part the stochastic calculus is briefly developed and the second part contains the definitions of homogeneous equations and solutions to such equations and the main theorem about relationship between stochastic and deterministic equation.

Definition 1. Let $H \in (0,1)$ be a constant. *A (scalar) fractional Brownian motion (fBm) with Hurst index H* is a continuous centered stochastic Gaussian process $(B^{(H)}(t))_{t\geq 0}$ with covariance function

$$\mathbb{E}\left[B^{(H)}(t)B^{(H)}(s)\right] = \frac{1}{2}(t^{2H} + s^{2H} - |t-s|^{2H}).$$

Fractional Brownian motion is not a semimartingale and its paths do not have bounded variation so it was necessary to develop a suitable integration theory.

Stochastic calculus

From now on we suppose that the Hurst index $H > 1/2$. We utilize theory described in [*Mishura*, 2008] which uses fractional calculus and we define the stochastic integral pathwise. We now briefly introduce the stochastic integral according to the paper [*Maslowski, Nualart*, 2003].

Let $\alpha \in (0,1)$. We say that function $f : [0,T] \to \mathbb{R}$ has a Weyl derivative $D^{\alpha}_{0+}f$,

$$D^{\alpha}_{0+}f(t) = \frac{1}{\Gamma(1-\alpha)}\left(\frac{f(t)}{t^{\alpha}} + \alpha\int_0^t \frac{f(t)-f(\lambda)}{(t-\lambda)^{\alpha+1}}\,\mathrm{d}\lambda\right),$$

if the integral on the right hand side converges for a.a. $t \in (0,T)$, where Γ is the Gamma function. Similarly we define $D^{\alpha}_{T-}f(t)$ as

$$D^{\alpha}_{T-}f(t) = \frac{1}{\Gamma(1-\alpha)}\left(\frac{f(t)}{(T-t)^{\alpha}} + \alpha\int_t^T \frac{f(t)-f(\lambda)}{(\lambda-t)^{\alpha+1}}\,\mathrm{d}\lambda\right).$$

Let $W^{\alpha,1}(0,T)$ be a space of measurable functions $f:[0,T]\to\mathbb{R}$ such that

$$|f|_{\alpha,1}:=\int_0^T\left(\frac{|f(s)|}{s^\alpha}+\int_0^s\frac{|f(s)-f(\lambda)|}{(s-\lambda)^{\alpha+1}}\,\mathrm{d}\lambda\right)\mathrm{d}s<\infty,$$

where $0<\alpha<1/2$ is fixed. Further let g be a continuous function on $[0,T]$ such that $\Lambda_\alpha(g)<\infty$, where

$$\Lambda_\alpha(g):=\frac{1}{\Gamma(1-\alpha)\Gamma(\alpha)}\sup_{0<s<t<T}\left(\frac{|g(t)-g(s)|}{(t-s)^{1-\alpha}}+\int_s^t\frac{|g(\lambda)-g(s)|}{(\lambda-s)^{2-\alpha}}\,\mathrm{d}\lambda\right).$$

Now we can define the generalized Stieltjes integral $\int_0^T f\,\mathrm{d}g$ of f with respect to g as

$$\int_0^T f\,\mathrm{d}g:=\int_0^T D^\alpha_{0+}f(s)D^{1-\alpha}_{T-}g_{T-}(s)\,\mathrm{d}s,\tag{1}$$

where $g_{T-}(t)=g(t)-g(T)$. Under the assumptions stated above the integral $\int_0^T f\,\mathrm{d}g$ does not depend on the choice of α, the integral $\int_0^t f\,\mathrm{d}g$ exists for all $t\in[0,T]$ and it holds that

$$\int_0^t f\,\mathrm{d}g=\int_0^T f1_{(0,t)}\,\mathrm{d}g.$$

We have the following estimate

$$\left|\int_0^t f\,\mathrm{d}g\right|\le\Lambda_\alpha(g)|f|_{\alpha,1},\qquad t\in[0,T].$$

Let $C^\alpha([0,T])$ denote the space of α-Hölder continuous functions defined on $[0,T]$. Then for $\varepsilon>0$ we have $C^{\alpha+\varepsilon}([0,T])\subset W^{\alpha,1}(0,T)$ and $\Lambda_\alpha(g)<\infty$ for every $g\in C^{1-\alpha+\varepsilon}([0,T])$. For more details on the space $W^{\alpha,1}$ and related spaces we refer to [*Mishura*, 2008].

Now we can define the stochastic integral with respect to $B^{(H)}$.

Definition 2. Let $B^{(H)}$ be a fractional Brownian motion with Hurst index H defined on the probability space $(\Omega,\mathcal{F},\mathbb{P})$ and let $H\in(1/2,1)$. Assume that f is a random process on $(\Omega,\mathcal{F},\mathbb{P})$ such that $f\in W^{\alpha,1}(0,T)$ $\mathbb{P}$-a.s., where $\alpha\in(1-H,1/2)$. We understand the integral

$$\int_0^T f(s)\,\mathrm{d}^\circ B^{(H)}(s)$$

pathwise in the sense of formula (1).

The integral is well defined since $\Lambda_\gamma(B^{(H)})<\infty$ $\mathbb{P}$-a.s. for all $\gamma\in(1-H,1/2)$.

Homogeneous equation

Definition 3. We say that a function F is *homogeneous of degree* $m\ge1$, if for all $\lambda>0$

$$F(\lambda x,\lambda y,\lambda z,\ldots)=\lambda^m F(x,y,z,\ldots),\quad x\in\mathbb{R},y\in\mathbb{R}^d,z\in\mathbb{R}^{d\times d},\ldots\tag{2}$$

We say that the equation (3) is *homogeneous of degree* $m\ge1$, if the function F is homogeneous of degree m.

Consider an equation

$$v_t=F(v,Dv,D^2v,\ldots),\ t>0,\ x\in\mathbb{R}^d,\tag{3}$$

with initial condition $v(0,x) = v_0(x)$. The unknown in the equation (3) is the function $v = v(t,x)$, function F is given, $v_t = \partial v/\partial t$ and $D^k v$ is k-th derivative of v with respect to x. By the symbol $F(v, Dv, D^2v, \ldots)$ we mean that F depends on v and on a finite number of derivatives $Dv, D^2v, \ldots, D^m v, m \in \mathbb{N}$.

Further, consider a stochastic version of the equation (3) for unknown random field $u = u(t,x), t > 0,\ x \in \mathbb{R}^d$, of the following form

$$du = F(u, Du, D^2u, \ldots)dt + u(f(t)\,\mathrm{d}^\circ B^{(H)}(t) + g(t)\,\mathrm{d}t), \tag{4}$$

with the same initial condition as in (3), i.e. $u(0,x) = u_0(x) = v_0(x)$, where $f \in C^{\alpha+\varepsilon}([0,T])$ for all $T > 0$, for some $0 < \varepsilon < \min\{1-\alpha,\ \alpha,\ H+\alpha-1\}$, $\alpha \in (1-H, 1)$ and $g \in L^\infty_{loc}(\mathbb{R}_+)$.

By the notion of solution to the equations (3) and (4) we understand the following.

Definition 4. Function $v : \mathbb{R}_+ \times \mathbb{R}^d \to \mathbb{R}$ is *a classical solution* to the equation (3), if the following is satisfied:

1. v is continuous,
2. all partial derivatives of function v with respect to x involved in (3) exist and are continuous in variable x and are γ-Hölder continuous in variable t on $[0,T]$ for all $T > 0$, where $\gamma > 1 - H$.
3. The following equality

$$v(t,x) = v_0(x) + \int_0^t F(v(s,x), Dv(s,x), D^2v(s,x), \ldots)\,\mathrm{d}s$$

holds for all $(t,x) \in \mathbb{R}_+ \times \mathbb{R}^d$.

Definition 5. Let $\tau : \Omega \to \mathbb{R}_+ \cup \{+\infty\}$ be a stopping time. The random field $u = u(t,x)$ is a *classical solution* to the equation (4), if there exists a set $\Omega^* \subset \Omega$, $\mathbb{P}(\Omega^*) = 1$, and on the set $((0,\tau]] = \{(t,x,\omega) : t < \tau(\omega),\ \omega \in \Omega^*,\ x \in \mathbb{R}^d\}$ the following conditions are satisfied:

1. u is continuous in variable x and γ-Hölder continuous in variable t for every $(t,x,\omega) \in ((0,\tau]]$, for some $\gamma > 1 - H$,
2. all partial derivatives of function u with respect to x involved in (4) exist and are continuous in variable x and are γ-Hölder continuous in variable t on $[0,T]$ for all $0 < T < \tau$, for some $\gamma > 1 - H$,
3. the equality

$$u(t,x) = u_0(x) + \int_0^t F(u(s,x), Du(s,x), \ldots)\,\mathrm{d}s + \int_0^t u(s,x)(f(s)\,\mathrm{d}^\circ B^{(H)}(s) + g(s)\,\mathrm{d}s)$$

holds for every $(t,x,\omega) \in ((0,\tau]]$.

In the work [*Bártek*, 2010] it has been shown that the existence of the solution to (3) implies the existence of the solution to (4) and a formula for solutions to (4) was given. Now we will show that this formula can be inverted and the existence of solution to (4) implies the existence of solution to (3) as well. In particular, this means that if the deterministic equation has a unique solution, do does the stochastic one.

Define processes

$$h(t) = \exp\left(\int_0^t g(s)\,\mathrm{d}s + \int_0^t f(s)\,\mathrm{d}^\circ B^{(H)}(s)\right) \quad \text{and} \quad \mathcal{H}_m(t) = \int_0^t h^{m-1}(s)\,\mathrm{d}s \tag{5}$$

and their counterparts

$$\begin{aligned} z(t) &= \exp\Big(-\int_0^t g(s)\,\mathrm{d}s - \int_0^t f(s)\,\mathrm{d}^\circ B^{(H)}(s)\Big) = \frac{1}{h(t)}, \\ \mathcal{R}_m(t) &= \mathcal{H}_m^{-1}(t). \end{aligned} \tag{6}$$

The process h is called a geometric fractional Brownian motion and plays the key role in the correspondence of solutions to the equations (3) and (4). It is obvious that $h(0) = 1$ and by a straightforward application of the Itô formula (see for example [*Mishura*, 2008], Chapter 2) we obtain the following equality for h

$$h(t) = 1 + \int_0^t h(s)f(s)\,\mathrm{d}^\circ B^{(H)}(s) + \int_0^t h(s)g(s)\,\mathrm{d}s.$$

Now we can state the main theorem.

Theorem 1. Assume that function F in the equation (3) is continuous and homogeneous of the degree m and let u be a solution to its stochastic version (4). Then function v defined by the relation

$$v(t,x) = z(\mathcal{R}_m(t))u(\mathcal{R}_m(t),x)$$

is a solution to the equation (3).

Before the proof the theorem we formulate a technical lemma that will be used later in the proof. For $T > 0$ we approximate $B^{(H)}(t)$ by

$$\psi_n(t) := B^{(H)}(t_k^n) + \frac{n}{T}(B^{(H)}(t_{k+1}^n) - B^{(H)}(t_k^n))(t - t_k^n),$$

where $t \in [t_k^n, t_{k+1}^n],\ t_k^n = \frac{k}{n}T$.

Lemma 1. (i) For a.a. $\omega \in \Omega$ and for any $0 < \varkappa < H$ there exists a constant $C = C(\varkappa,\omega)$ such that the functions $\{\psi_n(\cdot,\omega)\}_{n\in\mathbb{N}}$ are $\varkappa$-Hölder continuous with norm less than or equal to C.
(ii) For a.a. $\omega \in \Omega$ there exists a constant $K = K(\varkappa,\omega,T)$ such that

$$|D_{T-}^{1-\alpha}\psi_{n,T-}(s,\omega)| \le K(\varkappa,\omega,T),$$

where $\psi_{n,T-}(s) = \psi_n(T-) - \psi_n(s)$.

Proof. (i) It is known that for a.a. $\omega \in \Omega$ the trajectories of fractional Brownian motion $B^{(H)}$ are Hölder continuous up to order H, i.e.

$$\forall\ 0 < \varkappa < H\ \exists\ K_\omega > 0 \quad \sup_{t\in[0,T]} |B^{(H)}(t)| + \sup_{s,t\in[0,T]} \frac{|B^{(H)}(t) - B^{(H)}(s)|}{|t-s|^\varkappa} < K_\omega.$$

Using the definition of ψ_n we see that

$$\forall\ n \in \mathbb{N} \quad \sup_{t\in[0,T]} |\psi_n(t)| \le \sup_{t\in[0,T]} |B^{(H)}(t)| < K_\omega.$$

Further we have for $s,t \in [t_k^n, t_{k+1}^n],\ 1 \le k \le n$,

$$\begin{aligned} |\psi_n(t) - \psi_n(s)| &= |(B^{(H)}(t_{k+1}^n) - B^{(H)}(t_k^n))n(t-s)T^{-1}| \le |t-s|nK_\omega T^{-1}|t_{k+1}^n - t_k^n|^\varkappa \\ &= |t-s|K_\omega T^{\varkappa-1}n^{1-\varkappa} \le K_\omega|t-s|^\varkappa|t-s|^{1-\varkappa}n^{1-\varkappa}T^{\varkappa-1} \le K_\omega|t-s|^\varkappa \end{aligned}$$

and we obtain for $s \in [t_i^n, t_{i+1}^n],\ t \in [t_j^n, t_{j+1}^n],\ 1 \le i \le j \le n$,

$$\begin{aligned} |\psi_n(t) - \psi_n(s)| &= |\psi_n(t) - \psi_n(t_j^n) + \psi_n(t_j^n) - \psi_n(t_i^n) + \psi_n(t_i^n) - \psi_n(s)| \\ &\le 3K_{\omega,\varkappa}|t-s|^\varkappa. \end{aligned}$$

(ii) For every $n \in \mathbb{N}$ we have the following estimate:

$$\begin{aligned}
|D_{T-}^{1-\alpha}\psi_{n,T-}(s)| &= \left|\frac{1}{\Gamma(\alpha)}\left(\frac{\psi_n(s)-\psi_n(T)}{(T-s)^{1-\alpha}} + (1-\alpha)\int_s^T \frac{\psi_n(s)-\psi_n(r)}{(r-s)^{2-\alpha}}\,\mathrm{d}r\right)\right| \\
&\leq \frac{1}{\Gamma(\alpha)}\left(C_\omega(\varkappa)(T-s)^{\varkappa+\alpha-1} + (1-\alpha)\int_s^T C_\omega(\varkappa)(r-s)^{\varkappa+\alpha-2}\,\mathrm{d}r\right) \\
&= \frac{C_\omega(\varkappa)(T-s)^{\varkappa+\alpha-1}}{\Gamma(\alpha)}\left(1+\frac{1-\alpha}{\varkappa+\alpha-1}\right).
\end{aligned}$$

□

We now prove Theorem 1.

Proof. We assume that functions f and g are constant. The proof of the general case is analogous, but more technical. Let u be a solution to (4). We define u_n as

$$u_n'(t) := F(u(t)) + gu(t) + fu(t)\psi_n'(t), \quad u_n(0) = u_0,$$

$$z_n = \exp\{-gt - f\psi_n(t)\},$$

and

$$v_n(t) = z_n(\mathcal{R}_m(t))u_n(\mathcal{R}_m(t)).$$

The functions v_n are differentiable a.e. and we get

$$v_n(t) = v(0) + \int_0^t A_n(s)\,\mathrm{d}s + B_n^{(1)}(t) + B_n^{(2)}(t),$$

where

$$\begin{aligned}
A_n(s) &= h^{1-m}(\mathcal{R}_m(s))z_n(\mathcal{R}_m(s))F(u(\mathcal{R}_m\gamma(s))), \\
B_n^{(1)}(t) &= \int_0^t h^{1-m}(\mathcal{R}_m(s))z_n(\mathcal{R}_m(s))g[u(\mathcal{R}_m(s)) - u_n(\mathcal{R}_m(s))]\,\mathrm{d}s, \\
B_n^{(2)}(t) &= \int_0^t h^{1-m}(\mathcal{R}_m(s))z_n(\mathcal{R}_m(s))f\psi_n'(s)(\mathcal{R}_m(s))[u(\mathcal{R}_m(s)) - u_n(\mathcal{R}_m(s))]\,\mathrm{d}s.
\end{aligned}$$

Using the standard fractional calculus, Lemma 1 and properties of u, u_n and z_n we get that for a.a. t

$$v_n(t) \to v(t), \quad A_n(s) \to F(v(s)),$$

and

$$B_n^{(1)}(t) \to 0, \quad B_n^{(2)}(t) \to 0,$$

as $n \to \infty$. Therefore, $v(t) = v(0) + \int_0^t F(v(s))\,\mathrm{d}s$ holds and $v(t,x) = z(\mathcal{R}_m(t))u(\mathcal{R}_m(t),x)$ is a classical solution to (3). □

Remark. i) It can be shown by similar methods that the statement of Theorem 1 is also true for weak solutions to equations (3) and (4).

ii) The right definition of process $h(t)$ requires h to be a solution to the stochastic differential equation

$$dh(t) = h(t)g(t)\,\mathrm{d}t + h(t)f(t)\,\mathrm{d}B^{(H)}(t).$$

Therefore, for a different stochastic integral, for example the Skorokhod integral, it has in general a different form than (5). Nevertheless, in case of Skorokhod integral the statement of Theorem 1 seems to be valid only in the Wiener process case $H = 1/2$.

Acknowledgments. This work was supported by the GA UK grant no. 167510 and by the grant SVV 261315/2010 and by the Czech Science Foundation grant no. P201/10/0752.

References

Bártek J.: *Homogeneous Stochastic Differential Equations*, WDS 2010 - Proceedings of Contributed Papers, Part I, pp. 195-200, 2010.

Biagini F., Hu Y., Øksendal B., Zhang T.: *Stochastic Calculus for Fractional Brownian Motion and Applications*, Springer, London, 2008.

Duncan T. E., Maslowski B., Pasik-Duncan, Stochastic equations in Hilbert space with a multiplicative fractional Gaussian noise, *Stochastic Process. Appl.* 115(8) 1357–1383, 2005.

Lototsky, S. V.: *A random change of variables and applications to the stochastic porous medium equation with multiplicative time noise*, Communications on Stochastic Analysis 1(3) 343–355, 2007.

Maslowski B., Nualart D.: *Evolution equations driven by a fractional Brownian motion*, Journal of Functional Analysis 202 277–305, 2003.

Mishura Y. S.: *Stochastic Calculus for Fractional Brownian Motion and Related Processes*, Springer – Verlag, Berlin, 2008.

Tindel S., Tudor C. A., Viens F.: *Stochastic evolution equations with fractional Brownian motion*, Probab. Theory Related Fields 127(2) 186–204, 2003.

Testing the Parametric Form of the Volatility in Continuous Time Diffusion Models

D. Stibůrek

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. This work presents tests for the parametric form of the volatility function in a stochastic differential equation $dX_t = b(t, X_t)dt + \sigma(t, X_t)dW_t$. We can consider the hypothesis of constant volatility or general form of volatility. In the first case we can use two statistics which have approximately Gaussian distribution or supremum of some Gaussian process. One of these methods can be extended in the general case. The hypotheses can be also tested by the parametric bootstrap methods. Proposed tests are numerically illustrated and compared with respect to numerical problems.

Introduction the problem

Consider a random process X_t, which is generated by a stochastic differential equation

$$dX_t = b(t, X_t)dt + \sigma(t, X_t)dW_t, \tag{1}$$

where W_t is a standard Wiener process defined on an appropriate probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{0\le t\le 1}, P)$ with corresponding filtration $\mathcal{F}_t^W = \sigma(W_s, 0 \le s \le t)$. In this work we would like to know some information about the covariance function $\sigma(t, X_t)$. In particular, we will investigate the testing of hypothesis

$$H_0 : \sigma^2(t, X_t) = \sum_{i=1}^{d} \theta_j \sigma_j^2(t, X_t), \tag{2}$$

where $\sigma_1^2, \ldots, \sigma_d^2$ are known functions and $\theta_1, \ldots, \theta_d$ are unknown parameters. Consider the case of discretely observed data on a fixed time span, say $[0, 1]$.

It was pointed out [*Corradi*, 1999] that this model is appropriate, for example, for analyzing the pricing options. These authors also discuss the problem of testing for a parametric form of the volatility function in this model. This problem is also discussed by [*Woerner*, 2003]. If the assumption of the parametric model cannot be justified, nonparametric estimates for the drift and variance of the diffusion can be used (e.g. [*Genon-Catalot*, 1992], [*Jiang*, 1997]), though they are less efficient from an asymptotic point of view.

Assumptions and notation

Let the functions $b : [0, 1]\times\mathbb{R} \to \mathbb{R}$ and $\sigma : [0, 1]\times\mathbb{R} \to \mathbb{R}$ satisfy the standard conditions for existence and uniqueness a solution of (1) (Lipschitz and linearly bounded growth). Suppose that $\sigma, \sigma_1, \ldots, \sigma_d$ are strictly positive and that $\sigma_1, \ldots, \sigma_d$ are linearly independent on every compact set $[0, 1] \times [a, b], a < b$ and that they satisfy the same conditions as σ. We assume additionally that σ is twice continuously differentiable with respect to the second argument (denoted by $\frac{\partial}{\partial y}$) such that for some constant $F > 0$ and all $i, j \in \{1, \ldots d\}$

$$\sup_{s,t\in[0,1]} \mathrm{E}\left[\left(\frac{\partial}{\partial y}\sigma(s, X_t)\right)^4\right] < F, \sup_{s,t\in[0,1]} \mathrm{E}\left[\left(\frac{\partial^2}{\partial y^2}\sigma(s, X_t)\right)^4\right] < F,$$

$$\sup_{s,t\in[0,1]} \mathrm{E}\left[\left(\frac{\partial}{\partial y}\left\{\sigma_i(s, X_t)\sigma_j(s, X_t)\right\}\right)^4\right] < F, \sup_{s,t\in[0,1]} \mathrm{E}\left[\left(\frac{\partial^2}{\partial y^2}\left\{\sigma_i(s, X_t)\sigma_j(s, X_t)\right\}\right)^4\right] < F.$$

Set $\xi = X_0$ and assume also that $\mathrm{E}\,|\xi|^8 < \infty$. We will denote by B_t the standard Brownian bridge. Consider the following stochastic process

$$N_t \;:=\; \int_0^t \left\{\sigma^2(s, X_s) - \sum_{j=1}^{d} \theta_j^{\min} \sigma_j^2(s, X_s)\right\} ds, \tag{3}$$

where

$$\boldsymbol{\theta}^{\min} := \operatorname{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^d} \int_0^1 \left\{ \sigma^2(s, X_s) - \sum_{j=1}^{d} \theta_j \sigma_j^2(s, X_s) \right\}^2 ds. \tag{4}$$

It should be noted that under the hypothesis (2) the process N_t is equal to the *zero process*. Let $\langle ., . \rangle_2$ be the standard inner product on $L_2[0,1]$ (i.e. $\langle x, y \rangle_2 = \int_0^1 x(t)y(t)dt$). Random variable $\boldsymbol{\theta}^{\min}$ can be expressed as $\boldsymbol{\theta}^{\min} = \boldsymbol{D}^{-1}\boldsymbol{C}$, where

$$\boldsymbol{D} = (D_{ij})_{1\le i,j\le d}, D_{ij} := \left\langle \sigma_i^2, \sigma_j^2 \right\rangle_2, \boldsymbol{C} = (C_1, \ldots, C_d)^T, C_i := \left\langle \sigma^2, \sigma_i^2 \right\rangle_2.$$

In practice X_s cannot be observed for each $s \in [0,1]$, so we must observe the trajectory only in discrete network of points (e.g. in $X_{\frac{k}{n}}, k = 0, 1, \ldots, n$). For estimating $\boldsymbol{\theta}^{\min}$ we use the least squares estimation method with

$$\begin{aligned} \boldsymbol{Y} &= \left(\hat{\sigma}^2 \left(\frac{1}{n}, X_{\frac{1}{n}} \right), \ldots, \hat{\sigma}^2 \left(\frac{n}{n}, X_{\frac{n}{n}} \right) \right)^T = \left(n \left(X_{\frac{1}{n}} - X_0 \right)^2, \ldots, n \left(X_{\frac{n}{n}} - X_{\frac{n-1}{n}} \right)^2 \right)^T, \\ \boldsymbol{X} &= \left(\sigma_j^2 \left(\frac{i}{n}, X_{\frac{i}{n}} \right) \right)_{i=1,\ldots,n}^{j=1,\ldots,d}. \end{aligned}$$

If we set

$$\begin{aligned} \hat{\boldsymbol{D}} &= (\hat{D}_{ij})_{1\le i,j\le d}, \hat{D}_{ij} := \frac{1}{n} \sum_{k=1}^{n} \sigma_i^2 \left(\frac{k}{n}, X_{\frac{k}{n}} \right) \sigma_j^2 \left(\frac{k}{n}, X_{\frac{k}{n}} \right), \\ \hat{\boldsymbol{C}} &= (\hat{C}_1, \ldots, \hat{C}_d)^T, \hat{C}_i := \sum_{k=2}^{n} \sigma_i^2 \left(\frac{k-1}{n}, X_{\frac{k-1}{n}} \right) \left(X_{\frac{k}{n}} - X_{\frac{k-1}{n}} \right)^2, \end{aligned}$$

we will estimate $\boldsymbol{\theta}^{\min}$ by

$$\hat{\boldsymbol{\theta}} = \left(\hat{\theta}_1, \ldots, \hat{\theta}_d \right)^T = \operatorname{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^d} (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\theta})^T (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\theta}) = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} = \hat{\boldsymbol{D}}^{-1}\hat{\boldsymbol{C}}. \tag{5}$$

The process N_t will be estimated by

$$\hat{N}_t := \int_0^t \widehat{\sigma^2(s, X_s)} ds - \sum_{j=1}^{d} \hat{\theta}_j \int_0^t \widehat{\sigma_j^2(s, X_s)} ds = \hat{B}_t^0 - \hat{\boldsymbol{B}}_t^T \hat{\boldsymbol{\theta}}, \tag{6}$$

where

$$\hat{B}_t^0 := \sum_{k=1}^{\lfloor nt \rfloor} \left(X_{\frac{k}{n}} - X_{\frac{k-1}{n}} \right)^2, \hat{\boldsymbol{B}}_t^T = (\hat{B}_t^1, \ldots, \hat{B}_t^d), \hat{B}_t^i := \frac{1}{n} \sum_{k=1}^{\lfloor nt \rfloor} \sigma_i^2 \left(\frac{k}{n}, X_{\frac{k}{n}} \right).$$

Testing for homoscedasticity

Firstly we will investigate how to test the hypothesis

$$H_0 : \sigma^2(t, X_t) = \sigma^2 \tag{7}$$

for some $\sigma > 0$. Set

$$A_t^{(n)} = \sqrt{n}(\hat{N}_t - N_t), \quad (t \in [0,1]).$$

Theorem 1. *Under the above assumptions and the hypothesis (7), the process* $\left(A_t^{(n)} \right)_{t\in[0,1]}$ *converges weakly on* $D[0,1]$ *to a process*

$$\left(\sqrt{2}\sigma^2 B_t \right)_{t\in[0,1]}. \tag{8}$$

Proof. See [*Dette*, 2008] theorem 3.6 part (b) and the pages 25-26. □

Under the hypothesis (7) ($d = 1$ and $\sigma_1 = 1$) using (5) we will get $\hat{\theta} = \sum_{k=2}^{n} \left(X_{\frac{k}{n}} - X_{\frac{k-1}{n}} \right)^2$. If we use (6), we will get

$$\begin{aligned} \hat{N}_t &= \sum_{k=2}^{\lfloor nt \rfloor} \left(X_{\frac{k}{n}} - X_{\frac{k-1}{n}} \right)^2 - \frac{1}{n} \sum_{l=1}^{\lfloor nt \rfloor} 1 \cdot \sum_{k=2}^{n} \left(X_{\frac{k}{n}} - X_{\frac{k-1}{n}} \right)^2 \\ &= \sum_{k=2}^{\lfloor nt \rfloor} \left(X_{\frac{k}{n}} - X_{\frac{k-1}{n}} \right)^2 - \frac{\lfloor nt \rfloor}{n} \sum_{k=2}^{n} \left(X_{\frac{k}{n}} - X_{\frac{k-1}{n}} \right)^2. \end{aligned}$$

To use the result (8) we also need a consistent estimator of $\gamma^2 = \sqrt{2}\sigma^2$. The suitable estimator will be

$$\hat{\gamma}_n = \sqrt{2} \sum_{k=2}^{n} \left(X_{\frac{k}{n}} - X_{\frac{k-1}{n}} \right)^2.$$

For testing the hypothesis (7) we will use a Mapping theorem (see [*Billingsley*, 1999]) to a statistic

$$N^{(n)} := \sqrt{n} \sup_{t \in [0,1]} \left| \frac{\hat{N}_t}{\hat{\gamma}_n} \right| \xrightarrow{D} \sup_{t \in [0,1]} |B_t| \,. \tag{9}$$

The hypothesis of constant volatility is rejected if $N^{(n)}$ exceeds the corresponding quantile of the distribution of the supremum of Brownian bridge on the interval [0,1]. Remember that such distribution can be written as

$$\mathrm{P}\left(\sup_{t \in [0,1]} |B_t| > x \right) = \sum_{k=1}^{\infty} (-1)^{k+1} e^{-2k^2 x^2} \tag{10}$$

for $x > 0$. This result can be found in [*Billingsley*, 1999].

Another method how to test the hypothesis (7) is to use a statistic

$$\hat{N}^2 = \left\{ \frac{n}{3} \sum_{k=2}^{n} \left(X_{\frac{k}{n}} - X_{\frac{k-1}{n}} \right)^4 - \left(\sum_{k=2}^{n} \left(X_{\frac{k}{n}} - X_{\frac{k-1}{n}} \right)^2 \right)^2 \right\}.$$

Similar argumentations show (see [*Dette*, 2003]) that under the hypothesis (7)

$$\sqrt{n} \hat{N}^2 \xrightarrow{D} \mathrm{N}(0, \delta^2),$$

where $\delta^2 = \frac{8}{3}\sigma^8$. As a consistent estimator of δ^2 we use

$$\hat{\delta}_n^2 = \frac{8}{3} \left(\sum_{k=2}^{n} \left(X_{\frac{k}{n}} - X_{\frac{k-1}{n}} \right)^2 \right)^4,$$

so we have

$$\frac{\sqrt{n} \hat{N}^2}{\hat{\delta}_n} \xrightarrow{D} \mathrm{N}(0, 1).$$

We will reject the hypothesis (7) if this statistic exceeds the corresponding quantile of the standard normal distribution.

Testing for the parametric form of the volatility

For testing more general hypothesis (2) we will need the following theorem.

Theorem 2. *Under the above assumptions, the process* $\left(A_t^{(n)} \right)_{t \in [0,1]}$ *converges weakly on* $D[0,1]$ *to a process* $(A_t)_{t \in [0,1]}$ *which is Gaussian conditioned on the* σ*-field* $\mathcal{F}$ *and for all* k *the finite dimensional distributions* $(A_{t_1}, \ldots, A_{t_k})^T$ *have the covariance matrix*

$$2\boldsymbol{V} \int_0^1 \boldsymbol{\Sigma}_{t_1,\ldots,t_k}(s, X_s) ds \; \boldsymbol{V}^T, \tag{11}$$

where

$$\boldsymbol{\Sigma}_{t_1,\dots,t_k}(s,X_s)=\begin{pmatrix}\boldsymbol{E} & \boldsymbol{F}\\ \boldsymbol{F}^T & \boldsymbol{G}\end{pmatrix},$$

with

$$\begin{aligned}\boldsymbol{E} &= \left(\sigma^4(s,X_s)1_{[0,t_i\wedge t_j)}(s)\right)_{i=1,\dots,k}^{j=1,\dots,k},\ \boldsymbol{F}=\left(\sigma_j^2(s,X_s)\sigma^4(s,X_s)1_{[0,t_i)}(s)\right)_{i=1,\dots,k}^{j=1,\dots,d},\\ \boldsymbol{G} &= \left(\sigma_i^2(s,X_s)\sigma_j^2(s,X_s)\sigma^4(s,X_s)\right)_{i=1,\dots,d}^{j=1,\dots,d},\end{aligned}$$

$$\tilde{\boldsymbol{V}}=-\begin{pmatrix}\boldsymbol{B}_{t_1}^T\boldsymbol{D}^{-1}\\ \vdots\\ \boldsymbol{B}_{t_k}^T\boldsymbol{D}^{-1}\end{pmatrix},\ \boldsymbol{V}=(\boldsymbol{I}_k|\tilde{\boldsymbol{V}}).$$

Proof. See [*Dette*, 2008] theorem 3.5 and its analogue on pages 23-24. □

This theorem shows that the limiting distribution of $A_t^{(n)}$ is Gaussian (conditioned on the process $(X_t)_{t\in[0,1]}$). To obtain critical values we will use a parametric bootstrap. Before that we also need to estimate the conditional variance

$$s_t^2=2(1,-\boldsymbol{B}_t^T\boldsymbol{D}^{-1})\int_0^1\boldsymbol{\Sigma}_t(s,X_s)ds\ (1,-\boldsymbol{B}_t^T\boldsymbol{D}^{-1})^T.$$

To estimate these elements we will use empirical approach. Define

$$\begin{aligned}\tilde{E} &= \sum_{k=1}^{\lfloor nt\rfloor}\left(X_{\frac{k}{n}}-X_{\frac{k-1}{n}}\right)^4,\ \tilde{\boldsymbol{F}}=\left(\sum_{k=1}^{\lfloor nt\rfloor}\sigma_i^2\left(\frac{k-1}{n},X_{\frac{k-1}{n}}\right)\left(X_{\frac{k}{n}}-X_{\frac{k-1}{n}}\right)^4\right)_{i=1,\dots,d},\\ \tilde{\boldsymbol{G}} &= \left(\sum_{k=1}^{n}\sigma_i^2\left(\frac{k-1}{n},X_{\frac{k-1}{n}}\right)\sigma_j^2\left(\frac{k-1}{n},X_{\frac{k-1}{n}}\right)\left(X_{\frac{k}{n}}-X_{\frac{k-1}{n}}\right)^4\right)_{i=1,\dots,d}^{j=1,\dots,d}.\end{aligned}$$

So the estimator of the matrix $\int_0^1\boldsymbol{\Sigma}_t(s,X_s)ds$ will be

$$\widehat{\int_0^1\boldsymbol{\Sigma}_t(s,X_s)ds}=\begin{pmatrix}\tilde{E} & \tilde{\boldsymbol{F}}\\ \tilde{\boldsymbol{F}}^T & \tilde{\boldsymbol{G}}\end{pmatrix}.$$

The corresponding estimator of the conditional variance we may express as

$$\hat{s}_t^2=2(1,-\hat{\boldsymbol{B}}_t^T\hat{\boldsymbol{D}}^{-1})\widehat{\int_0^1\boldsymbol{\Sigma}_t(s,X_s)ds}\ (1,-\hat{\boldsymbol{B}}_t^T\hat{\boldsymbol{D}}^{-1})^T.$$

This yields a statistic

$$Z_n=\sup_{t\in[0,1]}\left|\frac{\sqrt{n}\hat{N}_t}{\hat{s}_t}\right|. \tag{12}$$

The procedure of testing the hypothesis (2) is to compute Z_n from the original data, then generate data $X_{i/n}^{*(j)}$ $(i=1,\dots,n,\ j=1,\dots,B)$ from stochastic differential equation (1) with $b(t,x)\equiv 0$ and $\sigma^2(t,x)=\sum_{j=1}^d\hat{\theta}_j\sigma_j^2(t,x)$. Then we compute the corresponding bootstrap statistics $Z_n^{*(1)},\dots,Z_n^{*(B)}$ by (12) and compare Z_n with the corresponding quantiles of that bootstrap distribution.

Let's introduce another bootstrap method for testing the hypothesis (2), which is based on a random variable

$$N^2=\min_{\theta_1,\dots,\theta_d\in\mathbb{R}}\int_0^1\left\{\sigma^2(t,X_t)-\sum_{j=1}^d\theta_j\sigma_j^2(t,X_t)\right\}^2dt. \tag{13}$$

Under the hypothesis (2) N^2 is equal to the zero proces a.s. It can be proved [*Achieser*, 1956] that the variable N^2 can be written as follows

$$N^2 = C_0 - (C_1, \dots, C_d)\boldsymbol{D}^{-1}(C_1, \dots, C_d)^T, \tag{14}$$

where

$$\begin{aligned} C_0 &:= \left\langle \sigma^2, \sigma^2 \right\rangle_2 = \int_0^1 \sigma^4(t, X_t)dt,\ C_i := \left\langle \sigma^2, \sigma_i^2 \right\rangle_2 = \int_0^1 \sigma^2(t, X_t)\sigma_i^2(t, X_t)dt, \quad 1 \le i \le d, \\ \boldsymbol{D} &= (D_{ij})_{1\le i,j\le d}, D_{ij} := \left\langle \sigma_i^2, \sigma_j^2 \right\rangle_2 = \int_0^1 \sigma_i^2(t, X_t)\sigma_j^2(t, X_t)dt, \quad 1 \le i, j \le d. \end{aligned}$$

The empirical version of N^2 (which is also the least squares estimator) is defined by

$$T_n := \frac{1}{3}T_{2n} - \boldsymbol{T}_{1n}^T \hat{\boldsymbol{D}}_n^{-1} \boldsymbol{T}_{1n},$$

where

$$\begin{aligned} T_{1n,j} &:= \sum_{k=1}^{n-1} \sigma_j^2\left(\frac{k}{n}, X_{\frac{k}{n}}\right)\left(X_{\frac{k+1}{n}} - X_{\frac{k}{n}}\right)^2, \boldsymbol{T}_{1n} = (T_{1n,j})_{j=1,\dots,d}, \\ T_{2n} &:= \sum_{k=1}^{n-1} n\left(X_{\frac{k+1}{n}} - X_{\frac{k}{n}}\right)^4, \hat{\boldsymbol{D}}_n := \frac{1}{n}\boldsymbol{X}^T\boldsymbol{X}. \end{aligned}$$

Let's also work with the least squares estimator $\hat{\boldsymbol{\theta}}$. It can be shown (see [*Dette*, 2006]) that under the hypothesis (2) (because of $N^2 = 0$)

$$\sqrt{n}\frac{T_n}{\delta} \xrightarrow{D} \mathrm{N}(0, 1),$$

where $\delta^2 = \frac{8}{3}\int_0^1 \sigma^8(t, X_t)dt$. The consistent estimator of δ^2 (see [*Dette*, 2006]) can be expressed as

$$\hat{\delta}_n^2 = \frac{8}{3}\frac{1}{n}\sum_{i=1}^{n}\left\{\sum_{j=1}^{d}\hat{\theta}_j\sigma_j^2\left(\frac{i}{n}, X_{\frac{i}{n}}\right)\right\}^4.$$

Consequently (under the hypothesis (2)) we will get

$$Y_n = \sqrt{n}\frac{T_n}{\hat{\delta}_n} \xrightarrow{D} \mathrm{N}(0, 1). \tag{15}$$

The first possibility is to compare this statistic with the quantile of the standard normal distribution. Second possibility is to use the bootstrap method with Y_n (similarly as it was with Z_n).

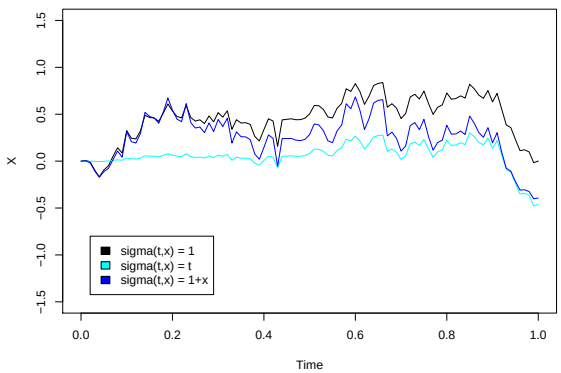


Figure 1. Example of trajectories starting from zero for different volatility functions and zero drift.

	$N^{(n)}$						$\hat{N}^2$					
b/n	25	50	100	200	500	1000	25	50	100	200	500	1000
0	0.013	0.021	0.030	0.035	0.026	0.040	0.027	0.040	0.052	0.062	0.053	0.059
1	0.012	0.025	0.031	0.034	0.028	0.041	0.032	0.037	0.056	0.058	0.056	0.059
$x+1$	0.013	0.022	0.029	0.041	0.027	0.041	0.029	0.030	0.051	0.056	0.056	0.059
t	0.014	0.022	0.031	0.035	0.028	0.041	0.027	0.039	0.053	0.061	0.052	0.058

Table 1. Approximation of the nominal level of the tests for different drift functions with $\sigma = 1$, based on $N^{(n)}$ and $\hat{N}^2$.

	$N^{(n)}$						$\hat{N}^2$					
σ/n	25	50	100	200	500	1000	25	50	100	200	500	1000
$x+1$	0.380	0.582	0.767	0.874	0.971	0.990	0.255	0.441	0.619	0.743	0.902	0.962
t	0.763	0.996	1.000	1.000	1.000	1.000	0.317	0.620	0.890	0.992	1.000	1.000
$1+xt$	0.331	0.555	0.735	0.830	0.928	0.964	0.195	0.346	0.511	0.644	0.783	0.851

Table 2. Empirical power for different volatility functions with $b(t,x) = x + t$, based on $N^{(n)}$ and $\hat{N}^2$.

	Y_n, $\sigma(t,x) = 1$						$\sigma(t,x) = x+1$					
σ_1/n	25	50	100	200	500	1000	25	50	100	200	500	1000
1	0.028	0.033	0.054	0.062	0.053	0.060	0.233	0.422	0.617	0.743	0.901	0.963
$x+1$	0.187	0.307	0.492	0.659	0.847	0.936	0.023	0.024	0.023	0.036	0.045	0.044
t	0.259	0.432	0.721	0.921	1.000	1.000	0.497	0.602	0.699	0.779	0.842	0.903
$x+t$	0.374	0.532	0.769	0.930	0.996	0.999	0.281	0.340	0.418	0.513	0.566	0.642

Table 3. Empirical power for different volatility functions with $b(t,x) = x + t$, $d = 1$ and $\sigma(t,x)$ (is done) based on Y_n.

Graphical and numerical illustrations

All numerical results were computed from 1000 generations and with significance level $\alpha = 0.05$.

As we can see — the drift function is irrelevant because of using drift-invariant statistics. For approximation of the nominal level of the test it is better to use the test based on the statistic $\hat{N}^2$. For testing the hypothesis of constant volatility it seems to be better to use the statistic $N^{(n)}$, which has greater power. For testing the general hypothesis (2) we can use the test based on the statistic Y_n, which can be easily implemented and computed. Power of this test depends (not surprisingly) on alternatives, which can be similar to the null hypothesis as it is shown in Figure 1. The properties of the tests based on the parametric bootstrap method may be more efficient, but they are difficult from numerical point of view. For the next extension of our results we need next simulation study (e.g. for random times). In the next work we are going to investigate some nonparametric methods of testing or estimating the parameters of other processes.

Acknowledgments. The work was supported by the grant SVV 261315/2011.

References

Achieser N.J., (1956): Theory of approximation. Dover Publications Inc., New York.

Billingsley P., (1999): Convergence of Probability Measures. NY: John Wiley & Sons, Inc., New York.

Corradi V. and White H., (1999): Specification tests for the variance of a diffusion. Journal of Time Series Analysis 20, 253-270.

Dette H., Podolskij M. and Vetter M., (2006): Estimation of integrated volatility in continuous time financial models with applications to goodness-of-fit testing. Scandinavian Journal of Statistics 33, 259-278.

Dette H. and Podolskij M., (2008): Testing the parametric form of the volatility in continuous time diffusion models—an empirical process approach. Journal of Econometrics 143, 56-73.

Dette H. and Wilkau C., (2003): On a test for a parametric form of volatility in continuous time financial models. Finance and Stochastics 7, 363-384.

Genon-Catalot V., Laredo C. and Picard D., (1992): Non-parametric estimation of the diffusion coefficient by wavelet methods. Scandinavian Journal of Statistics 19, 317-335.

Jiang J.G. and Knight J.L., (1997): A nonparametric approach to the estimation of diffusion processes with an application to a short-term interest rate model. Econometric Theory 13, 647-667.

Woerner J.H.C., (2003): Estimation of integrated volatility: a unifying approach to model selection and estimation in semimartingale models. Statistics and Decisions 21, 47-68.

WDS'11 Proceedings of Contributed Papers, Part I, 19–24, 2011. ISBN 978-80-7378-184-2 © MATFYZPRESS

Tools for Decision Making under Uncertainty

V. Sečkárová
Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.
Institute of Information Theory and Automation, Prague, Czech Republic.

Abstract. Decision making is a process used in many parts of life to determine an optimal choice with respect to a particular subjective aim for a particular decision maker. In this paper we focus on two often considered distinct aims, namely maximizing of an utility function (e.g. an investment profit) and getting a more reliable global description of considered situation based on observed data (e.g. the final outcome of databases merging). In both cases we face the problem, that the data are unreliable, since they contain uncertainty caused by their source (i.e. human being). If we are looking for the optimum of the former aim, a game theory reformulation of the decision making task brings a smoother way to reach it. If the latter aim is considered, a merging procedure (also called fusion) processing the data should help us. This aim was chosen as the author's present work is connected with it and she has to inspect the state of the art. This paper describes four recently developed methods dealing with decision making under uncertainty in two considered directions and one tool used for comparison of the fusion algorithms.

Introduction

In common life almost every minute we have to make a decision satisfying our subjective aims. The procedure leading to the best decision among the possible ones is called decision making. Here we focus on two distinct aims, which one may have almost surely considered in the past. The first one is the maximization of an utility function (e.g. investment profit), the second one is to get a more reliable and correct description of considered environment (e.g. about the occurrence of considered events). Since in both cases the observations are required, handling of data sources (e.g. human beings, sensors) is important part of decision making. Every source has its limiting abilities, e.g. precision of the measuring sensor or human ability to provide complete information, therefore given data contain a part reflecting the source's uncertainty. The seminal work of Wald on decision making under uncertainty dates back to 50's (see Wald [1950]), since then people try to deal with this issue by developing different methods.

In the case when maximal profit is of interest a game theory formulation of original decision making task should properly treat the uncertainty and give us a smoother and more consistent way leading to a solution (see Neumann and Morgenstern [1944]). The uncertainty here is caused by a non-cooperative relation among the sources, which means they have no chance to observe the choices of other sources. In Reneke [2009] author suggests a two player game, DM (a decision maker) and incompletely observable and unpredictable source called NATURE. The uncertainty is treated with functions from the sigmoid family. To find the optimal choice (i.e. investment alternative), authors introduce a decision variable based on second order statistics of the score for particular investment alternative. In Madani and Lund [2011] a higher number of non-cooperative sources is considered. But none of them is of type NATURE from the previous case. Here the uncertainty is handled by interpretation of the matrix (subject to criteria and alternatives) of sources' payoff via ordinal rank matrix and by Monte Carlo simulations. To obtain the optimal game solution, the ordinal rank matrix is transformed into game theory terms (via transition matrix) and several stability definitions are applied.

If we are interested in a global description of considered events (hypotheses) and we have corresponding observations, their fusion leads to a satisfactory solution. The uncertainty comes through the impreciseness of the data given by sources. In Fassinut-Mombot and Choquel [2004] a method using basic terms of information theory is introduced, e.g. entropy and mutual information. Authors suggest to reduce the space of given observations for a particular hypothesis via introducing the notion of source's redundancy (equivalently via source's complementarity). To do that a probabilistic representation of the observations obtained via maximum entropy principle (Shore and Johnson [1980]) is used. The optimal hypothesis is determined by conditional entropy. Note that here the uncertainty is measured via Shannon's entropy (see Shannon [1948]). In Pavlin et al. [2010] the hypotheses are represented via random variables and the conditional probabilities, then modeled by Bayesian networks and described by factor graphs. Authors introduce a processing unit operating on the subset of all variables, which satisfies

a cooperative scenario within the sources operating at least one common variable. Then the updated versions of (marginal) posterior probabilities of random variables are computed. Finally, we take a look at an information-based quality measure (see Qu et al. [2002]), which serves for comparison of the fusion algorithms by computing a mutual information between the input observations and output. Although this concept is simple and seems to work well, under specific conditions it gives unsatisfactory results, as shown in Chen et al. [2008].

In the following sections we give a brief review of these methods together with their subjective critique.

Game theory reformulation

Reformulation under uncertainty and risk

In this section we take a look on the formulation proposed in Reneke [2009]. Here, the decision maker's attention is paid to long term investments connected with increasing oil prices and environmental degradation (i.e. degradation on air). The aim is to construct a sequence of rational investment decisions, which keeps a balance between decision maker's payoff and risk (its gain and loss).

Since the oil prices and air degradation are almost unpredictable in long time period, the construction of such a sequence is a difficult problem. To solve this author suggests a two-player game between the decision maker and NATURE. The strategies of these players are:

- DECISION MAKER (DM) – rational investment decisions,
- NATURE – future oil prices and environmental (air) degradation.

The uncertainty here comes through the fact, that at the time DM makes the investment decision, NATURE's strategy is unknown to it. Following the Knight's distinctions (see Knight [1921]), the uncertainty is considered as an unquantifiable variable (there is no assumption on existence of describing distribution), while the risk is estimated in terms of a quantifiable variable (with a distribution). Their proposed procedure is the following:

- construct an outcome (denoted by Z) of a random performance of one of DM's investment alternatives; this outcome depends on time and NATURE's strategies (via the investment alternative),
- construct the score of the game (denoted by V); this depends on the outcome from the previous step and on a discount rate r,
- for later purposes: compute the expectation and the variance of V (both are still depending on NATURE's strategies),
- make an assumption on the form of NATURE's strategies – each is described by single parameter sigmoid function; the original uncertainties are now reduced to uncertainties in the pace of change – the timing (determined by a particular parameter) of NATURE's strategy performance is unknown (there are two different parameters, because two original uncertainties are modeled separately).

The computations now depend on the choice of timing parameters; in the paper authors considered three values for each parameter and computed parts of decision variable (introduced later) for all possible pairs of parameters. Since the uncertainties have already been modeled, the main focus is now on the construction of decision variable.

Since the outcome Z is normally distributed with mean μ and standard deviation σ dependent on DM's alternative, the bad outcome is introduced as $\{Z \leq \mu - \alpha\sigma\}$, where α is some positive number determined by DM's risk tolerance. The previously mentioned investment risk is then interpreted as the probability of a bad outcome and is taken as a constant. Then for each DM's alternative we can compute the corresponding μ_i and σ_i. If we have to decide between two alternatives i and j and the criterion $\mu_i - \sigma_i > \mu_j - \sigma_j$ holds for all considered possibilities of timing parameters, we should take the alternative i. The only arising problem is that the inequality between alternatives can change at another time point, so the choice of the best alternative is ambiguous. The authors suggest two criteria, both following the relation between alternatives: one on the whole set of possibilities for timing parameters, the other based on particular values of timing parameter. If the arising set of possible investment alternatives has more than one element you will have to use the least squares method to find the best one.

Here only the case of two uncertainties is treated. Multiple uncertainty situation can also be addressed, since every larger problem can be decomposed into (already proposed) smaller problems.

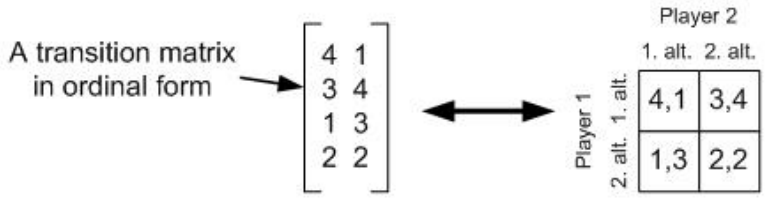


Figure 1. Relationship between multi-criteria decision making form and game theory form.

Conclusion: Through the paper many strict assumptions were made: e.g. the uncertainty is modeled by a specific one-parameter function (to simplify the computation of the variables based on it), where also the parameter values were set. Finally, we get a deterministic model, describing a specific situation, which obviously can not be used generally.

A Monte Carlo game theory approach

In this section we take a look on another formulation of decision making task under uncertainty (see Madani and Lund [2011]). In contrary to the previous case, the proposed method:

- treats multiple sources, multiple criteria and obviously multiple alternatives,
- considers only sources with the ability to provide the information about possible alternatives; none of them is unpredictable as the source NATURE (see previous section).

Again, the sources are non-cooperative, which means none of them can see the choices of the other sources through the game. The non-cooperative scenario was preferred to a perfect cooperation because the latter allows agreement only on one alternative and a cooperative outcome – any kind of disagreement leading to a non-cooperative outcome is disregarded. As we will see later, the proposed method considers both types of outcomes.

If we assume the situation with two DMs, one criterion and two alternatives, then the main idea of this game-theory reformulation is:

1. Construct a performance (or utility) matrix P for all considered alternatives under particular criterion of a particular source (here we get $2 \times (1 \times 2) = 2 \times 2$ matrix). This is a conventional form of multi-criteria decision making; for the considered situation there are only two outcomes, which occur when both sources agree on the same alternative.

2. Construct a new 2×2 matrix corresponding to P, where the ordinal ranks are used instead of utilities – the elements are ranks of a particular alternative with respect to the criterion of a concrete DM.

3. Construct a transition matrix to convert the proposed problem into a game theoretic form. A transition matrix is now a matrix with the number of rows corresponding to the number of all possible combinations (strategies) of DMs' alternatives and with the number of columns corresponding to the number of players. Each row describes a possible outcome, each column represents a player. The whole matrix represents the payoff of two players from four possible outcomes – the elements of the matrix are represented by ordinal ranks (e.g. a higher rank coincides with higher payoff). Thus we have now a 4×2 matrix (see Fig. 1 on the left). The transition matrix corresponds to a 2×2 game, shown in Fig. 1 on the right.

Now we want to find the possible results of the game. This step has the following pattern:

- use several stability definitions (Madani and Hipel [2011]) to determine, which possible outcome is the most likely to occur,
- find the outcome determined by majority of stability conditions – this one has a higher chance to be a final outcome of the game.

So far we did not stress that the final results based on ordinal representation of information (see step 2.) are less sensitive to uncertainty provided by DMs. It is because the results are insensitive to the changes in the performance as long as the rankings do not change. By transformation into game theory terms there is no need to weight DMs and criteria (to determine which one is more or less useful), which also reduces the influence of uncertainty on results. The last step to eliminate the remaining uncertainty consists of Monte Carlo simulations (multiple computations), where authors:

- choose a random set of performances,
- randomly select utilities of the four available strategies for each considered DM,
- construct the transition matrix,
- solve the game with 6 non-cooperative stability definitions.

Conclusion: This approach brings the simplification in the representation of original situation by introducing the ordinal rank matrix followed by the game theory reformulation. In contrast with the previous method, it has just one special assumption – a non-cooperative scenario between the sources, which in fact generalizes a multi-criteria decision making. The only problem brings the last part, the elimination of uncertainty via Monte Carlo simulations, which is computationally very time-demanding and is to be improved.

Information fusion

Information fusion (or merging) is another way of how the decision making faces uncertainty. It is a process of combining information from heterogeneous sources in order to get more reliable information describing the whole considered environment (e.g. events or hypotheses). The main issue of the information fusion is how to treat the incompleteness, impreciseness and uncertainty contained in processed information. In this section we briefly describe three methods introducing different kinds of fusion processes.

But first, let us recall some of the notions of information theory:

- entropy H(X) measures uncertainty contained in the random variable X; usually it refers to Shannon's entropy (see Shannon [1948]) and evaluates the expected value of information contained,
- maximum entropy principle MEP (see Shore and Johnson [1980]) states, that from all of the distributions describing the considered environment and satisfying particular constraints the one with the highest entropy should be chosen,
- conditional entropy $H(Y|X)$ measures the amount of information provided by the output Y when the input X of the fusion system is known,
- mutual information $I(X,Y)$ measures the statistical dependence between two random variables X, Y and the amount of information that one variable contains about the others.

Entropy fusion model

Fassinut-Mombot and Choquel [2004] introduce a method using all terms mentioned above to determine the best information description of the considered environment. The main idea is to reduce the combination space (space of all inputs X) by introducing a notion of source's redundancy and complementarity. In particular, authors use mutual information and conditional entropy to make a decision, which information sources to merge. They follow the decomposition of the entropy of an output Y as follows:

$$I(X,Y) + H(Y|X) = H(Y) = \text{constant} \quad \text{(see Fig. 2)},$$

where $I(X,Y)$ allows us to measure the redundancy of transmitted information and $H(Y|X)$ allows us to measure the complementarity of the information. In order to optimize the fusion system:

- maximize the mutual information $I(X,Y)$ between all inputs X and the output vector Y (which coincides with minimization of the redundancy of information sources),
- or equivalently minimize the conditional entropy $H(X|Y)$ (which coincides with maximization of the complementarity between output and inputs).

The main steps of the method are:

- Modeling step:
  - construct the set including all elementary events relevant to the given problem (which stands for the set of all possible hypotheses),

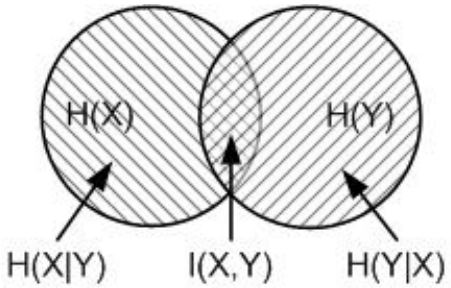


Figure 2. Relationship among simple entropy, conditional entropy and mutual information.

  - represent the knowledge via conditional probability of hypothesis' characteristic conditioned on observed data; to choose the optimal one use MEP (note, that by using MEP also the uncertainty is respected),

- Combination step: use conditional entropy of each hypothesis conditioned on the corresponding set of observations for a reduction of the combination space – choose the observations for which the conditional entropy is minimal,

- Decision making step: choose the hypothesis with the lowest conditional entropy (conditioned on the corresponding reduced set of observations); if ambiguity occurs, measure a quality of the decision based on mutual information between the corresponding reduced combination space and hypothesis to determine the optimal hypothesis.

Conclusion: This work brings interesting procedure for fusion of information sources. However, there is a concern about the applicability. In the beginning a very strong assumption is made: the set of all possible hypotheses is given. If the hypotheses are not specifically given by somebody interested in the result of the merging, their formulation might be a very difficult task.

A mutli-agent systems approach

The problem proposed in the previous section should be overcome by the method proposed in Pavlin et al. [2010]. When using real-world describing sources for decision making, it is impossible to determine the set of possible events. Therefore it is suggested that the critical hidden events must be expressed via fusion of an enormous amount of uncertain and heterogeneous information (e.g. obtained from sensors). The technique consists of the following (simplified) steps:

- assume that the relevant states (realizations of events) can be described through a finite number of random variables,

- model the hypotheses (its realizations cover hidden and observed states) by conditional probability conditioned on observations; use the interpretation via Bayesian networks to compute the probabilities,

- interpret the current situation via probabilistic graphical models: the evidence of instantiated variables should bring the simplification in computation of the joint probability over the set of all hypotheses,

- introduce a fusion agent – a processing unit operating on the variables in its Bayesian network (a subset of all considered hypotheses) and cooperating with other agents,

- update the probability of a particular random variable by computing the posterior probability based on information from sources containing this variable in their set of hypotheses.

Conclusion: Since this method treats a very general situation, it is obvious that its structure is more complicated. One can complain, that under the amount of allowed uncertainty in this case, it is hard to obtain, e.g., the probabilities based on often unreliable data. Despite that, under several assumptions like conditional independence, this method brings useful result.

Information based quality measure and its theoretical analysis

Because many methods of information fusion are being developed, one can not be sure which one to use in order to get the best output. Therefore, there arises the need to measure the quality of the output based on particular input. One of such quality measures was introduced in Qu et al. [2002]. It is based on mutual information between the output F and inputs A, B from the sources respectively as follows:

$$Q = I(f, a) + I(f, b)$$

and it is shown on an example of image fusion of two image sources, where a, b denote corresponding pixels in the image given by the first and the second source, respectively; f stands for the fused pixel. The symbol $I(f, a)$ (similarly $I(f, b)$) refers to mutual information:

$$I(f, a) = \sum_{f,a} p(f, a) \log \frac{p(f, a)}{p(f)p(a)},$$

where $p(f, a)$ stands for a joint distribution of random variables A, F and $p(f)p(a)$ is the distribution associated with the independence of A and F.

Several fusion schemes were compared by using this quality measure in Qu et al. [2002]. In a recent paper (Chen et al. [2008]), the weighted averaging of two images was analyzed. For the analytical studies the intensity of each pixel is introduced and is modeled it by Gaussian random vector and Gaussian noise. The study shows that under image degradation the quality measure Q gives antipodal results to what is usually expected. In particular, if we add more noise to the source images, the quality measure Q increases (although we would expect the opposite).

Conclusion: Be careful when comparing different fusion techniques via quality measures. The main issue is to make sure we compare the methods at the same image level, at different levels it has different meaning (see Qu et al. [2002]). Also pay attention to the results the quality measures are giving, since occurrence of an unexpected behavior for a specific situation is highly possible.

Acknowledgments. This work was supported by the grants GAČR 102/08/0567, MŠMT 1M0572 and UK SVV 261315/2011.

References

Chen, Y., Xue, Z., and Blum, R., Theoretical analysis of an information-based quality measure for image fusion, *Information Fusion*, *9*, 161–175, 2008.

Fassinut-Mombot, B. and Choquel, J., A new probabilistic and entropy fusion approach for management of information sources, *Information Fusion*, *5*, 35–47, 2004.

Knight, F., *Risk, uncertainty and profit*, Library of Economics and Liberty. Retrieved May 20, 2011 from the World Wide Web, 1921.

Madani, K. and Hipel, K., Non-cooperative stability definitions for strategic analysis of generic water resources conflicts, *Water resource management*, pp. 1–29, 2011.

Madani, K. and Lund, J., A Monte-Carlo game theoretic approach for Multi-Criteria Decision Making under uncertainty, *Advances in Water Resources*, *34*, 607–616, 2011.

Neumann, J. V. and Morgenstern, O., *Theory of games and economic behavior*, Princeton University Press, 1944.

Pavlin, G., de Oude, P., Maris, M., Nunnik, J., and Hood, T., A multi-agent systems approach to distributed bayesian information fusion, *Information Fusion*, *11*, 267–282, 2010.

Qu, G., Zhang, D., and Yan, P., Information measure for performance of image fusion, *Electronics Letters*, *38*, 313–315, 2002.

Reneke, J. A., A game theory formulation of decision making uncer conditions of uncertainty and risk, *Nonlinear Analysis: Theory, Methods & Applications*, *71*, e1239–e1246, 2009.

Shannon, C., A Mathematical Theory of Communication, *Bell System Technical Journal*, *27*, pp. 379–423, 1948.

Shore, J. E. and Johnson, R. W., Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy., *IEEE Trans. Inf. Theory*, *26*, 26–37, 1980.

Wald, A., *Statistical Decision Functions*, John Wiley, New York, London, 1950.

 ISBN 978-80-7378-184-2

Ensemble Kalman Filter

I. Kasanický

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic,
Institute of Computer Science, Academy of Sciences of the Czech Republic.

K. Eben

Institute of Computer Science, Academy of Sciences of the Czech Republic.

Abstract. The ensemble Kalman filter (EnFK) has recently become one of the most popular methods for high dimensional data assimilation and it is widely used in many disciplines such as discretization of partial differential equations in geophysics or image reconstruction. The EnKF has been originally proposed as a Monte Carlo approximation to the extended Kalman filter, which is in practice inapplicable when dimension of the input data is huge.
Numerous publications have studied applications and asymptotic results of the EnKF, while properties of the Kalman filter (KF) and EnKF for the infinite dimensional Hilbert spaces are still under the development.
The main purpose of this paper is a brief description of the high dimensional EnKF with references to the most valuable publications. Also, a short introduction to problems related with KF and EnKF on infinite dimensional Hilbert spaces is included.

Introduction

The ensemble Kalman filter (EnKF) is a sophisticated sequential data assimilation method, used especially for high dimensional data. It is a Monte Carlo approximation of the Kalman filter (KF), where the true covariance matrix in the KF is replaced by the sample covariance matrix computed from the ensemble. Therefore it could be implemented very efficiently. This method was published for the first time in *Evensen* [1994].

Data assimilation is a statistical method for estimating the true state of a system (typically dynamic system evolving in time) by merging various measurements irregularly distributed in space and time, with a prior knowledge of the state. It has been widely used in many disciplines such as image reconstruction, weather prediction or wildfire modelling.

The paper is structured as follows. The first section is meant as a brief introduction to data assimilation problem, it presents necessary assumptions needed and ends with definition of the sequential estimation formula. In the next section the KF's equations are reminded. The third section is used to derive the EnKF from the KF with references to relevant papers. The last section presents a summary of the main questions, which arise when we try to use KF on general Hilbert space.

For readers interested in implementation of the EnKF we recommend papers *Evensen* [2003] and *Mandel* [2009], where a discussion about efficient implementation of EnKF is included. Very good, but also very extensive, is the book *Evensen* [2009]. Many scripts, usually written in Matlab, with further discussion and development could be found on the website http://enkf.nersc.no/, which is administrated directly by Geir Evensen.

Preliminaries

Suppose we want to predict temperature in Europe. In that case we would probably use some geophysical model and make prediction of the temperatures in every point in some three-dimensional grid covering Europe, say, for the next day. The quality of the prediction depends

very much on the error of initial conditions which the model starts from. Usually we have some prior estimate of initial conditions, e.g. from a previous run of the model. The initial conditions can be improved with the help of available measurements.

It is obvious that it is not possible to have real measurements in all points of the grid, so the dimension of the measured data would be much lower than dimension of our model state. Thus we come to an estimation problem, which in geophysical sciences is called data assimilation. This terminology reflects the dominant role of the model which has to keep physical balance and the perturbation caused by statistical estimation procedure needs to be limited in practice.

So we assume that we have available data in some discrete time steps $t_1, \ldots, t_K$ and will denote $\mathbf{D}_{t_i}$, $\mathbf{D}_{t_i} \in \mathbb{R}^m$ data observed at the time t_i. Similarly we will denote $\mathbf{X}_{t_i}$ a vector of size n, which contains all the values describing our model state at the time t_i. The dependence between $\mathbf{D}_{t_i}$ and $\mathbf{X}_{t_i}$ is characterized by the function

$$h_{t_i} : \mathbf{X}_{t_i} \to h_{t_i}(\mathbf{X}_{t_i}) \sim \mathbf{D}_{t_i},$$

which is deterministic, but it could be different in the different time steps. This function is usually called observation operator. The length of $\mathbf{D}_{t_i}$ is often much lower then the length of $\mathbf{X}_{t_i}$, $m \ll n$.

The model state is a random vector and we will assume, that its distribution has

- a bounded second moment and
- a density on $\mathbb{R}^n$.

Model states are evolved in time using the known operator (function)

$$\mathcal{M} : (\mathbf{X}_{t_i}, t_i, t_{i+1}) \to \mathbf{X}_{t_{i+1}}.$$

It could be for example numerical solution to some differential equations.

Our goal is to estimate the model state at some future time $\mathbf{X}_{t_K}$ using all available data until time t_{K-1}. We will assume, that the model is Markovian of order 1

$$p(\mathbf{X}_{t_K}|\mathbf{X}_{t_1}, \ldots, \mathbf{X}_{t_{K-1}}) = p(\mathbf{X}_{t_K}|\mathbf{X}_{t_{K-1}}), \tag{1}$$

where $p(\cdot)$ denotes density function. The main theoretical background of data assimilation methods is the Bayes theorem, which states that

$$p(\mathbf{X}^a_{t_K}) = p(\mathbf{X}^f_{t_K}|\mathbf{D}_{t_K}) \propto p(\mathbf{D}_{t_K}|\mathbf{X}^f_{t_K})p(\mathbf{X}^f_{t_K}), \tag{2}$$

where $\mathbf{X}^f_{t_K}$ represents the forecast model state, in Bayesian statistics called also prior state, and $\mathbf{X}^a_{t_K}$ represents posterior model state, often called "analysis" by the meteorologists. Using Bayesian rule (2) together with Markov characteristics (1) we obtain the sequential estimation formula

$$p(\mathbf{X}^a_{t_K}) \propto p(\mathbf{D}_{t_K}|\mathbf{X}^f_{t_K})p(\mathbf{X}^f_{t_K}|\mathbf{D}_{t_1}, \ldots, \mathbf{D}_{t_{K-1}}). \tag{3}$$

The Kalman Filter

One solution to the problem describe above is well known for almost 50 years. It was first proposed by *Kalman* [1960] and *Kalman and Bucy* [1961] and is known as Kalman Filter.

The KF restricts model function to be linear, so it could be rewritten using an $n \times n$ matrix $\mathbf{M}_{t_i}$ and n-dimensional vector $\mathbf{b}_{t_i}$ in the form

$$\mathcal{M}(\mathbf{X}_{t_i}, t_i, t_{i+1}) = \mathbf{M}_{t_i}\mathbf{X}_{t_i} + \mathbf{b}_{t_i}.$$

The linearity of $\mathcal{M}$ is quite restrictive and in many application the nonlinear operator $\mathcal{M}$ is replaced by linear approximation. In this case the matrix $\mathbf{M}_{t_i}$ is replaced by Jacobian of the model $\mathcal{M}$ evaluated at time t_i and the vector $\mathbf{b}_{t_i}$ is replaced by the model evaluated at zero, that means by $\mathcal{M}(\mathbf{0}, t_i, t_{i+1})$. The KF also requires observation function to be linear and then it could be rewritten in similar matrix form

$$h_{t_i}(\mathbf{X}_{t_i}) = \mathbf{H}_{t_i}\mathbf{X}_{t_i} + \mathbf{h}_{t_i},$$

where $\mathbf{H}_{t_i}$ is $m \times n$ matrix and $\mathbf{h}_{t_i}$ is vector of length m. Conditional distribution of the data available at time t_i under condition of forecast model state is assumed to be normal

$$\mathbf{D}_{t_i}|\mathbf{X}^f_{t_i} \sim N(\mathbf{H}_{t_i}\mathbf{X}^f_{t_i}, \mathbf{R}_{t_i}) \tag{4}$$

with some known non-singular covariance matrix $\mathbf{R}_{t_i}$. The density of the distribution (4) is called data likelihood. Distribution of model state is also assumed to be normal

$$\mathbf{X}^f_{t_i} \sim N(\boldsymbol{\mu}^f_{t_i}, \mathbf{Q}^f_{t_i}), \tag{5}$$

where non-singular covariance matrix $\mathbf{Q}^f_{t_i}$ is counted in each time step. We assume that $\mathbf{Q}^f_{t_1}$ is known.

Under all these conditions, the distribution of analysis (posterior) model state remains normal. It can be shown that the posterior mean $\boldsymbol{\mu}^a_{t_i}$ and covariance matrix $\mathbf{Q}^a_{t_i}$ are given by (6)–(8) written below:

$$\mathbf{K}_{t_i} = \mathbf{Q}^f_{t_i}\mathbf{H}^\top_{t_i}(\mathbf{H}_{t_i}\mathbf{Q}^f_{t_i}\mathbf{H}^\top_{t_i} + \mathbf{R}_{t_i})^{-1}, \tag{6}$$

$$\boldsymbol{\mu}^a_{t_i} = \boldsymbol{\mu}^f_{t_i} + \mathbf{K}_{t_i}(\mathbf{D}_{t_i} - \mathbf{H}_{t_i}\boldsymbol{\mu}^f_{t_i}), \tag{7}$$

$$\mathbf{Q}^a_{t_i} = (\mathbf{I} - \mathbf{K}_{t_i}\mathbf{H}_{t_i})\mathbf{Q}^f_{t_i}. \tag{8}$$

The matrix $\mathbf{K}_{t_i}$ is called gain matrix and all three equations together are called Kalman formula update. So the KF could be summarized in a two step recursive algorithm:

- analysis step

$$\boldsymbol{\mu}^a_{t_i} = \boldsymbol{\mu}^f_{t_i} + \mathbf{K}_{t_i}(\mathbf{D}_{t_i} - \mathbf{H}_{t_i}\boldsymbol{\mu}^f_{t_i}),$$
$$\mathbf{Q}^a_{t_i} = (\mathbf{I} - \mathbf{K}_{t_i}\mathbf{H}_{t_i})\mathbf{Q}^f_{t_i},$$

- forecast step

$$\boldsymbol{\mu}^f_{t_{i+1}} = \mathbf{M}_{t_i}\boldsymbol{\mu}^a_{t_i} + \mathbf{b}_{t_i},$$
$$\mathbf{Q}^f_{t_{i+1}} = \mathbf{M}^\top_{t_i}\mathbf{Q}^a_{t_i}\mathbf{M}_{t_i}.$$

Proofs of all equations and a detailed treatment of the KF could be found in many statistical books, our notation conforms with e.g. that of *Beezley* [2009].

The KF is well known, its theoretical properties have been studied for a long time and under normality assumption and some other assumptions concerning independence of errors it has optimum properties. As such it is frequently used in many engineering applications. In Earth sciences, however, the length of the state vector is very high. For example in weather prediction, a 3D grid covering Europe in horizontal resolution of 10 km may have dimensions between 100 and 200 in each direction and e.g. 30 - 50 vertical levels. With 6 state variables this causes, that length of the model state vector $\mathbf{X}_{t_i}$ is about 5×10^6. In such a case it is not possible to store the covariance matrices $\mathbf{Q}^f_{t_i}$ and $\mathbf{Q}^a_{t_i}$ in any computer memory and the formulas (6)–(8) start to be intractable in practice.

The Ensemble Kalman Filter

The basic idea behind the ensemble Kalman Filter is a low rank approximation of the covariance matrix $\mathbf{Q}^f_{t_i}$. To define the EnKF we will have to restate some properties. At the time t_i we will now work with a random sample

$$\mathbf{X}^f_{t_i1}, \ldots, \mathbf{X}^f_{t_iN},$$

which in earth science is usually called "ensemble" and is often perceived as a set of possible scenarios, like possible evolutions of the atmospheric states. At the time t_i we have the forecast ensemble, where each member of the ensemble is a column vector of size n and N is a number of ensemble members, $N \ll n$. Each ensemble member contains the whole model state. The analysis ensemble arises as result of application of formulas (6)–(8) on each of the members of the forecast ensemble, taking advantage of the low rank of the sample covariance matrix as a approximation to $\mathbf{Q}^f_{t_i}$. We generate random perturbations of the input data

$$\mathbf{D}_{t_ij} = \mathbf{D}_{t_i} + \mathbf{V}_{t_ij} \quad \forall j = 1, \ldots, N,$$

where $\mathbf{V}_{t_ij}$ are simulated from normal distribution independently of each other and of the forecast ensemble,

$$\mathbf{V}_{t_ij} \sim N(\mathbf{0}, \mathbf{R}_{t_i}) \quad \forall j = 1, \ldots, N.$$

These randomly perturbed data are then used to updating model state in analysis step.

Let $\bar{\mathbf{X}}^f_{t_i}$ be the mean of the forecast ensemble

$$\bar{\mathbf{X}}^f_{t_i} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{X}^f_{t_ij}$$

and $\mathbf{C}^f_{t_i}$ the sample covariance matrix

$$\mathbf{C}^f_{t_i} = \frac{1}{N} \sum_{j=1}^{N} (\mathbf{X}^f_{t_ij} - \bar{\mathbf{X}}^f_{t_i})(\mathbf{X}^f_{t_ij} - \bar{\mathbf{X}}^f_{t_i})^\top.$$

The EnKF then replaces the true forecast mean by the mean of the forecast ensemble and the true covariance by the sample covariance matrix in Kalman update formulas (6)–(8). The formulas are applied on each ensemble member.

$$\begin{aligned}
\mathbf{E}_{t_i} &= \mathbf{C}^f_{t_i}\mathbf{H}^\top_{t_i}(\mathbf{H}_{t_i}\mathbf{C}^f_{t_i}\mathbf{H}^\top_{t_i} + \mathbf{R}_{t_i})^{-1}, \\
\mathbf{X}^a_{t_ij} &= \mathbf{X}^f_{t_ij} + \mathbf{E}_{t_i}(\mathbf{D}_{t_ij} - \mathbf{H}_{t_i}\mathbf{X}^f_{t_ij}) && \text{(analysis step)}, \\
\mathbf{X}^f_{t_{i+1}j} &= \mathbf{M}_{t_i}\mathbf{X}^a_{t_ij} + \mathbf{b}_{t_i} && \text{(forecast step)}.
\end{aligned}$$

Matrix $\mathbf{E}_{t_i}$ is called sample gain Kalman matrix and it is an approximation to Kalman gain matrix (6) used in the KF. It is important to realize, that while the forecast mean and covariance matrix in the KF were deterministic, the mean and sample covariance matrix used for EnKF are random quantities.

The crucial property of the EnKF is that we don't need to store the matrices $\mathbf{C}^f_{t_i}$ in memory of the computer. Let $\mathbf{u}$ be any vector of dimension n, it easy to see that

$$\begin{aligned}
\mathbf{C}^f_{t_i}\mathbf{u} &= \left(\frac{1}{N} \sum_{j=1}^{N} (\mathbf{X}^f_{t_ij} - \bar{\mathbf{X}}^f_{t_i})(\mathbf{X}^f_{t_ij} - \bar{\mathbf{X}}^f_{t_i})^\top \right) \mathbf{u} \\
&= \frac{1}{N} \sum_{j=1}^{N} \underbrace{\left((\mathbf{X}^f_{t_ij} - \bar{\mathbf{X}}^f_{t_i})^\top \mathbf{u} \right)}_{\text{scalar product}} (\mathbf{X}^f_{t_ij} - \bar{\mathbf{X}}^f_{t_i}),
\end{aligned}$$

so we simplify the computation of $\mathbf{C}^f_{t_i}\mathbf{u}$ to computation of N scalar products of n-dimensional vectors, which is computationally feasible. Similarly we could simplify the computation of the sample Kalman gain matrix by using the equality

$$\mathbf{H}_{t_i}\mathbf{C}^f_{t_i}\mathbf{H}^\top_{t_i} = \frac{1}{N}\sum_{j=1}^{N} \underbrace{\left(\mathbf{H}_{t_i}(\mathbf{X}^f_{t_ij} - \bar{\mathbf{X}}^f_{t_i})\right)}_{m\times 1} \underbrace{\left(\mathbf{H}_{t_i}(\mathbf{X}^f_{t_ij} - \bar{\mathbf{X}}^f_{t_i})\right)^\top}_{1\times m}.$$

The rank of the sample covariance matrix $\mathbf{C}^f_{t_i}$ is at most $N-1$ and thus the the perturbation of the forecast ensemble

$$\mathbf{X}^a_{t_i1} - \mathbf{X}^f_{t_i1}, \ldots, \mathbf{X}^a_{t_iN} - \mathbf{X}^f_{t_iN}$$

is contained in the space spanned by the columns of $\mathbf{C}^f_{t_i}$ and any uncertainty outside of this subspace is simply ignored. It has been shown, for example by *Anderson* [2007], that this could cause the EnKF to diverge from the optimal solution. One of the methods, proposed to solve this problem, is called Localized Ensemble Filter and for more informations about it we recommend papers *Anderson* [2007] and *Ott* [2004].

It has been also shown that perturbations of the input data, in the presented form of the EnKF, brings artificially generated noise into the filter. There are versions of EnKF which avoid perturbations of the data. For example *Evensen* [2004] achieves this by adding another step into Kalman formula update

$$\begin{aligned}
\bar{\mathbf{X}}^a_{t_i} &= \bar{\mathbf{X}}^f_{t_i} + \mathbf{E}_{t_i}(\mathbf{D}_{t_i} - \mathbf{H}_{t_i}\bar{\mathbf{X}}^f_{t_i}) && \text{(added step)},\\
\mathbf{X}^a_{t_ij} &= \bar{\mathbf{X}}^a_{t_ij} + (\mathbf{X}^f_{t_ij} - \bar{\mathbf{X}}^f_{t_i})\tilde{\mathbf{E}}_{t_i} && \text{(analysis step)},\\
\mathbf{X}^f_{t_{i+1}j} &= \mathbf{M}_{t_i}\mathbf{X}^a_{t_i} + \mathbf{b}_{t_i} && \text{(forecast step)},
\end{aligned}$$

where matrix $\tilde{\mathbf{E}}_{t_i}$ is determined by solving the equation

$$\mathbf{C}^a_{t_i} = (\mathbf{I} - \mathbf{E}_{t_i}\mathbf{H}_{t_i})\mathbf{C}^f_{t_i},$$

where $\mathbf{C}^a_{t_i}$ is sample covariance matrix of the analysis ensemble $\mathbf{X}^a_{t_i1}, \ldots, \mathbf{X}^a_{t_iN}$. Detailed treatment of this situation is described in *Beezley* [2009].

Another large class of ensemble filters is formed by the so called square root filters which try to generate the perturbations of model states in some more efficient manner than purely random perturbations in the classical EnKF, for details see *Tippett et al.* [2003].

For a long time there was a lack of asymptotic properties of EnKF and many people were using it only by assuming that ensemble members are independent random vectors, which is not correct. This gap has been recently fulfilled mainly by papers *Le Gland, Monbet and Tran* [2009] and *Mandel, Cobb and Beezley* [2011]. The second paper is using a weak law of large numbers for exchangeable random variables (invariant under permutation) to prove that sample covariance matrix converges to true covariance matrix in probability as N goes to ∞. Also L^p bounds and convergence of ensemble members are proved. The first paper gives similar results, but the proofs are much more complicated. All convergences are meant under the assumption that m, n are fixed, or bounded, and for $N \to \infty$.

The KF and EnKF on Hilbert spaces

The motivation for extending the EnKF to general Hilbert space (complete vector space with inner product) is to get more insight into the behaviour of the filter when the dimension m of the input data grows. The asymptotic results, published in the cited papers, assumed that m and n were fixed or bounded. The question is, whether the convergence of EnKF to KF is not ruined by raising data dimension and whether is possible to find general convergence bound,

that could be applied on any Hilbert space.
First we need to define random variable on Hilbert spaces. Assume that $\mathcal{W}$ is a infinite dimensional with inner product $\langle \cdot, \cdot \rangle$, we can define a random element on $\mathcal{W}$ as a measurable function

$$X : (\Omega, \mathcal{S}, \mathrm{P}) \to (\mathcal{W}, \mathcal{B}(\mathcal{W})),$$

where $(\Omega, \mathcal{S}, \mathrm{P})$ is a probability space and $\mathcal{B}(\mathcal{W})$ are Borel sets defined on $\mathcal{W}$. It is also well known, see for example *Bosq* [2000], that mean value $\mathrm{E}[X] \in \mathcal{W}$ could be defined as a solution of equation

$$\langle u, \mathrm{E}[X] \rangle = \mathrm{E}[\langle u, X \rangle] \quad \forall u \in \mathcal{W}.$$

This solution exists for all random elements from $L^1(\Omega, \mathcal{W})$ and it is unique. Similarly covariance operator could be defined.

In the ideal situation we would just apply Kalman update formula (6)–(8) to mean and covariance operator. However it brings many questions such as

- is Bayes theorem still valid on infinite dimensional spaces? And how to define density on such spaces?
- When computing Kalman gain matrix (6), is expression $(\mathbf{H}_{t_i}\mathbf{Q}^f_{t_i}\mathbf{H}^\top_{t_i} + \mathbf{R}_{t_i})^{-1}$ well defined? Doest it exist? If yes, is it bounded?
- When using the EnKF on $\mathcal{W}$, how to define random perturbation?

Answers to these questions are still unknown and will be object of our next research.

References

Anderson J., Exploring the need for localization in ensemble data assimilation using a hierarchical ensemble filter, *Physica D. Nonlinear Phenomena*, 230(1-2), 99-111, 2007.

Beezley J., High-dimensional data assimilation and morphing ensemble Kalman filters with applications in wildfire modeling. Ph.D. Thesis, Department of Mathematical and Statistical Sciences, University of Colorado Denver, 2009. http://www.math.ucdenver.edu/ jbeezley/jbeezley_thesis.pdf.

Bosq D., Linear Processes in Function Spaces, Springer, 2000.

Evensen G., Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, *J. Geophys. Res.*, 99, 10143–10162, 1994.

Evensen G., The Ensemble Kalman Filter: theoretical formulation and practical implementation, *Ocean Dynamics*, 53, 343–367, 2003.

Evensen G., Sampling strategies and square root analysis schemes for the EnKF, *Ocean Dynamics*, 54, 539–560, 2004.

Evensen G., The Ensemble Kalman Filter 2nd ed., Springer, 2009.

Kalman R., A new approach to linear filtering and prediction problems, *Transaction of the ASME - Journal of Basic Engineering*, 82(1), 35–45, 1960.

Kalman R. and Bucy R., New results in filtering and prediction theory, *Transaction of the ASME - Journal of Basic Engineering*, 83, 95–108, 1961.

Le Gland F., Monbet V., Tran V.-D.,Large Sample Asymptotics for the Ensemble Kalman Filter, Technical report, INRIA - Université de Rennes, 2009. http://hal.inria.fr/inria-00409060/PDF/RR-7014.pdf

Mandel J., A Brief Tutorial on the Ensemble Kalman Filter, eprint arXiv:0901.3725, 2009. http://arxiv.org/abs/0901.3725

Mandel J., Cobb L., Beezley J., On the convergence of the ensemble Kalman filter, University of Colorado Denver CCM Report 278, revised 2011, http://arxiv.org/abs/0901.2951

Ott E. et al., A local ensemble Kalman filter for atmospheric data assimilation, *Tellus A*, 56, 415–428,2004.

Tippett, M. et al., Ensemble Square Root Filters, *Mon. Wea. Rev.*, 131, 1485–1490, 2003.

 ISBN 978-80-7378-184-2

On Moment Estimation Methods for Spatial Cox Processes

J. Dvořák

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. Large data sets in the form of point patterns are frequently encountered in practice and need to be analyzed, e.g. by fitting parametric models. We consider stationary spatial Cox point processes and give overview of moment estimation methods suitable for fitting this class of models to the data – the minimum contrast method and the composite likelihood and the Palm likelihood approaches. These methods represent a simulation-free faster-to-compute alternative to the computationally intense maximum likelihood estimation.

Introduction

In this paper we will consider the problem of model fitting for a certain class of point processes in $\mathbb{R}^2$. A point process can be regarded as a random element whose values are point patterns, i.e. locally finite configurations of points, see e.g. Fig. 1.

Fitting a parametric model to observed data is often important part of the statistical inference. For clustered point patterns modelled by Cox processes the classical maximum likelihood method is computationally prohibitive. The reason is the need to repetitively evaluate mean values of complicated high-dimensional integrals, which are part of the normalizing constant in the likelihood. Alternative methods of parameter estimation for Cox processes based on their second-order characteristics were proposed in the past few years.

In this article we will restrict ourselves to the case of Cox process models. We will discuss the difficulties of the maximum likelihood approach and review the alternative methods of parameter estimation which are more suitable for the Cox processes.

Background

In this section we introduce some basic notation from the theory of spatial point processes. For a detailed introduction into the topic [*Daley, Vere-Jones*, 2003] or [*Diggle*, 2003] can be consulted.

Let X be a point process in $\mathbb{R}^2$. If B is a Borel set in $\mathbb{R}^2$ we denote by $|B|$ its area and by $X(B)$ the number of points of X in B. For a given point $x \in \mathbb{R}^2$, let $\mathrm{d}x$ be an infinitesimal region containing the point x.

We define the intensity function $\lambda(x)$ of X as the occurence rate of points of X in the location x, i.e. the limit

$$\lambda(x) = \lim_{|\mathrm{d}x| \to 0} \frac{\mathbb{E}\left[X(\mathrm{d}x)\right]}{|\mathrm{d}x|},$$

where the area of $\mathrm{d}x$ approaches 0.

Heuristic interpretation is that the value $\lambda(x)$ determines the probability that a point of X is observed in an infinitesimally small region $\mathrm{d}x$. More formally: $\lambda(x)|\mathrm{d}x| \approx \mathbb{P}(X(\mathrm{d}x) = 1)$.

A Poisson point process is characterized by its intensity function $\lambda(x)$ and the following properties:

- for every Borel set $B \subset \mathbb{R}^2$ it holds that $X(B)$ is a random variable with Poisson distribution with mean value equal to $\int_B \lambda(u)\mathrm{d}u$,
- for every $k = 2, 3, \ldots$, if $B_1, \ldots B_k \subset \mathbb{R}^2$ are pairwise disjoint Borel sets then the random variables $X(B_1), \ldots, X(B_k)$ are independent.

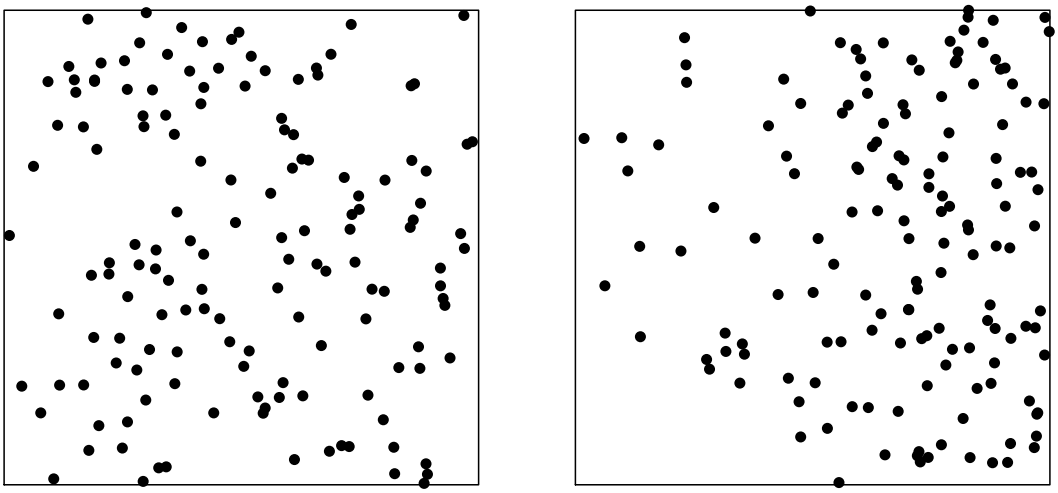


Figure 1. Sample realizations of Poisson point processes in window $W = [0,1]^2$. Left: homogeneous Poisson process with intensity function $\lambda((x_1, x_2)) = 150$, right: inhomogeneous Poisson process with spatially varying intensity function $\lambda((x_1, x_2)) = 300x_1$.

Fig. 1 shows two examples of processes with constant and spatially varying intensity function. They are realizations of a homogeneous Poisson process ($\lambda(x) = \lambda$ for all $x \in \mathbb{R}^2$) and an inhomogeneous Poisson process ($\lambda(\cdot)$ is non-constant function of the location).

A more complicated class of models are Cox processes. A Cox process is a doubly stochastic process – it is driven by a random conditional intensity function $\Lambda(\cdot)$ and given the realization $\Lambda = \lambda$, it is an inhomogeneous Poisson process with intensity function equal to λ.

Different types of clustering can be modeled by Cox processes (see Fig. 2). Examples of Cox processes include:

- Log-Gaussian Cox process – the random conditional intensity function is defined as $\Lambda(x) = \exp\{Y(x)\}$, where $\{Y(x), x \in \mathbb{R}^2\}$ is a Gaussian random field.
- Thomas process – first a Poisson process of parent points is generated and then each parent point is replaced by a cluster of offspring points. The number of offsprings is Poisson distributed and the locations of the offspring points of one parent are independent with bivariate isotropic normal distribution centered at the parent point.

Now let's assume that X is a stationary process, i.e. its distribution is translation-invariant. We shall define some important second-order properties of the process X describing the inter-point interactions.

The second-order intensity function of the process X is defined by

$$\lambda_2(x, y) = \lim_{|\mathrm{d}x|,|\mathrm{d}y|\to 0} \frac{\mathbb{E}\left[X(\mathrm{d}x)X(\mathrm{d}y)\right]}{|\mathrm{d}x||\mathrm{d}y|}.$$

Let us assume that the process X is simple, i.e. two points of X cannot occur at the same location. Then, for $x \neq y$, $\lambda_2(x, y)|\mathrm{d}x||\mathrm{d}y|$ can be regarded as the approximate probability that $\mathrm{d}x$ and $\mathrm{d}y$ each contains a point of X. From the assumption of stationarity of X we see that $\lambda_2(x, y) = \lambda_2(0, y - x) = \lambda_2(y - x)$.

The second-order reduced intensity function λ_0 of the process X is defined indirectly by the decomposition $\lambda_2(x, y) = \lambda\lambda_0(y - x)$. Its heuristic interpretation is the occurence rate of a point of X at a location x provided that another point of the process X is at the origin o, i.e.

$$\lambda_0(x)|\mathrm{d}x| \approx \mathbb{P}\left(X(\mathrm{d}x) = 1 | X(\{o\}) = 1\right).$$

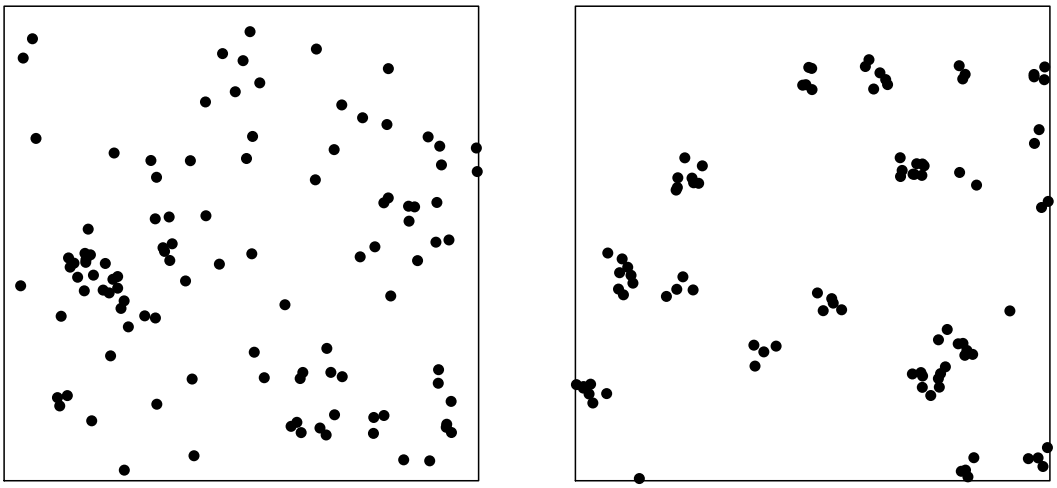


Figure 2. Sample realizations of Cox point processes in window $W = [0,1]^2$ showing different types of clustering. Left: log-Gaussian Cox process, right: Thomas process.

In the literature λ_0 is also called the conditional intensity or Palm intensity since it is in fact the intensity function of the Palm distribution of the process X.

Two popular point process characteristics can be defined using λ_0. They are the pair correlation function

$$g(x,y) = g(y-x) = \lambda_2(y-x)/\lambda^2 = \lambda_0(y-x)/\lambda,$$

and the K-function defined for $r > 0$ by

$$\lambda K(r) = \int_{B(o,r)} \lambda_0(u)\mathrm{d}u = \mathbb{E}\left[X(B(o,r)\backslash\{o\})|X \cap \{o\} \neq \emptyset\right],$$

where $B(o,r)$ is the ball centered at the origin with radius r. Note that $\lambda K(r)$ can be interpreted as the mean number of further points from X in $B(x,r)$ centered at a typical point x of the process X.

Model fitting for Cox processes

Assume that we have observed a realization $\mathbf{x} = (x_1, \ldots, x_n)$ of the point process X in a compact observation window W, where $x_i, i = 1, \ldots, n$, denote the locations of the observed points. We will assume a parametric model for X and the vector of unknown parameters will be denoted by θ.

Let us first discuss the classical maximum likelihood estimation. To use this method we will consider the density $f_\theta(\mathbf{x})$ of X with respect to a stationary unit-rate Poisson process (i.e. $\lambda(u) = 1$). The maximum likelihood estimate $\hat{\theta}$ is then obtained as the value of θ maximizing $f_\theta(X)$.

For a Poisson proces with the intensity function $\lambda_\theta(x)$ the density is (see e.g. [*Møller, Waagepetersen*, 2007])

$$f_\theta(\mathbf{x}) = \exp\left\{|W| - \int_W \lambda_\theta(u)\mathrm{d}u\right\} \prod_{i=1}^{n} \lambda_\theta(x_i).$$

As long as a suitable parametrization of λ_θ is available the density f_θ has a tractable form and the estimate $\hat{\theta}$ can be easily obtained.

For a Cox process driven by random conditional intensity function $\Lambda_\theta = (\Lambda_\theta(u))_{u\in W}$ the density is (see e.g. [*Møller, Waagepetersen*, 2007])

$$f_\theta(\mathbf{x}) = \mathbb{E}_\theta \left[\exp\left\{ |W| - \int_W \Lambda_\theta(u)du \right\} \prod_{i=1}^{n} \Lambda_\theta(x_i) \right].$$

In order to obtain the maximum likelihood estimate of θ it is needed to repetitively evaluate the expectation including a complicated integral term with respect to possible values of the random conditional intensity Λ_θ. One can of course take advantage of MCMC or other techniques and use approximations of the likelihood function $f_\theta(\mathbf{x})$, see e.g. [*Møller, Waagepetersen*, 2003]. This approach is usually computationally very demanding and thus faster, simulation-free alternatives were sought.

Three such methods will be reviewed in the following subsections. They are in fact moment estimation methods because they are based on the second-order moment characteristics of the process X.

In the following we will assume that X is a stationary Cox point process with second-order intensity function $\lambda_2(\cdot;\theta)$ – and also the pair correlation function and the K-function derived from it – in a closed form with respect to the vector of unknown parameters θ.

Minimum contrast method

This method was in the context of spatial statistics first introduced in [*Diggle*, 1983]. It can be based either on the K-function or the pair correlation function g, see e.g. [*Diggle*, 2003]. In the version using the g-function this method requires isotropy of the process X in addition to its stationarity. In this case the g-function is a function of a scalar argument.

The vector of parameters θ is estimated by minimizing the discrepancy measure

$$\int_r^R [\hat{K}^c(u) - K^c(u;\theta)]^2 \mathrm{d}u \qquad \text{or} \qquad \int_r^R [\hat{g}^c(u) - g^c(u;\theta)]^2 \mathrm{d}u$$

between the nonparametric estimate of $\hat{K}$ or $\hat{g}$ and its theoretical value $K(\cdot;\theta)$ or $g(\cdot;\theta)$, respectively.

The constants c, r and R are used to control the sampling fluctuations in the estimates of K and g. In [*Diggle*, 2003] it is recommended that for fitting aggregated point patterns using the K-function the constant $c = 0.25$ is used and that for data on a unit square R should be no larger than 0.25. The remaining constant r can be set to zero or a small positive value, e.g. the minimum observed interpoint distance. For the pair correlation function g there is no standard recommendation available.

The asymptotic properties of the minimum contrast method are discussed in [*Heinrich*, 1992].

Composite likelihood method

Composite likelihood approach is based on adding together individual log-likelihoods for single points or pairs of points of the process X to form a composite log-likelihood. We consider the version suggested in [*Guan*, 2006] which uses the second-order intensity function λ_2 to obtain the density for two points of X occuring at locations x and y:

$$h(x,y;\theta) = \frac{\lambda_2(x-y;\theta)}{\int_W \int_W \lambda_2(u-v;\theta)\mathrm{d}u\mathrm{d}v}.$$

We are interested only in those pairs of points which are not separated by a large distance. This is motivated by the fact that distant pairs of points are often nearly independent and they do not carry much information about the parameters, but increase the variability of the

estimator. Moreover, computational demands can be greatly reduced by disregarding the distant pairs of points.

This reasoning leads to the following form of the composite log-likelihood:

$$\log CL(\theta) = \sum_{x \neq y \in X \cap W, |x-y|<R} \log \frac{\lambda_2(x-y;\theta)}{\int_W \int_W I(|u-v|<R)\lambda_2(u-v;\theta)\mathrm{d}u\mathrm{d}v},$$

where only pairs of points within distance R are considered and I denotes the indicator function.

Applying the inner region correction leads us to a somewhat simplified formula

$$\log CL(\theta) = \sum_{x \in X \cap (W \ominus R), y \in X \cap W, 0<|x-y|<R} \log \frac{\lambda_2(x-y;\theta)}{\lambda^2 |W \ominus R| K(R)},$$

where $W \ominus R = \{w \in W : B(w,R) \subset W\}$, i.e. $W \ominus R$ is the window W eroded by distance R. The vector $\hat{\theta}$ maximizing $\log CL(\theta)$ is then taken for the estimate of θ.

Asymptotic properties of the composite likelihood estimates are discussed in [*Guan*, 2006]. Other versions of the composite likelihood method can be found in [*Guan*, 2006; *Møller, Waagepetersen*, 2007].

Palm likelihood method

Consider the processes $Y_x, x \in X$, of differences of the points of the process X:

$$Y_x = \{y - x, x \neq y \in X \cap W\}, x \in X.$$

Following the reasoning in [*Tanaka et al.*, 2007; *Prokešová, Jensen*, 2010] each Y_x can be approximated by an inhomogeneous Poisson process with (first-order) intensity function equal to $\lambda_0(\cdot;\theta)$. The approximation is in the Poisson distribution of the process, not in the intensity function, which is exactly $\lambda_0(\cdot;\theta)$.

Treating $Y_x, x \in X$, as independent, identically distributed replications, we can form a superposition $Y = \bigcup_{x \in X} Y_x$ and approximate it as an inhomogeneous Poisson process with the intensity function (exactly equal to) $X(W)\lambda_0(\cdot;\theta)$.

We make the inference about the process X through the properties of the process of differences Y. Forming a Poisson log-likelihood of Y (and calling it the Palm log-likelihood of X, because it is expressed in terms of the Palm intensity function $\lambda_0(\cdot;\theta)$ of the process X) we get

$$\log PL(\theta) = \sum_{x \neq y \in X \cap W, |x-y|<R} \log(X(W)\lambda_0(x-y;\theta)) - X(W) \int_{\mathbb{R}^2} I(|u|<R)\lambda_0(u;\theta)\mathrm{d}u.$$

Here, we are again interested only in pairs of points within a predefined distance R. Again, the maximizer of $\log PL(\theta)$ is taken for the estimate of θ.

Applying the inner region correction and dropping the terms which do not depend on θ we get the following expression for the Palm log-likelihood:

$$\log PL(\theta) = \sum_{x \in X \cap (W \ominus R), y \in X \cap W, 0<|x-y|<R} \log \lambda_0(x-y;\theta) - X(W \ominus R)\lambda K(R).$$

Asymptotic properties of the Palm likelihood method are discussed in [*Prokešová, Jensen*, 2010].

Research plans

No direct comparison of the moment estimation methods for stationary spatial Cox processes is available in the literature. The formulae for the asymptotic variance of these estimators are very complicated and depend on the fourth-order moment measures of the processes in question. Thus, direct comparison of the efficiency of these estimators is not possible.

The author's current research concentrates on assessment of empirical properties of these methods and comparison of their performance through a simulation study. Different types of Cox process models including the Thomas process and the log-Gaussian Cox process are considered to assess the influence of different types of clustering.

Research plans for the next years are aimed to estimation methods for non-stationary point processes.

Acknowledgments. The author would like to express his thanks to RNDr. Michaela Prokešová, PhD. for her generous support and invaluable comments. The work was supported by the grants GAČR P201/10/0472 and SVV 261315/2011.

References

Daley, D. J. and Vere-Jones, D., An Introduction to the Theory of Point Processes. Volume 1: Elementary Theory and Methods, 2nd ed., New York: Springer-Verlag, 2003.

Diggle, P. J., Statistical Analysis of Spatial Point Patterns, London: Academic Press, 1983.

Diggle, P. J., Statistical Analysis of Spatial Point Patterns, 2nd ed., New York: Oxford University Press, 2003.

Guan, Y., A composite likelihood approach in fitting spatial point process models, *J. Amer. Statist. Assoc.*, 101, 1502 – 1512, 2006.

Heinrich, L., Minimum contrast estimates for parameters of spatial ergodic point processes, in: *Transactions of the 11th Prague Conference on Random Processes, Information Theory and Statistical Decision Functions*, Prague: Academic Publishing House, 1992.

Møller, J. and Waagepetersen, R. P., Statistical Inference and Simulation for Spatial Point Processes, Florida: Chapman and Hall/CRC, 2003.

Møller, J. and Waagepetersen, R. P., Modern statistics for spatial point processes, *Scand. J. Statist.*, 34, 643 – 684, 2007.

Prokešová, M. and Jensen, E. B. V., Asymptotic Palm likelihood theory for stationary spatial point processes, submitted, 2010.

Tanaka, U., Ogata, Y. and Stoyan, D., Parameter estimation and model selection for Neyman-Scott point processes, *Biom. J.*, 49, 1 – 15, 2007.

Dependencies in Stochastic Geometry—A Simulation Study

O. Šedivý

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. Dependency of a random set on some accompanying spatial covariates can be studied by the means of inverse regression. This attitude is well established in classical statistics, where the explained variable has numerical response. The basic idea of inverse regression is to exchange the roles of explained variable Y and vector of explanatory variables X. Focusing on the properties of X as Y varies has an immediate advantage in reducing the dimensionality problem - it can be carried out by regressing each coordinate of X against Y. Recently, several authors have shown that inverse regression can also be applied in spatial statistics. The response is considered as binary with value 1 in the points of the process and 0 elsewhere. The background of sliced inverse regression with applications in stochastic models dependent on a multidimensional Gaussian random field is presented in the paper. In particular, Voronoi tessellation is repeatedly generated by Poisson point process with intensity being a function of the random field. Then the point, fibre or surface process of the tessellation generators, edges or faces, respectively, is of our interest.

Introduction

In classical regression setting, an explained variable Y is a function of vector of explanatory variables $\mathbf{X} = (X_1, \ldots, X_p)$. We suppose that this dependency can be expressed through linear combinations of $X_1, \ldots, X_p$, which means that Y is independent of $\mathbf{X}$ given $B^T\mathbf{X}$ for a matrix $B_{p\times c}$, $c \leq p$, formally written by relation

$$Y \perp\!\!\!\perp \mathbf{X} \,|\, B^T\mathbf{X}. \tag{1}$$

Given this model, $B^T\mathbf{X}$ captures all the information of $\mathbf{X}$ about Y. The linear space $\mathcal{S}(B)$ spanned by columns of B is called a *sufficient dimension reduction subspace*. If the intersection of all such subspaces is also a sufficient dimension reduction subspace, it is called a *central subspace*. The main goal of classical regression models is to estimate the matrix B with a specific form of the dependency.

In [*Li*, 1991], a different attitude to the dimension reduction problem was introduced. Instead of regressing Y on $\mathbf{X}$, a so called *inverse regression curve* $\mathbb{E}(\mathbf{X} \,|\, Y)$ is examined as Y varies. This conditional expectation can be estimated using the basic idea of slicing. The range of Y is divided into several slices and the sample mean of explanatory variables is calculated in each of them. Under the model (1) and the linearity condition introduced later, the inverse regression curve lies in a c-dimensional affine subspace of R^p. To locate its main orientation, a weighted principal component analysis can be conducted.

A way of application of sliced inverse regression (SIR) in spatial point processes was suggested in [*Guan*, 2008]. Although any numerical response values are missing, we can consider a set of binary observations with value 1 in the points of the process and 0 elsewhere which naturally consists of two slices. In fact, only the slice with the response equal to 1 is used; it stands for the realization of the point process.

The present paper gives a generalization of SIR to fibre or surface process and extension of this idea to slicing according to some geometrical quantities.

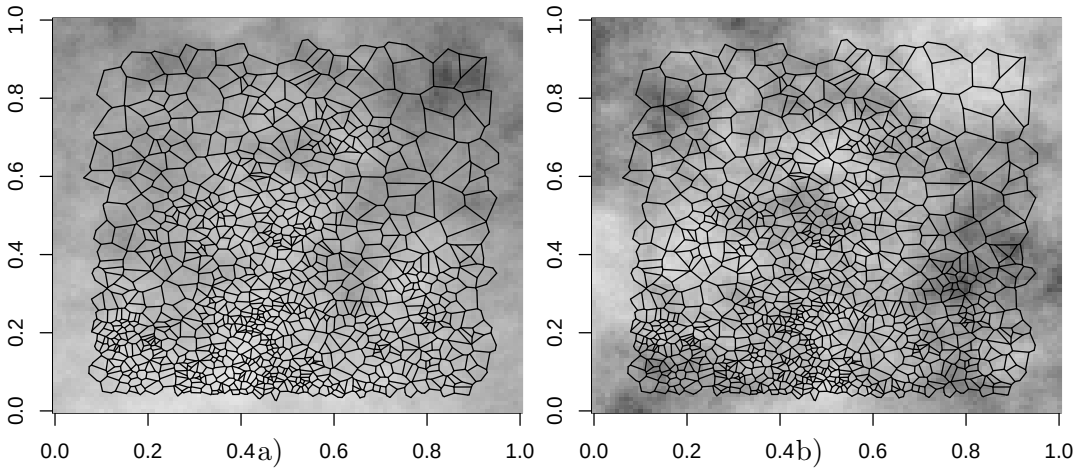


Figure 1. Simulation of a Gaussian random field $\mathbf{X}$ with a Poisson-Voronoi tessellation a) depending on the random field, b) independent of the random field. In figure a) higher intensity of the tessellation cells is markable in lighter areas (grey level tends to white - higher values of $\mathbf{X}$).

Theoretical background

Let $(\Omega, \mathcal{A}, P)$ be a probability space. Denote M^d the set of all locally finite measures on $(\mathbb{R}^d, \mathcal{B}^d)$ equipped by σ-algebra

$$\mathcal{M}^d = \sigma\{\mu \in M^d;\ \mu \mapsto \mu(A) \text{ measurable } \forall A \in \mathcal{B}^d, A \text{ bounded}\}.$$

Then

$$\Psi : (\Omega, \mathcal{A}, P) \to (M^d, \mathcal{M}^d)$$

is a random measure on $\mathbb{R}^d$ with intensity measure $\Lambda(\cdot) = \mathbb{E}\Psi(\cdot)$. Let $\mathcal{H}^k$ be the Hausdorff measure of order k in $\mathbb{R}^d$. The concept of random $\mathcal{H}^k$-sets in $\mathbb{R}^d$ as random closed sets which are $\mathcal{H}^k$-rectifiable was introduced in [*Zähle*, 1982]. A random $\mathcal{H}^k$-set Y such that $\Psi_Y(.) = \mathcal{H}^k(Y \cap .)$ is a locally finite measure in $\mathbb{R}^d$, will be called a point, fibre, surface process for $k = 0, 1, d-1$, respectively.

In our setting, let $Y \subset \mathbb{R}^d$ be a random point, fibre or surface process dependent on $\mathbf{X}$ which is a p-dimensional Gaussian random field in $\mathbb{R}^d$, $d = 2, 3$, according to the model (1). The aim is to estimate the central subspace $\mathcal{S}_{Y|\mathbf{X}}$ from data and, possibly, to test its dimension. In the case of a point process Y with intensity λ realized in an observation window W, the statistics

$$\tilde{B}_{SIR} = \frac{1}{\Psi_Y(W)} \sum_{s \in Y \cap W} \mathbf{X}(s),$$

was proposed in [*Guan et al.*, 2010]. It is simply the sample mean of the random field in all the points of the process in the observation window W. It is there shown that providing ergodicity of both, Y and $\{\mathbf{X}(s) : s \in Y\}$, $\tilde{B}_{SIR}$ converges in probability to

$$B_{SIR} = \left[\int \mathbb{E}\left\{\lambda(s)\mathbf{X}(s)\right\} ds\right] \left[\int \mathbb{E}\left\{\lambda(s)\right\} ds\right]^{-1}.$$

The following condition is sufficient to formulate the key result which justifies using SIR in this framework.

Linearity condition *For all points $s \in \mathbb{R}^d$ and any constant vectors $b_1, b_2 \in \mathbb{R}^p$ there exist constants $a_0, a_1 \in \mathbb{R}$ such that $\mathbb{E}\,(b_1^T\mathbf{X}(s) \,|\, b_2^T\mathbf{X}(s)) = a_0 + a_1 b_2^T\mathbf{X}(s)$.*

It is satisfied if the distribution of $\mathbf{X}(s)$ is normal or elliptically symmetric.

Theorem [*Guan*, 2008] *Under the model (1) and the linearity condition, $\mathcal{S}(B_{SIR}) \subseteq \mathcal{S}_{Y|\,\mathbf{X}}$.*

The theorem says that the directions estimated by SIR lie in the central subspace.

The same idea introduced for point processes can be applied to $\mathcal{H}^k$-sets in general. In the sense of sliced inverse regression the following slicing mechanism can be provided. Consider a marked point, fibre or surface process (Y, Γ) where the real marks Γ represent some geometrical quantity (nearest neighbour distance of points, lengths of edges etc.). Partitioning the range of Γ into H disjoint intervals (slices) induces partition of Y into subsets $Y = Y_1 \cup \ldots \cup Y_H$. SIR is then based on the conditional expectation

$$m_j = \mathbb{E}(\mathbf{X}(s) \,|\, s \in Y_j) = \frac{1}{\mathcal{H}^k(Y_j)} \int_{Y_j} \mathbf{X}(s)\mathcal{H}^k(ds)$$

in each slice $j = 1, \ldots, H$ which can be estimated using sample mean in the set of random test points. This leads to the following algorithm.

Simulation and SIR evaluation

1. Simulate a p-dimensional Gaussian random field $\tilde{\mathbf{X}} = (\tilde{X}_1(s), \ldots, \tilde{X}_p(s)), s \in [0,1]^2$ or $[0,1]^3$ (unit square or cube), with independent elements $\tilde{X}_1, \ldots, \tilde{X}_p$, each having covariance function $C(s,t) = \exp(-||s-t||)$.

2. Simulate a 2- or 3-dimensional inhomogeneous Poisson point process Φ with intensity

$$\lambda(s) \quad = \quad a \exp\{-\tilde{X}_1(s)\} \tag{2}$$

   where a is a constant so that the expected number of points is equal to 1000 in $[0,1]^2$ or 10000 in $[0,1]^3$, respectively.

3. Build up a Poisson-Voronoi tessellation Ξ generated by Φ with an edge effects correction[1].

4. Compute the standardized version $\mathbf{X} = \hat{\Sigma}^{-\frac{1}{2}}(\tilde{\mathbf{X}} - \bar{\tilde{\mathbf{X}}})$ of Gaussian random field $\tilde{\mathbf{X}}$, where $\bar{\tilde{\mathbf{X}}}$ is the sample mean and $\hat{\Sigma}$ is the sample covariance matrix of $\tilde{\mathbf{X}}$.

5. Choose a set of random test points[2] $Z = \{z_1, \ldots, z_n\}, Z \subset \Xi$. Divide Z into H subsets (slices) $Z_1, \ldots, Z_H$ according to the range of the following geometrical quantities:

   - in $[0,1]^2$, lengths of the corresponding tessellation edges
   - in $[0,1]^3$, areas of the corresponding tessellation faces

6. Express the weighted covariance matrix

$$V = \sum_{j=1}^{H} \hat{p}_j \hat{m}_j \hat{m}_j^T,$$

   where $\hat{m}_j = \frac{1}{card(Z_j)} \sum_{z \in Z_j} \mathbf{X}(z)$ and $\hat{p}_j = \frac{card(Z_j)}{card(Z)}$.

[1] Edge effects correction omits cells close the border of the observation window which could be influented by the points outside of the window.

[2] Random test points can be achieved from independent realizations. To avoid a strong dependency when sampling from a single large realization, at first a random set of tessellation edges/faces is chosen proportionally to their lengths/areas and then only a midpoint is selected in each of them.

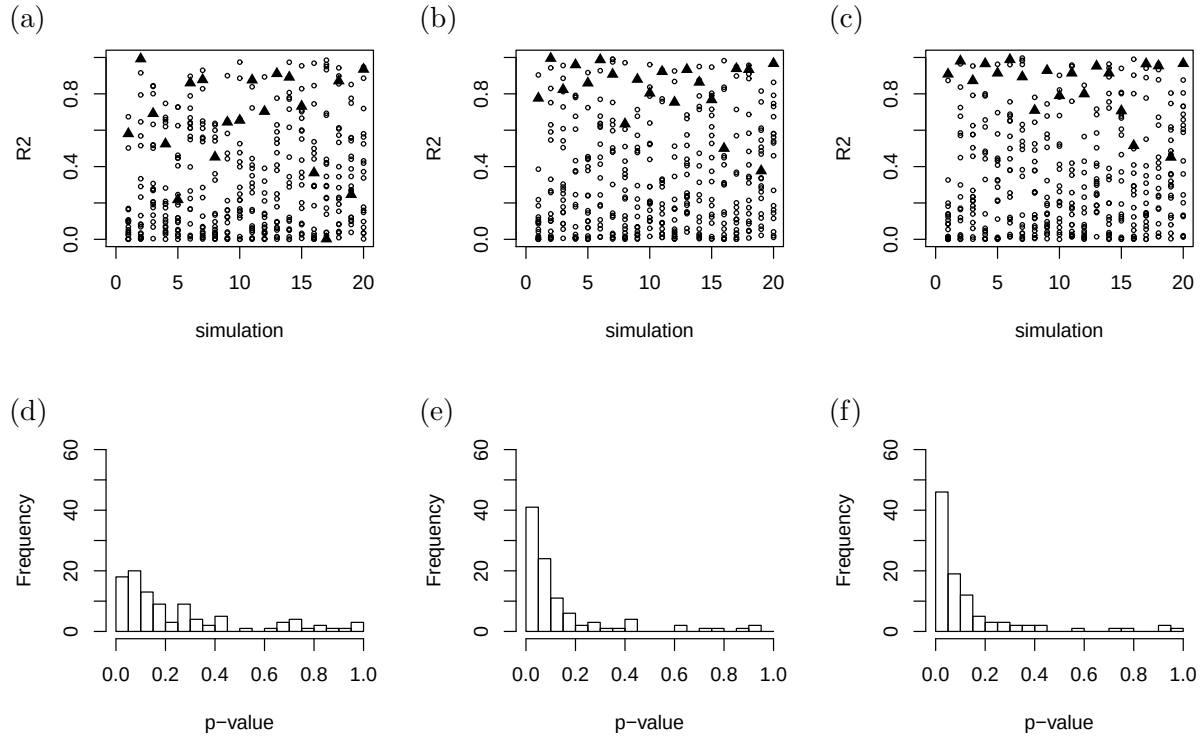


Figure 2. Results for tessellation edges in $[0,1]^2$. Upper row: The coefficients R^2 from each simulation when the number of slices is 1 (a), 2 (b) and 4 (c). Filled triangle stands for the dependent case, circles for the independent cases. Based on 20 simulations $S_1, \ldots, S_{20}$. Lower row: Histograms of p-values for H_0 when the number of slices is 1 (d), 2 (e) and 4 (f), based on 100 simulations $S_1, \ldots, S_{100}$.

7. Evaluate c largest eigenvalues $\lambda_1, \ldots, \lambda_c$ and the corresponding eigenvector $\eta_1, \ldots, \eta_c$ of V. The number c is deterministic or it can be a subject of further sequential testing, see [*Guan*, 2008]. In the application mentioned further, $c = 1$.

8. Finally obtain the column vectors $\hat{\beta}_i = \Sigma^{-\frac{1}{2}}\eta_i, i = 1, \ldots, c$, of the matrix $\hat{B}_{SIR}$. This retransformation is necessary to fit the output vectors to the original random field $\tilde{\mathbf{X}}$. For details see [*Cook*, 1998].

The degree of fit between the estimated vector $\hat{\beta}$ and the theoretical vector β is evaluated using the squared correlation coefficient

$$R^2(\hat{\beta}) = \frac{(\beta\hat{\beta}^T)^2}{(\beta\beta^T)(\hat{\beta}\hat{\beta}^T)}. \tag{3}$$

Results

This section presents the simulation results for the fibre process of tessellation edges in $[0,1]^2$ (figure 2) and the surface process of tessellation faces in $[0,1]^3$ (figure 3).

In both cases the total number of 100 simulations $S_1, \ldots, S_{100}$ were realized, each of them consisting of twenty 3-dimensional random fields

$$\mathbf{X}^{(1)} = (X_1^{(1)}, X_2^{(1)}, X_3^{(1)}) \quad \ldots \quad \mathbf{X}^{(20)} = (X_1^{(20)}, X_2^{(20)}, X_3^{(20)}).$$

and one Voronoi tessellation depending just on $\mathbf{X}^{(1)}$. Intensity of its generating point process is given by (2) with the first element of $\mathbf{X}^{(1)}$, while it is independent of the next 19 random fields. Each vertical line of marks in figures 2a), 2b), 2c) and 3a), 3b), 3c) stands for the

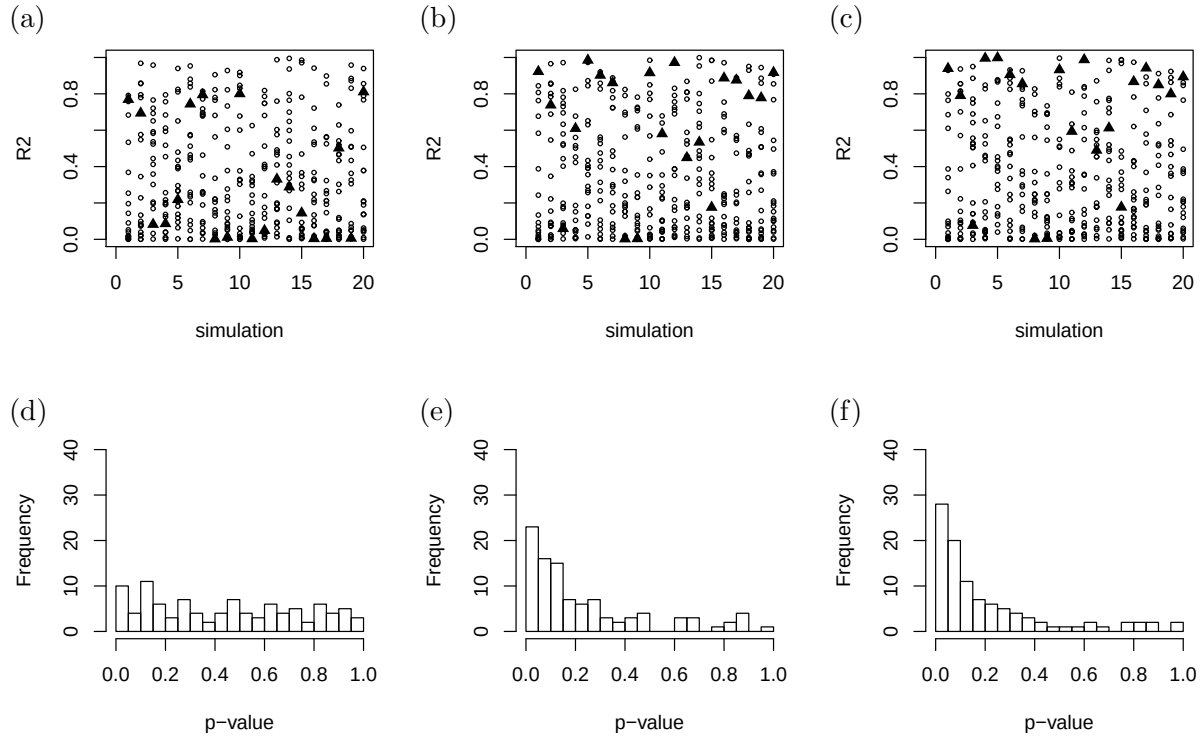


Figure 3. Results for tessellation faces in $[0,1]^3$. Upper row: The coefficients R^2 from each simulation when the number of slices is 1 (a), 2 (b) and 4 (c). Filled triangle stands for the dependent case, circles for the independent cases. Based on 20 simulations $S_1, \ldots, S_{20}$. Lower row: Histograms of p-values for H_0 when the number of slices is 1 (d), 2 (e) and 4 (f), based on 100 simulations $S_1, \ldots, S_{100}$.

coefficients R^2 (3) obtained from the application of SIR to the data $(Y, \mathbf{X}^{(1)}), \ldots, (Y, \mathbf{X}^{(20)})$, consecutively. For better illustration, just the first 20 simulations $S_1, \ldots, S_{20}$ are figured. Filled triangle representing the dependent case $(Y, \mathbf{X}^{(1)})$ is expected to have the highest value of R^2. Both figures reveal considerable refinement with increasing number of slices.

Repeating the simulation with independent realizations of the random field allows to evaluate the p-value of the test of orthogonality hypothesis $H_0 : R^2 = 0$ in a following way

$$\text{p-value} = \frac{1}{n} card\{R_i^2 \geq R_1^2\},$$

where R_i^2 is the statistics obtained from the data $(Y, \mathbf{X}^{(i)})$ and n is the number of realizations, here $n = 20$. The lower rows d), e), f) in both figures display histograms of p-values obtained from 100 simulations. With increasing number of slices most of the p-values are concentrated close to zero.

Conclusion

Application of inverse regression techniques in stochastic geometry is possible, although any primary numeric response is missing. The article presents the way of using the simplest of these methods, sliced inverse regression, in a particular stochastic model. Slicing is performed according to suitable geometrical properties. Improvement of the results with increasing number of slices is well remarkable. However, due to its simplicity, SIR has several limitations. For instance, it works well for monotonic trends but cannot estimate highly symmetric relationships. Other similar methods which are considered to be more powerful have already been developed. Their extension into stochastic models is being examined.

References

Cook, R. D., Regression Graphics: Ideas for Studying Regressions Through Graphics, Wiley, 1998.

Li, K.-Ch., Sliced Inverse Regression for Dimension Reduction, *J. Am. Stat. Assoc.*, Vol. 86, No. 414, 316-327, 1991.

Guan, Y., On Consistent Nonparametric Intensity Estimation for Inhomogeneous Spatial Point Processes, *J. Am. Stat. Assoc.*, Vol. 103, No. 483, 1238-1246, 2008.

Guan, Y., H. Wang, Sufficient Dimension Reduction for Spatial Point Processes Directed by Gaussian Random Fields, *J. R. Statist. Soc. B*, Vol. 72, Part 3, 367-387, 2010.

Zähle, M., Random Processes of Hausdorff Rectifiable Closed Sets., *Math. Nachr.*, Vol. 108, 49-72, 1982.

 ISBN 978-80-7378-184-2

How to Construct Borel Measurable PLIFs?

P. Kříž

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. The concept of the Probability Limit Identification Function (PLIF) will be introduced with the presentation of known results concerning existence and measurability of the PLIF. Ongoing research on the construction of the Borel measurable PLIFs for special sets of measures will be discussed.

Motivation

Motivation for the probability limit identification function (PLIF) came from the estimation theory. Let us have a measurable space $(\Omega, \mathcal{A})$ with collection of probabilities $(P_\theta; \theta \in \Theta)$. Further assume some parametric function $g(\theta)$ the value of which is to be estimated by the estimators $E_n = e_n(X_1, \ldots, X_n)$, where X_i are observations and e_n are measurable functions. In statistics, consistent estimators play important role, because they converge to the unknown parameter (or parametric function) with increasing number of observations. Estimation theory distinguishes weakly and strongly consistent estimators. Recall that the sequence of estimators is weakly consistent, if it converges in probability to the unknown parametric function, i.e.

$$P_\theta[|E_n - g(\theta)| > \epsilon] \overset{n\to\infty}{\longrightarrow} 0 \quad \forall \epsilon > 0, \forall \theta \in \Theta,$$

and the sequence of estimators is strongly consistent, if it converges almost surely to the unknown parametric function, i.e.

$$P_\theta[E_n \overset{n\to\infty}{\longrightarrow} g(\theta)] = 1 \quad \forall \theta \in \Theta.$$

The advantage of strongly consistent estimators is the fact, that the value of the parametric function $g(\theta)$ can be easily identified on the basis of the observations of estimators E_n simply by taking their limit. However, in case of weakly consistent estimators the parametric function $g(\theta)$ may not be the limit of the observations of estimators E_n as this limit may not exist with positive probability. Identification of parametric function on the basis of the observations of weakly consistent estimators is more complicated and it is provided by the probability limit identification function (to be correctly defined below), i.e. the function

$$f : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}$$

with the property:

$$P_\theta\left\{\omega \in \Omega : f\Big(E_1(\omega), E_2(\omega), \ldots\Big) = g(\theta)\right\} = 1 \quad \forall \theta \in \Theta.$$

In this context, the PLIF is the function, which almost surely identifies the value of the probability limit of a random sequence converging in probability on the basis of the values of its coordinates. The probability limit identification function thus identifies unknown parameters from weakly consistent estimators.

In the following section, we will define PLIF more correctly and introduce special PLIF for 0-1 valued sequences (SPLIF). We will also summarize known results concerning PLIFs. In the next section, we will present the way how to construct Borel measurable SPLIFs for convergent sequences, distributions of which are absolutely continuous with respect to a specific measure. In the last section, we will discuss further possibilities of constructing Borel measurable SPLIFs.

Definition

We shall now present the notation used in this article. The symbol $\mathbb{R}$ will stand for the real line. The set of all real-valued infinite sequences will be denoted as $\mathbb{R}^{\mathbb{N}}$ and the set of 0-1 valued sequences will be refered to as $\{0,1\}^{\mathbb{N}}$. If not specified, we will be assuming, that $\mathbb{R}^{\mathbb{N}}$ (resp. $\{0,1\}^{\mathbb{N}}$) is equipped with the product topology created from standard Euclidean (resp. discrete) topologies. Having topological spaces T, S, the symbol $\mathcal{P}(T)$ will represent the set of all Borel probability measures on T and $\mathcal{B}(T,S)$ will denote the set of all Borel mappings from T to S. The projection mapping to n-th coordinate will be labeled π_n, i.e. $\pi_n(x_1, x_2, \dots) = x_n$. We can now constitute $\mathcal{E}$ - the set of probability distributions of real-valued random sequnces converging in probability. More rigorously

$$\mathcal{E} = \{\mu \in \mathcal{P}(\mathbb{R}^{\mathbb{N}}) : \exists p_\mu \in \mathcal{B}(\mathbb{R}^{\mathbb{N}}, \mathbb{R}), \pi_n \xrightarrow{\mu} p_\mu \text{ as } n \to \infty\},$$

where symbol $\xrightarrow{\mu}$ stands for convergence in measure μ. Analogically, we may define $\mathcal{E}^*$ as the set of distributions of 0-1 valued sequences converging in probability to 0 or 1, i.e.

$$\mathcal{E}^* = \{\mu \in \mathcal{P}(\{0,1\}^{\mathbb{N}}) : \exists p_\mu \in \{0,1\}, \pi_n \xrightarrow{\mu} p_\mu \text{ as } n \to \infty\}.$$

Let us define the probability limit identification function more precisely. It will be useful to define it in terms of measures.

Definition 1. Function $f : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}$ is the probability limit identification function (**PLIF**) on $\mathcal{F} \subset \mathcal{E}$, if the following holds for all $\mu \in \mathcal{F}$

$$\{x \in \mathbb{R}^{\mathbb{N}} : f(x) \neq p_\mu(x)\} \quad \text{is a } \mu\text{-null set,}$$

where a μ-null set is a set contained in a Borel set with μ-mass equal to zero. In case of measurable f, the above property reads as follows

$$\mu\{x \in \mathbb{R}^{\mathbb{N}} : f(x) = p_\mu(x)\} = 1 \quad \forall \mu \in \mathcal{F}.$$

For more details on the convergence in probability of measures and definition of PLIF in the terms of measures, see [*Štěpán*, 1973].

The PLIF identifies probability limits of real-valued random sequences. It is useful to introduce its analogue for 0-1 valued random sequences converging in probability to a constant 0 or 1, the so called special probability limit identification function (SPLIF).

Definition 2. We shall call the function $f : \{0,1\}^{\mathbb{N}} \to \{0,1\}$ the special probability limit identification function (**SPLIF**) on $\mathcal{F}^* \subset \mathcal{E}^*$, if for all $\mu \in \mathcal{F}^*$

$$\{x \in \{0,1\}^{\mathbb{N}} : f(x) \neq p_\mu\} \quad \text{is a } \mu\text{-null set,}$$

i.e. it is contained in a Borel set with μ-mass equal to zero. If f is measurable, we can rewrite the above condition as follows

$$\mu\{x \in \{0,1\}^{\mathbb{N}} : f(x) = p_\mu(x)\} = 1 \quad \forall \mu \in \mathcal{F}^*.$$

Having a function f as in the above definition of PLIF (resp. SPLIF), it will be useful to call the maximum subset of $\mathcal{E}$ (resp. $\mathcal{E}^*$) on which the function f is the PLIF (resp. SPLIF) as *the domain of identification of* f. In other words, the domain of identification of f is the set of measures from $\mathcal{E}$ (resp. $\mathcal{E}^*$) whose probability limits are identified by f.

The concept of the PLIF was first introduced in [*Simons*, 1971]. The SPLIF was defined in the same article and it was also shown there, that the existence of (Borel measurable) SPLIF on

$\mathcal{E}^*$ implies the existence of the (Borel measurable) PLIF on $\mathcal{E}$. The construction of PLIF from SPLIF was based on transformation of a real-valued sequence into countable many 0-1 valued sequences, SPLIF was then applied on each sequence resulting in countable sequence of zeros and ones and finally this 0-1 valued sequence was transformed to a single real number, which is the result of the PLIF applied on the starting real-valued sequence. This construction preserves Borel measurability. Thus we can restrict ourselves on studying special sets $\mathcal{F}^*$ of probability measures on $\{0,1\}^{\mathbb{N}}$ possessing Borel measurable SPLIF. However, the existence of SPLIF on $\mathcal{E}^*$ was not proved by Simons.

The existence of the PLIF on $\mathcal{E}$ under the assumption of continuum hypothesis was proved in [*Štěpán*, 1973]. The continuum hypothesis enables us to index measures in $\mathcal{E}$ by countable ordinals, application of transfinite induction together with the fact, that each sequence converging in probability has a subsequence converging almost surely yields the existence of the PLIF. In [*Kříž et al*, 2010] the construction of PLIF was extended to general separable metrizable spaces with demonstration of its application for functional representations in stochastic analysis. However, this PLIF is not Borel measurable. In fact, Borel measurable PLIF on $\mathcal{E}$ does not exist, which follows from the main result of the article [*Blackwell*, 1980], which asserts that there are no Borel measurable SPLIFs on $\mathcal{E}^*$. The proof presented there is based on the Oxtoby 0-1 category law. For Borel measurable SPLIF f, it follows from the 0-1 law, that $f^{-1}(1)$ (or $f^{-1}(0)$) contains a dense G_δ set and thus we can construct a random sequence with values in this set converging in probability to 0 (or 1) which contradicts the existence of the Borel measurable SPLIF on $\mathcal{E}^*$.

However, Borel measurable SPLIFs exist for special subsets $\mathcal{F}^* \subset \mathcal{E}^*$. Simons in [*Simons*, 1971] gave the example of estimation problem, where (Borel measurable) PLIF exists, but there is no strongly consisten estimator. The example is based on the sequences of replicated independent Bernoulli variables. In the same article, it is also described, that for each countable set $\mathcal{F}^* \subset \mathcal{E}^*$ there exists measurable SPLIF. Having only one measure, we can construct measurable SPLIF as the limit of a certain subsequence, chosen as subsequence converging almost surely. Applying induction by selecting consecutive sub-subsequences gives us measurable SPLIF for a finite set $\mathcal{F}^*$. Now having infinite sequence of measures from $\mathcal{E}^*$ we can construct measurable SPLIFs for each finite subsequence and take limes superior of these SPLIFs which results in Borel measurable SPLIF for the original infinite sequence of measures.

Measurable SPLIF for dominated sets

The idea of further effort is to extend the SPLIFs from countable sets to some larger subsets $\mathcal{F}^* \subset \mathcal{E}^*$ dominated by a particular measure (i.e. sets of measures that are absolutely continuous with respect to a certain measure). This description of $\mathcal{F}^*$ might be useful as dominated measures are often dealt with in estimation theory. A key to construction of measurable SPLIF for dominated sets may be the following theorem, which can be found in [*Lehmann et al.*, 2005] (Theorem A.4.1, page 698):

Theorem 3. *A family $\mathcal{P}$ of probability measures over a Euclidean space $(\mathbb{R}, \mathcal{B})$ is dominated by a σ-finite measure if and only if it is separable with respect to the total variation distance, i.e. the metric defined as follows*

$$d(P,Q) = \sup_{A \in \mathcal{B}} |P(A) - Q(A)|. \tag{1}$$

Note that according to the Borel isomorphism theorem, Euclidean space $(\mathbb{R}, \mathcal{B})$ and space $(\{0,1\}^{\mathbb{N}}, \mathcal{B})$, are Borel isomorphic. It easily follows that the Borel isomorphism preserves the metric d and thus the theorem 3 holds for the space $\{0,1\}^{\mathbb{N}}$ with Borel σ-algebra as well.

Preceding theorem has the immediate consequence:

Theorem 4. *Let $\mathcal{F}^* \subset \mathcal{E}^*$ be dominated by a σ-finite measure, i.e. there exists σ-finite measure ν such that*

$$\forall \mu \in \mathcal{F}^* : \mu \ll \nu.$$

Then there exists Borel measurable SPLIF on $\mathcal{F}^*$.

Proof. The theorem 3 asserts that the set $\mathcal{F}^*$ is separable with respect to the metric d. Assume f is a Borel measurable SPLIF for the corresponding countable dense subset. It suffices to show, that the domain of identification of f is closed with respect to d. Assume the sequence $\mu_1, \mu_2, \dots$ of measures from the domain of identification of f, i.e.

$$\mu_n\{x : f(x) = p_{\mu_n}\} = 1 \quad \forall n \in \mathbb{N}.$$

Further assume

$$d(\mu_n, \mu) \to 0 \quad \text{as} \quad n \to \infty$$

for a measure $\mu \in \mathcal{E}^*$. Denote p_μ corresponding limit of μ (necessarily constant 0 or 1). It follows from the definition, that

$$\forall n : \mu_n\{x : x_k \neq p_{\mu_n}\} \overset{k\to\infty}{\longrightarrow} 0, \quad \mu\{x : x_k \neq p_\mu\} \overset{k\to\infty}{\longrightarrow} 0.$$

Further, we get for arbitrary $n, k \in \mathbb{N}$:

$$\begin{aligned}
&\mu\{x : f(x) \neq p_\mu\} \leq \mu\{x : p_\mu \neq x_k\} + \mu\{x : x_k \neq f(x)\} \\
&\leq \mu\{x : p_\mu \neq x_k\} + \mu_n\{x : x_k \neq f(x)\} + d(\mu_n, \mu) \\
&\leq \mu\{x : p_\mu \neq x_k\} + \mu_n\{x : x_k \neq p_{\mu_n}\} + \mu_n\{x : f(x) \neq p_{\mu_n}\} + d(\mu_n, \mu)
\end{aligned}$$

It follows that for arbitrary $\epsilon > 0$, there exist $n, k \in \mathbb{N}$ such that

$$\mu\{x : f(x) \neq p_\mu\} \leq \epsilon$$

and this concludes the proof, because

$$\mu\{x : f(x) = p_\mu\} = 1.$$

□

Note that the previous theorem does not provide us with any specific set $\mathcal{F}^*$. To obtain such set, we have to find appropriate dominant σ-finite measure. One possibility is to take Haar measure on topological group $\{0,1\}^{\mathbb{N}}$ endowed with the product topology of the discrete spaces $\{0,1\}$ and with coordinatewise aditive operation defined as follows:

$$\Big(x_k; k \in \mathbb{N}\Big) + \Big(y_k; k \in \mathbb{N}\Big) := \Big((x_k + y_k) \mod 2; k \in \mathbb{N}\Big) \in \{0,1\}^{\mathbb{N}}$$

Haar measure on this compact group is finite, translation invariant measure such that the measure of each non-empty open set is positive. This measure (normalized to 1) is in fact the joint distribution of the independent random variables uniformly distributed on $\{0,1\}$. As Haar measure is an analogue to a Lebesgue measure, it could produce useful set of dominated measures, denote it $\mathcal{H}^*$. However, we still have not found appropriate description of the set $\mathcal{H}^*$. We may also try to describe directly the corresponding set of measures $\mathcal{H} \subset \mathcal{E}$ in terms of dominated measures.

We often deal with measures from $\mathcal{E}^*$ which are not diffusion, i.e. measures μ with the property

$$\exists x \in \{0,1\}^{\mathbb{N}} : \mu(\{x\}) > 0.$$

For example when constructing PLIF from SPLIF as described above, we must apply a SPLIF on such non-diffusion measures. But non-diffusion measures are obviously not absolute continuous with respect to Haar measure. However, we can easily solve this issue due to decomposition of measures and tail property of a SPLIF. Recall, that the function defined on sequences (in our case 0-1 valued) has the tail property, if the value of the function applied on a sequence does not depend on any finite initial segment of the sequence (see [*Blackwell*, 1980]). First note, that each measure from $\mathcal{E}^*$ can be decomposed in the following way:

$$\mu = \alpha_0 \mu_{dif} + \sum_{n \in \mathbb{N}} \alpha_n \delta_{x^n}, \tag{2}$$

where μ_{dif} is a diffusion measure, δ_{x^n} are Dirac measures and α_n are positive constants. Moreover, each $x^n \in \{0,1\}^{\mathbb{N}}$ in the previous decomposition must be constant sequence except for at most finite set of coordinates. Further, having a SPLIF f for a certain set $\mathcal{F}^*$, we may assume without loss of generality, that

$$f(0,0,0,\dots) = 0, \quad f(1,1,1,\dots) = 1. \tag{3}$$

Finally we can assume, that a SPLIF f has a tail property. If not, use a construction of a tail SPLIF as described in [*Blackwell*, 1980]. Recall that a function on $\{0,1\}^{\mathbb{N}}$ is a tail function, if changing a finite number of any coordinates in argument does not affect the value of the function. Thus, we obtain Borel measurable SPLIF f for $\mathcal{F}^*$ with the property

$$f(x_1, x_2, \dots, x_n, 0, 0, \dots) = 0, \quad f(x_1, x_2, \dots, x_n, 1, 1, \dots) = 1, \quad \forall x_i \in \{0,1\}, \forall n. \tag{4}$$

Such SPLIF thus identifies limits for Dirac measures in the decomposition 2 implying the following assertion.

Lemma 5. *If there exists a Borel measurable SPLIF for a set $\mathcal{F}^*$, then there exists a Borel measurable SPLIF for a set*

$$\{\mu \in \mathcal{E}^* : \mu_{dif} \in \mathcal{F}^*\}.$$

We may thus restrict ourselves without loss of generality on studying Borel measurable SPLIFs for diffusion measures only.

Conclusion and discussion

As described above, PLIFs (and SPLIFs) play important role in identifying parameters from weak consistent estimates. Even though universal Borel measurable SPLIF for all sequences does not exist, we presented construction of Borel measurable SPLIF for sequences, whose distributions are dominated by a σ-finite measure and suggested a Haar measure as dominant measure. However, we have not fully described resulting set in $\mathcal{E}$ having Borel measurable PLIF. There may also be other appropriate candidates for dominant measures.

We have dealt with distance in total variation on set of measures. But we may work with other topologies, e.g. topology of weak convergence. With respect to this topology, space of all probability measures is separable, there is no need for dominance. On the other hand, domain of identification of measurable SPLIFs may not be closed. Additional conditions are required so that existence of SPLIF is preserved when taking weak limits of measures.

Further idea is to utilize separation theorems. Assume the following decomposition

$$\mathcal{E}^* = \mathcal{E}_0^* \cup \mathcal{E}_1^*, \qquad \mathcal{E}_i^* = \{\mu \in \mathcal{E}^* : \mu \xrightarrow{P} i\}. \tag{5}$$

The task is to find appropriate compact convex sets

$$\mathcal{F}_i^* \subset \mathcal{E}_i^*, \quad i = 0, 1$$

which may be then separated.

The constructions of SPLIFs above were based on limits of subsequences and the fact, that each sequence converging in probability has a subsequence converging almost surely. But we may also try to construct SPLIFs on the basis of limits of averages, that is

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} x_i$$

and apply results of ergodic theory of theory of Markov chains. It might be useful to combine these results with the simple observation, that if a measure μ from $\mathcal{E}_i^*$ (see 5) is in the domain of identification of a SPLIF f, then all measures from $\mathcal{E}_i^*$ absolutely continuous with respect to μ are in the domain of identification of f as well.

Acknowledgments. The work was supported by the grant SVV 261315/2011 and by GA UK under Contract 2673/2011.

References

Blackwell, D., There are no Borel SPLIFs, *Ann. Probability*, 8, 1189–1190, 1980.

Kříž, P., Štěpán, J., Probability Limit Identification Functions on Separable Metric Spaces, *Acta Universitatis Carolinae - Mathematica et Physica*, 2, 29–36, 2010.

Lehmann, E.L., Romano, J.P., Testing Statistical Hypotheses, *Springer* 3rd Ed., 2005.

Simons,G., Identifying Probability Limits, *Ann. Math. Statist.*, 42, 1429–1433, 1971.

Štěpán, J., The Probability Limit Identification Function Exists Under The Continuum Hypothesis, *Ann. Probability*, 1, 712–715, 1973.

Change Detection in Autoregressive Time Series with Martingale Difference White Noise

K. Starinská

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. In this paper we study asymptotic properties of an efficient score statistic for testing changes in the parameter values of an autoregressive time series. We assume that the error process is a martingale difference sequence with known variance instead of independent identically distributed (i.i.d.) random variables. The procedure allows testing changes in the mean and in the autoregressive parameters separately. We present the invariance principle and the law of the iterated logarithm for martingale difference sequences and linear processes, and show that the asymptotic distribution of statistics under consideration is the same as in the case of i.i.d. errors.

Introduction

Detecting change in the parameters of an autoregressive time series can prevent us from wrong forecasts and data analysis. There were different methods developed for testing if any change occurred during observed time period, see *Csörgő and Horváth* [1997] for a review. Changes in autoregressive processes were studied e.g. by *Davis et al.* [1995], *Hušková et al.* [2007] or *Gombay* [2008]. We were inspired by *Gombay* [2008] where an efficient score statistic is built with the assumption of i.i.d. residuals. Instead of assuming that residuals are i.i.d. we will consider martingale difference sequence and we will prove the same asymptotic properties for testing the change in the mean and autoregressive parameters by this statistic as in *Gombay* [2008].

Let the sequence $\{Y_i\}$ satisfy the autoregressive model of order p

$$Y_i - \mu = \sum_{j=1}^{p} \phi_j (Y_{i-j} - \mu) + \varepsilon_i, \tag{1}$$

where $\{\varepsilon_i\}$ are martingale differences with $\mathrm{E}[\varepsilon_i^2] = \sigma^2$ and $\xi = (\mu, \phi_1, \ldots, \phi_p)'$ is the vector of parameters which can change. The parameter σ^2 is considered known and unchanging constant.

To derive the efficient score statistic we assume that the errors $\{\varepsilon_k\}$ are normally distributed. We denote the conditional density of Y_i under $Y_{i-1}, \ldots, Y_{i-p}$ as $f(Y, \xi) = f(Y_i | Y_{i-1}, \ldots, Y_{i-p})$ for $i = 1, \ldots, n$ and the logarithmic likelihood function of $Y_{-p+1}, \ldots, Y_0, Y_1, \ldots, Y_k$ as $\ell_k(\xi)$. In case that $\{\varepsilon_i\}$ is not a Gaussian white noise, the likelihood function is quasi-likelihood function and we need to specify the finiteness of some moments. The components of the efficient score vector are

$$\begin{aligned} \frac{\partial}{\partial \mu} \ell_k(\xi) &= \frac{1 - \sum_{j=1}^{p} \phi_j}{\sigma^2} \sum_{i=1}^{k} \left[Y_i - \mu - \sum_{j=1}^{p} \phi_j (Y_{i-j} - \mu) \right], \\ \frac{\partial}{\partial \phi_s} \ell_k(\xi) &= \frac{1}{\sigma^2} \sum_{i=1}^{k} \left[Y_i - \mu - \sum_{j=1}^{p} \phi_j (Y_{i-j} - \mu) \right] (Y_{i-s} - \mu), \ s = 1, \ldots, p. \end{aligned} \tag{2}$$

If there is no change in the autoregressive time series we can substitute $Y_{-p}, \ldots, Y_0$ by any finite value, because we are considering stationarity and large sample of observations. For building the test statistic we will need the information matrix

$$I(\xi) = \left(- \mathrm{E} \left[\frac{\partial^2 \ln f(Y, \xi)}{\partial \xi_i \partial \xi_j} \right] \right)_{i,j=1}^{p+1} = \begin{pmatrix} \frac{1}{\sigma^2} (1 - \sum_{j=1}^{p} \phi_j)^2 & 0 \\ 0 & \frac{1}{\sigma^2} \Gamma \end{pmatrix}, \tag{3}$$

where Γ is the covariance matrix of vector $(Y_1, \ldots, Y_p)^T$. The design of the information matrix allows us to test the change separately for the mean and autoregressive parameters. This is because we are using the centered form of the AR(p) model.

Denote $\hat{\xi}_n = (\hat{\mu}_n, \hat{\boldsymbol{\phi}}_n')'$ the maximum likelihood estimators of ξ, where $\hat{\boldsymbol{\phi}}_n = (\hat{\phi}_{n1}, \ldots, \hat{\phi}_{np})'$ and consider statistic

$$\hat{\mathbf{B}}(u) = n^{-1/2} I^{-1/2}(\hat{\xi}_n) \begin{pmatrix} \frac{\partial}{\partial \mu} \ell_{[nu]}(\hat{\xi}_n) \\ \nabla_\phi \ell_{[nu]}(\hat{\xi}_n) \end{pmatrix}, \quad u \in [0,1], \tag{4}$$

where $[x]$ is the integer part of x and $\nabla_\phi \ell$ is a vector of partial derivations of the function ℓ according to the elements of ϕ.

Preliminary results

To show the convergence of $\hat{\mathbf{B}}(u)$ to the Brownian bridge we need some theorems such as the invariance principle or the law of the iterated logarithm for martingale differences. The following theorem can be found in *Eberlein* [1986].

Theorem 1. *Let $\{\varepsilon_k, k \geq 1\}$ be a sequence of random variables with zero mean satisfying*

$$\| \mathrm{E}[S_n(m)|\mathcal{F}_m] \|_1 = O(n^{1/2-\theta}) \tag{5}$$

uniformly in m for some $0 < \theta < 1/2$, where $S_n(m) = \sum_{k=m+1}^{m+n} \varepsilon_k$ and $\| \cdot \|_1$ is L^1-norm and $\mathcal{F}_m$ is σ-algebra generated by $\varepsilon_1, \ldots, \varepsilon_m$. We assume that there exists a finite number $T > 0$ such that uniformly in m

$$n^{-1} \mathrm{E}[S_n(m)^2] - T = O(n^{-\rho}) \tag{6}$$

for some $\rho > 0$. Moreover suppose that there exists $\eta > 0$ such that uniformly in m

$$\| \mathrm{E}[S_n(m)^2|\mathcal{F}_m] - \mathrm{E}[S_n(m)^2] \|_1 = O(n^{1-\eta}) \tag{7}$$

and that there exists a constant $M < \infty$ and $\delta > 0$ such that $\mathrm{E}[|\varepsilon_k|^{2+\delta}] \leq M$. Then there exists a standard Brownian motion $\{W(t), t \geq 0\}$ such that

$$\sum_{\nu=1}^{[t]} \varepsilon_\nu - \sqrt{T} W(t) = O(t^{1/2-\kappa}) \quad a.s. \tag{8}$$

for some $\kappa > 0$.

We may simplify this theorem for our purpose. Let $\{\varepsilon_k, k \geq 1\}$ be a stationary martingale difference sequence. If we use the martingale difference property $\mathrm{E}[\varepsilon_m|\mathcal{F}_{m-1}] = 0$, condition (5) is trivially satisfied since

$$\begin{aligned} \mathrm{E}\left[S_n(m)\middle|\mathcal{F}_m\right] &= \mathrm{E}\left[\mathrm{E}\left[\varepsilon_{m+1} + \ldots + \varepsilon_{m+n}\middle|\mathcal{F}_{m+n-1}\right]\middle|\mathcal{F}_m\right] = \\ &= \mathrm{E}\left[\mathrm{E}\left[\varepsilon_{m+1} + \ldots + \varepsilon_{m+n-1}\middle|\mathcal{F}_m\right]\middle|\mathcal{F}_{m+n-2}\right] = \ldots = \mathrm{E}[\varepsilon_{m+1}|\mathcal{F}_m] = 0. \end{aligned}$$

We know that the martingale differences are uncorrelated and thus we have $n^{-1}\mathrm{E}[S_n(m)^2] = n^{-1}\mathrm{E}\left[\sum_{k=m+1}^{m+n} \varepsilon_k^2 + 2\sum_{k=m+1}^{m+n}\sum_{l=k+1}^{m+n} \varepsilon_k \varepsilon_l\right] = n^{-1}\mathrm{E}\left[\sum_{k=m+1}^{m+n} \varepsilon_k^2\right] = \sigma^2$, where σ^2 is the variance of the martingale differences. Now we choose $T = \sigma^2$ and the condition (6) is satisfied.

Assuming that $\mathrm{E}[\varepsilon_k^2|\mathcal{F}_{k-1}] = \sigma^2$ we get (7) trivially. Therefore, we may state a simplified version of Theorem 1 for the martingale difference sequence as follows.

Theorem 2. *Let $\{\varepsilon_k, k \geq 1\}$ be a stationary martingale difference sequence with zero mean and variance σ^2. Assume that $\mathrm{E}[\varepsilon_k^2|\mathcal{F}_{k-1}] = \sigma^2$ and that $\mathrm{E}[|\varepsilon_k|^{2+\delta}] < \infty$ for some $\delta > 0$. Then there exists a Brownian motion $\{W(t), t \geq 0\}$ such that*

$$\sum_{\nu=1}^{[t]} \varepsilon_\nu - \sigma W(t) = O(t^{1/2-\kappa}) \quad a.s. \tag{9}$$

for some $\kappa > 0$.

We also need an invariance principle and the law of the iterated logarithm for the sequence $Y_k - \mu$, which we assume to satisfy the conditions to be a linear process.

Theorem 3. *Let us define*

$$X_k = \sum_{i=0}^{\infty} c_i \varepsilon_{k-i}, \tag{10}$$

where $\{\varepsilon_k\}$ *are stationary martingale differences,* $\mathrm{E}\,\varepsilon_k = 0$, $\mathrm{E}[\varepsilon_k^2|\mathcal{F}_{k-1}] = \sigma^2$, $0 < \sigma^2 < \infty$ *and* $\mathcal{F}_{k-1}$ *is* σ*-algebra generated by* $\varepsilon_{k-1}, \varepsilon_{k-2}, \ldots$. *Assume that* $\sum_{j=1}^{\infty} j|c_j| < \infty$ *and* $\mathrm{E}\,|\varepsilon_k|^\kappa < \infty$ *for some* $\kappa > 2$. *Then there exists a Brownian motion* $\{W(t), t \geq 0\}$, *such that*

$$\sum_{k=1}^{[t]} X_k - \sigma_1 W(t) = o(t^{1/\nu}), \qquad a.s. \tag{11}$$

for some $\sigma_1 > 0$ *and some* $\nu > 2$.

Moreover, if $\kappa > 4$, *then there exists a Brownian motion* $\{W(t), t \geq 0\}$, *such that*

$$\sum_{k=1}^{[t]} (X_k X_{k-s} - \mathrm{E}[X_k X_{k-s}]) - \sigma_2 W(t) = o(t^{1/\nu}), \qquad a.s. \tag{12}$$

for some $\sigma_2 > 0$ *and some* $\nu > 2$, *where* $s \geq 0$ *is fixed.*

Proof. Steps of this proof are identical to *Gombay and Horváth* [2009], where the same result with i.i.d. $\{\varepsilon_t\}$ is proved. The only difference is that we use the result (9) for martingale differences. It follows from the proof that by using this theorem for (1) we get $\sigma_1^2 = \sigma^2(\sum_{j=1}^{\infty} c_j)^2$. □

Next theorem can be found in *Zhao and Woodroofe* [2008].

Theorem 4. *Let* $\ldots, \varepsilon_{-1}, \varepsilon_0, \varepsilon_1, \ldots$ *be an ergodic stationary martingale difference sequence with mean 0 and variance 1. Define a linear process*

$$X_k = \sum_{j=0}^{\infty} c_j \varepsilon_{k-j},$$

where $\sum_{k=0}^{\infty} c_k^2 < \infty$. *Suppose* $c_k = O[1/(kL(k))]$, *where* $L(k)$ *is a positive, nondecreasing, slowly varying function. If* $\sum_{n=2}^{\infty} (\log k)^{3/2}/(kL(k)) < \infty$, *then for some* $0 < \sigma < \infty$, *with probability* 1

$$\limsup_{n\to\infty} \frac{\sum_{k=1}^{n} X_k}{\sqrt{2n \log\log n}} = \sigma.$$

Let $\{Y_t\}$ satisfy (1) and the roots of the characteristic polynomial $\phi(z) = 1 - \sum_{j=1}^{p} \phi_j z^j$ lies outside the unit circle. Then $\{X_t\}$, where $X_t = Y_t - \mu$, is linear process (10) with coefficients c_j such that $|c_j| < K\rho^j$ for $0 < \rho < 1$ and a constant $K > 0$ (for more details see *Prášková* [2005]). Then condition $\sum_{j=0}^{\infty} j|c_j| < \infty$ in Theorem 3 is fulfilled.

Moreover, $\sum_{j=0}^{\infty} c_j^2 < \infty$ and with $L(x) = (\log x)^\beta$ for $\beta > 5/2$ the conditions of Theorem 4 are satisfied.

Further we consider an AR(p) sequence with martingale difference white noise and without any change in the parameters of the model. The version with i.i.d. white noise is presented in *Gombay and Serban* [2009].

Theorem 5. *Let us assume that* $\{Y_i\}$ *satisfy AR(p) as in* (1) *and that there is no change in any of the parameters. Let* $\{\varepsilon_t\}$ *be a stationary and ergodic martingale difference sequence with* $\mathrm{E}[\varepsilon_i^2] = \sigma^2$ *and* $\mathrm{E}\,|\varepsilon_i|^\kappa < \infty$ *for some* $\kappa > 4$. *Assume that the characteristic polynomial* $\phi(z) = 1 - \sum_{j=1}^{p} \phi_j z^j$ *satisfies* $\phi(z) \neq 0$ *for all* $|z| \leq 1$. *Then the following results hold:*

(i) Under the hypothesis $\sigma^2 = \sigma_0^2$ *and* $\phi = \phi_0$

$$|\hat{\mu}_k - \mu| = O\left(\sqrt{k^{-1} \log\log k}\right) \quad a.s.$$

(ii) Under the hypothesis $\mu = \mu_0$ *and* $\sigma^2 = \sigma_0^2$

$$\|\hat{\phi}_k - \phi\| = O\left(\sqrt{k^{-1} \log\log k}\right) \quad a.s.$$

Proof. (i) Under the assumption of this theorem the maximum likelihood estimation of μ is

$$\hat{\mu}_k = \frac{1}{k(1-\sum_{j=1}^p \phi_j)} \sum_{i=1}^k (Y_i - \sum_{j=1}^p \phi_j Y_{i-j}) = \mu + \frac{1}{k(1-\sum_{j=1}^p \phi_j)} \sum_{i=1}^k \varepsilon_i.$$

Using the law of the iterated logarithm for martingale differences from *Stout* [1970] we have $|\hat{\mu}_k - \mu| = O(\sqrt{k^{-1}\log\log k})$ almost surely.

(ii) In this part of the proof we will work with the vector form of our model, i.e. $\mathbf{Z}_k = \mathbf{X}_k\boldsymbol{\phi} + \boldsymbol{\epsilon}$, where

$$\mathbf{Z}_k = \begin{pmatrix} Y_1 - \mu \\ \vdots \\ Y_k - \mu \end{pmatrix}, \ \mathbf{X}_k = \begin{pmatrix} Y_0 - \mu & \dots & Y_{-p+1} - \mu \\ \vdots & \vdots & \vdots \\ Y_{k-1} - \mu & \dots & Y_{k-p} - \mu \end{pmatrix}, \ \boldsymbol{\phi} = \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_p \end{pmatrix} \text{ and } \boldsymbol{\epsilon} = (\varepsilon_1, \dots, \varepsilon_k)'.$$

Then the estimator of $\boldsymbol{\phi}$ is $\hat{\boldsymbol{\phi}}_k = (\mathbf{X}_k'\mathbf{X}_k)^{-1}\mathbf{X}_k'\mathbf{Z}_k$ and $\hat{\boldsymbol{\phi}}_k - \boldsymbol{\phi} = (\mathbf{X}_k'\mathbf{X}_k)^{-1}\mathbf{X}_k'\boldsymbol{\epsilon}$.

The mth element of the vector $\mathbf{X}_k'\boldsymbol{\varepsilon}$ is $\sum_{i=1}^k (Y_{i-m} - \mu)\varepsilon_i$, $m = 1, \dots, p$ and it has a martingale differences property. The covariance matrix of this vector is $k\sigma^2\Gamma$ by the stationarity property. We may use the law of the iterated logarithm for this sequence, hence $\|(k\sigma^2\Gamma)^{-1/2}\mathbf{X}_k'\boldsymbol{\epsilon}\| = O(\sqrt{\log\log k})$ almost surely.

The element at the mth row and lth column of $\mathbf{X}_k'\mathbf{X}_k$ is $\sum_{i=1}^k (Y_{i-m} - \mu)(Y_{i-l} - \mu)$, thus by the ergodic theorem (*Davidson* [1994]) we have $(1/k)\mathbf{X}_k'\mathbf{X}_k \overset{\text{a.s.}}{\to} \Gamma$. Then

$$\|\hat{\boldsymbol{\phi}}_k - \boldsymbol{\phi}\| = \left\| k^{-1/2} \left(k^{-1}\mathbf{X}_k'\mathbf{X}_k\right)^{-1} \sigma\Gamma^{1/2} \left((\sigma^2 k)^{-1/2}\Gamma^{1/2}\mathbf{X}_k'\boldsymbol{\epsilon}\right) \right\| \overset{\text{a.s.}}{=} O\left(\sqrt{k^{-1}\log\log k}\right).$$

□

Instead of using the maximum likelihood estimator of μ we can use $\bar{\mu}_k = \frac{1}{k}\sum_{i=1}^k Y_i$. This estimator is easier to compute and it does not depend on the autoregressive parameters. Since $\hat{\mu}_k = \mu + \frac{1}{k(1-\sum_{j=1}^p \phi_j)} \sum_{i=1}^k \varepsilon_i$, we have $\mu = \hat{\mu}_k - \frac{1}{k(1-\sum_{j=1}^p \phi_j)} \sum_{i=1}^k \varepsilon_i$. Then

$$\bar{\mu}_k = \frac{1}{k}\sum_{i=1}^k (Y_i - \mu) + \mu = \hat{\mu}_k - \frac{1}{k(1-\sum_{j=1}^p \phi_j)} \sum_{i=1}^k \varepsilon_i + \frac{1}{k}\sum_{i=1}^k (Y_i - \mu) = \hat{\mu}_k + O(\sqrt{k^{-1}\log\log k}).$$

The second and third term are both of order $O(\sqrt{k^{-1}\log\log k})$ by the law of the iterated logarithm.

Main Result

The following theorem says that asymptotic properties of $\hat{\mathbf{B}}(u)$ established by *Gombay* [2008] remain valid under more general assumptions.

Theorem 6. *Let $\{Y_i\}$ be a sequence satisfying model (1) where $\{\varepsilon_i\}$ is a stationary, ergodic martingale difference sequence with $\mathrm{E}[\varepsilon_i^2] = \sigma^2$ and $\mathrm{E}\,|\varepsilon_i|^\kappa < \infty$ for some $\kappa > 4$. Assume that $\mathrm{E}[\varepsilon_k^2|\mathcal{F}_{k-1}] = \sigma^2$. Furthermore, assume that the characteristic polynomial $\phi(z) = 1 - \sum_{j=1}^p \phi_j z^j$ has roots outside the unit circle. Then there exists a $(p+1)$-dimensional Gaussian process $\boldsymbol{B}(u)$ with independent Brownian bridge components $B^{(j)}(u)$, $j = 1, 2, \dots, p+1$, such that*

$$\max_{1\le j\le p+1} \sup_{0\le u\le 1} |\hat{B}^{(j)}(u) - B^{(j)}(u)| = o_p(1).$$

Sketch of the proof. We will consider here the change in the mean only. Then the test statistic has form

$$\hat{B}^{(1)}(u) = \frac{1}{\sqrt{n}}\frac{1}{\sigma}\sum_{i=1}^{[nu]} \left(Y_i - \hat{\mu}_n - \sum_{j=1}^p \hat{\phi}_{jn}(Y_{i-j} - \hat{\mu}_n) \right) = \frac{1}{\sqrt{n}}\frac{1}{\sigma}\sum_{i=1}^{[nu]} \hat{\varepsilon}_i,$$

where $\hat{\varepsilon}_i = Y_i - \hat{\mu}_n - \sum_{j=1}^p \hat{\phi}_{jn}(Y_{i-j} - \hat{\mu}_n)$. Similarly we will have $\varepsilon_i = Y_i - \mu - \sum_{j=1}^p \phi_j(Y_{i-j} - \mu)$. Since $\frac{\partial}{\partial\mu}\ell_n(\hat{\xi}_n) = 0$, $\sum_{i=1}^n \hat{\varepsilon}_i = 0$ and we can write

$$\hat{B}^{(1)}(u) = \frac{1}{\sigma}n^{-1/2} \left(\sum_{i=1}^{[nu]} \hat{\varepsilon}_i - \frac{[nu]}{n}\sum_{i=1}^n \hat{\varepsilon}_i \right).$$

After some computations we get

$$\hat{B}^{(1)}(u) = \frac{1}{\sigma} n^{-1/2} \left[\left(\sum_{i=1}^{[nu]} \varepsilon_i - \frac{[nu]}{n} \sum_{i=1}^{n} \varepsilon_i \right) - \sum_{j=1}^{p} (\hat{\phi}_j - \phi_j) \left(\sum_{i=1}^{[nu]} (Y_{i-j} - \mu) + \frac{[nu]}{n} \sum_{i=1}^{n} (Y_{i-j} - \mu) \right) \right].$$

According to Theorem 2 the first term converges in distribution to $\sigma B^{(1)}(u)$, where $B^{(1)}(u)$ is the Brownian bridge.

Further, we know from Theorem 5 that $\|\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}\|$ is of order $O(\sqrt{n^{-1} \log\log n})$. When we apply Theorem 3 we can see that the term in brackets of the second term is bounded in probability, and thus, we can conclude that the remainder term is of order $o_p(1)$. Thus we have

$$\hat{B}^{(1)}(u) \to B^{(1)}(u)$$

in distribution and

$$\sup_{0 \le u \le 1} \hat{B}^{(1)}(u) \to \sup_{0 \le u \le 1} B^{(1)}(u).$$

The proof concerning the autoregressive parameters is omitted and will be published elsewhere. □

As we mentioned before, it is possible to use this statistic for testing the change in one parameter or in any of the parameters. Let us denote by α the required significance level. Then if we test the change in parameter ξ_j for some $j = 1, \ldots, p+1$ we say, that there is a change if

$$\sup_{0 \le u \le 1} |\hat{B}^{(j)}(u)| \ge C(\alpha)$$

holds, where $C(\alpha)$ is a critical value obtained from

$$\mathrm{P}\left(\sup_{0 \le t \le 1} |B(t)| \ge x \right) = 2 \sum_{k=1}^{\infty} (-1)^{k+1} \exp\{-2k^2 x^2\}, \tag{13}$$

where $\{B(t), t \ge 0\}$ is a Brownian bridge and $x > 0$. This critical values can be found in some statistical tables (for example see *Owen* [1966]).

If we are trying to detect change in a agroup of d parameters ($d = 1, \ldots, p+1$) and we want to keep the significance level α we use

$$\max_{j=1,\ldots,p+1} \sup_{0 \le u \le 1} |B^{(j)}(u)| \ge C(\alpha^*),$$

where $\alpha^* = 1 - (1-\alpha)^{1/d}$. If this inequality holds, we say that there is a change in at least one of the tested parameters. We are using α^*, because due to independent components of limiting process

$$\begin{aligned}
\alpha &= \mathrm{P}_H\left(\max_{j=1,\ldots,d} |B^{(j)}| > C \right) = 1 - \mathrm{P}_H\left(\max_{j=1,\ldots,d} |B^{(j)}| \le C \right) = 1 - \mathrm{P}_H\left(|B^{(j)}| \le C, \forall j = 1, \ldots, d \right) = \\
&= 1 - \prod_{j=1}^{d} \mathrm{P}_H\left(|B^{(j)}| \le C \right) = 1 - \prod_{j=1}^{d} (1 - \mathrm{P}_H\left(|B^{(j)}| > C \right)) = 1 - \left[1 - \mathrm{P}_H\left(|B^{(j)}| > C \right) \right]^d = \\
&= 1 - (1 - \alpha^*)^d,
\end{aligned}$$

where $\mathrm{P}_H(A)$ is the probability of A under the hypothesis that there is no change in the parameters value of AR(1).

To see how the score statistic works we run a small simulation study. We generated 700 sequences of AR(1) with the parameters $\mu = 4$, $\phi_1 = 0.5$, $\sigma^2 = 3$ and the martingale difference white noise sequence for three different length $n = 100$, 250 and 500. The empirical distribution functions of $\hat{B}^{(1)}$ (change in the mean) for these sequences are in Figure 1 with the theoretical distribution function (solid line).

We are also interested in the ability to detect changes if they appear. We set the change-points and for every change-point we generated 700 realizations of AR(1) with change in mean ($\mu = 4 \rightsquigarrow 2.5$) and autoregressive coefficient ($\phi_1 = 0.5 \rightsquigarrow 0.2$). The variance of the white noise sequence remained the same ($\sigma^2 = 3$). The length of generated sequences was $n = 160$. In Figure 2 we see the relative number of detected changes in every parameter separately and also for all the parameters together. Score statistic for the mean (solid line) is strongest when the change is in the middle of the observed sequence and gets weaker as the change goes to the beginning or end of the sequence. The same result holds for testing

change in any of the parameters (dot-dash line). Testing change in autoregressive parameter (dotted line) is not very successful and we think it is because the generated change in this parameter was small. Even we assume that the variance is not changing, we believe that the result of Theorem 6 can be extended for testing changes in the variance. Because of this we were detecting changes in σ^2(dashed line). Since there was no change, the test statistic was not significant.

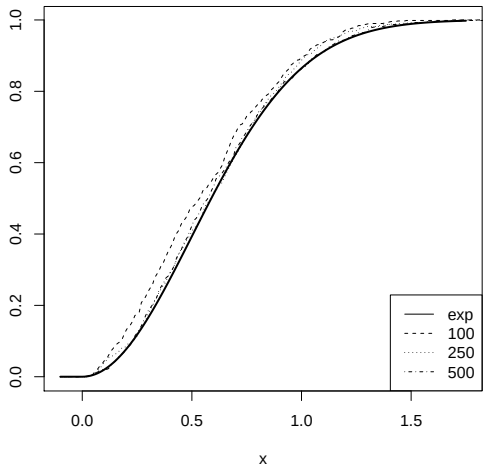


Figure 1.: Empirical distribution functions of $\hat{B}^{(1)}$ for different length of AR(1) with martingale difference white noise. Solid line denotes the theoretical distribution function.

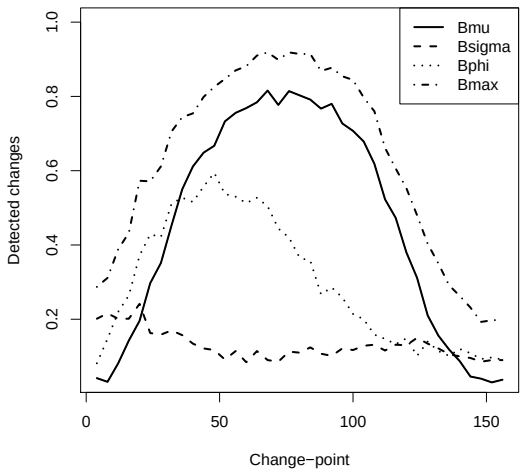


Figure 2.: Relative number of detected changes if the change appeared in mean and the autoregressive parameters at various times of AR(1).

Acknowledgments. The work was supported by the grant SVV261315/2011 and grants GAČR 201/09/0775 and GAČR 201/09/J006.

References

Csörgő M and Horváth L., Limit Theorems in Change-Point Analysis, *Chichester: Wiley*, 1997.

Davidson J., Stochastic Limit Theory: Advanced Texts in Econometrics, Oxford University Press, USA, p.200, 1994.

Davis R.A., Huang D. and Yao Y.-C., Testing for a Change in the Parameter Value and Order of an Autoregressive Model, *Ann. Statist.*, 23, 282-304, 1995.

Eberlein E., On Strong Invariance Principles Under Dependence Assumptions, *Ann. Probab.*, 14, 260-270, 1986.

Gombay E., Change Detection in Autoregressive Time Series, *J. Multivar. Anal.*, 99, 451-464, 2008.

Gombay E. and Horváth L., Sequential Tests and Change Detection in the Covariance Structure of Weakly Stationary Time Series, *Communications in Statistics - Theory and Methods*, 38, 2872-2883, 2009.

Gombay E. and Serban D., Monitoring Parameter Change in AR(p) Time Series Models, *J. Multivar. Anal.*, 100, 715-725, 2009.

Hušková M., Prášková Z. and Steinebach J., On the Detection of Changes in Autoregressive Time Series I & II, *J. Stat. Plann. Inference*, 137, 2007.

Owen D. B., Handbook of Statistical Tables, Russian title: *Sbornik statističeskich tablic*, Moscow, 1966.

Prášková Z. and Lachout P., Základy Náhodných Procesů (in Czech), Nakladatelství Karolinum, Praha, 2005.

Stout W. F., The Hartman-Wintner Law of the Iterated Logarithm for Martingales, *Ann. Math. Stat.*, 41, 2158-2160, 1970.

Zhao O. and Woofroofe M., Law of the Iterated Logarithm for Stationary Processes, *Ann. Probab.*, 36, 127-142, 2008.

Interaction of Incompressible Flow with an Elastic Wall

M. Hadrava

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. The present paper is devoted to the numerical solution to flow in time-dependent domains with elastic walls. This problem has several applications in engineering and medicine. The flow is described by the system of Navier-Stokes equations supplemented with suitable initial and boundary conditions. A part of the boundary of the region occupied by the fluid is represented by an elastic wall, whose deformation is driven by a hyperbolic partial differential equation with initial and boundary conditions. Its right-hand side represents the force by which the fluid flow acts on the elastic wall. A numerical method for solving this coupled problem is elaborated, based on the finite element method and the Arbitrary Lagrangian Eulerian (ALE) formulation of the equations describing the flow. The formulation and analysis of the problem together with discretization, algorithmization and programming of modules, which were added to an existing software package, is presented. The method that was worked out is applied to solving test problems.

Introduction

Interaction of flow with an elastic structure plays an important role in contemporary research and industry. Among the most significant areas of application one can mention aerospace engineering, civil engineering, car industry or medicine. The presented work is concerned with medical applications by modeling fluid flow in a channel with elastic walls, which may represent elastic walls in human vocal chords or vessels.

In the article we describe a numerical method which was derived to obtain an approximate solution to the plane model presented below. The fluid flow is described by the system of incompressible Navier-Stokes equations supplemented with the continuity equation. Movement of the elastic wall of the two-dimensional channel occupied by the fluid is described by the string hyperbolic partial differential equation. The domain occupied by the fluid is time-dependent and therefore the ALE method is applied to transforming the problem to a fixed reference domain. Both the fluid and the structure problem are then semi-discretized in time by the backward-difference formula of second order and the resulting equations are discretized in space using a conforming finite element method. Some numerical results are presented at the end of the paper.

Governing equations

Structure of the domain occupied by the fluid

We deal with incompressible flow in a bounded plane domain $\Omega_t \subset \mathbb{R}^2$ depending on time $t \in [0, T)$, $T > 0$. Figure 1 shows an example of such domain with two elastic walls. For the sake of simplicity,

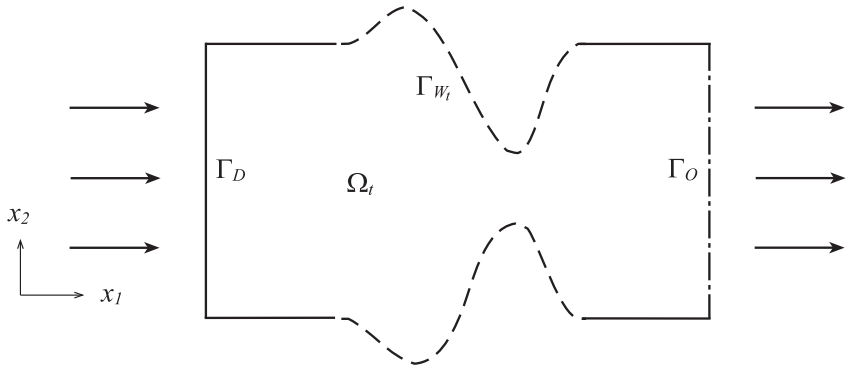


Figure 1. Plane channel with elastic walls.

only a single elastic wall is considered. The boundary $\partial\Omega_t$ of the domain occupied by the fluid is split according to prescribed boundary conditions into three disjoint parts, $\partial\Omega_t = \Gamma_D \cup \Gamma_O \cup \Gamma_{W_t}$. The part of the boundary Γ_D represents inlet and fixed impermeable parts of the walls. Γ_O represents the outlet of the channel. Finally, Γ_{W_t} represents the elastic wall and is time-dependent.

The Navier-Stokes equations

The fluid flow is described by the following system of equations and boundary and initial conditions:

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u} + \nabla p - \nu\Delta\boldsymbol{u} = 0 \qquad \text{in } \mathcal{M}, \tag{1}$$

$$\mathrm{div}\boldsymbol{u} = 0 \qquad \text{in } \mathcal{M}, \tag{2}$$

$$\boldsymbol{u} = \boldsymbol{u}_D \qquad \text{on } \Gamma_D \times (0,T), \tag{3}$$

$$-p\boldsymbol{n} + \nu\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} = -p_{ref}\boldsymbol{n} \quad \text{on } \Gamma_O \times (0,T), \tag{4}$$

$$\boldsymbol{u} = \boldsymbol{w} \qquad \text{on } \Gamma_{W_t} \times (0,T), \tag{5}$$

$$\boldsymbol{u} = \boldsymbol{u}_0 \qquad \text{in } \Omega_0, \tag{6}$$

where the velocity of viscous incompressible flow is denoted by $\boldsymbol{u}$ and the kinematic pressure of the fluid is denoted by p. We consider the incompressible Navier-Stokes equations in the form (1) supplemented with the continuity equation (2). Here the constant $\nu > 0$ denotes the kinematic viscosity of the fluid. This couple of equations is valid in the domain $\mathcal{M} = \{(\boldsymbol{x},t); \boldsymbol{x} \in \Omega_t, t \in (0,T)\}$. On Γ_D the Dirichlet boundary condition (3) is prescribed, where $\boldsymbol{u}_D$ is a given function. On Γ_O the so-called "do-nothing" boundary condition (4) is prescribed, where $\boldsymbol{n}$ denotes the unit outer normal to Ω_t and p_{ref} denotes the reference pressure prescribed on the outlet. On the moving wall Γ_{W_t} the Dirichlet boundary condition (5) is prescribed, where $\boldsymbol{w}$ denotes the velocity of the elastic wall deformation. A rigorous definition of the quantity $\boldsymbol{w}$ is given below (cf. eq. (11)). The system of equations is finally completed by the initial condition (6), where $\boldsymbol{u}_0$ is a given function.

String equation

Deformation of the elastic wall is described by the following initial boundary-value problem:

$$\frac{\partial^2\eta}{\partial t^2} - a\frac{\partial^2\eta}{\partial x_1^2} + b\eta - c\frac{\partial^3\eta}{\partial t\partial x_1^2} + d\frac{\partial\eta}{\partial t} = H \quad \text{in } Q, \tag{7}$$

$$\eta = \eta_0,\ \frac{\partial\eta}{\partial t} = \eta_1 \quad \text{in } (0,L) \text{ at } t = 0, \tag{8}$$

$$\eta = 0,\ \frac{\partial\eta}{\partial t} = 0 \quad \text{in } (0,T) \text{ at } x_1 = 0 \text{ and } x_1 = L, \tag{9}$$

where the string hyperbolic partial differential equation (7) is valid in the domain $Q = (0,L)\times(0,T)$, $L > 0$. Here the function η denotes the deformation of the wall in the direction of the x_2-axis, a, b, c, d are positive real constants characterizing properties of the wall and the function H represents the x_2-component of the force by which the fluid acts on the elastic wall. Equation (7) is supplemented with initial conditions (8), where η_0 and η_1 are given functions, and homogeneous boundary conditions (9). The derivation of the presented model can be found in [*Zaušková*, 2006]. We assume that the elastic wall Γ_{W_t} can be parametrized by a smooth function $\sigma = \sigma_0 + \eta$, $\sigma : Q \to \mathbb{R}$, where σ_0 parametrizes Γ_{W_t} at $t = 0$ and η represents the deformation of the elastic wall. The right-hand side H of eq. (7) is defined by the relation

$$H = \frac{1}{\rho_w h_w}\sum_{j=1}^{2} n_j \tau_{j2},$$

where $\boldsymbol{n} = (n_1, n_2)$ is the unit outer normal to Ω_t, $\rho_w > 0$ is the constant density of the elastic wall, $h_w > 0$ is its constant thickness and $\tau = (\tau_{ij})_{i,j=1}^2$ is the fluid stress tensor (cf. [*Feistauer*, 1993]). Finally, the velocity of the elastic wall deformation is defined by

$$\boldsymbol{w} = \left(0, \frac{\partial\eta}{\partial t}\right). \tag{11}$$

From the presented equations it follows that the fluid flow problem depends on the solution to the string deformation problem through boundary condition (5) and the string deformation problem depends on the solution to the fluid flow problem through the right-hand side H of equation (7).

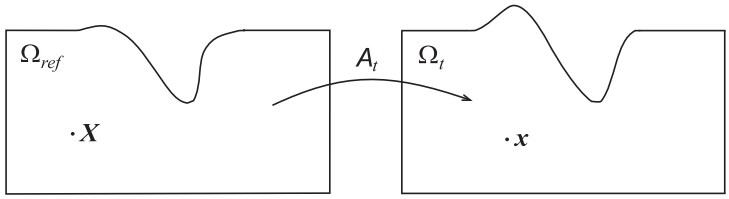


Figure 2. ALE mapping $\mathcal{A}_t$.

ALE formulation

In order to simulate flow in a time-dependent domain the ALE method is employed. Let us denote the *reference configuration* by $\Omega_{ref} = \Omega_0$, i.e. the reference configuration is the computational domain Ω_t at time $t = 0$. A smooth, one-to-one mapping of $\overline{\Omega}_{ref}$ onto $\overline{\Omega}_t$ at time t (the so-called *current configuration*) is denoted by $\mathcal{A}_t$ (cf. [*Nomura and Hughes*, 1992], see Fig. 2), i.e.

$$\mathcal{A}_t : \overline{\Omega}_{ref} \to \overline{\Omega}_t,\ \mathcal{A}_t : \boldsymbol{X} \mapsto \boldsymbol{x} = \mathcal{A}_t(\boldsymbol{X}).$$

The ALE mapping $\mathcal{A}_t$ serves as a basis for the definition of the *domain velocity*

$$\begin{aligned} \tilde{\boldsymbol{w}}(\boldsymbol{X},t) &= \frac{\partial}{\partial t}\mathcal{A}_t(\boldsymbol{X}), \quad \boldsymbol{X} \in \Omega_{ref},\ t \in (0,T), \\ \boldsymbol{w}(\boldsymbol{x},t) &= \tilde{\boldsymbol{w}}(\boldsymbol{X},t), \qquad \boldsymbol{X} = \mathcal{A}_t^{-1}(\boldsymbol{x}),\ \boldsymbol{x} \in \Omega_t,\ t \in (0,T). \end{aligned}$$

We use the symbol $\boldsymbol{w}$ to denote the ALE velocity since its restriction on Γ_{W_t} is the velocity of the elastic wall (see eq. (11)). The *ALE derivative* of a function $f = f(\boldsymbol{x},t)$ is defined as

$$\frac{D^{\mathcal{A}} f}{Dt}(\boldsymbol{x},t) = \frac{\partial \tilde{f}}{\partial t}(\boldsymbol{X},t), \quad \tilde{f}(\boldsymbol{X},t) = f(\mathcal{A}_t(\boldsymbol{X}),t),\ \boldsymbol{X} = \mathcal{A}_t^{-1}(\boldsymbol{x}) \in \Omega_{ref},\ \boldsymbol{x} \in \Omega_t,\ t \in (0,T).$$

From the chain rule it follows that $\dfrac{D^{\mathcal{A}} f}{Dt} = \dfrac{\partial f}{\partial t} + (\boldsymbol{w}\cdot\nabla) f$, which yields the ALE form of the Navier-Stokes equations

$$\frac{D^{\mathcal{A}} \boldsymbol{u}}{Dt} + ((\boldsymbol{u}-\boldsymbol{w})\cdot\nabla)\,\boldsymbol{u} + \nabla p - \nu\Delta\boldsymbol{u} = 0. \tag{14}$$

Discretization of the Navier-Stokes equations

Time discretization

For the time semi-discretization of equations (2) and (14) the second-order backward difference formula is applied. We introduce a uniform partition $0 = t_0 < \cdots < t_N = T$, $t_k = k\tau$, of the time interval $[0,T]$ with a constant time step $\tau > 0$. The exact solution to the Navier-Stokes system $(\boldsymbol{u}, p)$ at time t_n is approximated by the couple $(\boldsymbol{u}^n, p^n)$. The time derivative of $\boldsymbol{u}$ is discretized as

$$\frac{D^{\mathcal{A}} \boldsymbol{u}}{Dt}(\boldsymbol{x}, t_{n+1}) \approx \frac{3\boldsymbol{u}^{n+1}(\boldsymbol{x}) - 4\hat{\boldsymbol{u}}^n(\boldsymbol{x}) + \hat{\boldsymbol{u}}^{n-1}(\boldsymbol{x})}{2\tau},\ \boldsymbol{x} \in \Omega_{t_{n+1}}, \tag{15}$$

where $\hat{\boldsymbol{u}}^j = \boldsymbol{u}^j \circ \mathcal{A}_{t_j} \circ \mathcal{A}_{t_{n+1}}^{-1}$, $j = n-1, n$. Replacing the ALE derivative of the fluid velocity $\boldsymbol{u}$ with the term on the right-hand side of approximation (15) yields a system of stationary PDEs

$$\frac{3\boldsymbol{u}^{n+1} - 4\hat{\boldsymbol{u}}^n + \hat{\boldsymbol{u}}^{n-1}}{2\tau} + \left((\boldsymbol{u}^{n+1} - \boldsymbol{w}^{n+1})\cdot\nabla\right)\boldsymbol{u}^{n+1} + \nabla p^{n+1} - \nu\Delta\boldsymbol{u}^{n+1} = 0, \quad \mathrm{div}\boldsymbol{u}^{n+1} = 0 \tag{16}$$

for unknown functions $\boldsymbol{u}^{n+1}$, p^{n+1}. By $\boldsymbol{w}^{n+1}$ we denote the approximation of the function $\boldsymbol{w}(\cdot, t_{n+1})$.

Space discretization

Weak formulation. Space discretization is carried out using a conforming finite element method (FEM). In this section we write simply $\boldsymbol{u}$, p, Ω and $\boldsymbol{w}$ instead of $\boldsymbol{u}^{n+1}$, p^{n+1}, $\Omega_{t_{n+1}}$ and $\boldsymbol{w}^{n+1}$. At first, equations (16) are multiplied by test functions $\boldsymbol{v} \in \mathcal{X}$ and $q \in \mathcal{Q}$ respectively, where

$$\mathcal{X} = \left\{\boldsymbol{v} \in (H^1(\Omega))^2; \boldsymbol{v}|_{\Gamma_D \cup \Gamma_{W_t}} = 0\right\}$$

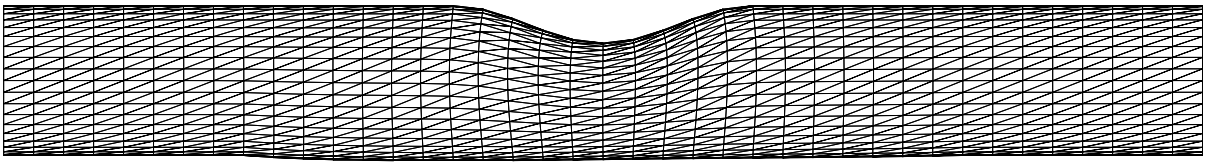


Figure 3. Triangular mesh $\mathcal{T}_h$ of the domain Ω.

and $\mathcal{Q} = L^2(\Omega)$, $H^1(\Omega)$ is the Sobolev space of $L^2(\Omega)$ functions with first derivatives in $L^2(\Omega)$ and $L^2(\Omega)$ is the Lebesgue space of square-integrable functions on Ω. We further write $\mathcal{W} = (H^1(\Omega))^2$. Integrating the resulting equations over Ω, summing them and using Green's theorem for the terms containing $\Delta\boldsymbol{u}$ and ∇p, the weak formulation of equations (16) is obtained: Find $U = (\boldsymbol{u}, p) \in \mathcal{W} \times \mathcal{Q}$ such that

$$a(U, U, V) = f(V) \quad \text{for all} \quad V = (\boldsymbol{v}, q) \in \mathcal{X} \times \mathcal{Q}. \tag{18}$$

We assume that the function $\boldsymbol{u}$ satisfies the boundary conditions (3) and (5). The terms $a(U, U, V)$ and $f(V)$ are defined by

$$a(U^*, U, V) = \frac{3}{2\tau}(\boldsymbol{u}, \boldsymbol{v})_\Omega + (((\boldsymbol{u}^* - \boldsymbol{w}) \cdot \nabla)\boldsymbol{u}, \boldsymbol{v})_\Omega - (p, \mathrm{div}\boldsymbol{v})_\Omega + \nu((\boldsymbol{u}, \boldsymbol{v}))_\Omega + (\mathrm{div}\boldsymbol{u}, q)_\Omega,$$
$$f(V) = \frac{1}{2\tau}(4\hat{\boldsymbol{u}}^n - \hat{\boldsymbol{u}}^{n-1}, \boldsymbol{v})_\Omega - \int_{\Gamma_O} p_{ref}\boldsymbol{v} \cdot \boldsymbol{n}\,\mathrm{d}S,$$

where $U^* = (\boldsymbol{u}^*, p) \in \mathcal{W} \times \mathcal{Q}$, $U = (\boldsymbol{u}, p) \in \mathcal{W} \times \mathcal{Q}$, $V = (\boldsymbol{v}, q) \in \mathcal{X} \times \mathcal{Q}$ and $(f, g)_\Omega = \int_\Omega fg\,\mathrm{d}\boldsymbol{x}$ denotes the scalar product in $L^2(\Omega)$ or in $(L^2(\Omega))^2$ and $((\boldsymbol{u}, \boldsymbol{v}))_\Omega$ denotes the bilinear form

$$((\boldsymbol{u}, \boldsymbol{v}))_\Omega = \int_\Omega \nabla\boldsymbol{u} \cdot \nabla\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} = \sum_{i,j=1}^{2} \int_\Omega \frac{\partial u_i}{\partial x_j}\frac{\partial v_i}{\partial x_j}\,\mathrm{d}\boldsymbol{x},$$

where $\boldsymbol{u} = (u_1, u_2)$ and $\boldsymbol{v} = (v_1, v_2)$. The couple $(\boldsymbol{u}, p)$ represents the approximate solution to the fluid flow problem at the time level t_{n+1}.

Finite element method. We approximate the spaces $\mathcal{W}$, $\mathcal{X}$ and $\mathcal{Q}$ from the weak formulation (18) by finite dimensional subspaces $\mathcal{W}_h$, $\mathcal{X}_h$ and $\mathcal{Q}_h$, $h \in (0, h_0)$, $h_0 > 0$, where $\mathcal{X}_h = \left\{\boldsymbol{v}_h \in \mathcal{W}_h; \boldsymbol{v}_h|_{\Gamma_D \cup \Gamma_{W_t}} = 0\right\}$. We define the discrete problem to find $U_h = (\boldsymbol{u}_h, p_h) \in \mathcal{W}_h \times \mathcal{Q}_h$ such that the equation

$$a(U_h, U_h, V_h) = f(V_h)$$

is satisfied for all $V_h = (\boldsymbol{v_h}, q_h) \in \mathcal{X}_h \times \mathcal{Q}_h$ and $\boldsymbol{u}_h$ approximately satisfies prescribed boundary conditions. Because of the stability of the method, finite-dimensional spaces $\mathcal{X}_h$, $\mathcal{Q}_h$ that satisfy the Babuška-Brezzi condition (cf. [*Brezzi and Falk*, 1991]) are chosen, i.e. there exists a positive constant $c > 0$ such that

$$\sup_{0 \neq \boldsymbol{v} \in \mathcal{X}_h} \frac{(p, \mathrm{div}\boldsymbol{v})_\Omega}{|\boldsymbol{v}|} \geq c||p||$$

holds for all $p \in \mathcal{Q}_h, h \in (0, h_0)$. Here $|\cdot|$ denotes the $(H^1(\Omega))^2$ semi-norm defined by $|\boldsymbol{v}| = ((\boldsymbol{v}, \boldsymbol{v}))_\Omega$ and $||\cdot||$ denotes the $L^2(\Omega)$ norm defined by $||p|| = (p, p)_\Omega$.

In practical realization the domain Ω is assumed to be a polygonal approximation of the region occupied by the fluid at time t_{n+1}. $\mathcal{T}_h$ denotes the triangulation of the domain Ω with standard properties from FEM. Figure 3 shows an example of a triangular mesh of the domain Ω. We employ the Taylor-Hood P^2/P^1-elements (cf. [*Brezzi and Falk*, 1991]), where the pressure p is approximated by a continuous function p_h, which is linear on each triangle $K \in \mathcal{T}_h$, and the fluid velocity $\boldsymbol{u}$ is approximated by a continuous function $\boldsymbol{u}_h$, which is quadratic in each component on each triangle $K \in \mathcal{T}_h$. The couple $(\mathcal{X}_h, \mathcal{Q}_h)$ defined in this way satisfies the Babuška-Brezzi condition (cf. [*Brezzi and Falk*, 1991]).

Linearization. Resulting strongly non-linear problem is linearized by application of the Oseen iterative process (cf. [*Feistauer*, 1993], page 599). Each Oseen iteration is equivalent to the system of linear algebraic equations with a non-symmetric matrix

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}+\mathbf{C} \\ \mathbf{B}^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{U} \\ \boldsymbol{P} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F} \\ \boldsymbol{G} \end{pmatrix}, \tag{22}$$

where the vector $\boldsymbol{U}$ denotes the coefficients of an approximation of the discrete fluid velocity with respect to the chosen basis of the space $\mathcal{W}_h$ and the vector $\boldsymbol{P}$ denotes the coefficients of an approximation of the discrete pressure with respect to the chosen basis of the space $\mathcal{Q}_h$.

The solution to the system (22) is realized by the direct solver UMFPACK (cf. [*Davis and Duff*, 1997]), which works sufficiently fast for systems with up to 10^5 equations.

Discretization of the string equation

In this section we denote $x = x_1$. The string equation (7), which is of the second order in time, is transformed to a couple of first order differential equations

$$\frac{\partial \xi}{\partial t} - a\,\frac{\partial^2 \eta}{\partial x^2} + b\eta - c\frac{\partial^2 \xi}{\partial x^2} + d\xi = H, \quad \frac{\partial \eta}{\partial t} = \xi, \tag{23}$$

where ξ denotes the velocity of the elastic wall deformation. Equations (23) are semi-discretized in time with the aid of the backward-difference formula of second order of accuracy, similarly as in the time semi-discretization of the Navier-Stokes equations. The resulting equations are then discretized in space employing the conforming finite element method with linear elements. The resulting stiffness matrix $\mathbf{S}$ is block-banded, indefinite and non-symmetric. Since the number of degrees of freedom in this case is low, the solution to the wall deformation problem can be obtained with the aid of a direct solver.

Numerical experiments

The following numerical experiments were obtained using a modified software package FEMFLUID (cf. [*Sváček*, 2007]). Numerical methods for solving the flow problem and the wall deformation problem separately were presented in the preceding sections. Prior to presenting results of the main test problem we introduce a method to deal with the interaction problem. A technique based on the so-called *predictor-corrector* method is employed. At the time level t_n the right-hand side H of eq. (7) is computed. The obtained result is then used to compute an approximation to the wall deformation η at time t_{n+1} and subsequently an approximation of the domain occupied by the fluid at time t_{n+1} is obtained. Now we solve numerically the flow problem at the time level t_{n+1} and use the obtained results to update the approximation of the right-hand side H of eq. (7). If H changes by more than a prescribed tolerance, we repeat the whole process to obtain a better approximation of the wall deformation η and subsequently a better approximation of the flow problem solution.

In the presented numerical experiment it is assumed that the upper wall movement is prescribed by a sufficiently smooth time-periodic function. The upper wall movement induces movement of the lower elastic wall. The main parameters of the numerical experiment are given as follows: $\tau = 0.01$, $T = 40$, $\nu = 0.01$. The corresponding Reynolds number Re is defined by the relation $Re = UL/\nu$ and therefore equals 100. Here U denotes the characteristic velocity (in our case the prescribed flow velocity at the channel inlet) and L denotes the characteristic length (in our case the width of the channel). Figure 4 shows the evolution of the deformation of the elastic wall in time. Figure 5 shows the graph of the movement of two fixed points of the elastic wall with the first coordinate given by $x_1 = -1.4$ and $x_1 = 0$. Finally, Figure 6 shows the velocity field at time $t = 7.9$. The triangular mesh used in this example has

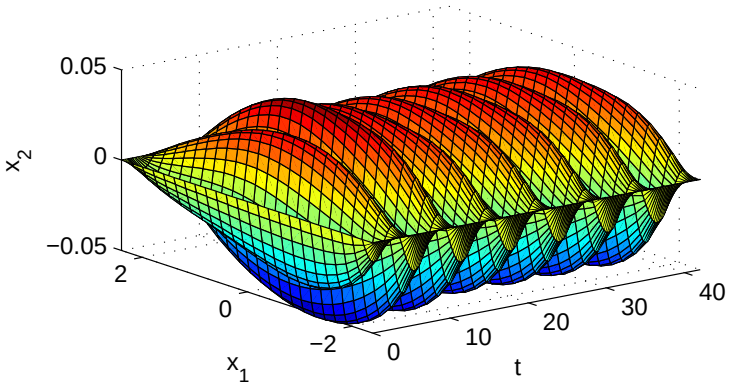


Figure 4. Evolution of the deformation of the elastic wall in time.

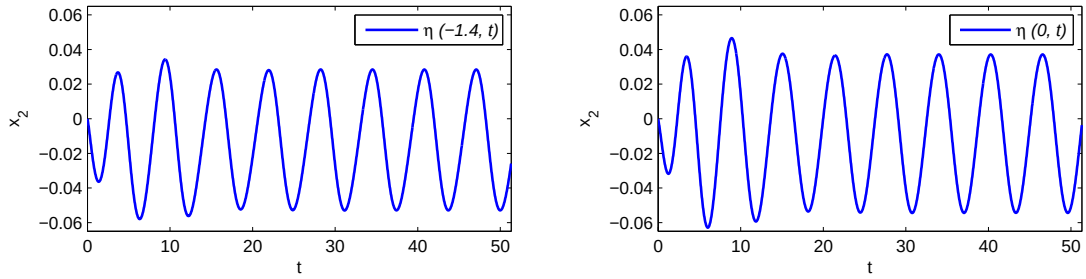


Figure 5. Graph of the movement of a point with the first coordinate $x_1 = -1.4$ (left) and $x_1 = 0$ (right).

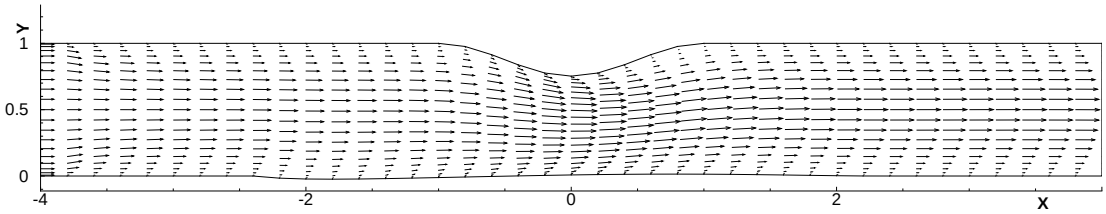


Figure 6. Velocity field $\boldsymbol{u}$ at time $t = 7.9$. Here $\mathbf{X} = x_1$ and $\mathbf{Y} = x_2$.

1600 elements. The number of degrees of freedom in the fluid flow problem is 7503, while the number of degrees of freedom in the elastic wall deformation problem is 46. The computational time on a standard dual-core laptop was approximately 12 hours. Details about the numerical experiment can be found in [*Hadrava*, 2010].

Conclusion

We developed a numerical method and a program code for solving interaction between the two-dimensional viscous incompressible fluid flow in time-dependent domains and elastic walls. In this paper we focused on a single elastic wall located on the lower side of the channel. A modification which would account for a couple of elastic walls is fairly straight-forward. The resulting algorithm was programmed in the C language. The results that we obtained suggest that the developed numerical scheme for the solution to the problem of interaction is sufficiently robust and it is possible to extend its scope of application by further development.

In the future work more numerical experiments will be evaluated to estimate the rate of convergence of the numerical method. The author would also like to extend the presented model to a more complicated problem of the interaction between incompressible or compressible flow and an elastic vibrating two-dimensional body with two degrees of freedom. The model will then be applied to computing airflow around a profile of a vibrating airfoil, which can be deformed, move in the vertical direction and rotate around its elastic axis.

Acknowledgments. The present work was supported by the grant SVV-2011-263316 of Charles University in Prague.

References

Brezzi, F. and Falk, R. S., Stability of higher-order Hood-Taylor methods. *SIAM J. Numer. Anal.*, no. 28, pp. 581–590, 1991.

Davis, T. A. and Duff, I. S., An unsymmetric-pattern multifrontal method for sparse LU factorization. *SIAM Journal on Matrix Analysis and Applications*, vol 18, no. 1, pp. 140–158, 1997.

Feistauer, M., Mathematical Methods in Fluid Dynamics. Longman Scientific & Technical, Harlow, 1993.

Hadrava, M., Numerické řešení proudění v časově závislých oblastech s elastickými stěnami (in Czech). Master Thesis, Faculty of Mathematics and Physics, Charles University in Prague, 2010.

Nomura, T. and Hughes, T. J. R., An arbitrary Lagrangian-Eulerian finite element method for interaction of fluid and a rigid body. *Comp. Methods Appl. Mech. Engrg.*, no. 95, pp. 115–138, 1992.

Sváček, P., FEMFLUID, 2007. [Available online at `http://marian.fsik.cvut.cz/~svacek/femfluid.html`.]

Zaušková, A., 2D Navier-Stokes equations in a time dependent domain. PhD Dissertation, Faculty of Mathematics, Physics and Informatics, Comenius Univ. Bratislava, 2006.

Numerical Simulation of Interaction of Fluid Flow and An Elastic Body

A. Kosík

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. We are concerned with the numerical simulation of interaction of incompressible flow with elastic bodies. Our goal is to simulate airflow in human vocal folds and their flow-induced vibrations. We consider two-dimensional viscous incompressible flow in a time-dependent domain. The fluid flow is described by the Navier-Stokes equations in the arbitrary Lagrangian-Eulerian formulation. The flow problem is coupled with the elastic behaviour of the solid bodies. The dynamical problem of deformation of the elastic body is formulated. The developed solution of the coupled problem based on the finite element method is demonstrated by numerical experiments.

Introduction

In many cases we solve either problems of the fluid mechanics or problems of the solid mechanics. But in many applications we are interested in the behaviour of both the fluid flow and the solid and moreover their interaction. A lot of examples we find in biomechanics. Here we are concerned with the phonation onset in biomechanics of voice, which is an important characteristic of human voice production. The phonation onset can be characterized as a state of the system when it is loosing the aeroelastic stability, i.e. when the airflow parameters like subglottal pressure or airflow rate cross a limit value and the system becomes unstable by flutter. Frequency-modal analysis of a simplified three-mass model of the vocal fold in interaction with a potential flow separated at the superior edge of the vocal fold showed that the two eigenfrequencies are coupled when the instability occurs [*Horáček et al.*, 2002]. A similar linear stability analysis was performed on a 2-D continuum model of vocal folds in potential flow by [*Zhang*, 2009]. Here the instability threshold is studied in the time domain using 2-D FE model of the vocal folds coupled with 2-D FE model of viscous incompressible flow.

Fluid flow in a moving domain

We consider incompressible viscous flow in a bounded domain $\Omega_t^f \subset \mathbb{R}^2$ depending on time $t \in [0, T]$. By $\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{x}, t)$ we denote the velocity and by $p = p(\boldsymbol{x}, t)$ the kinematic pressure (i.e.,

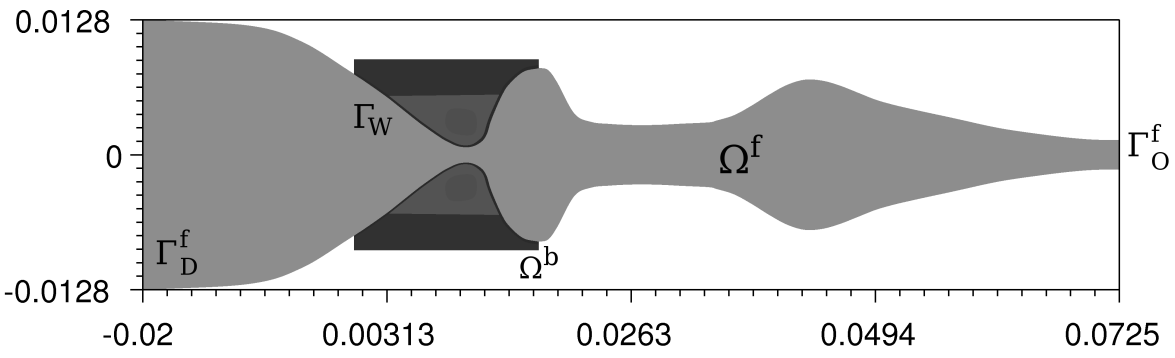


Figure 1. The model of vocal fold.

pressure divided by the density ϱ^f of the fluid), $\boldsymbol{x} \in \Omega_t^f$, $t \in (0,T)$ and ν denotes the kinematic viscosity. Incompressible viscous flow is described by the system of the Navier-Stokes equations equipped with initial and boundary conditions [*Feistauer*, 1993].

$$\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{v} + \nabla p - \nu \Delta \boldsymbol{v} = 0 \quad \text{in } \Omega_t^f, \tag{1}$$

$$\nabla \cdot \boldsymbol{v} = 0 \quad \text{in } \Omega_t^f, \tag{2}$$

In order to simulate flow in a moving domain, we employ the Arbitrary Lagrangian-Eulerian (ALE) method. This method is based on a special mapping of the reference configuration Ω_0^f onto the deformed, actual configuration Ω_t^f. We reformulate the Navier-Stokes equations in the ALE form [*Nomura*, 1992]:

$$\frac{D^{\mathcal{A}}}{Dt}\boldsymbol{v} + ((\boldsymbol{v} - \boldsymbol{w}) \cdot \nabla)\, \boldsymbol{v} + \nabla p - \nu \Delta \boldsymbol{v} = 0 \quad \text{in } \Omega_t^f, \tag{3}$$

$$\nabla \cdot \boldsymbol{v} = 0 \quad \text{in } \Omega_t^f. \tag{4}$$

Here $\frac{D^{\mathcal{A}}}{Dt}$ is so-called ALE derivative and $\boldsymbol{w}$ denotes the domain velocity. System (3) - (4) is equipped with the initial condition

$$\boldsymbol{v}(\boldsymbol{x}, 0) = \boldsymbol{v}_0, \quad \boldsymbol{x} \in \Omega_0^f, \tag{5}$$

and boundary conditions. We assume that $\partial\Omega_t^f = \Gamma_D^f \cup \Gamma_O^f \cup \Gamma_{W_t}$, where Γ_D^f, Γ_O^f and Γ_{W_t} are mutually disjoint. On Γ_D^f, representing the inlet and impermeable fixed walls, we prescribe the Dirichlet boundary condition, on the impermeable moving walls Γ_{Wt} we assume that the fluid velocity $\boldsymbol{v}$ is equal to the domain velocity of the elastic body and on the outlet Γ_O^f we prescribe the so-called "do-nothing" boundary condition:

$$\boldsymbol{v}|_{\Gamma_D^f} = \boldsymbol{v}_D \quad \text{on } \Gamma_D^f, \tag{6}$$

$$\boldsymbol{v}|_{\Gamma_{Wt}} = \boldsymbol{w}|_{\Gamma_{Wt}} \quad \text{on } \Gamma_{W_t}, \tag{7}$$

$$-(p - p_{ref})\boldsymbol{n} + \nu \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{n}} = 0 \quad \text{on } \Gamma_O^f. \tag{8}$$

Here $\boldsymbol{n}$ is the unit outer normal to $\partial\Omega_t^f$ and p_{ref} is a prescribed reference outlet kinematic pressure.

There are several possibilities how to carry out the space-time discretization. In order to develop a stable, accurate scheme, which can easily treat complicated boundaries, we apply the finite element method (FEM). Here, the Taylor-Hood P_2/P_1 finite element pair satisfying the Babuška-Brezzi condition is used. In order to avoid spurious oscillations in approximate solutions in the case of high Reynolds numbers we apply the *streamline diffusion method* together with *div-div* stabilization of the pressure. For the time discretization we use a second-order two-step backward difference formula using the computed approximate solution $\boldsymbol{v}^{n-1}$ in $\Omega_{t_{n-1}}^f$ and $\boldsymbol{v}^n$ in $\Omega_{t_n}^f$ for the calculation of $\boldsymbol{v}^{n+1}$ in the domain $\Omega_{t_{n+1}}^f$. Here the solutions from previous time steps are transformed via ALE mapping on the actual computational domain. For the numerical solution of incompressible flow we use the program FEMFLUID developed by P. Sváček. The space-time discretization is specified in [*Sváček*, 2004].

The linear problem of elasticity

In what follows, $\Omega^b \subset \mathbb{R}^2$ will be a bounded domain representing the elastic body of the form of vocal folds. We denote by $\boldsymbol{u}(\boldsymbol{x}, t)$, $\boldsymbol{x} \in \Omega^b$, $t \in (0,T)$, the displacement of the body and the strain tensor as

$$e_{ij}(\boldsymbol{u}) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \quad i,j = 1,2. \tag{9}$$

The deformation of the vocal folds is modelled by the generalized Hooke's law for isotropic bodies [*Nečas et al.*, 1981]

$$\tau_{ij}^b = \lambda \mathrm{div}\, \boldsymbol{u}\delta_{ij} + 2\mu e_{ij}, \quad i,j = 1,2, \tag{10}$$

where $(\tau_{ij}^b)_{i,j=1}^2$ denotes the stress tensor and λ and μ are the Lamé coefficients related to the Young modulus E and to the Poisson ratio σ as

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu}, \quad \sigma = \frac{\lambda}{2(\lambda + \mu)}. \tag{11}$$

The dynamic equations of an elastic body have the form

$$\varrho^b \frac{\partial^2 u_i}{\partial t^2} + C\varrho^b \frac{\partial u_i}{\partial t} - \sum_{j=1}^{2} \frac{\partial \tau_{ij}^b}{\partial x_j} = 0, \quad \text{on } (0,T)\times\Omega^b, \quad i = 1,2. \tag{12}$$

The expression $C\varrho^b \frac{\partial u_i}{\partial t}$, where $C \geq 0$, is a dissipative damping of the system and ϱ^b denotes the density of the solid material. We complete the elasticity problem with initial and boundary conditions. The initial conditions read

$$\boldsymbol{u}(\cdot,0) = \boldsymbol{0}, \quad \frac{\partial \boldsymbol{u}}{\partial t}(\cdot,0) = \boldsymbol{0}, \quad \text{in } \Omega^b. \tag{13}$$

Further, let $\partial\Omega^b = \Gamma_W \cup \Gamma_D^b$, where Γ_W and Γ_D^b are two disjoints parts of $\partial\Omega^b$. Let surface force $\boldsymbol{T^n}$ be prescribed on the boundary Γ_W and let the part of the boundary Γ_D^b be fixed:

$$\sum_{j=1}^{2} \tau_{ij}^b n_j = T_i^{\boldsymbol{n}} \quad \text{on } \Gamma_W \times (0,T), \quad i = 1,2, \tag{14}$$

$$\boldsymbol{u} = 0 \quad \text{on } \Gamma_D^b \times (0,T). \tag{15}$$

We are looking for the displacement $\boldsymbol{u}$ satisfying equation (12) and the initial and boundary conditions (13) - (15).

Further, we carry out the space-time discretization. We reformulate the problem in a weak sense. For given $\boldsymbol{u}^0 \in \boldsymbol{H}^1(\Omega^b)$, $\boldsymbol{z}^0 \in \boldsymbol{L}^2(\Omega^b)$ we seek for the solution $\boldsymbol{u} \in \boldsymbol{L}^2(0,T;\boldsymbol{V})$, if it holds $\boldsymbol{u}' \in \boldsymbol{L}^2\left(0,T;\boldsymbol{L}^2(\Omega^b)\right)$, $\boldsymbol{u}'' \in \boldsymbol{L}^2(0,T;\boldsymbol{V}^*)$,

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}(\varrho^b \boldsymbol{u}(t),\boldsymbol{y})_{0,\Omega^b} + \frac{\mathrm{d}}{\mathrm{d}t}(C\varrho^b \boldsymbol{u}(t),\boldsymbol{y})_{0,\Omega^b} + a(\boldsymbol{u},\boldsymbol{y};t) = (\boldsymbol{T^n}(t),\boldsymbol{y})_{0,\Gamma_W}, \quad \forall \boldsymbol{y} \in \boldsymbol{V},\ t \in [0,T],$$

$$\boldsymbol{V} = V^2,\ V = \left\{\varphi \in H^1(\Omega^b) \mid \varphi|_{\Gamma_D^b} = 0\right\},$$

$$a(\boldsymbol{u},\boldsymbol{y};t) = \int_{\Omega^b} \sum_{i,j=1}^{2} \left(\lambda \mathrm{div}\boldsymbol{u}(t)\delta_{ij} + 2\mu e_{ij}\left(\boldsymbol{u}(t)\right)\right) \frac{\partial y_i}{\partial x_j}\, \mathrm{d}\boldsymbol{x}. \tag{16}$$

We apply the finite element method using continuous piecewise linear elements. Thus we seek the approximate solution $\boldsymbol{u_h}$ on the space $\boldsymbol{L}^2(0,T;\boldsymbol{V}_h)$, where $\boldsymbol{V}_h = V_h^2$,

$$V_h = \left\{\varphi_h \in C(\overline{\Omega}_b) \mid \varphi \text{ is linear on each triangle of the triangulation, } \varphi|_{\Gamma_D^b} = 0\right\} \subset V. \tag{17}$$

The semi-discretized problem can be written as a second order system of ordinary differential equations. For the time discretization we apply the Newmark scheme. In each time-step we get a linear algebraic system with symmetric positive definite matrix. The solution of this system was realized by the solver, which is based on the conjugate gradient method.

The coupled problem

Up to now we assumed that the fluid flow and the deformation of the elastic body were separated processes. On the common boundary of both domains we need to take into account mutual action of the fluid and the body. We denote as the common boundary set Γ_{W_t}, defined as

$$\Gamma_{W_t} = \left\{ \boldsymbol{x} \in \mathbb{R}^2;\ \boldsymbol{x} = \boldsymbol{X} + \boldsymbol{u}(\boldsymbol{X}, t),\ \boldsymbol{X} \in \Gamma_W \right\}. \tag{18}$$

The domain Ω_t^f is defined by the displacement $\boldsymbol{u}$ of the part Γ_W at time t. The ALE mapping $\mathcal{A}_t$ is defined with the aid of a special stationary linear elasticity problem. If we know the computational domain Ω_t^f obtained by the fluid at time t, we can solve the problem describing the flow and we can assign the surface force $\boldsymbol{T^n}$ acting on the body Ω^b on the part of the boundary Γ_W. This surface force is opposite to the force $\boldsymbol{T}^f$ acting by the airflow on the reference domain Ω_0^f.

$$\boldsymbol{T^n} = -\boldsymbol{T}^f. \tag{19}$$

We obtain the force $\boldsymbol{T}^f$ by the transformation of the force $\boldsymbol{T}_n^f$ defined on $\Gamma_{W_{t_n}}^f$ by the equation

$$T_{n\,i}^f = \sum_{j=1}^{2} \tau_{ij}^f \tilde{n}_j, \quad i = 1, 2, \tag{20}$$

where $\tilde{\boldsymbol{n}}$ is the unit outer normal to the boundary $\Gamma_{W_{t_n}}$ and τ_{ij}^f are components of the stress tensor τ^f. We can compute the stress in the actual configuration $\Omega_{t_n}^f$ by the equation:

$$\tau_{ij}^f = \varrho^f \left(-p\delta_{ij} + \nu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right), \quad i, j = 1, 2. \tag{21}$$

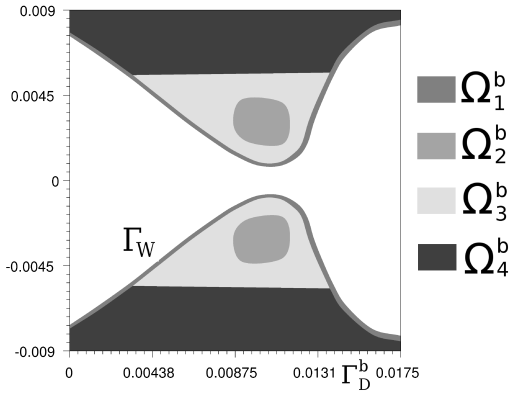
The ALE mapping describing the movement of the computational domain matches to the definition of deformation. Thus we use Piola's transformation for the acting force $\boldsymbol{T}_n^f$ onto the boundary of the reference domain Ω_0^f:

$$\boldsymbol{T}^f = |\mathrm{cof}(\nabla \mathcal{A}_{t_n})\boldsymbol{n}|\, \boldsymbol{T}_n^f \tag{22}$$

where $\mathrm{cof}(\nabla \mathcal{A}_{t_n})$ denotes the cofactor matrix of $\nabla \mathcal{A}_{t_n}$. Since $\tilde{\boldsymbol{n}} = \frac{\mathrm{cof}(\nabla \mathcal{A}_{t_n})\boldsymbol{n}}{|\mathrm{cof}(\nabla \mathcal{A}_{t_n})\boldsymbol{n}|}$ we get

$$\boldsymbol{T}^f = \tau^f \mathrm{cof}(\nabla \mathcal{A}_{t_n})\boldsymbol{n}. \tag{23}$$

The procedure of the implementation is following. First we approximate the computational domain $\Omega_{t_n}^f$ by the domain $\Omega_{t_{n-1}}^f$. The computation of the flow problem is carried out in the approximate domain $\Omega_{t_n}^f$. We get the approximate velocity $\boldsymbol{v}_h$ and pressure p_h. We use the obtained result to compute an approximation of the components of stress tensor τ_{ij}^f on the $\Gamma_{W_{t_n}}$. The stress tensor gives us the surface force $\boldsymbol{T}_n^f$ on the boundary $\Gamma_{W_{t_n}}$. We transform the surface force $\boldsymbol{T}_n^f$ to obtain the surface force $\boldsymbol{T}^f$ acting on the reference domain Ω^b on the part of boundary Γ_W. Further we solve the problem of elasticity and get the approximate displacement $\boldsymbol{u}_h$ at time t_n. Hence we are able to solve the flow problem in the updated domain $\Omega_{t_n}^f$ and repeat the computation of the elasticity problem with the updated approximation of the stress tensor τ_{ij}^f. If the approximation of the displacement $\boldsymbol{u}_h$ changes more than a prescribed tolerance, we repeat the whole process to obtain a better approximation of the flow problem solution and consequently a better approximation of the body displacement. Then we come to the next time step t_{n+1}.



subdomain	E	σ
Ω_1^b	25000	0.4
Ω_2^b	15000	0.4
Ω_3^b	5000	0.45
Ω_4^b	45000	0.35

Figure 2. Material characteristics.

Numerical experiments

We consider model of human vocal folds and vocal tract [*Feistauer et al.*, 2010]. The vocal folds represented by the domain Ω^b have different material characteristics in different subdomains. We use the same time step $\tau = 5 \cdot 10^{-5}$s for the solution of the coupled flow and elasticity problem and the input data $\nu = 1.5 \cdot 10^{-5}$ m^2.s^{-1}, $\varrho^f = 1.17$ kg.m^{-3}, $C = 0.1$ s^{-1}, the initial velocity $\boldsymbol{v}^0 = 0$ and the inlet and outlet pressure $p_{in} = 600$ Pa, $p_{out} = 0$ Pa.

As for the domain Ω^b we distinguish subdomains with different material characteristics. We assume that the material density $\varrho^b = 1040$ kg.m^{-3} is the same for all subdomains, but the values of the Young modulus E and the Poisson ratio σ are different, see Table 2 and Figure 1. On the part of boundary Γ_W we prescribe the conditions for the moving boundary. We acquire the surface forces $\boldsymbol{T^n}$ by solving the problem of fluid flow. The computational process starts by the solution of the flow problem in the domain $\Omega_{t_\alpha}^f$ at the initial time $t_\alpha = -2.10^{-3}$ s. At time $t = 0$ the structure was released and the solution of the complete fluid-structure interaction started.

Figure 3 shows velocity streamlines and displacement of the computational domain at several time instants.

Conclusion

The numerical method for solving the flow-induced vibrations of an elastic body in incompressible viscous flow has been developed and applied to the simulation of airflow in interaction with human vocal folds. The results are in good agreement with previous simulations using simplified models and with known physiological data (see, e.g. [*Horáček*, 2002]).

In future we will focus on the verification of our computations. We are going to perform more numerical experiments with various time steps and various number of finite elements. Another goal of our future work will be the simulation of the complete closure of the channel and the use of a non-linear elasticity model in the coupled problem.

Acknowledgments. This research was supported by the grant SVV-2011-263316 of the Charles University in Prague.

References

Feistauer, M.: Mathematical Methods in Fluid Dynamics, Longman Scientific & Technical, Harlow, 1993.

Feistauer, M., Česenek, J., Horáček, J., Kučera, V. & Prokopová, J., DGFEM for the numerical solution of compressible flow in time dependent domains and applications to fluid-structure interaction. In: *ECCOMAS CFD 2010, J.C.F. Pereira and A.Sequeira (Eds), Lisbon, Portugal, CDROM, ISBN 978-989-96778-1-4, 14-17 June 2010.*

Horáček, J., Švec, J. G.: Aeroelastic model of vocal-fold-shaped vibrating element for studying the phonation, Journal of Fluids and Structures 16 (7), 931-955. doi:10.1006/jfls.454, 2002

Nečas, J. & Hlaváček, I., Mathematical Theory of Elastic and Elasto-Plastic Bodies, An Introduction. Elsevier, Amsterdam, 1981.

Nomura, T. & Hughes, T. J. R.: An arbitrary Lagrangian-Eulerian finite element method for interaction of fluid and a rigid body, Comp. Methods Appl. Mech. Engrg., no. 95., 115-138, 1992.

Sváček P. & Feistauer M.: Application of a stabilized fem to problems of aeroelasticity. In M. Feistauer, V. Dolejší, and Najzar K., editors, Numerical Mathematics and Advanced Applications, ENUMATH2003, pages 796-805, Heidelberg, Springer, 2004.

Zhang, Z.: Characteristics of phonation onset in a two-layer vocal fold model, J. Acoust. Soc. Am. 125(2), 1091-1102, 2009.

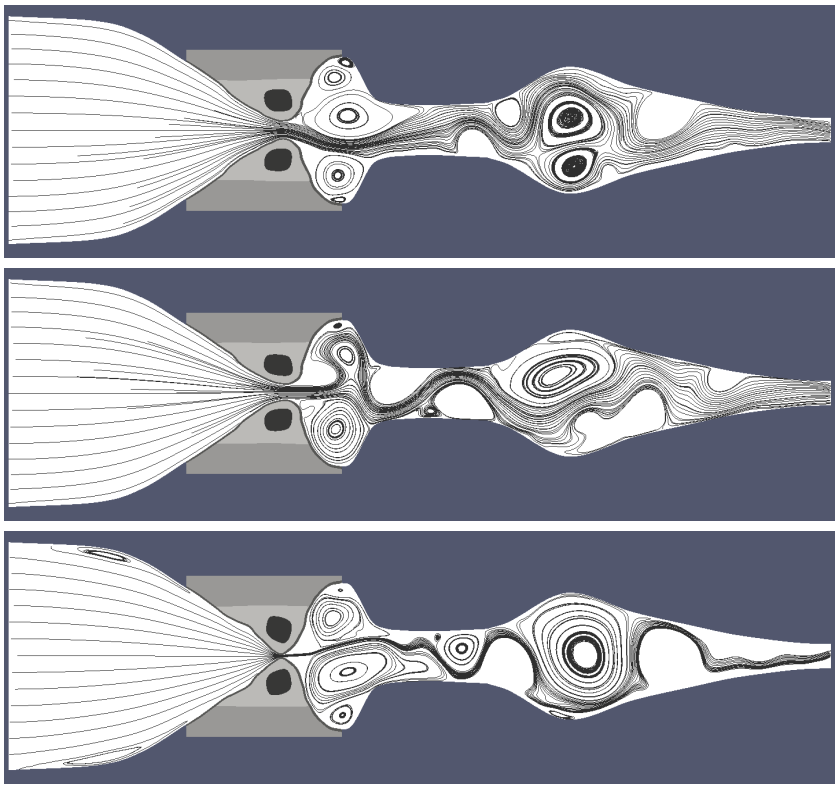


Figure 3. Streamlines at time instants t = 0.033, 0.034, 0.036 and 0.03725 s

 ISBN 978-80-7378-184-2

Comparison of Clenshaw-Curtis and Gauss Quadrature

M. Novelinková

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. In the present work, Gauss and Clenshaw-Curtis quadrature formulas are compared. It is well known, that Gauss quadrature converges for every continuous function f and has a factor-of-2 advantage in efficiency for finite n ($(n+1)$-point scheme integrates exactly polynomials of degree $2n+1$) On the other hand, Clenshaw-Curtis scheme integrates exactly polynomials of degree at most n, but converges also for every continuous function f. This scheme does not turn out to be half as efficient as the Gauss formula for most of the integrands, both quadratures reach almost the same accuracy. Moreover, using the fast Fourier transformation, the Clenshaw-Curtis scheme can be implemented in $\mathcal{O}(n \log n)$ operations, what makes the scheme more efficient. Gauss quadrature nodes and weights can be evaluated in $\mathcal{O}(n^2)$ operations solving a tridiagonal eigenvalue problem. However, there are some problems (boundary elements integrals) where the Gauss quadrature should continue to be preferred quadrature rule.

Introduction

In this article we will consider the following problem. We are given a continuous function f on the closed interval $[-1, 1]$ and we seek to approximate the integral

$$I = I(f) = \int_{-1}^{1} f(x)\mathrm{d}x$$

by sums

$$I_n = I_n(f) = \sum_{k=0}^{n} w_k f(x_k)$$

for various integers n, where the nodes x_k depend on n but not on the function f itself. The weights are defined uniquely by the property that I_n is wanted to be *interpolatory quadrature*, what means that it integrates exactly polynomials of degree at most n.

The most important approach to such a problem is through the automatic quadrature scheme, what is a set of quadrature formulas, each with its own error estimate. According to the specified tolerence the one of these formulas is chosen, without requiring an excessive number of values of the integrand. The quadrature formulas and the error estimate must be possible to evaluate using only values of the integrand in the interval of integration. Usually, the formulas are applied in sequence so it is very suitable that function values required by one formula can be used by the later ones.

There are several methods how to built a quadrature formula. The well known Newton-Cotes formulas are defined by taking the nodes to be equally spaced from -1 to 1. The properties of such schemes for $n \to \infty$ are very bad, some of the weights are negative and the formulas do not converge for a general continuous integrand f. They converge only if the function f is analytic in large region surrounding the interval of integration.

Another very significant example is the Gauss quadrature formula, which is defined by choosing the nodes optimally in the sense of maximizing the degree of polynomials that (1) can integrate exactly. Since there are $n+1$ nodes, the attainable degree is $n+1$ order higher. Thus the $(n+1)$-point Gauss formula integrates exactly polynomials of degree $2n+1$, what makes it the most accurate.

Clenshaw-Curtis scheme

There are two ways how to describe the idea behind the Clenshaw-Curtis scheme. Firstly, we construct the interpolatory polynomials that have the same values as the integrand in the zeros of the Tchebyshev polynomials. We observe that a sequence of such polynomials converges to the function almost everywhere for the piecewise continuous function (unlike the interpolatory polynomials agreeing at equidistant nodes). That means that the function $f(x)$ will be firstly approximated by an interpolating polynomial agreeing with it in the Tchebyshev points, and then this polynomial will be integrated.
The second, equivalent, way is to substituate the variable of integration

$$\int_{-1}^{1} f(x)\mathrm{d}x = \int_{0}^{\pi} f(\cos\theta)\sin\theta\mathrm{d}\theta. \tag{1}$$

If we knew the cosine transform of the function f

$$f(\cos\theta) = F(\theta) = \sum_{n=0}^{\infty} A_n \cos(n\theta) \tag{2}$$

then the integral (1) could be rewritten as follows

$$\int_{0}^{\pi} \sum_{n=0}^{\infty} A_n \cos(n\theta)\sin\theta\mathrm{d}\theta = \sum_{\substack{n=0 \\ n \text{ even}}}^{\infty} \frac{2A_n}{1-n^2} \tag{3}$$

The cosine transform of the function f is, of course, not known, but if we compute the disctrete finite cosine transform of $F(\theta)$ sampled at equidistant points $\theta = \frac{\pi s}{N}$, $s = 0, 1, \ldots, N$, then we obtain[1]

$$a_n = 2\sum_{s=0}^{N}{}'' F\left(\frac{\pi s}{N}\right)\cos\left(\frac{\pi s n}{N}\right). \tag{4}$$

Then the inverse formula gives

$$F\left(\frac{\pi s}{N}\right) = \frac{1}{N}\sum_{n=0}^{N}{}'' a_n \cos\left(\frac{\pi s n}{N}\right). \tag{5}$$

Thus we can use this formula and approximate the integrand

$$F(\theta) \approx \sum_{n=0}^{N}{}'' \left(\frac{a_n}{N}\right)\cos(n\theta). \tag{6}$$

Finally, we obtain the Clenshaw-Curtis quadrature formula

$$\int_{0}^{\pi} f(\cos\theta)\sin\theta\mathrm{d}\theta \approx \sum_{\substack{n=0 \\ n \text{ even}}}^{N}{}'' \left(\frac{a_n}{N}\right)\frac{2}{1-n^2}. \tag{7}$$

The expansion to the cosine series exists and converges for every function f continuous with bounded variation. Many properties now follow from classical Fourier series results such as Parseval's theorem or the Nyquist sampling theorem. It was proved [*Imhof*, 1963] that the

[1] in these and later formulas the symbol $\sum''$ means that the first and the last terms of the sum have half weight

weights a_n are positive, the scheme integrates polynomials of degree n exactly and converges for all continuous function f (for details see [*Novelinkova*, 2010]). The high cost of the cosine transform was a serious drawback in using this type of quadrature formula. Furthermore, the conventional computation of the cosine transform using the recurrence relation [*Engels*, 1980] is numerically unstable, particularly at the low frequencies. Also in case the automatic scheme requires refinement of the sampling, massive storage is needed to save the integrand values after the cosine transformation is computed.
In the paper [*Gentleman*, 1972] a modification of the fast Fourier transform was introduced, what overcomes all the problems mentioned above, it is very resistant to rounding errors and can be implemented in $O(n \log n)$ [*Waldvogel*, 2006]. In order not to waste integrand values already obtained, the new choice of the number of nodes n should be a multiple of the previous one, usually this is taken as $n = 2^p$ [*Gentleman*, 1972].
There are many error estimates for this quadrature scheme proposed, for example see [*Clenshaw and Curtis*, 1960], [*O'Hara and Smith*, 1967], [*Gentleman*, 1972], all of them can be easily calculated, what is very important for checking the accuracy obtained.

Comparison

As written in the introduction section, the Gauss quadrature is positive and also converges for every continuous function $f \in \mathcal{C}([-1, 1])$. The nodes and weights of this formula can be evaluated in $\mathcal{O}(n^2)$ operations by solving a tridiagonal eigenvalue problem, as was proved in [*Golub and Welsch*, 1969].
The basic comparison is straightforward, the Gauss quadrature appear to have factor-of-2 advantage in efficiency for finite f, on the other hand, Clenshaw-Curtis scheme is easier and faster to implement.
However, the numerical comparison of the Clenshaw-Curtis and the Gauss quadrature schemes can reveal suprising results [*O'Hara and Smith*, 1967], [*Trefethen*, 2008]]: Clenshaw-Curtis scheme reaches almost the same accuracy for most of the integrands. The Gauss quadrature significantly outperforms the Clenshaw-Curtis quadrature only for functions analytic in a sizable neighborhood of $[-1, 1]$. For such functions, the convergence of both methods is very fast. Thus Clenshaw-Curtis quadrature essentially never requires many more evaluations than Gauss to converge to a prescribed accuracy. Even Clenshaw and Curtis themselves recorded the same effect [*Clenshaw and Curtis*, 1960]. There were several error estimations stated in [*O'Hara and Smith*, 1972] and [*Trefethen*, 2008] that explains the practical equivalence of these two quadrature rules.

For the theoretical explanation of such a behavior we will introduce following definitions. Given $f \in \mathcal{C}([-1, 1])$, and $n \geq 0$, let p_n^* be the unique best approximation to f on $[-1, 1]$ of degree $\leq n$ with respect to the norm $||.|| = ||.||_\infty$ and define $E_n^* = ||f - p_n^*||$. For a quadrature scheme with nonnegative weights and for any $f \in \mathcal{C}([-1, 1])$ the following statement holds

$$|I - I_n| \leq 4E_n^*, \tag{8}$$

and $I_n \to I$ as $n \to \infty$. Since both of the quadrature considered are nonnegative, this statement can be applied and thus we have that if the best approximants to f converge rapidly as $n \to \infty$ then I_n will converge rapidly to I. As in [*Trefethen*, 2008] this will be combined with results of the approximation theory to the effect that if f is smooth, its best approximants converge rapidly. The results derived from the Tchebyshev series for a function $f \in \mathcal{C}([-1, 1])$ will be considered.
Let us construct the Tchebyshev series for $f \in \mathcal{C}([-1, 1])$ as it was done in [*Trefethen*, 2008].

Then we obtain[2]

$$f(x) = \sum_{j=0}^{\infty}{}' a_j T_j(x), \qquad a_j = \frac{2}{\pi}\int_{-1}^{1} \frac{f(x)T_j(x)}{\sqrt{1-x^2}}\mathrm{d}x, \tag{9}$$

where the $T_j(x) = \cos(j\cos^{-1}x)$ are the Tchebyshev polynomials of degree j. The equal sign in the first formula is justified under the mild condition that f is Dini-continuous, in which case the series converges uniformly to f.
In the work [*Trefethen*, 2008] is showed, that if f is smooth, its Tchebysheff coefficients decrease rapidly. Two smoothness conditions are considered: a kth derivative satisfying a condition related to bounded variation, or analyticity in a neighborhood of $[-1, 1]$. Several theorems bounding the coefficients of the Tchebyshev expansion are proved, and finally the following theorem is stated. Let the norm $\|.\|_T$ be the Tchebyshev-weighted 1-norm defined by

$$\|u\|_T = \|\frac{u'(x)}{\sqrt{1-x^2}}\|_1.$$

Theorem 1. *Let Clenshaw-Curtis and Gauss quadrature be applied to a function $f \in \mathcal{C}([-1, 1])$. If $f, f', \ldots, f^{(k-1)}$ are absolutely continuous on $[-1, 1]$ and $||f^{(k)}||_T = V < \infty$ for some $k \geq 1$, then for all sufficiently large n*

$$|I - I_n| \leq \frac{32V}{15\pi k(2n+1-k)^k}. \tag{10}$$

"Sufficiently large n" means for the Clenshaw-Cutis scheme that $n > n_k$ for some n_k that depends on k but not f or V and for the Gauss quadrature $n \geq \frac{k}{2}$.

Let the norm $\|.\|_T$ be the Tchebyshev-weighted 1-norm defined by

$$\|u\|_T = \|\frac{u'(x)}{\sqrt{1-x^2}}\|_1.$$

The factor 2^{-k} in the error bound applies to Clenshaw-Curtis quadrature too. The crucial fact of the proof is that of aliasing. On the grid in $[0, 2\pi]$ of $2n$ equally spaced points $\theta_j = \pi j/n$, $0 \leq j \leq 2n-1$, the functions $\cos((n+p)\pi\theta_j)$ and $\cos((n-p)\pi\theta_j)$ are indistinguishable. Applying this fact to the variable $x = \cos\theta$ we obtain

Theorem 2. *For any integer p with $0 \leq p \leq n$*

$$T_{n+p}(x_j) = T_{n-p}(x_j) \tag{11}$$

on the Tchebyshev grid. Consequently for the Clenshaw-Curtis scheme

$$I_n(T_{n+p}) = I_n(T_{n-p}) = I(T_{n-p}) = \begin{cases} \frac{2}{1-(n-p)^2}, & \text{for } n \pm p \text{ is even} \\ 0, & \text{for } n \pm p \text{ is odd.} \end{cases} \tag{12}$$

The error in integrating T_{n+p} can be expressed

$$I(T_{n+p}) - I_n(T_{n+p}) = \begin{cases} \frac{8pn}{n^4-2(p^2+1)n^2+(p^2-1)^2}, & \text{for } n \pm p \text{ is even} \\ 0, & \text{for } n \pm p \text{ is odd.} \end{cases} \tag{13}$$

[2]the symbol $\sum'$ indicates that the term for $j = 0$ is multiplied by $1/2$

Clenshaw-Curtis formula has essentially the same performance for most integrands and can be implemented effortlessly by the fast Fourier transformation. This scheme applied to the functions analytic in a sizable neighborhood of $[-1, 1]$ exhibits a curious phenomenon, that explains also why Gauss quadrature outperforms this scheme in these cases.
When the number of nodes in the integration rule increases, the error of the Clenshaw-Curtis quadrature rule does not decay to zero evenly but in two distinct stages. In the work [*Weideman and Trefethen*, 2007] was proved that for the number of nodes n less than a critical value, the error behaves like $\mathcal{O}(\rho^{-2n})$, where ρ is a constant greater than 1. For these values the accuracy of Clenshaw-Curtis scheme is almost indistinguishable from the one of Gauss quadrature. With higher amount of nodes, the error decreases at the rate $\mathcal{O}(\rho^{-n})$. It means, that initially Clenshaw-Curtis scheme converges about as fast as the Gauss rule. The point in which the convergence switches from one rate to another can be seen also in the error curve constructed for this scheme as it was done in [*Weideman and Trefethen*, 2007]. It was proved also, the critical value of n where the kink occurs depends on the position of the singularity if the integrand.
Even though, the practical equivalency of these two schemes was stated in several papers, there are problems in which Gauss quadrature scheme should stay the preferred integration rule (for example [*Elliott, Johnston and Johnston*, 2008]).

Conclusion

In this paper, we have briefly introduced Clenshaw-Curtis scheme which belongs to the class of interpolatory quadrature schemes. We stated some of the most important characteristics as well as the inconvenience which caused that this scheme was not that much in use it the past. The comparison with the well-known Gauss formula was examined from several points of view and the theoretical explanation was given. In many cases, these two schemes can be considered as equivalent, however there are some problems in which the equivalency can not be applied. The outlook for the future work is to focus on the sensitivity of the schemes discussed.

Acknowledgments. The work was supported by the grant SVV-2011-263316

References

Clenshaw, C. W. and Curtis, A. R., *A method for numerical integration on an automatic computer*, Numer. Math., 2, pp. 197-205, 1960.

Elliott, D., Johnston, B. M., Johnston, P. R., *Clenshaw-Curtis and Gauss-Legendre quadrature for certain boundary element integrals*, SIAM J. SCI. COMPUT., 31, pp.510-530, 2008.

Engels, H., *Numerical quadrature and cubature*, London, Academia Press, 1980.

Gentleman, W. M., *Implementing Clenshaw-Curtis quadrature. I*, Comm. ACM, 15, pp. 337-342, 1972.

Gentleman, W. M., *Implementing Clenshaw-Curtis quadrature. II*, Comm. ACM, 15, pp. 343-346, 1972.

Golub, G. H. and Welsch, J. H., *Calculation of Gauss quadrature rules*, Math. Comp., 23, pp. 221-230, 1969.

Imhof, J. P., *On the method for numerical integration of Clenshaw and Curtis*, Numer. Math., 5, pp. 138-141, 1963.

Novelinkova, M., *Non-interpolatory quadratures and refined interpolatory quadratures*, Diploma thesis, 2010.

O'Hara, H. and Smith, F. J., *Error estimation in the Clenshaw-Curtis quadrature formula*, Compu. J., 11, pp.213-219, 1968

Trefethen, L. N., *Is Gauss quadrature better than Clenshaw-Curtis?*, SIAM Rev., 50, pp.67-87, 2008.

Waldvogel, J., *Fast construction of the fejer and Clenshaw-Curtis quadrature rules*, BIT, 46, pp.195-202, 2006.

Weideman, J. A. C. and Trefethen, L. N., *The kink phenomenon in Fejér and Clenshaw-Curtis quadrature*, Numer. Math., 107, pp.707-727, 2007.

The Story of a Right Wavelet Conoid

J. Doležal

Department of Mathematics and Descriptive Geometry, VŠB-TU Ostrava, Czech Republic.

Abstract. Conoids are a special kind of warped ruled surfaces. This paper presents one of them, its geometric creation and motivation for using it. The original idea of utilizing right wavelet conoid comes from Gaudí, it was then continued in the work of Santiago Calatrava and there is one implementation of the surface in the Czech Republic with a Dutch inspiration.

Introduction

A conoid is a ruled surface specified by three control elements: plane ρ, straight line a called the axis of the conoid, and curve c. The conoid is then formed by all straight lines (rulings) which are parallel to control plane ρ and intersect axis a and control curve c (Figure 1a).

If $a \perp \rho$, then the conoid is called the right conoid (Figure 1b). Therefore, all forming rulings of the surface are perpendicular to axis a. It also means that if axis a and control curve c have been specified, there is no need to specify control plane ρ, because it must be perpendicular to axis a and its absolute position is not important.

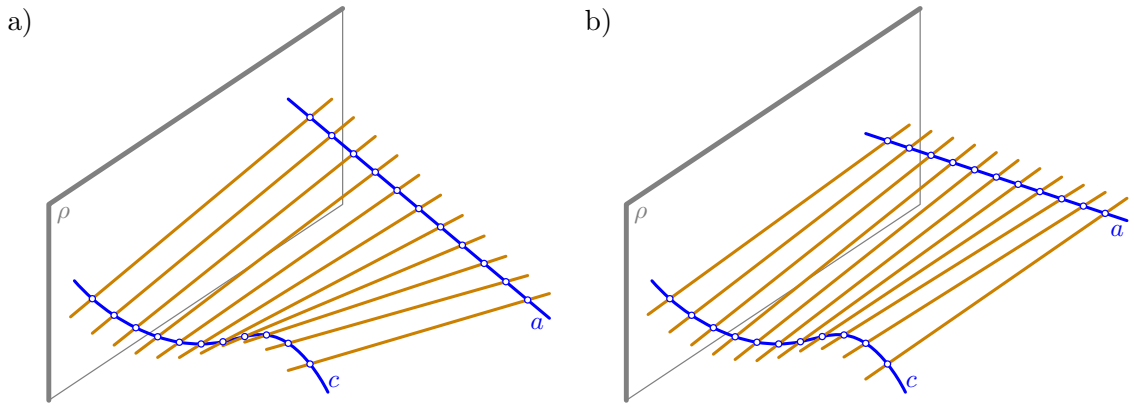


Figure 1. A general (a) and a right (b) conoid.

Right conoids (or their parts) are used in civil engineering. For example, a right conoid of a circle or a right parabolic conoid can be found as roofing constructions. A right conoid of a helix is often used for a spiral staircase. In this case an axis of the control helix is also the axis of the conoid. It is usually called a helicoid.

We will focus on one particular right conoid – a right wavelet conoid.

Right wavelet conoid

The control curve of this right conoid is a wavelet. For the description of the wavelet a sine curve will be chosen.

This surface is one of the first conoidal structures used for practical purposes. In the next three subsections we will describe its geometric creation, show a video tutorial and give a historical overview of its usage. All this is presented on a web page: `http://mdg.vsb.cz/jdolezal/StudOpory/Geometrie/Plochy/ZborcenePlochy/Konoidy/Realizace/PrimyVlnkovyKonoid.html`

Virtual animated model

The model of the surface and of its sequential creation is designed with the help of VRML (Virtual Reality Modeling Language), so that it can also become an element of the web page (to display it a relevant plug-in is needed). This model is conceived as a series of animations. In the model, orientation of the axes is as follows: x-axis from left to right, z-axis from bottom to top and y-axis from front to back.

In the first animation of the model, the sine curve is geometrically created. Let's have an initial circle with a given radius $r > 0$ and with its center at point $[-r; 0; 0]$ (Figure 2a). A copy of the initial circle then moves to a new position centered at point $[0; 0; r]$ (Figure 2b). In the next step the moved circle starts to roll along the x-axis. During this movement it straightens up (Figure 2c), simultaneously, the sine curve is drawn (Figure 2d). The wavelet-sine curve is described by the following parametric equations:

$$\begin{aligned} x &= ru \\ y &= 0 \\ z &= r \sin u, \end{aligned}$$

where $r > 0$ and $u \in \langle 0; 2\pi \rangle$.

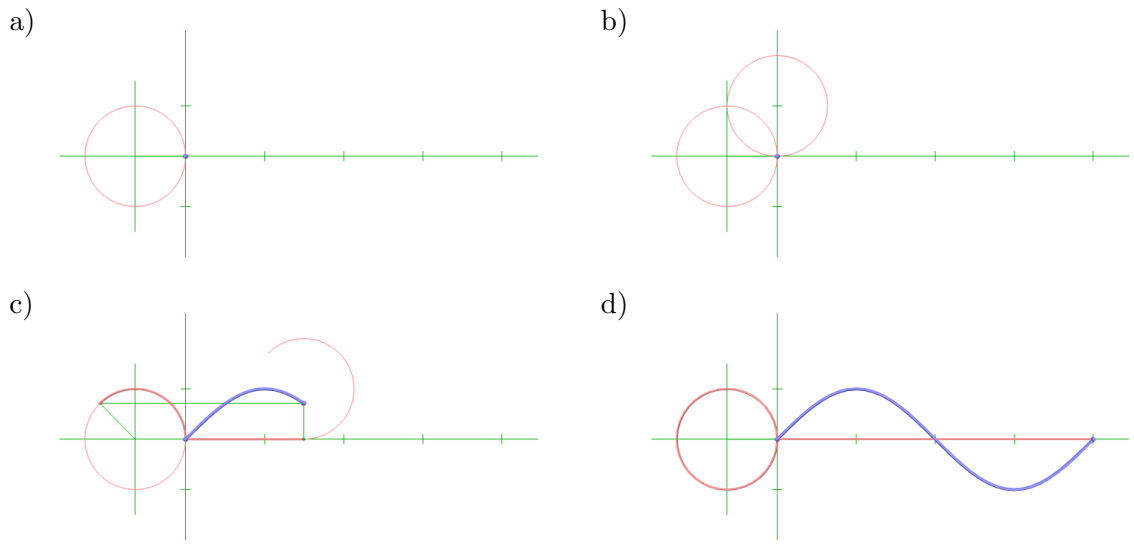


Figure 2. Individual steps of creating the wavelet-sine curve.

In the second animation, the created sine curve runs into the depth away from us along the y-axis (Figure 3a). We obtain a developable cylindrical surface with the control sine curve. All its rulings are parallel to the y-axis. They correspond to a new parameter v, where $v \in \langle 0; 1 \rangle$ and $a \neq 0$ is a constant:

$$\begin{aligned} x &= ru \\ y &= 2av \\ z &= r \sin u. \end{aligned}$$

For the third animation, parameter s is important. It controls flipping of the rear sine curve upside down with respect to the horizontal xy-plane (Figure 3b). As we can see in new corresponding equations

$$\begin{aligned} x &= ru \\ y &= 2av \\ z &= r(1 - 2sv) \sin u, \end{aligned}$$

the rulings of the surface remain rulings, only now they are not parallel but skew – the cylindrical surface has changed into a warped one.

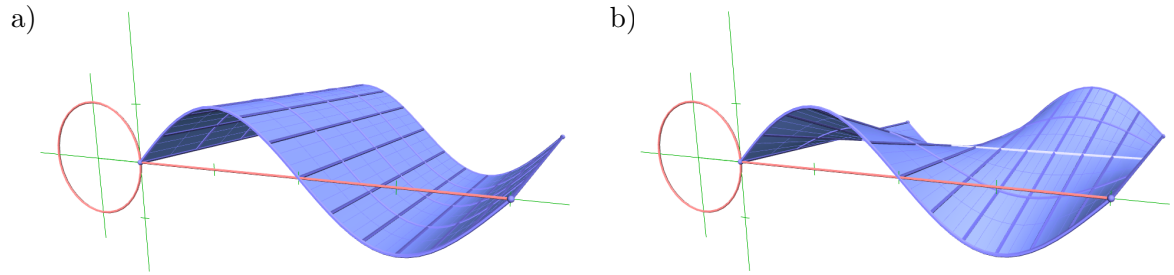


Figure 3. A cylindrical (a) and a conoidal (b) wavelet-sine surface.

The fourth animation is only a phase shift of the created conoid, so the last parameter p gets involved, where $p \in \langle 0; 2\pi \rangle$:

$$\begin{aligned} x &= ru \\ y &= 2av \\ z &= r(1 - 2sv)\sin(u + p). \end{aligned} \tag{1}$$

During this animation we can watch undulating of the surface. We may also change the appearance of the conoid into a form of beams of the same length. These beams are harmonically rotating around the axis of the conoid (Figure 4).

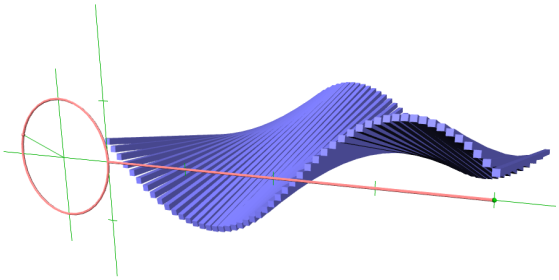


Figure 4. Screenshot of the model shaped to undulating form of rotating beams.

When the fourth animation is over, it is possible to display equations (1) in the model with six parametric sliders – changing the values of r, a, s and p parameters will allow you to watch the appropriate forming of the surface, and by setting the values of parameters u and v you can change corresponding u-ruling and v-wavelet curve (Figure 5).

Video tutorial

The above described model provides a lot of options allowing alteration of the model but in fact it is not a creative tool. Therefore, a learning video was created as a possibility for anyone who would like to design this conoidal surface on his own.

The first and longest part of the video was made as a screen record of step by step modeling of the surface with Google SketchUp, a free application (Figure 6a). The second part shows how previous modeling can be extended to the geolocated model of a real building with realistic photo textures (Figure 6b) and in the third part the whole model can be viewed in Google Earth application (Figure 6c).

The entire video is uploaded to YouTube server and as such is a part of the above mentioned web page devoted to this interesting right conoid.

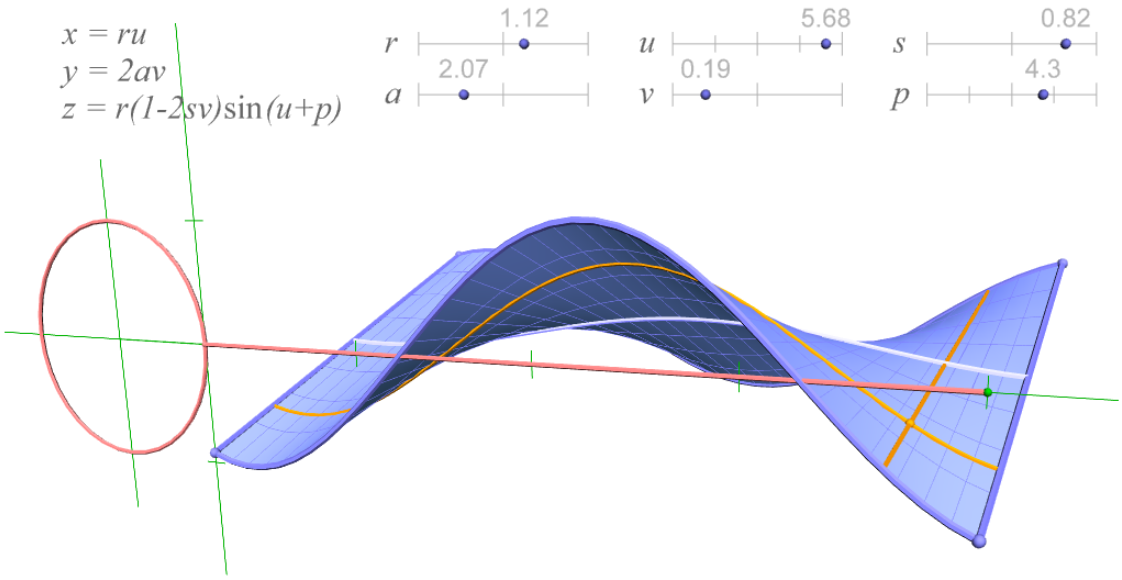


Figure 5. Parametric sliders used for better understanding of the relationship between the shaping of the surface and its mathematical expression.

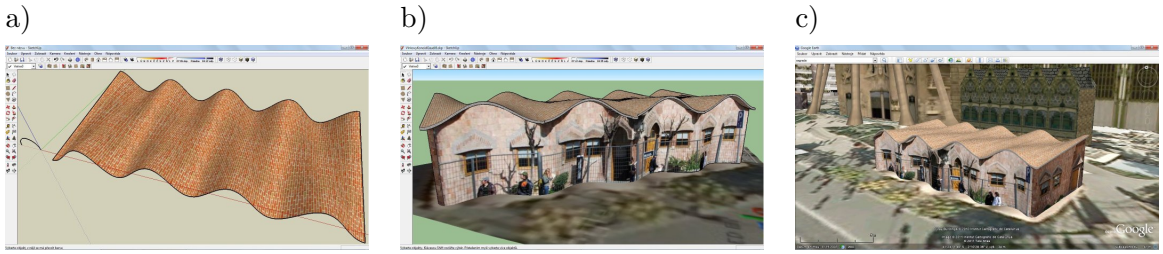


Figure 6. A model of the surface (a), its implementation in a real building using Google SketchUp (b) and sharing of the model in Google Earth application (c).

History of the surface

The right wavelet conoid was first used by the great Catalan architect Antoni Gaudí *(1852-1926)* and this is why the surface is sometimes called Gaudí's surface. The Temple of the Holy Family in Barcelona, called Sagrada Família in Spanish, is a famous work of Gaudí and several different kinds of ruled surfaces are implemented in this temple.

In 1906 a roof of a storehouse for plaster models was built near the Sagrada Família building site. The shape of the roof was a sine conoid (Figure 7a and in the background of Figure 7b). At the beginning of the Spanish Civil War in 1936 the whole complex of the buildings burned out and never been reconstructed to its original form.

Gaudí's temporary school building is much more famous – it was intended for children of Sagrada Família workers (Figure 7b). It was built in 1909, burned out in 1936, was reconstructed afterwards, burned out again in 1939 and was reconstructed again. Due to the proceeding work on the temple it was decided to remove and completely rebuild the school building – this was done in 2002 (Figure 7c and the model in Figures 6bc). There was also constructed a replica of the school building in a nearby town of Badalona.

What is interesting is that Gaudí used the conoidal shape not only for the roof but for walls, too. This allowed him to reduce the building costs to a great extent – it cost about 9000 pesetas at that time; current price of the building is estimated at 60 millions of pesetas.

Another world-renowned Spanish architect Santiago Calatrava *(*1951)* extended Gaudí's

a) b) c)

Figure 7. Gaudí's storehouse, studio and workshop (a); the original (b) and reconstructed (c) school building.

original idea into the motion using the sine phase shift. In 2000 he installed a moving sculpture called "Wave" in front of the Meadows Museum in Dallas (Figure 8a) and in 2004 he used the same idea as the "Nations Wall" for the Olympic Sports Complex in Athens (Figure 8b). In 2001 Calatrava designed a static variant of the right wavelet conoid as a roofing of Bodegas Ysios winery building in northern Spain (Figure 8c).

a) b) c)

Figure 8. The moving sculpture "Wave" in Dallas (a), the "Nations Wall" at the Olympic Sports Complex in Athens (b); the Bodegas Ysios winery building in northern Spain (c).

We can also find one implementation of this surface in the Czech Republic: a roofing of several buildings in Landal Marina holiday resort near the Lipno nad Vltavou village. In front of the central building the wavelet-sine curves coming from both sides rise to form a traditional Dutch female hat, a signature of Dutch studio Factor Architecten, designer of the building (designed in 2003, Figure 9).

Conclusion

This paper aims to show a complex illustration of using efficient geometric experience for effective technical practice, e.g. for magnificent architecture. Also it is a resume of the most relevant resources – historical, structural and purely mathematical.

All these results were compiled in the form of an attractive web presentation including the virtual 3D model, the guidance video and the list of realizations with many useful links for further study.

References

Crippaová, M.A.. *Antoni Gaudí: Od přírody k architektuře.* Praha: Taschen/Slovart, 2005. 96 s. ISBN 80-7209-674-5


Figure 9. Summer and winter photos of conoidally roofed buildings near Lipno nad Vltavou.

Giralt-Miracle D. *Gaudí. La búsqueda de la forma* [online]. Ayuntamiento de Barcelona, 2002 [cit. 2011-05-31]. <http://www.seacex.es/Spanish/Publicaciones/37/gaudi_creditos.pdf>

Las Escuelas Provisionales de la Sagrada Familia. In *Aproximaciones Geométricas* [online]. [200?] [cit. 2011-05-31]. <http://s3.amazonaws.com/mcneel/misc/docs/In_Situ_18.pdf>

Piska, R.; Medek V. *Deskriptivní geometrie II.* Praha/Bratislava: SNTL/SVTL, 1966. 316 s. ISBN 04-025-66

Santiago Calatrava [online]. [201?] [cit. 2011-05-31]. <http://www.calatrava.com>

Serrano, J.G.; Gómez, J. *L'obrador de Gaudí* [online]. Edicions UPC, 1996 [cit. 2011-05-31]. <http://books.google.com/books?id=rWedd1pr0igC>

Vítková, L. *Umělec* [online]. [2006] [cit. 2011-05-31]. Marina Lipno — Přístav uprostřed lesů. <http://www.divus.cz/umelec/article_page.php?item=1358>

Zerbst, R. *Antoni Gaudí i Cornet - život v architektuře.* Praha: Slovart, 2010. 239 s. ISBN 978-80-7391-388-5

Žára, J. *VRML 97: Laskavý průvodce virtuálními světy.* Brno: Computer Press, 1999. 238 s. ISBN 80-7226-143-6

WDS'11 Proceedings of Contributed Papers, Part I, 78–84, 2011. ISBN 978-80-7378-184-2 © MATFYZPRESS

Graph Theory at Czech Grammar Schools

D. Lessner
Department of Software and Computer Science Education, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic.

Abstract. Graph theory is not being taught on Czech grammar schools, just as Computer science. The reasons seem to be rather historical than rational. In this contribution, we discuss the usefulness of elementary Graph theory for grammar school students. Then we formulate possible targets and objectives of this education. Core of this contribution is topological sorting. It is an example of an advanced topic in graph theory, which is still comprehensible for grammar school students. Moreover, it is potentially useful in their life also out of school. We explain the problem, its applications, and three approaches to the solution along with comments regarding introduction of such topic into grammar schools. Last part of this contribution explains the critical path method, another advanced topic strongly bond to the idea of representing tasks as graphs.

Introduction

Computer science (CS) is not included in Czech grammar school[1] programmes. We believe it to be a mistake. Our research addresses this issue [*Lessner*, 2010]. We have identified graph theory as one of the main topics of a basic CS course, respecting both their scientific and educational value [*Lessner*, 2011]. In this contribution, we provide some more details to this specific topic. It shows our point of view on graphs for the sake of secondary education. The next step involves experimantal teaching in cooperation with a few grammar schools. As this is at its beginning, we do not suggest any particular way to teach graphs. It still needs evaluation and adjustments. However, some hints may be found in "The solution" part.

We begin this contribution clarifiyng what we consider to be the basics of graph theory. For grammar school students the classical $G = (V, E)$ approach is rather confusing (with G standing for the graph, V for vertices and E for edges, i.e. pairs of vertices). What they really need to know is that graphs are structures to represents relations between objects and that it is useful to try to draw them. Then we examine the situation of graph theory on grammar schools. Graphs are not included in the Framework Education Programme (fundamental curricular document for our grammar schools, [*VÚP*, 2007]) nor other programmes ([*CERMAT*, 2010]) explicitly[2], yet they are used in many areas intuitively. Sadly, there is no awareness of some unifying theory which would allow knowledge transfer between different areas using graphs. After explaining some more reasons why graph theory should be introduced to grammar school students we describe the main educational targets.

After these preparations, we describe a more advanced topic in closer detail. We have chosen topological sorting for this purpose. The problem is to find an order of vertices in oriented graph which would respect the edges orientation. The usual idea is that vertices are tasks and edges their dependencies. We will show three approaches to solve this problem and discuss the interesting points from the grammar school point of view. Developing the problem of topological sorting into an even more useful idea we introduce the notion critical path, as well as a way of finding and using it. At the end we summarize the whole contribution.

[1] Secondary education, i.e. 15 – 19 years old in their 10. – 13. year of study.

[2] This fact is no obstacle. We have a two level curricula system – schools create their own programmes based on the Framework and they are free to extend it as they like. That is why we do not recommend any specific way to include CS – conditions on individual schools differ.

Graph theory on grammar schools

In this section we discuss graph theory education on grammar school level in Czechia. At first we briefly revisit graph theory itself. Then we look into the recent situation on our grammar schools regarding graph theory. Then we can discuss the reasons why to teach about it as well as specific educational targets.

Here we revisit the basics about graph theory. At first we shall make clear that graph theory in computer science is not about graphs of functions or charts. Graph is an abstract structure to represent relations between pairs of objects. This is very general and leads to extreme applicability in a wide range of areas. Among the advantages we see also visuality. Formally, graphs are sets of vertices and edges (i.e. pairs of vertices). This would be represented by some sequence of symbols. However, it is much more useful for general population to represent graphs (i.e. relations between objects) as drawings. This way one can see much deeper into the structure of represented relations and find much easier what he seeks.

Among the basic notions to study are graph, vertices and edges of course, including orientation, degree and score. Higher terms are path, cycle, coloring and components. There are some basic graph types we distinguish, mainly complete graphs, trees and regular graphs. An important topic is graph representation for the sake of further algorithmic processing, e.g. in computers, where drawing is not really helpful. Among processing we are interested in various types of systematic traversing and searching. These may also lead to solve more advanced problems, like topological sorting, described later in this contribution.

Application of graphs involves nearly any area which deals with some kind of binary relations between objects. That is actually almost any area of human activity. Classic application is in maps an plans, where vertices represent places (e. g. towns) and edges their connections (e. g. roads). This is a very helpful metaphor used to think about graphs. Another applications rise from the same idea. We do not connect only places on a map, there are also pipes and tanks or electric circuitry. More applications are mentioned in the following subsection.

Graph theory in official curricula

Let us see the situation of graph theory on Czech grammar schools. Shortly, it is not there. This is surprising considering the usefulness of this topic and situation abroad [*Gal-Ezer et al.*, 1995]. On the other hand, it is no surprise at all, if we consider the overall situation and historical development. Last years, maths related education is going through severe reduction of both hours and topics to teach. Interested teachers and researchers focus rather on saving what is left than on introducing new topics. This shows also the historical background. Graph theory has simply never been a part of grammar school programmes. The field itself, regardless of how rich and important results, is much younger compared to the classical parts of grammar school mathematics.

Further, should it be officially included, it might be of no effect at all. Without introducing CS itself, graph theory would most likely become a part of mathematics. The most important approach to the structure of mathematical programme in Czechia is respecting historical development of mathematics. Therefore graphs would come after the other topics. In practice, the last chapters are being taught usually only in specialized seminars for students who have chosen mathematics as to be a part of their school leaving exams or might need advanced mathematics for their university studies. These seminars focus mainly on the most important chapter of the end of grammar school level mathematics, which is differential calculus. Next to it, graph theory would be very likely left for self-study.

Although graph theory is not explicitly mentioned anywhere in the grammar school curriculum, we may look for its usage, even in seemingly unrelated subjects. The finding is disturbing. Graphs are being used widely throughout virtually every school subject on intuitive level. We will mention only a few blatant and obvious examples. In organic chemistry, students draw molecular structures. Simple math is used to find formulas which can not exist (given the num-

ber of atoms and their connectivity). In biology, taxonomy and genetics uses trees. Metabolic network is a graph. Royal family trees are needed in history for some nontrivial cases. Usage in geography and cartography is clear. In language, syntactic structure of a sentence is represented using a tree. Introduction to combinatorics is based on routes and intersections. Last application to mention is one of the recent hits in primary and secondary education in Czechia, mind mapping. For a computer scientist, mind map is nothing special but a tree modification.

Despite of all these examples students encounter, they are unaware of any unifying theory behind. This limits their ability to transfer any knowledge or skill (e. g. tree characteristics or systematic searching) to some different area.

Reasons for including graphs into education at grammar schools

We have described applications of graphs at grammar schools. Considering how frequent they are, they stand as one of the reasons why to include graph theory into grammar school programmes. Here we describe also other reasons for it. Graphs represent a new structure to work with, different from numbers or other classical structures from grammar school mathematics. Graphs seemingly require a different way of thinking. This might provide poor skilled students with a new chance to catch up in mathematics. Even though the structure differ and it may be mostly about drawing, the reasoning behind is the same as everywhere else.

The same idea works with algorithms here. They seem to be somehow different, working with vertices and edges instead of numeric values. Yet, this is no fundamental difference here. Graphic algorithms give the opportunity to notice this and therefore understand better what an algorithm itself is.

It is important to remark that even though the basics are very intuitive, graph theory naturally provides advanced topics of various difficulty. Most of them have their important applications as well as theoretical impact. One of the many examples would be the notion of planarity – a natural and useful one, yet not at all trivial. Many students think of graphs as of their planar drawing. This is no mistake, yet they shall know this approach has its limits. It is crucial to discover that many different drawings of one graph exist and therefore a drawing is nothing more than a visualisation.

It is also important to experience that there are graphs with no correct drawing at all, yet algorithms (e.g. searching) work in the very same way. Another such topic is isomorphism of graphs. We believe it to be at the limits of grammar school students capabilities.

Concluded, the reasons for teaching graph theory is in their practical usefulness as well as in their challenging theoretical background.

Education targets

Here we introduce shortly the main targets specific for graph theory education, as we have developed them so far. They are compatible with the official grammar school programme [*VÚP*, 2007]. To begin with, students shall recognize the appropriate situations to apply graphs (i.e. not regarding city plans only), and they shall also apply them. This means they shall be able to read relations from given drawn graphs as well as denoting them (both by drawing and symbolically). They shall be able to use some unified terminology. This does not necessarily mean the official terms, especially in cases they are not natural in our language (as “graph” itself). The point is smooth communication when using graphs.

Furthermore, they should be able to use drawing and sketching to their advantage, yet understand the unavoidable need to rely on reasoning in certain situations. Further processing includes translating notations (natural language description, drawing, list, matrix) and systematic traversing. This does not mean they shall know the exact way of depth first search for example. The point is to make them understand a few key points which will make their approach systematic (e.g. marking open and closed vertices) and which will enable them reinvent the algorithm if needed.

On higher level, they should be aware of some meta knowledge and heuristics, and be able to use them appropriately. They should be aware that there are many ways to achieve the same goals (regarding both graph drawing and its algorithmic processing). This goal however does not imply these ways themselves to be the same as well. They may differ significantly, mostly in difficulty and efficiency. Influenced by all the examples in graph theory, students shall know it is good to work systematically and in phases, virtually every time when it is possible.

Further, they shall understand the importance of preprocessing data and other preparations preceding work itself. One of the many examples is topological sorting, discussed below in this contribution. On the same example, they shall learn that a greedy approach may work well. By greedy we mean fixating partial solutions and no turning back (unlike in backtracking). This usually require some more work with choosing which partial solution to pick. Usually the least injuring one is found and picked. Still, they need to understand (preferably by experience) that such an argument is not necessarily correct and they shall rather seek guarantee of a partial solution correctness (and guaranteed inclusion in a correct and complete solution). The greedy approach itself may turn out to be useless.

Educational targets described above combine both practical usability of graphs and theoretical knowledge. These two shall strengthen each other synergistically. Practice leads to problems, solved using theoretical results, enabling more practice, and so on. This enhances students lives, providing them with tools to solve problems (using graphs), and make them understand the importance of quality theoretical thinking at the same time.

Example: Topological sorting

We have chosen the problem of topological searching as an example for this contribution. It represents a problem which is understandable, challenging, yet well solvable and widely applicable. It is well known, therefore we will not dive into exact technical details. We rather save space for what is interesting in grammar school context. Detailed explanations can be found in [*Kučera*, 1983] and [*Töpfer*, 2007]. In this section we introduce the problem and the main motivation for its solving, three main approaches to find the solution and a few further topics.

The problem

On input we have a directed graph. We search for an order of vertices which would respect edge orientations. More formally, for a given $G = (V, E)$, we search for a $\phi : V \to \aleph$, such that $(v_1, v_2) \in E \Rightarrow \phi(v_1) < \phi(v_2)$. The usual motivation for this problem is about tasks (represented as vertices) and their time dependencies (edges), such as getting dressed, building a villa or constructing a plane. It is not wise to put on shoes before socks, yet shoes are without any direct relation to a tie. With a complex project, it is easier to find these dependent pairs than the overall order of tasks.

Such a motivation is relevant to grammar school students lives. Not only do they encounter such problems occasionally, it is also important that the problem setting is comprehensible for them. Moreover, they may expect such (and more complex) problems in their future.

The solution

We will discuss three main approaches to solve the problem of topological sorting. They differ in many aspects, including efficiency. The first approach to consider is based on brute force. That would be an average student's first choice to examine. Try a few permutations of vertices and examine whether they satisfy given conditions (partial order implied by the graph). Obviously, with $n!$ possible orders, this may take very long, and moreover, the student will soon feel that he examines needless number of possibilities. He might notice some repeating patterns or even partial solutions. Trying to combine and complete them may lead to a solution, as well

as to a few questions: How can we be sure it is correct? How do we do this systematically with different input data?

During their work on a specific input, they might discover parts of their solution which are not going to change anymore. Specifically, they should notice the special characteristics of the first (or last) task to be done: they have no predecessors (or successors). This may lead to find the desired order systematically instead of a sequence of trial and errors. Excluding already ordered vertices from the graph (as well as incidental edges), we may proceed recursively. The graph is smaller, the idea is the same. Task with no predecessor may be put as the first and this causes no mistake. This can be found out by a grammar school student, given there is enough time, appropriate sample problems and careful guidance by the teacher. From this, the algorithm may be formulated quite easily. These observations gave us also a necessary condition for existence of any solution. Whenever there is no vertice to be chosen (i.e. with no predecessor), there is no task to start with and therefore no solution at all.

This approach may still be refined. Searching for the first vertice (with zero predecessors) is unnecessarily time consuming if not thought through and with no preparation in each step. This is useful to be experienced by students: Having a smart idea leading to a solution undoubtedly, yet not efficiently. At the same time, it is a trap for students more experienced in programming and data structures.

We want to find the candidates quickly, so one of the ideas could be to sort the vertices according to their incoming degree and keep a sorted list. Actually, a more efficient way can be found, because only the first vertice is needed. Therefore some kind of heap could serve us well. However, heap still means wasting resources, because we know exactly the needed degree of the first vertice. Finally, it is enough to keep a list of candidates (vertice with no predecessor) and the incoming degree for each vertice. Keeping these two structures up to date can not be faster (we will always need to do at least one operation per vertice and per each of its edges).

The last possibility we describe here is unlikely to be found by students, as it is not very natural. It is based on depth first search (DFS). Therefore it may help to revise DFS. Moreover, this approach may illustrate one of the key concepts in computer science: try to reuse what we already know for solving a new situation. In this case however, it is not straightforward.

DFS has to be run on a graph with reversed edges. First it finds vertices with no predecessors (along edges) of this modified graph. Then DFS is run from these vertices one by one. Order of closing vertices is the demanded topological order. If an open vertice is encountered anytime, the original graph contains a cycle and therefore there is no correct topological order. Time complexity of the last two approaches is proportionate to the number of vertices and edges, if appropriate data structures are used.

Further topics

The topic of topological sorting brings an excellent opportunity to seek some further matter – both theory and applications. Orders in general may be introduced, as well as lattices and Hasse diagrams. The strongest topic would probably be deepening planning and scheduling. There is an important counterpart to this out of CS. One of the major areas of project management is basically planning and scheduling without the algorithms. We will mention here one possible and fruitful topic, the critical path method (CPM). For this we need to consider also time consumed by individual tasks, so we add it as weights to vertices.

Finding an order for given set of tasks with their dependencies is certainly useful, yet often insufficient in real life. Tasks can often be accomplished in parallel. This of course does not mean they can be accomplished all at once, there still are their dependencies. A reasonable question in this situation is about the minimum working time to accomplish all tasks. Length of the critical path is the answer. Also, this is a way to define critical path. Looking into it, we would find out it is the longest possible sequence of vertices connected by edges. Critical path is a good example that some useless looking problems, such as searching for the longest path in

a graph, may actually be of crucial importance in certain situations.

The total time taken by the project can not be shorter for a simple reason. Each task on the path has to be accomplished, in a time given by its weight. And it has to be accomplished after its direct predecessor and before the direct successor, not along any of them. This holds for all vertices on the path. This shows us the importance of finding all critical paths, not only the length. Should any task on any critical path be delayed, the whole project will inevitably end late. We should remark here that all this is not dependent on the topological order possibly found earlier, CPM only needs the graph itself.

Finding a critical path is surprisingly easy and comprehensible, and therefore suitable for a grammar school student with a very little computer science experience. We will calculate the shortest time each and every vertice may be finished, one by one, in the topological order. For a vertice with no predecessor, the time needed to finish is obviously only the time for the task itself. Whenever we take a vertice in topological order, all its predecessors have been processed already, therefore we know the earliest moment they could be accomplished. The earliest moment to finish the taken vertice one is obviously the time of its last finished predecessor plus the time for the vertice itself. This way, we simply run through the graph and end with an answer at hand. The educational benefit of this approach is the opportunity to reuse many crucial ideas from topological sorting.

Further work with real-life situations would show drawbacks of CPM, e.g. the lack of robustness against inaccurate or changing input data. This finding leads to the critical chain method. Another way to modify the problem and seek for another solution could be program evaluation and review technique (PERT) or hierarchical tasks using work breakdown structure (WBS). Of course, classically scientific questions may also be solved in the classroom, like what is the total number of all topological orders on given graph.

Summary

In this contribution, we have proposed a new topic for secondary education – graph theory. We have examined the current situation, shown why we consider it important and what benefits of including it do we expect. Graphs are a very helpful tool for dealing with relations between pairs of objects. This is the way we want grammar school students to see graphs. They shall be able to use graphs this way, including some basic traversing and searching.

After this we discussed topological sorting in detail. It represents an advanced topic, yet comprehensible for students and moreover, very useful. Many important aspects of graph theory and computer science may be illustrated, revisited and experienced using this problem. In the last part of this contribution we showed a possible continuation and extension of this topic. The critical path method is again a very useful concept, yet comprehensible by students and using some elegant ideas and reasoning.

Grammar school students are already using simple graphs, intuitively. Yet, they are not aware of any special science behind. We believe that showing it to them would improve their lives. They would be able to transfer and use their graph related knowledge and skills in new areas. They would also understand better the limits of drawings and common sense based approach.

Acknowledgment. The author thanks his supervisor RNDr. Tomáš Holan, Ph.D., for all the inspiration and support of works on this paper.

References

CERMAT: *Katalog požadavků zkoušek společné části maturitní zkoušky ZKUŠEBNÍ PŘEDMĚT: INFORMATIKA, vyšší úroveň obtížnosti.* Praha: Centrum pro zjišťování výsledků vzdělávání, 2010.

Gal-Ezer, J., Beeri, C., Harel, D., Yehudai, A., A High-School Program, in *Computer Science.* Computer, 1995, 28, 10, pp. 73-80.

Kučera, L.: *Kombinatorické algoritmy*, SNTL, Praha 1983

Lessner, D.: Computer Science Education on High Schools, in *WDS'10 Proceedings of Contributed Papers: Part I - Mathematics and Computer Sciences*, Prague, Matfyzpress. 2010, pp. 110–115.

Lessner, D.: Computer Science Curriculum Proposal for Czech Grammar Schools, in *Informačné Technológie – Aplikácie a Teória. Zborník príspevkov, ITAT 2011*, in printing.

Töpfer, P.: *Algoritmy a programovací techniky.* Prometheus, Praha 2007.

VÚP: *Rámcový vzdělávací program pro gymnázia.* Praha: Výzkumný ústav pedagogický v Praze, 2007.

The Change of Paradigm in the Matrix Theory

M. Štěpánová

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. The paper deals with the process of gradual acceptance of the matrix theory by mathematical community. This process began in the second part of the 19th century and continued till the second part of the 20th century. Until then, a majority of mathematicians formulated their results, which are presently well-incorporated in the matrix theory, in terms of determinants, bilinear and quadratic forms. The process resulted in new ways of expressing these results, requiring new terminology, new symbolism.

Introduction

More important than the question when this or that theorem or definition of the matrix theory was published for the first time is, perhaps, the question where we can find appropriate notions, notations and roots of ways of thought about problems that give the keystone on which the theory of matrices has built up.

Trace of the matrix theory goes back before Christ. The beginnings of this theory are adherent to solutions of systems of linear equations. The development of the theory of matrices is closely connected historically with the theory of bilinear and quadratic forms and above all with the theory of determinants. Many notations of the matrix theory are regarded as a product of the usage square and oblong schematics in the theory of the determinants and solutions of systems of linear equations.

First notions of matrix theory

We customarily consider the article *A memoir on the theory of matrices* to be the origin of the matrix theory. This article was published by the English mathematician and lawyer Arthur Cayley (1821–1895) in 1858. The theory of determinants surprisingly predated the theory of matrices. The year 1750 is regarded as the origin of the theory of determinants. For more information, read [*Bečvář*, 2007] or [*Štěpánová*, 2010].

Some notions of the matrix theory were known before the origin of this theory. Mathematicians used different terms for them than we use.

Firstly, we can find such notions in geometry. Equations of conics and quadrics were transformed to the sums of squares in the 17th century. For example, the French mathematician, scientist and philosopher René Descartes (1596–1650) studied these transformations. Some others tried to transform symmetrical bilinear and quadratic forms to canonical forms in the 18th century. Secondly, notions of the theory of matrices were taken in celestial mechanics. This branch studied *secular equations* (we call them *characteristic equations*), which described relationships between the planets. Astronomers and mathematicians studied roots of secular equations and their properties (multiplicity, eigenvector, ...) at that time. They knew roots of every secular equation (corresponding to a symmetric matrix) are real.

First square and oblong schematics

Oblong schematics were used by Chinese mathematicians two thousand years ago. Schematics appeared in the method so-called *the algorithm fang cheng* which solves the systems of linear equations and which is very similar to the Gaussian elimination method. For more information about this algorithm, read [*Hudeček*, 2008], [*Bečvář*, 2007] or [*Štěpánová*, 2010].

Apparently, the German mathematician and astronomer Carl Friedrich Gauss (1777–1855)

applied square schematics in mathematics for the first time. He used the square scheme to express the linear substitution in his masterpiece *Disquisitiones aritmeticae* in 1801. Moreover, he compounded two substitutions and represented the result of this operation by the scheme again. We find out (comparing the coefficients) that the scheme corresponds to the matrix multiplication which we use nowadays.

The German mathematician Ferdinand Gotthold Eisenstein (1823–1852) denoted the ternary quadratic form briefly by the square scheme for the first time. He wrote it in his article *Neue Theoreme der höheren Arithmetik* in 1847. It is a matrix of a form in the present sense. Eisenstein also used a square scheme to express a linear substitution.

The origin of the notion and the term matrix

The notion *matrix* was built up in 1840s and 1850s in several articles which were written by A. Cayley and his friend James Joseph Sylvester (1841–1897). A. Cayley wrote about *the theory of matrices* for the first time and voiced his opinion that the matrix theory should precede the theory of determinants in his article *Remarque sur la notation des fonctions* in 1855. As already mentioned, in 1858 he published *A memoir on the theory of matrices* in which he defined the notion *matrix* for a square array of numbers. It was J. J. Sylvester in 1850 who established the conception of a rectangular matrix and used the term *matrix* for the first time in his article *Additions to the articles "On a new class of theorems", and "On Pascal's theorem"*. He defined a notion *a nullity of a matrix* in his article *On the properties of a split matrix* in 1882. A nullity of a matrix is an example of archaism. A nullity was defined only for a square matrix. The sum of the rank and the nullity of the square matrix is equal to the order of it.

The situation after the the year 1858

Cayley's *Memoir* piqued little immediate interest in Great Britain. Moreover, it was practically unknown elsewhere. A majority of mathematicians who worked in algebra focused their interest on the theory of determinants in the second half of the 19th century and they did not accept matrix theory for a long time. It resulted in expressing results, which belong presently to the theory of matrices, in terms of determinants and in terms of bilinear and quadratic forms as well. There were a few mathematicians who used symbolism of matrices in the second half of the 19th century. The most important were A. Cayley, J. J. Sylvester, Arthur Buchheim (1859–1888) and Eduard Weyr (1852–1903). Moreover, A. Buchheim died prematurely.

Eduard Weyr was an exception on the European continent. He published seven works on matrices in 1880s. He introduced his very modern and original theory of canonical forms which we called the *Weyr's theory of characteristic numbers*. As already mentioned, he used formulations in terms of matrices. Weyr was one of the first mathematicians who contributed to connections the matrix theory with the theory of bilinear and quadratic forms. For example, see [*Weyr*, 1889].

We will introduce the process of acceptance of the matrix theory. We will present several quotations which will demonstrate a slow adoption of new terminology, symbolism and approach to this matter.

The German mathematician Georg Ferdinand Frobenius (1849–1917) formulated his famous results, which are well-incorporated in the matrix theory nowadays, in terms of bilinear and quadratic forms for a long time.

In 1879, in two articles *Über homogene totale Differentialgleichungen* and *Theorie der linearen Formen mit ganzen Coefficienten*, he defined a rank of a square matrix and a rank of a rectangular matrix without matrix terminology.

Wenn in einer Determinante alle Unterdeterminanten $(m+1)$*ten Grades verschwinden, die mten Grades aber nicht sämmtlich Null sind, so nenne ich m den Rang der Determinante.* ([*Frobenius*, 1968], I., p. 435)

Gegeben sei ein endliches System $\mathbb{A}$ *von Grössen* $a_{\alpha\beta}$ $(\alpha = 1, \dots m;\ \beta = 1, \dots n)$*, die nach Zeilen und Colonnen geordnet sind. Wenn in demselben alle Determinanten* $(l+1)$*ten Grades verschwinden, die lten Grades aber nicht sämmtlich Null sind, so heisst* l *den Rang des Systems.* ([*Frobenius*, 1968], I., p. 484)

Georg Ferdinand Frobenius accepted the language of matrices in 1896 when he published the article *Über vertauschbare Matrizen*. On the first page of his work, he wrote :

Ich werde mich daher hier der symbolischen Bezeichnung für die Zusammen setzung der Matrizen (Formen) bedienen, die ich in meiner (im Folgenden mit L. citirten) Arbeit Über lineare Substitutionen und bilineare Formen ... auseinandergesetzt habe. ([*Frobenius*, 1896], p. 601)

To realize the change of Frobenius's style of writting before and after 1896, the following list of five and five works can be used.

Ueber lineare Substitutionen und bilineare Formen, *Theorie der linearen Formen mit ganzen Coefficienten*, *Ueber die Elementarteiler der Determinanten*, *Ueber die congredienten Transformationen der bilinearen Formen*, *Zur Theorie der Scharen bilinearer Formen – Über Matrizen aus positiven Elementen*, *Über Matrizen aus nicht negativen Elementen*, *Über die mit einer Matrix vertauschbaren Matrizen*, *Über unitäre Matrizen* and *Über den Rang einer Matrix*.

A rank of a matrix

The English mathematician William Kingdon Clifford (1845–1879) wrote 5-page work titled *A fragment on matrices*, which was published posthumously in 1882. He separated 3×3 singular matrices into *matrices indeterminate in the second degree* (nowadays a matrix of a rank 1) and *matrices indeterminate in the first degree* (a matrix of a rank 2). We can consider this formulation to be the prime beginning of the notion of a rank of a matrix. Of course, Clifford used the language of determinants:

... An indeterminate matrix (or more definitely, a matrix indeterminate in the first degree) is a matrix the determinant of which vanishes, but for which the first minors do not all of them vanish ... A matrix indeterminate in the second degree is a matrix for which all the first minors vanish, or what is the same thing, one for which the second and third rows are mere multiples of the first row. ([*Clifford*, 1968], p. 337)

As mentioned above, the definition of the rank of a matrix was written by G. F. Frobenius in 1879 but he did not use the language of matrices. The definition of the nullity of a matrix was given by J. J. Sylvester. Although Sylvester was one of few mathematicians acquainted with the matrix theory and working in it, he formulated the definition of the nullity in terms of determinants:

... the nullity of a matrix of the order ω *being regarded as unity, when its determinant simply is zero, as* 2 *when each first minor simply is zero, as* 3 *when each second minor is zero ... as* $(\omega - 1)$ *when each quadratic minor is zero and as* ω *(or absolute) when every elements is zero.* ([*Sylvester*, 1904–1912], III., p. 646)

The theory of systems of linear equations

Not surprisingly, the notion of the rank of a matrix is closely connected with the theory of systems of linear equations.

Henry John Stanley Smith (1826–1883), who was a British mathematician, defined notions *an unaugmented array* and *an augmented array* in 1861 in his work titled *On systems of linear indeterminate equations and congruences*.

The English mathematician, writer and photographer Charles Lutwidge Dodgson (1832–1898), who is better known by the pseudonym Lewis Carroll as the author of *Alice's adventures*

in wonderland, called the same notions by the terms *V-Block* and *B-Block* in the book *Elementary Treatise on Determinants with their Application to Simultaneous Linear Equations and Algebraical Geometry* in 1867. He used the term *principal Minors* for subdeterminants of a matrix that have the highest order as possible and the term *evanescent* for a matrix whose *principal Minors* are equal to zero. He wrote:

If there be given n Equations, not all homogeneous, containing Variables: a test for their being consistent is that either, first, there is one of them such that, when it is taken along with each of the remaining Equations successively, each pair of Equations, so formed, has its B-Block evanescent; or, secondly there are m of them, where m is one of the number 2.....n, which contain at least m Variables, and have their V-Block not evanescent, and are such that, when they are taken along with each of the remaining Equations successively, each set of Equations, so formed, has its B-Block evanescent. ([*Dodgson*, 1867], p. 61)

Dodgson expressed (not very briefly and understandably) a necessary and sufficient condition for a solvability of a system of linear equations. Nowadays, it is often called *Frobenius theorem.* Dodgson defined *a Block* (a matrix) and then he created its *Minor* (a subdeterminant):

If mn quantities be so placed as to form m rows and n columns: they are said to form a Block; and the mn quantities are called the Elements of such a Block. ...

... If, in a given Block, any rows, and as many columns, be selected: the square Block formed of their common Elements is called a Minor of the given Block. ([*Dodgson*, 1867], p. 6–7)

Italian mathematicians Alfredo Capelli (1855–1910) and Giovanni Garbieri (1847–1931) expressed a necessary and sufficient condition for a solvability of a system of linear equations without using the theory of determinants.

Capelli published his article *Sopra la compatibilità o incompatibilità di più equazioni di primo grado fra più incognite* in 1892. He was the first mathematician, who briefly and understandably formulated a necessary and sufficient condition for a solvability of a system of linear equations:

Dato un sistema qualunque di m equazioni di 1° *grado con n incognite*

$$\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n & = & \alpha_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n & = & \alpha_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & & \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n & = & \alpha_m \end{array}$$

affinchè esso sia compatibile con valori finiti delle incognite è necessario e sufficiente che le due matrici

$$(A)\quad \begin{array}{cccc} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{array} \qquad e \qquad (B)\quad \begin{array}{ccccc} a_{11} & a_{12} & \ldots & a_{1n} & \alpha_1 \\ a_{21} & a_{22} & \ldots & a_{2n} & \alpha_2 \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ a_{m1} & a_{m2} & \ldots & a_{mn} & \alpha_m \end{array}$$

abbiano la stessa caratteristica. ([*Capelli*, 1892], p. 55)

Literature on matrices

The absence of any widespread use of the matrix algebra during the 19th century caused that all over the world there was no literature on matrices before 1900. Of course, we can find some exceptions. In books, the notion of a matrix usually appeared together with another notions (for example a bilinear and quadratic form, a determinant, a linear transformation, a system of linear equations, ...).

The situation changed after 1900. *Introduction to Higher Algebra*, for example, is one of the first textbook on matrices. It was published by the American mathematician Maxime Bôcher (1867–1918) in 1907. Nevertheless, he started with the notion determinant and then he precisely defined the notion of a matrix.

We assume that the reader is familiar with the determinant notation, and will merely recall to him that by a determinant of the nth order

$$\left\| \begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array} \right\|$$

we understand a certain homogeneous polynomial of the nth degree in the n^2 elements a_{ij}. By the side of these determinants it is often desirable to consider the system of the n^2 elements arranged in the order in which they stand in the determinant, but not combined into a polynomial. Such a square array of n^2 elements we speak of as a matrix. In fact, we will lay down the following somewhat more general definition of this term:

DEFINITION 1. *A system of mn quantities arranged in a rectangular array of m rows and n columns is called a matrix. If $m = n$, we say that we have a square matrix of order n.* ([*Bôcher*, 1907], p. 20–21)

Subsequently, he stressed the difference between a matrix and a determinant:

Even when a matrix is square, it must be carefully noticed that it is not a determinant. In fact, a matrix is not a quantity at all, but a system of quantities. This difference between a square matrix and a determinant is clearly brought out if we consider the effect of interchanging columns and rows. This interchange has no effect on a determinant, but gives us a wholly new matrix. ...

Although, as we have pointed out, square matrices and determinants are wholly different things, every determinant determines a square matrix, the matrix of the determinant, and conversely every square matrix determines a determinant, the determinant of the matrix. ([Bôcher, 1907], p. 21)

Rudolf Hans Heinrich Beck (1876–1942) wrote his textbook *Einführung in die Axiomatik der Algebra* in 1926, in which matrices are in the whole text. Moreover, chapters on matrices are before chapters on determinants and the rank of a matrix is not defined by determinants.

Cuthbert Edmund Cullis (1875?–1954) wrote a three-volume textbook *Matrices and determinoids* (1913, 1918, 1925), which became the first book having the word *matrix* in its title.

First successful monographs on matrices were published in the 1920s and in the 1930s. We will list the most important of them: Herbert Westren Turnbull (1885–1961) wrote his monograph *The Theory of Determinants, Matrices and Invariants* in 1928, H. W. Turnbull and Alexander Craig Aitken (1895–1967) published their book *An Introduction to the Theory of Canonical Matrices* in 1932, Cyrus Colton MacDuffee (1895–1961) wrote his book *The Theory of Matrices* in 1933 and Joseph Henry Maclagen Wedderburn (1882–1948) published his monograph *Lectures on Matrices* in 1934.

Bohumil Bydžovský wrote his textbook *Základy teorie determinantů a matic a jich užití* [*The fundaments of the theory of determinants and matrices and their applications*] in 1930. It is the first Czech book and one of the first books all over the word, which has the word *matrix* in the title.

Acceptance of matrices by physic

Physics did not use square or rectangular schematics until 1925, when Werner Karl Heisenberg (1901–1976) published his work *Über quantentheoretische Umdeutung kinematischer und mechanischer Beziehungen*. He described a movement by observable physical quantities that have been expressed by the sets of complex numbers. Other physicists (E. Schrödinger, M. Born, P. Jordan, F. London, H. Weyl, J. von Neumann, P. A. M. Dirac, ...) then accepted the matrix language and symbolism.

Conclusion

Althought, Cayley's *Memoir*, which was published in 1858, is regarded as the origin of the matrix theory, the real early history of the matrix theory goes back two thousand years. Moreover, this part of linear algebra and its language has absorbed into the mathematical mainstream during the first half of the 20th century.

Cayley's *Memoir* was unknown for a long time. Many mathematicians preferred the symbolism and the terminology of the theory of determinants, the theory of bilinear and quadratic forms to matrix language during the second half of the 19th century. The matrix theory became an independent theory in the 1930s. We can see the change of paradigm in literature very well.

Nowadays, the theory of matrices is a standard tool in other branches of science. It has its own language: its own terminology and symbolism as well as ways of expressing results.

Acknowledgments. My thanks are due to my supervisor doc. RNDr. Jindřich Bečvář, CSc., for encouragement in the prosecution of this work and his help during my study.

This article was supported by GA ČR P401/10/0690 *Prameny evropské matematiky* and development project *Doktorské studium oboru M8*.

References

Bôcher, M., *Introduction to Higher Algebra*, The Macmillan Company, New York, 1907.

Bečvář, J., *Z historie lineární algebry*, Matfyzpress, Praha, 2007.

Capelli, A., Sopra la compatibilità o incompabilità di più equazioni di primo grado fra più incognite, *Rivista di Matematica*, 2, 54–58, 1892.

Cayley, A., A Memoir on the Theory of Matrices, *Philosophical Transactions of the Royal Society of London*, 148, 17–37, 1858.

Clifford, W. K., *Mathematical Papers*, Macmillan and Co., London, 1968.

Dodgson, C. L., *Elementary Treatise on Determinants with their Application to Simultaneous Linear Equations and Algebraical Geometry*, MacMillan and Co., London, 1867.

Hudeček, J., *Matematika v devíti kapitolách*, Matfyzpress, Praha, 2008.

Frobenius, G. F., Über vertauschbare Matrizen, *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften zu Berlin*, 601–614, 1896.

Frobenius, G. F., *Gesammelte Abhandlungen I.–III.*, Springer-Verlag, Berlin, Heidelberg, New York, 1968.

Sylvester, J. J., *The Collected Mathematical Papers I.–IV.*, Cambridge University Press, Cambridge, 1904–1912.

Štěpánová, M., From the Algorithm Fang Cheng to the Matrix Theory, in J. Šafránková and J. Pavlů (eds.): *WDS'10 Proceedings of Contributed Papers: Part I – Mathematics and Computer Sciences*, Prague, Matfyzpress, 127–132, 2010.

Weyr, E., *O theorii forem bilinearných*, Spisův poctěných jubilejní cenou Královské české Společnosti nauk v Praze č. II, Praha, 1889.

Josef Úlehla and His Calculus Textbook

L. Vízek

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. Josef Úlehla (1852–1933) was a Czech teacher of primary and secondary schools. His subject was mainly mathematics and natural sciences, but he was interested in many other areas and wrote about them too. This text describes the biography of Josef Úlehla and analyses the context, historical background as well as reviews of his calculus textbook entitled *Počet infinitesimální* (*Infinitesimal Calculus*).[1]

Introduction

Josef Úlehla was an important Czech teacher of primary and secondary schools and one of the pioneers of education reforms. He was a contemporary of T. G. Masaryk (*1850, the first Czech president) or Alois Jirásek (*1851, a Czech writer, an author of historical novels). He worked in the second half of 19th century and early decades of 20th century.

Josef Úlehla was born in Podivín on the 16th March 1852. During the studies he passed several primary schools around his home, then he continued at the grammar school in Strážnice and Brno. In summer 1872 Josef Úlehla finished his studies as a graduate of Teacher College in Brno, where he gained the authorization for teaching at primary and secondary schools. In the same year he started his pedagogical career. He taught at many primary and secondary schools in Moravia. In 1897 he became the director of a secondary school in Klobouky, at the same position he worked in Jaroměřice nad Rokytnou (1905) and in Strážnice[2] (also in 1905) until his retirement. In addition, he performed as an inspector of Czech schools in Vienna (1912–1914) and after the First World War, in time of the First Czechoslovak Republic, helped to establish the new school in Lipov.

Josef Úlehla died on 22th December 1933.

Josef Úlehla never taught at the university in his whole life. This fact highlights his professional work. We know him as an author of many pedagogical articles, which were published in Czech pedagogical magazines *Komenský*, *Pedagogické rozhledy*, *Učitel*, *Věstník Ústředního spolku učitelských jednot na Moravě* etc., and he also wrote a lot of monographs. For example we can mention his book *Dějiny mathematiky*,[3] which deals with the development of mathematics.

A few articles have been about the life and work of Josef Úlehla written, but recent history of mathematics is missing the overall mapping and evaluation of his profession work. This article discusses Úlehla's calculus textbook entitled *Počet infinitesimální* and puts it into the broader historical context.

Počet infinitesimální

Josef Úlehla published his textbook *Počet infinitesimální* in 1906. In 1944 the book was published again. First we describe some other similar books which were edited in that time.

Other contemporary textbooks of calculus

Fixing the Czech language in Czech lands in the 19th century benefited from the "success" of the Czech National Revival and became an important prerequisite for the publication of mathematical textbooks in the Czech language.

---

[1] Úlehla J.: *Počet infinitesimální*. Dědictví Komenského, Praha, I. ed., 1906, *Vyšší matematika bez učitele, počet infinitesimální*. Česká grafická unie, Praha, II. ed., 1906.

[2] Podivín, Strážnice, Klobouky, Jaroměřice nad Rokytnou are small towns in south Moravia, the area of Úlehla's life and work.

[3] Úlehla J.: *Dějiny mathematiky*. Dědictví Komenského, Praha, part I, 1901 and part II, 1913. This book discusses the article Vízek L.: *Josef Úlehla (1852–1933) a jeho Dějiny mathematiky*. In J. Bečvář, M. Bečvářová (ed.): 32. mezinárodní konference Historie matematiky, Jevíčko, 26. 8. až 30. 8. 2011, Matfyzpress, Praha, 2011, str. 275–284.

Figure 1. The main title of the first and second editions of the textbook.

The oldest comprehensive grammar school mathematics textbooks were published since 1860. The first and the only Czech study text of this kind on "higher mathematics," as the parts of mathematical analyse were called, is *Přídavek k algebře pro vyšší gymnázia*[4] (*The Addition to Algebra for Higher High Schools*) by Václav Šimerka (1819–1887). He published this thin book (only 56 pages) as an appendix to his grammar school textbook *Algebra čili počtářství obecné pro vyšší gymnasia*[5] (*The Algebra or else General Counting for Higher High Schools*). In the mentioned addition Václav Šimerka described the foundations of differential and integral calculus.

The first university study texts of mathematical analysis published František Josef Studnička (1836–1903) and Eduard Weyr (1852–1903). František Josef Studnička taught since 1864 at the Prague Polytechnic, there he published *Základové vyšší mathematiky*[6] (*The Basics of Higher Mathematics*), in which besides the basics of calculus wrote about differential equations and its applications in technical fields. Younger Eduard Weyr issued, for example, *Počet differenciálný*[7] (*Differential Calculus*), which used to be a substitution for the older textbooks by František Josef Studnička.

The analysis of *Počet infinitesimální* by Josef Úlehla

After textbooks, whose authors are mentioned above, Josef Úlehla published his own title. In the preface to this textbook he explained his motivation for writing:[8]

V literatuře naší není dosud elementární knihy, která by stručně učila základům počtu diferenciálního a integrálního. Šimerkův Přídavek k algebře jest příliš stručný, učebnice Studničkova a Weyrova jsou nesnadny pro začátečníky.

[4] Šimerka V.: *Přídavek k algebře pro vyšší gymnázia*. Dr. E. Grégr, Praha, 1864.
[5] Šimerka V.: *Algebra čili počtářství obecné pro vyšší gymnasia*. Dr. E. Grégr, Praha, 1863.
[6] Studnička F. J.: *Základové vyšší mathematiky*. F. J. Studnička, Praha, part I, 1868, part II, 1871 and part III, 1867.
[7] Weyr Ed.: *Počet differenciálný*. Jednota čes. mathematiků, Praha, 1902.
[8] I. ed., preface, page I.

There is not an elementary book in our literature, which would briefly taught the basics of differential and integral calculus. Šimerka's addition to algebra is too short, and Weyr's and Studnička's textbooks are difficult for beginners.

Josef Úlehla knew the textbooks published by his contemporaries and he shortly reviewed them. He wanted by his textbook to help the graduates of Teacher Colleges, where, according to his own experience, the study of mathematics was neglected. The book was designed for autodidacts. This primary affected its style, but it also helped to its re-edition in 1944, which is described below.

The textbook is divided into two parts, entitled *Diferenciální počet* and *Integrální počet* (*Differential Calculus* and *Integral Calculus*). The first part describes derivatives of functions of one variable, calculation rules for derivations, derivatives of basic functions and Maclaurin's, Newton's and Taylor's series. In other chapters the readers can learn about logarithms, trigonometric and hyperbolometric functions and their derivatives. Finaly Josef Úlehla detailed analysis of function, he wrote about searching the maximum, the minimum or asymptotes. Basically the author used the concept of Gottfried Leibnitz, who divided the length of the curve into small parts, differentials. By contrast, František Josef Studnička worked with the Newton's theory of flux.

At the beginning of the part *Počet integrální* (*Integral Calculus*) Josef Úlehla briefly notes, that the differentiation and the integration are the other operations and remarks:[9]

Není obecných pravidel pro integrování, jsou jen pravidla zvláštní, jednotlivá.

There are no general rules for integration, the rules are just special, individual.

The textbook continues with the list of integrals of elementary functions (without any proofs). It contains many specific examples of calculating the indefinite integrals, explanation of the substitution method and the integration by parts. The largest passage is about definite integral, where the author describes the calculation of the length of the curve, the area under the graph, the surface area and volume of the solid of revolution etc. The second part ends with short chapter dealing with differential equations and their applications in physic. This chapter was attached to the text by František Nachtikal (1874–1939). The last pages show the solution of algebraic equations of the three degree and bring the list of analytic expressions of some special curves.

The review of Úlehla's textbook

There were written three reviews on the first issue of *Počet infinitesimální*. The first of them was published in the magazine *Komenský*.[10] and it was unsigned. The other two can be found in *Škola měšťanská*[11] and *Pedagogické rozhledy*,[12] they are signed by initials K. B. According to the style of these texts we can presume, that they were written by the same author. All reviews are very positive. They mentioned Úlehla's affable style of explaining and described the contents of the textbook. Except some misprints, the reviews did not mention about any negatives.

The second issue of Úlehla's textbook

The book was very popular, because of its simple and affable style. Its popularity led to its second edition. During the German occupation of Czech lands in the Second World War, the Czech universities were closed. Therefore people became autodidacts, when they wanted to study something at the highest level of education. The war also complicated the publication of new textbooks. František Navara a Miroslav Litomiský prepared Úlehla's *Počet infinitesimální* for re-publishing in 1944. This time it was titled *Vyšší matematika bez učitele* (*Higher Mathematics without The Teacher*).

The text of the second edition is the same in its contents, but it is not only a reprint. The type was made new. The redactors corrected the mistakes in the text, replaced some expressions of the older terminology by new terms and added the list of contemporary literature, from which the readers could study.

[9] I. ed., page 53.
[10] Komenský 35(1907), issue 8, March 14, 1907, p. 123–124.
[11] Škola měšťanská 9(1907), issue 7, April 4, 1907, annex, p. 21–22.
[12] Pedagogické rozhledy 20(1906–1907), issue 8, May, 1907, p. 614–616.

Conclusion

The positives of the Úlehla's calculus were specified in reviews. They declare that the Úlehla's concept of a simple book was successful. His textbook is like the "cookbook." One of the negatives could be that Josef Úlehla didn't work with the contemporary theory of mathematical analysis including the "ε and δ arithmetic." He wrote just the result without any proofs or exact definitions. Josef Úlehla also didn't add any list of literature. He didn't mention the books, from which he studied, or the books, which could be useful for the readers in their further studies.

Nowadays Josef Úlehla and his publications can inspire. The author's life shows us, how the teacher, who started at a small primary school can be active and what he can do not only for students, but also for his subject. Úlehla's work invites us to his world and can open the doors of the history of our "Queen of the Sciences."

Acknowledgments. This work was supported by the grant GA ČR P401/10/0690 *Prameny evropské matematiky*, the development project *Doktorské studium oboru M8* and the project *Specifický vysokoškolský výzkum 2011-261-315*.

References

Bečvářová M.: *Česká matematická komunita v letech 1848 až 1918.* Matfyzpress, Praha, 2008.
Komenský, časopis paedagogický. Orgán spolku moravských učitelů v Olomouci, Olomouc, 1873 and later.
Kopáč J.: *Josef Úlehla a moravské učitelstvo*. Universita J. E. Purkyně v Brně, Brno, 1967.
Pedagogické rozhledy. Dědictví Komenského, Praha, 1887 to 1932.
Studnička F. J.: *Základové vyšší mathematiky*. F. J. Studnička, Praha, part I, 1868, part II, 1871 and part III, 1867.
Šimerka V.: *Přídavek k algebře pro vyšší gymnázia*. Dr. E. Grégr, Praha, 1864.
Táborský F.: *Několik listů Josefa Úlehly.* Radhošť, Praha, 1934.
Úlehla J.: *Dějiny mathematiky.* Dědictví Komenského, Praha, part I, 1901 and part II, 1913.
Úlehla J.: *Počet infinitesimální.* Dědictví Komenského, Praha, 1906.
Úlehla J.: *Vyšší matematika bez učitele.* Dědictví Komenského, Praha, 1944.
Věstník Ústředního spolku učitelských jednot na Moravě, Brno, 1902 až 1937.
Weyr Ed.: *Počet differenciálný*. Jednota čes. mathematiků, Praha, 1902.

WDS'11 Proceedings of Contributed Papers, Part I, 95–100, 2011. ISBN 978-80-7378-184-2 © MATFYZPRESS

Decomposing Boolean Formulas into Connected Components

T. Balyo

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. The aim of this contribution is to find a way how to improve efficiency of current state-of-the-art satisfiability solvers. The idea is to split a given instance of the problem into parts (connected components) which can be solved separately. For this purpose we define component trees and a related problem of finding optimal component trees. We describe how this approach can be combined with standard satisfiability solver decision heuristics to improve them. The proposed ideas were implemented and experimentally evaluated on a large set of benchmark problems. We provide results of these experiments.

Introduction

Boolean satisfiability (SAT) is one of the most important problems of computer science. SAT is well known in theoretical computer science since it was the first known example of an NP-complete problem [*Cook*, 1971]. SAT has many practical applications mainly in artificial intelligence. One of the first and most successful applications was solving automated planning problems via reduction to SAT [*Kautz, Selman*, 1992]. Other examples of applications are automated reasoning [*Robinson, Voronkov*, 2001] and hardware verification [*Velev, Bryant*, 2003]. Being NP-complete, SAT can not be solved in polynomial time unless P=NP. However this bound holds only for the worst case scenario. Many formulas which we need to solve for the applications of SAT can be solved in reasonable time using the current state-of-the-art SAT solvers.

One of the ways of improving efficiency of a SAT solving procedure is to divide the problem into smaller independent subproblems. The subproblems can be solved separately (in parallel). For problems with exponential time complexity this approach can help us to achieve exponential speedup. This idea has already been used to design efficient SAT solver decision heuristics [*Balyo, Surynek*, 2009;*Pipatsrisawat, Darwiche*, 2001] and also to improve satisfiability model counting (♯SAT) algorithms [*Bayardo, Pehousek*, 2000]. In this paper we will further investigate the possibilities of using connected components of SAT problems to enhance SAT solvers.

SAT Definitions

A Boolean variable is a variable with two possible values: $true$ and $false$. A literal is a Boolean variable or its negation. A clause is a disjunction (OR) of literals. A conjunctive normal form (CNF) formula is a conjunction (AND) of clauses. In the rest of the text by formula we always mean a CNF formula. A truth assignment ϕ for a formula F is a function $\phi : Vars(F) \rightarrow \{true, false\}$ which assigns truth values to each variable of F. Similarly a partial truth assignment assigns truth values to some of the variables of F. We say that a (partial) truth assignment ϕ satisfies a variable x if $\phi(x) = true$; a positive literal of the variable x if $\phi(x) = true$; a negative literal of x if $\phi(x) = false$; a clause if it satisfies any of its literals and a CNF formula if it satisfies all of its clauses. We say that a formula is satisfiable if there is a (partial) truth assignment that satisfies it. Satisfiability (SAT) is the problem of determining whether a given formula is satisfiable.

Solving SAT

There are several algorithms for SAT solving of various kinds, but the most successful ones are based on the Davis Putnam Logemann Loveland (DPLL) procedure[*Biere et. al.*, 2009]. DPLL is a depth first search of the space of partial truth assignments. We start with an empty partial truth assignment and try to extend it into a satisfying truth assignment. The search can be stopped if all the clauses are satisfied and we can immediately backtrack if there is a clause with all literals falsified by the current partial truth assignment. DPLL uses two additional enhancements: pure literal elimination and unit propagation. If a variable has only positive occurrences or only negative occurrences, then the literals

of this variable are called pure literals. Such a variable can be immediately assigned to the proper value and make all its occurrences true. A clause is called unit if all but one of its literals are false and the remaining literal is unassigned. This literal has to be assigned to be true in order to satisfy the clause. This assignment can cause another clause to become unit and forces another assignment. The cascade of such assignments is called unit propagation. We present the pseudocode of DPLL as Algorithm 1.

Algorithm 1 DPLL($clauses, vars, assignment$) : *boolean*

if $\forall c \in clauses$ *assignment* satisfies c **then**
  return $true$
end if
if $\exists c \in clauses$ *assignment* makes c $false$ **then**
  return $false$
end if
$assignment = assignment \cup$ unitPropagation($clauses$, $assignment$)
$assignment = assignment \cup$ pureLiteralElimination($clauses$, $assignment$)
select x such that $x \in vars \wedge assignment(x) = NULL$
return DPLL($clauses, vars \setminus \{x\}, assignments \cup \{x = true\}$)
  or DPLL($clauses, vars \setminus \{x\}, assignments \cup \{x = false\}$)

The performance of DPLL very much depends on the selection of decision variables. We use decision heuristics to select these variables. There are many very good decision heuristics already used by state-of-the-art SAT solvers. Our goal is to design a new one, which similarly to the divide and conquer principle will try to split the formula into parts and solve those independently. In order to exactly define this idea we will use a graph derived from a formula called an interaction graph [*Biere et. al.*, 2009].

Definition 1 *An interaction graph of the formula F is an undirected graph $G(V, E)$, where V is the set of variables of F and $(x, y) \in E$ if and only if there is a clause $c \in F$ that contains literals of x and y.*

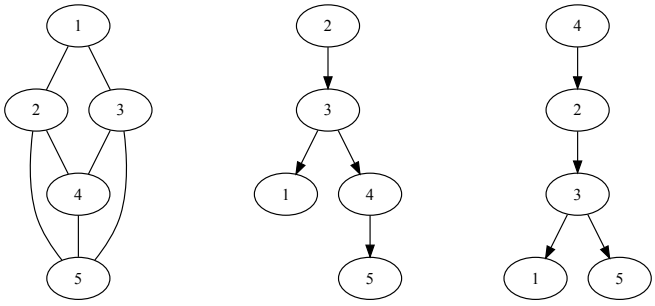


Figure 1. The interaction graph for the formula $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3) \wedge (x_2 \vee \neg x_4 \vee \neg x_5) \wedge (\neg x_4 \vee \neg x_5 \vee x_3)$ and two of its component trees.

An example of a formula and its interaction graph is given on Figure 1 (left). We will be interested in the connected components of interaction graphs, because subformulas corresponding to different connected components can be solved separately since they have no common variables. If we consider the worst case time complexity of solving general SAT instances, then solving formulas by components can give us exponential speedup. For example if the interaction graph of a formula with n variables has two equally large connected components of $n/2$ vertices, then this formula can be solved in $2^{n/2} + 2^{n/2}$ time instead of 2^n, which is $2^{(n/2)-1}$ times faster.

Unfortunately, interaction graphs of SAT formulas rarely have more than just one connected component. However if we consider the DPLL algorithm and observe that extending a partial truth assignments is equivalent to removing vertices from the interaction graph, we can see that the interaction graph can get disconnected during the solving. To precisely describe this behavior we define the dynamic interaction graph.

Definition 2 *A dynamic interaction graph of the formula F and its partial truth assignment ϕ is an undirected graph $G(V, E)$, where V is the set of variables of F which have no assigned truth values by ϕ and $(x, y) \in E$ if and only if there is a clause $c \in F$ that contains literals of x and y.*

If G is the dynamic interaction graph for a formula F and its arbitrary partial truth assignment ϕ, then the subformulas corresponding to the connected components of G can be solved independently for any partial truth assignment that extends ϕ. We will use this property to design our decision heuristic. We want to assign those variables first, that will disconnect the interaction graph as quickly and as uniformly as possible. The remaining problem is to find those variables. For this purpose we need to define component trees and the component tree problem.

The component tree problem

Definition 3 *A $rootPath(v)$ of a vertex v in a rooted tree T is the set of vertices on the path from v to the root of T (including both v and the root). Let G_{-S} be a graph formed from G by removing the set of vertices S and all incident edges from G. A rooted tree T is a component tree for a connected graph G if G and T have the same vertices and for each vertex v that has at least 2 sons the following holds: The vertices in the subtrees of the sons of v are in different connected components of $G_{-rootPath(v)}$.*

An example of a graph and two of its component trees is on Figure 1. It is obvious from the given example that there can be several different component trees for a given graph. We need to compare different component trees, so we define the component value.

Definition 4 *The component value $C(v)$ of a vertex v in a component tree is defined as $C(v) = 1$ if v is a leaf and $C(v) = 2 \times \sum_{s \in sons(v)} C(s)$ otherwise. The component value of a tree is the component value of its root.*

The component values of the trees from the example on Figure 1 are 12 and 16 respectively. Component trees with lower component values will be preferred. We will call T an optimal component tree for a graph G if there is no other component tree for G that has lower component value than T. There can be many optimal component trees for a given graph. For example a clique on n vertices has $n!$ optimal component trees with 2^{n-1} being their component value.

The component tree problem is the problem of finding an optimal component tree for a given graph. The decision version is determining if there is a component tree of a given component value for a given graph. The decision version is clearly in NP, since the component tree itself is the certificate. It is unknown to the author of this paper whether it is NP-hard and thus NP-complete. However, if we do not require an optimal component tree, we can obtain a component tree easily by depth first search.

If we perform a depth first search (DFS) on a graph, then the tree edges of the DFS spanning tree form a component tree. Such a component tree can be very far from optimal. There are some examples of graphs, where this algorithm cannot find an optimal component tree no matter in what order we process the vertices [*Balyo*, 2010]. For these reasons we designed another algorithm called the component tree builder (CTB).

The pseudocode of CTB is given as Algorithm 2. It builds the component tree by connecting small component trees (initially consisting of only one vertex) into bigger ones. The $rootOf(v)$ method returns the root of the component tree to which v currently belongs. Every possible component tree can be constructed by this algorithm and thus also the optimal ones [*Balyo*, 2010]. The quality of the returned component tree depends on the order in which we process the vertices in the main loop. For the ordering of the vertices we will use the following greedy heuristics: select a vertex so that after its addition, the total increase of the component value is minimal. We experimentally compared CTB with the greedy heuristics and the DFS algorithm, where the next vertex is selected randomly. The results are presented on Figure 2.

The component trees often contain long segments of vertices with only one son. We call them linear segments. An exact definition follows.

Definition 5 *Let $L = x_1, x_2, \ldots, x_n$ be a sequence of vertices in a component tree T. If x_1 has no brother and $\forall i \in \{1, \ldots, n-1\}$ x_{i+1} is the only son of x_i then L is a linear segment of T.*

From the definition of the component tree we can easily prove that the order of vertices in the linear segments is not important. We can permute the vertices inside all the linear segments and we get a valid component tree with the same shape and thus with the same component value. This allows us to

Algorithm 2 The component tree builder algorithm

INPUT $G(V, E)$
$V' = \emptyset$, $E' = \emptyset$
for all $v \in V$ **do**
  $R = \emptyset$
  for all $s \in neighbor(G, v)$ **do**
    if $s \in V'$ **then**
      $R = R \cup \{rootOf(s)\}$
    end if
  end for
  $V' = V' \cup \{v\}$
  for all $r \in R$ **do**
    $E' = E' \cup \{(v \rightarrow r)\}$
  end for
end for
OUTPUT $G(V', E')$

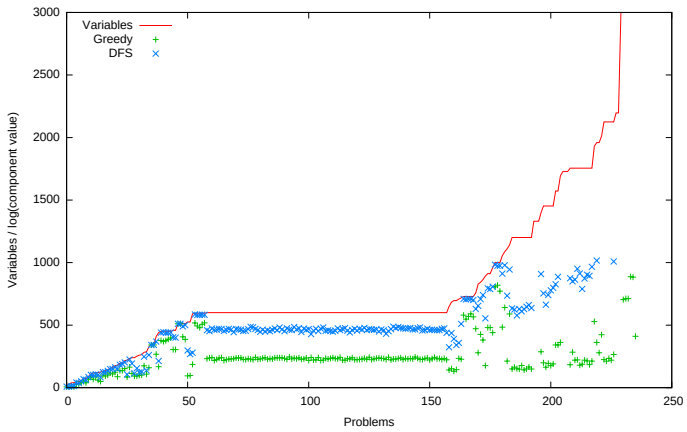


Figure 2. Comparison of the component tree construction algorithms on interaction graphs of SAT formulas. "Variables" denotes the number of variables of the formula, "Greedy" and "DFS" are the logarithms of the component values of the trees returned by the CTB algorithm using the greedy heuristics and the depth first search algorithm respectively.

look at the linear segments as sets of vertices and contract them into one vertex. The resulting tree is called a compressed component tree. An example of a component tree with its linear segments and its compressed component tree is given on Figure 3.

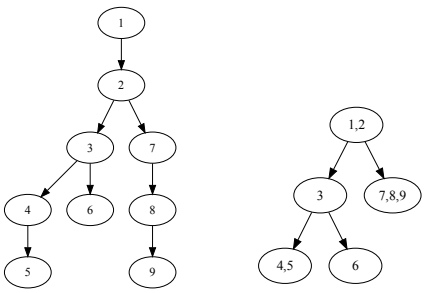


Figure 3. A component tree with linear segments $\{\{1, 2\}, \{3\}, \{4, 5\}, \{6\}, \{7, 8, 9\}\}$ and its compressed version.

SAT decision heuristics

Now we use the component tree to design decision heuristics for SAT solving. The algorithm is straightforward. We take the input formula and construct its interaction graph. We use our best available algorithm for component tree construction and find a component tree. Finally we contract its linear segments to get a compressed component tree. We will keep selecting decision variables from the root until all of them are assigned and the formula is disconnected. We proceed on the subtrees recursively to split the formula further. We will call this the component tree heuristics (CTH).

We have not yet specified the order of vertices in which we select them from the current compressed linear segment. If we pick them in a random order, the practical performance of the heuristics is very low. That is why we combine the component tree approach with state-of-the-art decision heuristics. These heuristics help us to order the variables within a linear segment. Now we will briefly describe some known decision heuristics and how they can be combined with CTH. Sometimes we will write "decision literal" instead of "decision variable", this means that we select the decision variable as well as the value that should be tried first. Some heuristics incorporate clause learning, which is described in [*Biere et. al.*, 2009].

Jeroslow-Wang (JW) [*Biere et. al.*, 2009] is a score-based decision heuristics. Each literal l has its score defined as $\sum_{cl \in F | l \in cl} 2^{-|cl|}$. The score is higher for literals with many occurrences in short clauses. An unassigned literal with the highest score is selected for assignment. The scores are computed once in the preprocessing phase. The combination with CTH is very simple, we select the literal with the highest score from our linear segment.

Dynamic Largest Individual Sum (DLIS) [*Biere et. al.*, 2009] is also score-based. The score of a literal is the number of not satisfied clauses containing it. This score is dynamic and thus needs to be recomputed when the partial truth assignment changes. The combination with CTH is analogous to JW.

Variable State Independent Decaying Sum (VSIDS) [*Biere et. al.*, 2009] is yet another score-based heuristic. For each literal l we define its score $s(l)$ and its occurrence count $r(l)$. At the beginning of solving, $s(l)$ is initialized to the number of clauses that contain l and $r(l)$ is set to 0 for each l. When a clause is added to the formula via clause learning, the occurrence counts of its literals are incremented by one. After each 255 decisions the scores of the literals are updated by the following formula: $s(l) = s(l)/2 + r(l)$ and $r(l)$ is set to 0 for each l. The literal with the highest score is selected for the decision. The combination with CTH is again analogous to JW (and DLIS).

BerkMin [*Biere et. al.*, 2009] also has literal scores based on the number of their occurrences. The decision literal is selected as the highest scored literal from the most recently learned currently not satisfied clause. The combination with CTH searches for the most recently learned unsatisfied clause that contains a variable from the current linear segment. Then we select the highest scored suitable literal from that clause.

Last Encountered Free Variable (LEFV) [*Balyo, Surynek*, 2009] defines no literal scores at all. It selects a literal from the clause which was the last not satisfied clause encountered during the most recent unit propagation. When combined with CTH, we select a literal from the last encountered clause that belongs to the current linear segment. If there is no such literal in the clause, we select a random literal from the linear segment.

Experiments

To measure the practical performance of our approach we implemented a SAT solver and all the above described heuristics. Our solver implements the conflict driven clause learning (CDCL) DPLL algorithm[*Biere et. al.*, 2009]. For clause learning we used the first UIP scheme[*Biere et. al.*, 2009]. Unit propagation was implemented using the 2-watched literals scheme[*Biere et. al.*, 2009] and our solver also incorporates restarting[*Biere et. al.*, 2009]. A detailed description of the solver is to be found in [*Balyo*, 2010].

We conducted experiments on various kinds of SAT benchmark problems. The formulas can be divided into two classes: random 3SAT formulas and structured formulas (modeling pseudo-practical problems). More exact description of the set of the used benchmark problems and their sources can be found in chapter 5 of [*Balyo*, 2010].

Selected results of our experiments are presented on Figure 4 for random formulas and on Figure 5 for structured ones. From the plots for random problems we can see that the performance difference is rather low for unsatisfiable formulas. Overall, there are many problems where the combined heuristics outperformed the original but in the majority of cases it did not.

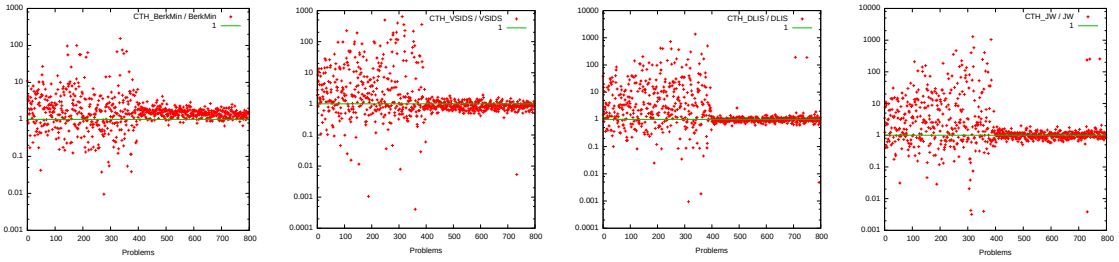


Figure 4. Comparison of the decision heuristics with their combined versions on random 3SAT formulas. The first 400 problems are satisfiable, the second half is unsatisfiable. If the cross is above the line then the combined version is weaker than the original.

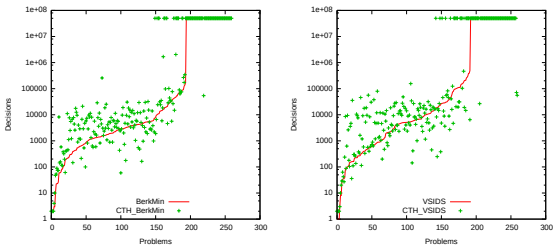


Figure 5. Comparison of the decision heuristics with their combined versions on random structured formulas. If the cross is above the line then the combined version is weaker than the original.

Conclusion

We defined a new general graph problem—the component tree problem and applied it to design a set of new decision heuristics for satisfiability solving via the DPLL algorithm. We experimentally compared the heuristics, but we did not manage to outperform the known heuristics used by state-of-the-art SAT solvers. Probably one of the reasons is that the component tree is a very rough approximation of the disconnection of Boolean formulas during DPLL. Component trees do not consider unit propagation, pure literal elimination or clause learning, which are important features of DPLL. We would like to remove these flaws in our future work.

Acknowledgments. I would like to thank my supervisor Roman Bartak and consultant Pavel Surynek for valuable discussions. This research is supported by the Science Foundation of the Charles University (grant No. 266111).

References

Balyo, T. and Surynek P., Efektivni heuristika pro SAT zalozena na znalosti komponent souvislosti grafu problemu, Proceedings of Znalosti 2009, 35–46, 2009.

Balyo, T., Solving Boolean satisfiability problems, Diploma Thesis, Charles University in Prague, 2010.

Bayardo, R. J. and Pehoushek J. D., Counting models using connected components, Proceedings of AAAI-00, 157–162, 2000.

Biere, A. and Heule, M. and van Maaren, H. and Walsh, T. (editors), Handbook of Satisfiability, IOS Press, 2009.

Cook, Stephen A., The complexity of theorem proving procedures, STOC, 151–158, 1971.

Kautz, Henry A. and Selman, Bart, Planning as satisfiability, ECAI, 359–363, 1992.

Pipatsrisawat K. and Darwiche A., A lightweight component caching scheme for satisfiability solvers, Lecture notes in computer science volume 4501, 294–299, Springer, 2007.

Robinson, John Alan and Voronkov, Andrei, editors, Handbook of Automated reasoning (in 2 volumes), Elsevier and MIT Press, 2001.

Velev, Miroslav N. and Bryant Randal E., Effective use of Boolean satisfiability procedures in the formal verification of superscalar and vliw microprocessors, Journal of Symbolic Computation, 35(2):73–106, 2003.

 ISBN 978-80-7378-184-2

Minimization of Matched Formulas

Š. Gurský

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. Decision problems of finding minimal representation of a given Boolean formula is known to be Σ_2 complete for formulas in conjunctive normal form and for two basic measures of minimality – number of occurences of literals or numbers of clauses. In this paper we study this problem restricted to a class of satisfiable Boolean CNF formulas called matched formulas. We present several variants of minimization and show that their complexity does not differ from general case. Then we show that minimization remains Σ_2 complete even restricted to matched formulas.

Introduction

Finding a minimal representation of a given object is a common task in many areas. In this paper the area of interest is Boolean minimization where we deal with Boolean functions represented in conjunctive normal form (CNF). The task is to find a shortest possible CNF representation for a given function with respect to a given measure.

This paper deals with decision problems connected with Boolean minimization and discusses their complexity. It is known that deciding whether there exists shorter CNF for function given as CNF is Σ_2 complete for general formulas. However, for certain classes of formulas the complexity of minimization is different.

In this paper we deal with class of satisfiable formulas where matching of variables to clauses exists. These formulas are called matched and can be recognized in polynomial time. Several subproblems connected to minimization are examined and their complexity is set. Then, building on a work by C. Umans, complexity of minimization of matched formulas is established.

Definitions

We assume unlimited supply of Boolean variables $x_1, x_2, \ldots$ that can have value either true or false. Literal is a variable or its negation. Clause is a disjunction of literals. Value of a clause is true, if any of its literals is true. Value of an empty clause is false. Conjunctive normal form (CNF) is a conjunction of clauses. CNF evaluates to true if all its clauses evaluate to true. Empty CNF evaluates to true.

We will also need dual objects to clauses and CNFs – terms and DNFs. Term is a conjunction of literals. An empty term's value is true. Disjunctive normal form (DNF) is a disjunction of terms. Empty DNF evaluates to false.

It is a well known fact, that any Boolean function can be represented by both CNF and DNF, often in more than one way. We will say, that two formulas φ and ψ are equivalent if they represent the same function.

Definition 1 (Incidence graph). For CNF φ, the *incidence graph* G is a bipartite graph, where vertices of one partity are clauses of φ and vertices of the other partity are its variables. A clause is connected to variables that it contains (that is clause C is connected to variable x if and only if it contains either literal x or literal $\overline{x}$).

Definition 2 (Matched CNF). CNF is called *matched* if its incidence graph has matching that covers all clauses. That is every clause can be assigned its own variable.

Matched DNFs can be defined in an analogous way.

Matched formulas were first defined in [*Franco and Van Gelder*, 2003] and the following simple observation was made.

Observation 1 ([*Franco and Van Gelder*, 2003]). Every matched CNF is satisfiable.

Proof. For clause C take its matched variable x. If C contains literal x, set x to true, othervise set it to false. This way all clauses can be satisfied. □

Minimization

General case

We will look at Boolean minimization as a decision problem. There are several flavors of the problem. In all of them we are given a formula and a limit and the question is whether there is an equivalent formula that has size smaller than the limit with respect to a given measure. Let us first look at nonrestricted cases of minimization problems.

The first natural limit is number of clauses in a CNF or number of terms in a DNF. Formal statements of problems follow.

Problem: MinCNFClauses
Input: CNF φ and integer k
Question: Is there a CNF ψ equivalent to φ that has at most k clauses?

Problem: MinDNFTerms
Input: DNF φ and integer k
Question: Is there a DNF ψ equivalent to φ that has at most k terms?

Lemma 1. Problems MinCNFClauses and MinDNFTerms are equivalent, that is one can be reduced to the other in polynomial time.

Proof. Given the instance φ and k of MinDNFTerms, we create CNF φ' that is negation of φ by switching ORs and ANDs and by negating each literal. φ' and k then represent an instance of MinCNFClauses and it is positive instance if and only if the φ and k were the positive instance of MinDNFTerms. That is because if φ has an equivalent DNF ψ that has at most k terms, then ψ' (negation of ψ) is CNF equivalent to φ' that has at most k clauses and vice versa. Reduction in the opposite direction is identical. □

Another measure of CNF or DNF length is number of occurences of literals.

Problem: MinCNFLit
Input: CNF φ and integer k
Question: Is there a CNF ψ equivalent to φ that has at most k occurences of literals?

Problem: MinDNFLit
Input: DNF φ and integer k
Question: Is there a DNF ψ equivalent to φ that has at most k occurences of literals?

Lemma 2. Problems MinCNFLit and MinDNFLit are equivalent in the same sense as in Lemma 1.

Proof. Proof of this lemma is identical to the proof of Lemma 1. □

From these two lemmas we see, that we can choose either CNF or DNF variant of any problem, find its complexity and then apply the result to the other normal form. The complexity of minimization of DNFs was determined by Christopher Umans in 2000.

Theorem 1 ([*Umans*, 1998, 1999, 2000]). The problems MinDNFTerms and MinDNFLit are both Σ_2 complete.

From lemmas 1, 2 and Theorem 1 we get the following corollary.

Corollary 1. Problems MinCNFClauses and MinCNFLit are also Σ_2 complete.

Matched formulas

In general case the minimization is Σ_2 complete. However, there are classes of formulas where minimization is easier. For example minimization of Horn formulas (those with at most one positive literal in each clause) is known to be NP complete [*Hammer and Kogan*, 1993]. For monotone formulas (those that have no negation) the minimization is trivial. In both of these classes of CNFs, the famous Satisfiability problem can be solved in polynomial time, so one can examine minimization in other classes of formulas with this property and expect minimization to be easier than in general case.

Let us concentrate on matched formulas. We will look at several smaller problems that are connected with minimization and determine the complexity of these problems. These problems deal with finding unnecessary parts of the formula — parts that can be removed from the formula without changing the function it represents.

Problem: Dependency on Variable
Input: Matched CNF φ and variable x.
Question: Does the function represented by φ depend on x?

Theorem 2. The problem Dependency on Variable is NP complete.

Proof. The problem is in NP. The certificate would be pair of assignments v_1 and v_2 that differ only in x, but $v_1(\varphi) \neq v_2(\varphi)$.

To show NP hardness, consider an instance of Satisfiability $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$. From this we construct $\psi = (C_1 \vee w_1) \wedge (C_2 \vee w_2) \wedge \cdots \wedge (C_n \vee w_n) \wedge (w_1 \vee w_2 \vee \cdots \vee w_n \vee x)$, an instance of Dependency on Variable. ψ is clearly matched (Clauses $(C_i \vee w_1)$ can be matched with w_i and the last clause can be matched with x). Suppose that φ is satisfiable with assignment v. Construct assignment v' by extending v such that w_i is false for every i and assignments v_1 and v_2 that both extend v' but v_1 sets x to true and v_2 sets x to false. These two assignments differ only in x but v_1 satisfies ψ and v_2 does not.

Suppose φ is not satisfiable. Then for assignment that sets every w_i to false ψ also evaluates to false and does not depend on value of x. For assignment that sets any of w_i to true the last clause of ψ is satisfied and ψ does not depend on x. φ is satisfied if and only if ψ depends on x. □

It should be noted that x occurs in φ from previous proof only once. Let us call DoV1 a restricted version of the previous problem, where we limit number of occurences of x to one. DoV1 is then also NP complete.

Problem: Removal of Variable
Input: Matched CNF φ and variable x.
Question: Can x be removed from φ without changing the function it represents?

Theorem 3. The problem Removal of Variable is coNP complete.

Proof. The problem is clearly in coNP. For negative answer the certificate would be an assignment on which the input formula φ and φ with x removed would have different value. coNP hardness can be shown by reduction from DoV1. Given instance φ and x of DoV1, where x occurs only once in φ, the corresponding instance of Removal of Variable is also φ and x. If φ depends on x, it cannot be removed, since the result would not contain x. If φ does not depend on x then removing x is same as setting x to false (it occurs only once in φ) and for all assignments this is same as setting x to true (φ does not depend on x), so x can be removed. Therefore x can be removed if and only if φ does not depend on it. □

Problem: Removal of Clause
Input: Matched CNF φ and its clause C.
Question: Can C be removed from φ without changing the function?

Theorem 4. The problem Removal of Clause is coNP complete.

Proof. This proof is almost identical to the proof of Theorem 3. Problem belongs to coNP since certificate for negative answer is an assignment where φ has value opposite to φ with C removed. To show coNP hardness we again start from instance of DoV1 and ask whether we can remove clause containing x. If φ does depend on x the clause cannot be removed. If φ does not depend on x then setting x to true is same as setting x to false and removal of clause containing x is equivalent to setting x to true. Therefore φ does not depend on x if and only if clause containing x can be removed. □

The next problem is not exactly minimization subproblem, however, it is an interesting one and we can use the previous problem to assess its complexity.

Problem: Is Implicate
Input: Matched CNF φ and some clause C.
Question: Does every assignment that satisfies φ also satisfy C?

Theorem 5. The problem Is Implicate is coNP complete.

Proof. The problem is in coNP. The certificate for negative answer is an assignment which sets φ to true but C to false. The coNP hardness can be seen from the previous problem. Instance φ and C of problem Removal of Clause is a positive instance if and only if $\varphi \setminus C$ and C are positive instance of problem Is Implicate. □

And now we look at real minimization problems. For general formulas we had problems MinCNF-Clause, MinCNFLit and their respective DNF variants. For matched formulas we limit CNFs (DNFs) in problem description to matched formulas. The problem statements for DNF follow. For CNF the problems are analogous.
Problem: MinDNFTermMatch
Input: Matched DNF φ and integer k
Question: Is there DNF ψ that is equivalent to φ and has at most k clauses?

Problem: MinDNFLitMatch
Input: Matched DNF φ and integer k
Question: Is there DNF ψ that is equivalent to φ and has at most k occurences of literals?

Using the same technique as in the proof of Lemma 1 we can show that CNF variants, problems MinCNFClauseMatch and MinCNFLitMatch, have the same complexity as their DNF equivalents. Therefore, we can use version that we want, to establish the complexity for both CNF and DNF (note that we cannot say anything about equivalence of problems that deal with different measures of formula length).

Theorem 6. Both problems MinDNFTermMatch and MinDNFLitMatch are Σ_2 complete.

Sketch of proof. The proofs of both these statements follow very closely the original proofs of C. Umans for general case with certain ammendments. Booth proofs are quite long and can be found in full in [*Gurský*, 2010]. The original proof for MinDNFLit can be found in [*Umans*, 1998], the proof for MinDNFTerm can be found in [*Umans*, 1999], both in [*Umans*, 2000] and cleaned up versions in [*Gurský*, 2010]. We will only show how to get a proof of complexity of minimization of matched formulas from the original version.

In both proofs the instance of certain Σ_2 complete problem is reduced to instance of problem at hand. We present here only the formula in the resulting instance and a way it can be made matched formula without breaking the proof of complexity.

Let us first look at MinDNFLit and therefore at MinDNFLitMatch. In his proof Umans reduced an instance of another Σ_2 complete problem Shortest Implicant Core (proof of Σ_2 completness provided therein) to instance of MinDNFLit. The formula in the resulting instance of MinDNFLit is in the form $\varphi'' = t_l w_1 w_2 w_3 \dots w_{m'} \vee \bigvee_{i=1}^{m} s'_i$ where s'_i is in the form $s_i w_1 w_2 \dots w_{i-1} w_{i+1} \dots w_{m'}$ with w_i being variable for all i and t_l and s_i are terms for all i. Important fact is that in this formula we can find matching that matches each of s'_i terms with variable w_{i+1} (the last one with w_1) and the first term can be matched with any of the variables in t_l (it is not empty). Therefore this formula is matched and problem MinDNFLitMatch is Σ_2 complete.

In proof of MinDNFTerm, C. Umans reduces again problem Shortest Implicant Core (albeit a somewhat different version) to an instance of MinDNFTerm. The resulting instance in this case is a formula that consists of two parts. The first part are terms in form $s'_i = s_i z_1 z_2 \dots z_{i-1} z_{i+1} \dots z_m$ with s_i being term and z_i being variable for each i. Each s'_i can be then matched with z_{i+1}. The second part of formula has terms in form $u_{i,j} = (p_i)\bar{x_j} z_1 z_2 \dots z_m$, where all parts are variables. However, we can change p_i from being a variable to being a term of parity function on new set of variables $a_1, a_2 \dots$ such that there is enough of a-variables to provide matching for all terms. The parity is used since it cannot be shortened in any way. The formula length grows only polynomially (each term gets polynomial amount of new variables) and the resulting formula is again matched. So MinDNFLitMatch is also Σ_2 complete. □

Corollary 2. Problems MinCNFClauseMatch and MinCNFLitMatch are also Σ_2 complete.

Proof. Same as in lemmas 1 and 2. □

Conclusion

We saw, that the class of matched formulas has the same complexity of minimization and attached problems as a general formulas. This result is partially surprising, since the matched formulas have a simple structure and the "standard" hard problem (SATISFIABILITY for CNF) is trivial for them.

Future work can be done on discovering distinguishing property between classes of formulas with Σ_2 complete minimization and classes whose minimization is in lower complexity level.

References

Franco J., Van Gelder A., A perspective on certain polynomial-time solvable classes of satisfiability, in *Discrete Appl. Math.*, 125, 177-214, 2003

Gurský, Š., Časová zložitosť minimalizácie Booleovských funkcií, diploma thesis, Charles University, Faculty of Mathematics and Physics, Prague, 2010, available online at http://artax.karlin.mff.cuni.cz/ gurss5am/diplomka.pdf (accessed on 2011-06-11).

Hammer P., Kogan A., Optimal compression of propositional Horn knowledge bases: complexity and approximation, in: *Artif. Intell.*, 64, 131-145, 1993

Umans C., The minimum equivalent DNF problem and shortest implicants, in: Foundations of Computer Science, 1998. Proceedings.39th Annual Symposium on, 556 -563, 1998

Umans, C., Hardness of Approximating Σ_2^p Minimization Problems, in: *FOCS '99: Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, 465-474, 1999

Umans C., Approximability and completeness in the polynomial hierarchy, disertation thesis, University of California, Berkeley, 2000

WDS'11 Proceedings of Contributed Papers, Part I, 106–111, 2011. ISBN 978-80-7378-184-2 © MATFYZPRESS

A Worst Case Aware Version of Universal Hashing

M. Babka

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. In this work we propose and analyze a simple hash table giving a worst case time guarantee in case of Find operation with all operations running in constant expected amortized time. To achieve this goal we exploit known properties of some universal classes of functions. Since the mentioned guarantee strongly depends on the quality of the underlying universal system, we show various systems capable of interesting bounds. With reasonable assumptions we can also combine the proposed model with the "two choices paradigm" and obtain $\mathcal{O}(\log \log n)$ worst case bound. The main advantage of the method is its simplicity and genericness because it may be used with chaining as well as with open addressing.

Introduction

Dictionary is a data structure which allows storing and querying data associated with each stored key. Dictionaries are often implemented with *hash tables*. This implementation is utilized especially in the case when the set of the possible keys is not linearly ordered or when the performance of the dictionary is crucial to its application. With hash tables we can easily and efficiently perform operations *Find*, *Insert* and *Delete*. In this paper we are interested in dynamic dictionaries which allow update operations and provide a guaranteed sublinear lookup time.

Known results and methods

Hash tables are considered to be one of the most efficient solutions to the dictionary problem since the expected running time of the operations is usually constant. The first methods of *plain hashing*, formerly separate chaining and linear probing or more recently Hopscotch hashing [9], rely on the randomness provided by the input data. These methods are usually analyzed under the assumption of full randomness, which is discussed in more detail in [11].

In some situations we can not rely on the randomness of input and have to switch to a randomization provided by a uniform selection of a hash function from a universal family. This method is known as *universal hashing* and was pioneered by Carter and Wegman in [5]. It achieves Find operation running in expected constant time but in general no non trivial worst case bound is known.

Perfect hashing [8] is an extension of universal hashing which addresses the problem of worst case running time for Find operation in case of static dictionaries. In order to add update operations various dynamizations of perfect hashing [7] and a real time hash table [6] were proposed. These methods together with cuckoo hashing [14] guarantee a constant worst case lookup time and the other operations run in expected amortized constant time. Currently the most outstanding theoretical method is Backyard Cuckoo Hashing [2] which addresses the problem of worst case time for updates and memory consumption of cuckoo hashing. It achieves constant running times of the operations with high probability.

Our result

Our aim is to fit in the gap between perfect hashing and common universal hashing. We provide a hash table which uses linear space and provides a worst case guarantee for Find operation. In fact, we analyze a simple modification of universal hashing with separate chaining and show that the method can be generalized to linear probing. We bound lengths of chains by a convenient probabilistic limit function and rehash the table whenever there exists a chain violating this condition. We show that this approach leads to a constant expected amortized time for each operation with a worst case guarantee for lookup.

The main advantage of the model is its simplicity, no need for a perfect hash function when compared to dynamic perfect hashing and is dynamic by design – resizing and updates cause no harm. To compare with cuckoo hashing, we have chosen a different approach which unfortunately does not achieve a constant worst case running time. On the other hand, we can tune the space consumption and the speed loss is compensated by use of simpler universal classes.

The proposed technique is generic. It may be used with double hashing and linear probing, too. Furthermore using two level hashing is possible and significantly improves the worst case guarantee.

Notation

The set U denotes the universe, V denotes the set of addresses of the hash table. We refer to $S \subset U$ as to the stored set. Let m be the size of the hash table, $m = |V|$, and n be the number of stored elements, $n = |S|$, $n \ll |U|$. The load factor of the table is denoted by α where $\alpha = n/m$. The function $f\colon U \to V$ denotes the used hash function.

The length of the chain at the address $y \in V$ is denoted by $\mathbf{psl}(y, S, f) = |f^{-1}(y) \cap S|$. The length of the longest chain, $\mathbf{lpsl}(S, f)$, is defined as $\mathbf{lpsl}(S, f) = \max_{y \in V} \mathbf{psl}(y, S, f)$. From now we assume the uniform choice of a hash function f from a universal system. This choice forms the probability space.

Definition 1 (c-universal system [5]). *Let H be a multiset of hash functions from U to V. We say that H is a* c-universal system*, for $c > 0$, if for arbitrary different $x, y \in U$: $|\{f \in H \mid f(x) = f(y)\}| \leq c|H|/m$.*[1]

An equivalent restatement of the definition in probability terms requires that $\mathbf{Pr}\left(f(x) = f(y)\right) \leq c/m$. There are also various extensions of c-universality which include strong *k-universality* [16], which is also called *k-wise independence* [16], *strongly ω-universal* [16] systems and *uniform* [13] systems.

Definition 2. *Let $k > 0$ be an integer. System of functions H is*

- strongly k-universal *with constant c, for $c \geq 1$, if for each sequence of k pairwise different elements $x_1, \ldots, x_k \in U$ and arbitrary $y_1, \ldots, y_k \in V$: $\mathbf{Pr}\left(f(x_1) = y_1, \ldots, f(x_k) = y_k\right) \leq c/m^k$,*
- strongly k-universal *if it is strongly k-universal with $c = 1$,*
- strongly ω-universal *if it is strongly k-universal for each $k \in \{1, \ldots, |U|\}$,*
- (almost) uniform *if it is (almost) strongly n-universal.*

The strongly k-universal systems behave fully random up to k different elements. In case of more than k elements they provide only so called *limited randomness*. On the other hand functions chosen from strongly ω-universal systems behave like truly random functions. We often need truly random estimates only up to n elements. In this situation uniformity is as powerful as ω-universality.

In the following section we describe and analyze the proposed model. Then we deal with various well-known universal systems and connect them with the model. In conclusion we point out further extensions and improvements.

Model of Universal Hashing

In this section we propose and analyze the model in case of separate chaining. The way we manage to achieve worst case lookup time is to keep chains shorter than the prescribed limit. First we introduce the concept of the limit function which bounds the length of the longest chain with a prescribed probability. It turns out that the time needed to keep chains short can be amortized to a constant in expectation.

Limit function

Each limit function has to depend on the size of the table, the load factor and the probability with which the bound holds. The following definition of a limit function takes into account the dependencies.

Definition 3 (Limit function, trimming rate, suitable function). *The function $l\colon \mathbb{N} \times \mathbb{R}_0^+ \times (0,1) \to \mathbb{N}$ is called a* limit function *with* trimming rate p*, $p \in (0,1)$, if $\forall S \subseteq U\colon \mathbf{Pr}\left(\mathbf{lpsl}(S, f) > l(m, \alpha, p)\right) \leq p$.*

We say that a function $f \in H$ is suitable *for S and l if $\mathbf{lpsl}(S, f) \leq l(m, \alpha, p)$.*

Algorithms

The hash table may be adjusted by the following parameters.

- m_0, initial size of the hash table;
- α_l, α_u, $\alpha_l < \alpha_u$ – we keep $\alpha \in [\alpha_l, \alpha_u]$, the lower bound may be violated when $m = m_0$;
- α_m, $\alpha_m \in (\alpha_l, \alpha_u)$ – when rehashing because $\alpha \notin [\alpha_l, \alpha_u]$ we choose new m so that $\alpha \sim \alpha_m$;

[1] The set on the left side of the expression is also considered to be a multiset.

- α', $\alpha_u < \alpha'$, is the load factor for which the limit function is computed, thus $\mathbf{lpsl}(S, f) < l(m, \alpha', p)$.

The following three invariants briefly describe the model.

(1) **Universal class and limit function.** The used c-universal system H has a limit function l with a trimming rate p.

(2) **Load Factor Rule.** The load factor of the table is kept in the predefined interval $[\alpha_l, \alpha_u]$. For $m = m_0$ the lower bound may be violated.

(3) **Chain Length Limit Rule.** If there is a chain longer than $l(m, \alpha', p)$, then the table is rehashed using a new suitable function chosen uniformly at random from H.

For lower trimming rates the worst case guarantee tends to be weaker. By different choices of the trimming rate we set up the trade-off between the worst case guarantee and the time spent per update.

Algorithm 1 Implementation of the hash table.

```
procedure FIND(x)
    if x is inside chain t[f(x)] then
        return true                    ▷ successful case
    else
        return false                   ▷ unsuccessful case
    end if
end procedure

procedure INSERT(x)
    if x is not inside chain t[f(x)] then
        insert x into the chain t[f(x)]
        if n/m > α_u then
            REHASH(true)
        else
            l_c ← length of the chain t[f(x)]
            if l_c > l(m, α', p) then
                REHASH(false)
            end if
        end if
        return true                    ▷ successful case
    else
        return false                   ▷ unsuccessful case
    end if
end procedure

procedure DELETE(x)
    if x is inside chain t[f(x)] then
        delete x from chain t[f(x)]
        if n/m < α_l and m > m_0 then
            REHASH(true)
        end if
        return true                    ▷ successful case
    else
        return false                   ▷ unsuccessful case
    end if
end procedure

procedure REHASH(Resize)
                ▷ Also finds suitable hash function.
    if Resize then
        m ← n/α_m
    end if
    t_old ← t
    repeat
        create a new table t of size m
        choose a new function h from H
        for all x in the t_old do
            add x into the chain f(x) in t
        end for
    until lpsl < l(m, α', p)
end procedure
```

The variable t denotes the array used to store the chains.

Consequences of Trimming Long Chains

We deal with the effects of Chain Length Limit Rule on the expected length of a chain.

Definition 4 ((l,p) - trimmed system). *Let l be a limit function with a trimming rate p. The* (l,p)-trimmed system *is the multiset of functions* $H(p,l,S) = \{f \in H \mid \mathbf{lpsl}(S,f) \leq l(m,\alpha',p)\}$.

After each update we keep the function f to be from the system $H(p,l,S)$. On the other hand, it means that we do not use all the possible functions from H and thus we have to reevaluate $\mathbf{E}(\mathbf{psl})$. We show that (l,p)-trimmed systems are still universal but with a higher constant of universality.

Lemma 1. *If H is a c-universal system and l is a limit function with a trimming rate p, then $|H(p,l,S)| \geq (1-p)|H|$ and $H(p,l,S)$ is $c/(1-p)$-universal.*

Proof. From Definitions 4 and 3 it is clear that $\mathbf{Pr}\left(f \in H(p,l,S)\right) \geq 1-p$. Hence $|H(p,l,S)| \geq (1-p)|H|$. For different $x, y \in U$ it holds that

$$\mathbf{Pr}\left(f(x) = f(y) \mid f \in H(p,l,S)\right) = \frac{\mathbf{Pr}\left(f(x) = f(y),\ f \in H(p,l,S)\right)}{\mathbf{Pr}\left(f \in H(p,l,S)\right)}$$
$$\leq \frac{\mathbf{Pr}\left(f(x) = f(y),\ f \in H\right)}{1-p} = \frac{\mathbf{Pr}\left(f(x) = f(y) \text{ for } f \in H\right)}{1-p}.$$

The system $H(p,l,S)$ is thus $c/(1-p)$-universal. □

Lemma 2 gives an estimate on the expected number of trials needed to find a function $f \in H(p,l,S)$ if f is chosen uniformly from H.

Lemma 2. *If l is a limit function with a trimming rate p, then the expected number of trials needed to find a function from $H(p,l,S)$ is at most $1/(1-p)$.*

Proof. Because the subsequent choices are independent, the probability of having at least k trials is at most p^{k-1}. The expected number of trials is thus at most $\sum_{k=1}^{\infty} p^{k-1} = \sum_{k=0}^{\infty} p^k = 1/(1-p)$. □

Amortized Analysis

We analyze the amortized complexity of the model with the potential method. Let p_i be the potential of the hash table after performing the i^{th} operation and t_i be the running time of the operation. We define the *amortized complexity of the i^{th} operation* as $a_i = t_i + p_i - p_{i-1}$. The expected complexity of a sequence of k operations then equals $\mathbf{E}\left(T\right) = \sum_{i=1}^{k} \mathbf{E}\left(t_i\right) = \sum_{i=1}^{k} \mathbf{E}\left(a_i - p_i + p_{i-1}\right) = \sum_{i=0}^{k} \mathbf{E}\left(a_i\right) - \mathbf{E}\left(p_k\right) + \mathbf{E}\left(p_0\right)$.

Theorem 1. *If the hash table described in Algorithm 1 is initially empty, then the expected amortized time complexity of each operation is constant and Find operation runs in $\mathcal{O}(1 + l(m, \alpha', p))$ time.*

Proof. To prove the theorem assume that we are given a sequence of operations Insert, Delete and Find. We consider Insert and Delete successful or unsuccessful according to the comments in Algorithm 1.

First, we partition the analyzed sequence into two types of cycles. The first cycle type, α-cycle, is used to amortize violations of Load Factor Rule. With the second type, l-cycles, which are partitioning of α-cycles, we are able to distribute the time required to repair violations of Chain Length Limit Rule.

Definition 5 (α-cycle, l-cycle). *Cycles partition operations of the analyzed sequence. Each* α-cycle *ends just after the operation causing violation of Load Factor Rule. Each* l-cycle *ends after the $(\alpha' - \alpha_u)m$-th successful insertion in the l-cycle or after an operation violating Load Factor Rule (both cycles end).*

First, we define the potential used in the proof. Let $e = 1/(1-p)$, r be the number of tried hash functions because of a violation of Chain Length Limit Rule and c_l be the number of started l-cycles. Both r and c_l are counted from the initial state. Let i_α be the number of successful insertions performed in an α-cycle prior to the execution of an analyzed operation. Similarly, d_α and i_l denote the number of deletions and insertions in the α-cycle and l-cycle before the operation. We define $p_\alpha = 2ei_\alpha/(\alpha_u - \alpha_m) + 2ed_\alpha/(\alpha_m - \alpha_l)$ and $p_l = ei_l/(\alpha' - \alpha_u) + (ce - r)m$. The overall potential $p_o = p_\alpha + p_l$.

At the end of each α-cycle the size of the table has to change and Rehash operation is necessary. We pay Rehash from the potential p_α accumulated in the α-cycle. When an l-cycle starts we pay in advance for the expected number of trials in it. The potential required to prepay the cycle is accumulated during the previous l-cycle or in the initial potential.

Since Find and unsuccessful updates have no effect on the potential, we omit them from the analyzed sequence. The worst case complexity of Find follows directly from Chain Length Limit Rule because $\mathbf{lpsl}(S,f) \leq l(m,\alpha',p)$. The expected running time of Find and unsuccessful updates is $\mathcal{O}\left(1 + \mathbf{E}\left(\mathbf{psl}\right)\right)$. Since (l,p)-trimmed systems are $c/(1-p)$-universal, the expected length of a chain is bounded from above by $c\alpha/(1-p)$ and thus it is constant.

Now we analyze the sequence consisting of successful updates only. Observe that finishing an l-cycle, when the α-cycle continues, does not change the potential. Before starting a new cycle $i_l = (\alpha' - \alpha_u)m$. Putting $i_l = 0$ and incrementing c_l by one keeps the potential intact. Each operation is split into the actual update and into a possible call of Rehash. These parts are analyzed separately.

- **Update part.** The potential change caused by the update part is constant. For Insert we have $\Delta p_o = 2e/(\alpha_u - \alpha_m) + e/(\alpha' - \alpha_u)$ and for Delete $\Delta p_o = 2e/(\alpha_m - \alpha_l)$. Hence $\Delta p_o = \mathcal{O}(1)$.

- **α-cycle ends.** At the end of an α-cycle either $i_\alpha \geq (\alpha_u - \alpha_m)m$ or $d_\alpha \geq (\alpha_m - \alpha_l)m$. The potential change equals $\Delta p_\alpha + \Delta p_l \leq -2em + em \leq -em$. After rescaling the potential we are able pay the time $\mathcal{O}(em)$ which by Lemma 2 is equal to the expected running time of Rehash.
- **Chain Length Limit Rule is violated.** Each trial of a function corresponds to a single iteration of the repeat loop in Rehash procedure of Algorithm 1 and takes $\mathcal{O}(m)$ time. Because the value of r is increased by the number of trials, the potential p_l is decreased by Δrm. The time $\mathcal{O}(rm)$ taken by Rehash procedure is compensated by the loss of the potential.

Now we argue why the expected potential loss compensating the increase of r cannot be high. We show that $\mathbf{E}(p_o) \geq 0$ after any sequence of operations. To do so we analyze the sequence of sets stored after the operations in an l-cycle. If we omit deletes from the l-cycle, we get another sequence of sets created only by insertions. Consider the set S' at the end of the l-cycle in the insertion only sequence. Since there are at most $(\alpha' - \alpha_u)m$ insertions in the l-cycle and $\alpha' > \alpha_u$ we get that $|S'| \leq \alpha' m$. Clearly, for each set S represented in the dictionary during the l-cycle we have that $S \subseteq S'$. Therefore during the l-cycle we reject even smaller number of functions compared to the insertion only sequence. From Lemma 2 it follows that we expect at most e trials to find a suitable function for S'. Hence the expected number of trials during each l-cycle is at most e. Since c_l is incremented at the beginning of each l-cycle, we get that $\mathbf{E}(c_l e) \geq \mathbf{E}(r)$ and hence $\mathbf{E}(p_o) \geq 0$ after each operation.

The theorem now follows from the fact $\mathbf{E}(T) = \mathbf{E}(A) - \mathbf{E}(p_o) + \mathbf{E}(p_0) \leq \mathbf{E}(A) + \mathcal{O}(1)$ and because the amortized complexity of each operation is constant. □

Obtaining the limit function

So far we have studied the possibility of providing a worst case guarantee for Find if we are given a limit function. In this section we show various examples of sublinear limit functions.

Linear hash functions

Representing the universe U and the addresses of the hash table V as vector spaces is natural when keys and addresses are interpreted as zero-padded bitstrings of a fixed length. Alon, Dietzfelbinger, Bro Miltersen, Petrank and Tardos [1] found an interesting upper bound on $\mathbf{E}(\mathbf{lpsl})$ with the system of all linear functions between two vector spaces over the field $\mathbb{Z}_2$.

Theorem 2. *Suppose universal hashing with the system of linear functions from U to V. If $m \log m$ elements are stored in a table of size m, then $\mathbf{E}(\mathbf{lpsl}) = \mathcal{O}(\log m \log\log m)$.*

The major problem of Theorem 2 is a high multiplicative constant. However, it can be significantly reduced by a refinement of the original proof. The improved result is stated in Theorem 3 [4].

Theorem 3. *Assume universal hashing with the system of all linear transformations between vector spaces over $\mathbb{Z}_2$. Let $p \in (0,1)$ be the trimming rate and $\alpha > 0$. If $n = \alpha m$ elements are stored inside the hash table of size m, then*

$$\mathbf{E}(\mathbf{lpsl}) \leq 538\alpha \log n \log\log n + 44 \text{ and}$$

$$\mathbf{Pr}(\mathbf{lpsl} \geq a_{\alpha,p} \log m \log\log m + b_{\alpha,p} \log m) < p.$$

where the values a and b depend only on the choice of α and p.

Let us note that the exact constants for different trimming rates and load factors can be found by a simple computer program. For example the choice of $\alpha = 1.5$ and $p = 0.5$ yields $a = 57.29$ and $b = 0$. In addition, constant a can get arbitrarily close to 1 but such estimates hold only for large values of n.

Two choices paradigm

A recent study [3] of balls and bins systems discovered that hashing with at least two independent fully random hash functions brings remarkable results if it is done properly. If each stored element is put inside a least loaded chain of d ones, then with a high probability the most loaded bin contains $\ln\ln n / \ln d + \Theta(1)$ balls. Further improvements of the states result may be derived using witness tree analysis [15].

Theorem 4. *Let H be an ω-universal system and $d \in \mathbb{N}$, $d \geq 2$. Assume that each stored element x is placed into a least loaded chain of chains $f_1(x), \ldots, f_d(x)$ where hash functions $f_1, \ldots, f_d$ are chosen uniformly and independently from H. If $n \leq m$, then $\mathbf{Pr}\left(\mathbf{lpsl} > \frac{\ln\ln n}{\ln d} + 5\right) \in o(1)$.*

Bounds for k-wise independence with constant number of functions and for general $S \subset U$ do not follow from the stated result and are not trivial. On the other hand the stated result holds in case of uniform or almost uniform systems. In addition a convenient example of a uniform system appeared in [13]. The system consists of functions which can be computed in a constant time. Let us note that a slight increase of the trimming rate occurs when using the uniform system since its uniformity is probabilistic.

The problem of using k-wise independent functions is addressed by Woelfel [17]. This work shows how to use simple universal classes with the two choices paradigm. So we can conclude that there is a way how obtain an impressive doubly logarithmic worst case warranty with the two choices paradigm.

Improvements and conclusion

Our work shows a model of a hash table which guarantees a worst case running time of Find operation if a proper limit function is provided. The advantage of the construction is that it is fully dynamic and works with limit functions derived for static sets as well.

Finally we are able to remove the use of separate chaining. In case of linear probing when we have a limit function on the length of the probe sequence, then our approach works as well. In case of linear probing combined with the two choices if the input data is random enough, we can assume full randomness of the hash function and get the limit function $\mathcal{O}(\log \log n)$ as described in [12] and [10]. Naturally, without any further assumption we can use the mentioned almost uniform system and get a doubly logarithmic worst case limit.

In case of separate chaining the provided warranty can be improved by representing chains by another hash table. Since updates in both hash tables take expected amortized constant time, operations in the combined table are fast. To conclude better cache utilization and thus greater improvement should be obtained by use of linear probing. Experiments confirming the mentioned behavior are the topic of our further study.

Acknowledgments. We would like to thank Václav Koubek and Vladimír Čunát for helpful comments. The present work was supported by Grant Agency of Charles University under Contract 205-10/251473.

References

[1] Noga Alon, Martin Dietzfelbinger, Peter Bro Miltersen, Erez Petrank, and Gábor Tardos. Linear hash functions. *J. ACM*, 46(5):667–683, 1999.

[2] Yuriy Arbitman, Moni Naor, and Gil Segev. Backyard cuckoo hashing: Constant worst-case operations with a succinct representation. In *FOCS*, pages 787–796, 2010.

[3] Yossi Azar, Andrei Z. Broder, Anna R. Karlin, and Eli Upfal. Balanced allocations (extended abstract). In *STOC*, pages 593–602, 1994.

[4] Martin Babka. Properties of Universal Hashing. Master's thesis, Charles University in Prague, Czech Republic, 2010.

[5] Larry Carter and Mark N. Wegman. Universal classes of hash functions. *J. Comput. Syst. Sci.*, 18(2):143–154, 1979.

[6] Martin Dietzfelbinger and Friedhelm Meyer auf der Heide. A new universal class of hash functions and dynamic hashing in real time. In *ICALP*, pages 6–19, 1990.

[7] Martin Dietzfelbinger, Anna R. Karlin, Kurt Mehlhorn, Friedhelm Meyer auf der Heide, Hans Rohnert, and Robert Endre Tarjan. Dynamic perfect hashing: Upper and lower bounds. *SIAM J. Comput.*, 23(4):738–761, 1994.

[8] Michael L. Fredman, János Komlós, and Endre Szemerédi. Storing a sparse table with 0(1) worst case access time. *J. ACM*, 31:538–544, June 1984.

[9] Maurice Herlihy, Nir Shavit, and Moran Tzafrir. Hopscotch hashing. In *DISC*, pages 350–364, 2008.

[10] Ebrahim Malalla. *Two-way hashing with separate chaining and linear probing.* PhD thesis, Montreal, Que., Canada, Canada, 2004. AAINR06322.

[11] Kurt Mehlhorn. *Data Structures and Algorithms 1: Sorting and Searching*, volume 1 of *Monographs in Theoretical Computer Science. An EATCS Series.* Springer, 1984.

[12] Michael Mitzenmacher and Salil P. Vadhan. Why simple hash functions work: exploiting the entropy in a data stream. In *SODA*, pages 746–755, 2008.

[13] Anna Pagh and Rasmus Pagh. Uniform hashing in constant time and optimal space. *SIAM J. Comput.*, 38(1):85–96, 2008.

[14] Rasmus Pagh and Flemming Friche Rodler. Cuckoo hashing. In *ESA*, pages 121–133, 2001.

[15] Berthold Vöcking. How asymmetry helps load balancing. *J. ACM*, 50(4):568–589, 2003.

[16] Mark N. Wegman and Larry Carter. New classes and applications of hash functions. In *FOCS*, pages 175–182, 1979.

[17] Philipp Woelfel. Asymmetric balanced allocation with simple hash functions. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 424–434, 2006.

Neocognitron: A Survey of a Classical Hybrid Neural Network Model

M. Kukačka

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. The Neocognitron neural network was introduced in 1980, and since then, it has developed from a model of brain's visual cortex into an effective pattern recognition tool. The model has many noteworthy properties, among others the ability to learn and classify visual patterns without any need for data preprocessing and the major use of self-organization in its learning algorithm. During its development, the model has undergone many modifications and extensions, in some cases gaining interesting abilities, such as the ability to restore damaged patterns. This article provides a survey of the Neocognitron's basic functionality and the ideas of its most interesting modifications.

Introduction

The Neocognitron neural network model was introduced in 1980 by Kunihiko Fukushima as an improvement of an older Cognitron model. The architecture of the network was inspired by the classical article by Hubel and Wiesel, which described cell structures of visual cortex in cat's brain. The network was supposed to provide a model of this architecture and allow a deeper understanding of its function. However, over the years of Neocognitron development, the model turned from a tool for brain modeling and research into a pattern recognition model. The performance of this model has been demonstrated on hand-written character recognition tasks, without using any data pre-processing and with impressive results.

The model has undergone a long development, starting in 1980 with a basic version of the network and continuing in a large number of improvements and modifications, which are described in numerous articles. The network was extended in various manners, with neural circuits being added to provide new functionality, or removed, when another improvement made them redundant. This article aims to provide a description of the basic Neocognitron model together with an overview of the most significant modifications and their effect on the network's performance. The ideas used in the construction of the Neocognitron can be utilized in other models, their influence is found in some of the most successful pattern recognition models, such as Yann LeCun's LeNet [*LeCun et al.*, 1998].

Network structure

The network, in its basic form, consists of alternating layers of two types of cells - the "simple" cells, denoted as *S-cells*, and the "complex" cells, denoted as *C-cells*. The S-cells work as feature detectors, with each cell in the layer detecting the presence of a different feature in the layer's input data. Also, these cells are the only cells in the network with adaptable weights, therefore the learning of the network is performed only on the S-layers. Each S-cell is accompanied by a *V-cell*, which has the same input as the S-cell, simple averaging activation function and an inhibitory connection to the S-cell. The V-cell helps the S-cell discriminate the feature it has learned to recognize from all other features.

The C-cells' function is hiding the exact position of the detected feature and thus improving the network robustness to deformations of the pattern, such as scaling, shifting of the pattern's position, or noise. The S-C pair of layers compose a *stage*. The early versions of Neocognitron (e.g. the version described in [*Fukushima et al.*, 1982]) consisted of 3 such stages, later versions (e.g. [*Fukushima*, 2010] consist of four stages. Also, in the later versions of the network, namely in the network described in [*Fukushima*, 2003], another type of cell has been introduce - the so-called *G-cell*, used in the first layer of the network (before the first S-layer) for contrast extraction. The structure of the network, as described in [*Fukushima*, 2010], is shown in Figure 1.

All of the cells in the network have local input field, which means that input connections of a cell are connected only to a small area of the layer's input. The cells in higher layers of the network therefore react to a larger area of the network's input. The pattern recognition performed by the network is thus done by the means of detecting local features and combining them in higher layers into more complex

features. The result of pattern classification is read from the last layer, which should contain cells reacting to the whole area of network's input.

Since a local feature should be detected in the same way regardless of its position in the data, the cells are organized into *cell-planes*, which group cells reacting to the same feature. In case of S-layers, cells in each plane are constrained by the learning algorithm to have the same weights. This results in the cell-plane's output representing a map of occurrences of the feature detected by the plane's cell over all positions in the layer's input. In case of C-layers, all cells in a single cell-plane are connected only to cells in one corresponding plane in the underlying S-layer, thus representing the same feature as this S-plane. This results in the C-layer always having the same number of cell-planes as the S-layer of the same stage.

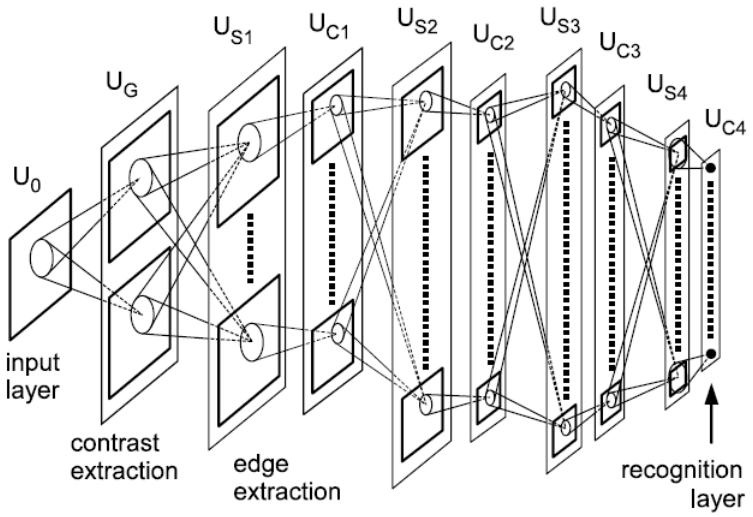


Figure 1. Structure of the Neocognitron network, as described in [*Fukushima*, 2010]. The image is taken from [*Fukushima*, 2010]. It shows the alternating of layer types, organization of cells into cell-planes, cells' local input fields and connections between the cell-planes.

S-cells

The feature-detecting S-cell has an input area of circular shape and pre-defined radius. The input area spans all cell-planes of the underlying layer, with the circular area positioned at the same location in each plane. If the cell-planes of the lower layer were organized as pages in a book, the input area of a cell would form a column, taking up a circular area of each "page" at the same position.

Besides the position of a cell in its cell-plane and the diameter of its input area, the cell is parametrized by values of its excitatory weights, the value of its inhibitory weight connecting it to its V-cell, and its threshold. The activation function of S-cells has changed in some respects during the model's development, here are three examples from [*Fukushima et al.*, 1982], [*Fukushima et al.*, 1997] and [*Fukushima*, 2010], respectively:

$$u = r \cdot \varphi\left[\frac{1 + \sum_i a_i x_i}{1 + \frac{r}{1+r} bv} - 1\right] \tag{1}$$

$$u = \frac{\theta}{1-\theta} \cdot \varphi\left[\frac{1 + \sum_i a_i x_i}{1 + \theta bv} - 1\right] \tag{2}$$

$$u = \frac{1}{1-\theta} \cdot \varphi\left[\frac{\sum_i a_i x_i}{bv} - \theta\right] \tag{3}$$

In these equations, r is the cell's *saturation parameter*, which was transformed in later versions of the model into *cell's threshold* θ, and it can be seen that $r = \theta/(1-\theta)$. The function φ is defined as $\varphi(x) = max(x, 0)$. Cell's input values are denoted by x_i and the corresponding excitatory weights are denoted by a_i, in both cases the cell's position in the cell-plane and the exact position of the input cell are contracted into one index i for the sake of simplicity. The weight of the inhibitory connection between the cell and its corresponding V-cell is denoted by b, and the output value of the V-cell is denoted by v.

To complete the description of an S-cell's output function, the activation function of the V-cell is defined as:

$$v = \sqrt{\sum_i c_i x_i^2} \tag{4}$$

where c_i denote the V-cell's fixed weights, which decrease with distance from the center of the cell's input area.

All of the versions of S-cell's activation function describe approximately the same behavior of the cell. The cell measures the similarity between its input vector and its weight vector, and if this similarity is greater than the cell's threshold, then the cell outputs positive value less than or equal to 1, otherwise the cell outputs 0. The S-cell's feature-detecting mechanism will be discussed in more detail below, in the section on vector notation.

C-cells and G-cells

The function of C-cells in the Neocognitron model is to hide the exact position of features, detected in the underlying S-layer, and in this way improve the network's robustness against pattern deformation and noise. C-cell grouped in one cell-plane have input connections coming only from a single S-plane. Therefore the C-planes map the same features as their corresponding S-planes, usually with lower spatial resolution.

The general rule is to activate a C-cell when at least one S-cell in its input area is activated. The degree of the C-cell's activation depends on the activation of S-cells in its input area, and on the used activation function, which, just like in the case of S-cells, has changed during the model's development. As an example, the functions used for activation of C-cells in [*Fukushima*, 1988a] and [*Fukushima*, 2010], are presented here. The general form of the function is the same in both cases:

$$u_c = \Psi\big(\sum_i d(i)u_s(i)\big) \tag{5}$$

However, in [*Fukushima*, 1988a], the Ψ function is defined as $\Psi(x) = \frac{\varphi(x)}{1+\varphi(x)}$, where, as above in the section on S-cells, $\varphi(x) = max(x, 0)$. In [*Fukushima*, 2010], the function is defined as $\Psi(x) = \sqrt{x}$.

The G-cells have been introduced into the model in [*Fukushima*, 2003]. In this version of the model, the G-cells form two cell-planes in the network's first layer, which precedes the first feature-detecting S-layer. In these cell-planes, contrast extraction is performed on the input data. The G-cells use the following function to perform this extraction:

$$u_G = \varphi\Big[(-1)^k \sum_i g_i u_i\Big] \tag{6}$$

where $k \in \{0, 1\}$ is the index of the G-plane and g_i are the cell's weights, which have fixed values of the Mexican hat function centered on the input area of the cell. The contrast extraction helps the S-cells in the first S-layer with detection of lines and edges, resulting in improvement of the network's performance.

The learning algorithm

The first version of Neocognitron, as described in [*Fukushima et al.*, 1982], used only *self-organization* for its learning. In later versions, several forms of *supervised* learning were introduced for different parts of the network. However, these supervised algorithms do not resemble the well-known back-propagation algorithm, they could rather be called "user-assisted" self-organization. With the Neocognitron's learning algorithm, the model is able to learn to distinguish between different categories in the data with little or no assistance from the user.

The key property of the learning algorithm is the way cells are selected for adaptation. Since S-cells grouped in a cell-plane are constrained to have the same weights, only a single cell can be selected from each cell-plane in one adaptation step - this cell is then called the *seed cell*. Also, the algorithm forces different cell-planes to learn to recognize different local features. This is achieved by allowing only a single cell from a *hypercolumn* - a group of cells with the same input area (and, equivalently, with the same position in their respective cell-plane) - to be adapted to a certain input area in one step. The seed-cell selection procedure for one learning step can be described as follows:

- in each hypercolumn, the cell with the highest output value is selected as the *seed candidate*
- in each cell-plane, the seed candidate with the highest output is selected to be the seed-cell. This ensures there can be at most one seed-cell in each cell-plane.
- the selected seed-cells' weights are adapted to its input values, and along with them, all cell in their cell-plane have their weights changed in the same manner
- if there is no seed-candidate in a cell-plane, then cells in this cell-plane are not adapted in this step

The weight adaptation is done according to these simple equations:

$$\Delta a_i = q \cdot c_i \cdot x_i \tag{7}$$
$$\Delta b = q \cdot v \tag{8}$$

where q is a learning parameter, a_i are the S-cell's excitatory weights, b is the cell's inhibitory weight, v is the output of the corresponding V-cell, c_i are the V-cell's weights, and x_i are the cell's input values. Later in development of the model, the learning parameter q has been removed from these equations, its role taken by the values of c_i weights. Also, in the more recent versions of the model, the inhibitory weight b is directly computed from excitatory weights a_i instead of being determined by adaptation. In [*Fukushima et al.*, 1997], the following equation is used for this:

$$b = \sqrt{\sum_i \frac{a_i^2}{c_i}} \tag{9}$$

Vector notation

If we describe the Neocognitron's learning algorithm with vector notation, we gain a deeper understanding of the mechanism used for feature detection within the network. Let us denote the sum of all training vectors, to which cells from a single cell-plane have been adapted, as the *reference vector* $\vec{X} = \sum_p \vec{x}^{(p)}$, where the index p goes over these local training vectors. Let us also define the *weighted inner product* as $(\vec{x}, \vec{y}) = \sum_i c_i x_i y_i$ and the corresponding norm as $||x|| = \sqrt{(x, x)}$.

The weights of cells in the network are initially set to small values, which can be ignored in the following description. Considering the weight adaptation equation $\Delta a_i = c_i x_i$, we see that the values of weights gain the value of $a_i = \sum_p c_i x_i^{(p)}$, therefore the dot product from equation 3 can be rewritten to $\sum_i a_i x_i = (\vec{X}, \vec{x})$. Also, using the definitions above, we can see that the inhibitory weight has the value of $b = \sqrt{\sum_i \frac{a_i^2}{c_i}} = ||X||$ and the output of the V-cell is equal to $v = \sqrt{\sum_i c_i x_i^2} = ||x||$.

We can now rewrite the equation 3 as

$$u = \frac{1}{1-\theta} \varphi \left[\frac{\sum_i a_i x_i}{bv} - \theta \right] = \frac{1}{1-\theta} \varphi(s - \theta) \tag{10}$$

where $s = \frac{\sum_i a_i x_i}{bv} = \frac{(\vec{X}, \vec{x})}{||X|| \cdot ||x||}$ is the *resemblance* of the cell's reference vector $\vec{X}$ and the presented local feature $\vec{x}$. It is very similar to the formula for cosine of an angle between the two vectors, with the exception of using the weighted inner product. However, it can be interpreted as such - a cosine-like description of the difference between the cell's reference vector $\vec{X}$ and the presented input vector $\vec{x}$. If this value is greater than the cell's threshold, meaning the input is within the cell's *tolerance area*, then the cell reacts to this input, otherwise the cell does not react (i.e. gives the output of zero). By changing the cell's threshold, we change the cell's tolerance to the deformation of the feature it has learned to recognize.

Supervised learning

Several forms of supervised learning have been introduced to the model in the course of its development. They generally consist of some intervention during the network's adaptation. For example, manually connecting several S-planes to a single C-plane has been used in the model's early development (e.g. [*Fukushima et al.*, 1988a]), in order to group the features detected by the S-planes into a single feature mapped by the C-plane. This was used to help the model recognize lines with certain orientation as a single feature, even though different parts of the line had different pixel patterns. This form of supervised learning is not used in the model's more recent versions.

In the model described in [*Fukushima et al.*, 1988a], supervised learning is used to help the last S-layer correctly classify the presented samples. Each cell-plane in the last C-layer contains only a single C-cell, which is influenced by the whole area of the network's input. Each C-cell in this layer corresponds to a distinct class of input samples. The presented sample can therefore be classified by determining which C-cell in the last layer has the highest output value. During the last S-layer's adaptation, a teacher assigns labels to the S-cells based on what class of samples they react to. This labeling can be done by connecting the S-plane to the C-plane corresponding to the label. If an already labeled S-cell wins the competition for a sample of different class, a new S-plane is created, containing an S-cell set to recognize this particular sample, and this S-plane is correctly labeled. A new S-plane is also added in this manner if no S-plane reacts to the presented sample. This method generally leads to creation of S-cells along the borders between classes, helping the network to determine these borders more precisely.

Winner-kill-loser

In the first version of Neocognitron, the number of cell-planes in each layer was fixed prior to the adaptation of the network. This caused the network to have limited capacity and it could learn to classify only a limited number of classes. In [*Fukushima et al.*, 1997], a method for adding new cell-planes to an S-layer during its adaptation is described. This allows the number of cell-planes in the S-layer to grow based on the number and distribution of features in the lower layer's output. The idea of this method is very simple: if there is a position in the S-layer's cell-planes, at which all cells are silent even though their shared input area contains non-zero values, then there is an undetected feature in this area. In this case, a new cell-plane is created to detect this feature. Instead of creating a new cell-plane, a cell-plane which has not yet been adapted can be set to react to this feature. This results in every feature appearing in the lower layer's output being "covered" by some S-cell's tolerance area. Such automated growth of feature-detecting layers improves the network's ability to describe the processed pattern, and thus improves its recognition rate.

To counteract the growing of number of S-planes, the *winner-kill-loser* adaptation rule has been proposed in [*Fukushima et al.*, 2010]. In this case, the seed-cells are selected in the order of their descending output values, and only if no seed-cell has yet been selected in the same cell-plane. This differs from the original seed-selection method, in which several seed-candidates could be selected from a single cell-plane if they had the highest output values in their respective hypercolumns. If there are other non-silent cells in a hypercolumn beside the selected seed-cell, they are removed along with their cell-planes. This results in every feature being covered by only a single cell's tolerance area, and therefore being represented by only one cell-plane. Of course, the removals may cause some features not being recognized by any cell - in this case, cell-planes will be added again for these features the next time they appear in the layer's input. This method helps to moderate the resulting size of the layer, optimizing the size of the network's layers and further improving its performance.

Selective attention and pattern restoration

One of the most interesting modifications, which were made to the Neocognitron model over the years of its development, is the addition of backward paths to the network, as described in [*Fukushima*, 1988b] and [*Fukushima et al.*, 1993]. These backward paths, together with several accessory circuits, allow the network to segment learned patterns from its input. This form of *selective attention* enables the network to focus on and recognize individual patterns in an image containing more than one pattern. Also, with the use of backward paths, the network is able to remove noise from the input image and restore damaged patterns according to their learned forms. Similar approach has been described in [*Fukushima*, 2005], where the backward signal is used to restore and recognize partly occluded patterns.

The backward signal originates from the recognition layer, the last C-layer, and only from active cells, representing the most prominent patterns in the network's input. Usually, only one cell in the recognition layer is active, representing the result of classification performed by the network. From this cell, the signal travels through backward cells which mirror the structure of the forward signal cells. In the lowest backward layer, mirroring the input layer of the network, the result of segmentation is formed. This recalled pattern is the focus of the network's attention.

The forward and backward signals interact with each other at each layer of the network. The backward signal influences the thresholds of forward cells, *facilitating* the forward signals. At the same time, the backward signal is allowed to continue in the downward direction only along paths corresponding to active forward signal, which in this way *gates* the backward signal.

Several accessory circuits help the network in segmenting the patterns and restoring damaged patterns. The *no-response detector* raises the network's tolerance to deformation of features in case no cell in the recognition layer is activated after presentation of the image. This allows the network to recognize even very deformed or noisy patterns. The *attention switcher* periodically interrupts the backward signal and raises the threshold of active cells. This allows a different pattern present in the input to be segmented out and processed.

Double thresholding

In [*Fukushima et al.*, 1996], the use of different thresholds for learning and recognition phases of the Neocognitron has been proposed. The use of higher threshold in S-cells during their adaptation makes them less tolerant to feature deformation. This leads to creation of higher number of S-planes and thus to more detailed description of the presented patterns. On the other hand, the use of lower thresholds in S-cells during the recognition phase improves their robustness against noise and deformation and improves the recognition rate.

This simple technique improves the network's recognition efficiency. On the other hand, it makes the space of network's parameters considerably larger and therefore more difficult to thoroughly search for optimal values.

Performance

In the early articles on the Neocognitron model, its recognition performance was demonstrated on a very small set of characters. This was caused by the hardware restrictions of computers of that time, which allowed only a small number of neurons to fit into the computer's memory, thus limiting the capacity of the network.

In more recent articles, the ETL1 database of handwritten characters has been used to demonstrate the capabilities of Neocognitron. The modifications of the model, especially its ability to increase the number of cell-planes in its layers based on the complexity of the training data, allowed the network to reach impressive recognition rates on this dataset. In [*Fukushima*, 2003], the recognition rate of 98.6% was reached on a testing set when using 3000 samples for training, demonstrating the network's generalization ability. With further improvements of the model, including the winner-kill-loser learning technique described above, the model variant introduced in [*Fukushima*, 2010] reached the recognition rate of 100% on the testing set.

Conclusion

The Neocognitron neural network model uses local features and several types of neurons to perform recognition of visual patterns. Its learning algorithm is based on self-organization and supplemented by a few supervised techniques. Due to this learning algorithm and the structure of the network, the model is able to reach impressive recognition rate on complex patterns without any need for data pre-processing.

The long development of the model brought many modifications to the network's structure, the function of its cells, and to the learning algorithm. Some of these modifications, together with the network's basic principles and mechanisms, have been described in this article. The Neocognitron model has inspired other successful neural network models, demonstrating the principles it utilizes can be combined with other learning and processing methods. These principles can certainly be used to advantage in design of future pattern recognition models.

Acknowledgments. The author would like to thank his advisor, doc. RNDr. Iveta Mrázová, CSc., for her advices and guidance. The presented work was supported by the Czech Grant Agency under the contract no. 201/09/H057 and by the Charles University Grant Agency under the contract no. 4361/2009 (grant no. 136109).

References

K. Fukushima, S. Miyake: "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position," *Pattern Recognition*, 15[6], pp. 455-469 (1982).

K. Fukushima: "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural Networks*, 1[2], pp. 119-130 (1988a).

K. Fukushima: "A neural network for visual pattern recognition," *IEEE Computer*, 21[3], pp. 65-75 (1988b).

K. Fukushima, T. Imagawa: "Recognition and segmentation of connected characters with selective attention," *Neural Networks*, 6[1], pp. 33-41 (1993).

K. Fukushima, M. Tanigawa: "Use of different thresholds in learning and recognition," *Neurocomputing*, 11[1], pp. 1-17 (1996).
K. Fukushima, K. Nagahara, H. Shouno: "Training neocognitron to recognize handwritten digits in the real world," *Proceedings, The Second Aizu International Symposium on Parallel Algorithms/Architectures Synthesis*, March 17-21, 1997, Aizu-Wakamatsu, Japan, pp. 292-298, IEEE Computer Society Press, Los Alamitos, CA.
K. Fukushima: "Neocognitron for handwritten digit recognition," *Neurocomputing*, 51, pp. 161-180 (2003).
K. Fukushima: "Restoring partly occluded patterns: a neural network model," *Neural Networks*, 18[1], pp. 33-43 (2005).
K. Fukushima: "Neocognitron trained with winner-kill-loser rule," *Neural Networks*, 23[7], pp. 926-938 (Sept. 2010).
Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Gradient-Based Learning Applied to Document Recognition, *Proceedings of the IEEE*, 86(11):2278-2324, 1998

Towards Believable Intelligent Virtual Agents with StateFull Hierarchical Reactive Planning

T. Plch

Charles University Prague, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. Intelligent Virtual Agents (IVAs) deployed within virtual worlds are required to exhibit believable rational behavior. The reactive action selection mechanism represents a popular choice for most Artificial Intelligence (AI) architectures, mainly due to simple implementation (e.g. Finite State Machines, If-Then rules etc.) and timely execution. We identified several problems of the commonly utilized Hierarchical Reactive Planning (HRP) approach summarized into three classes: a) intentions, b) planning, and c) transitional behaviors. We extended the HRP approach by a StateFull semantic layer on top of the reactive If-Then rule set thus creating StateFull HRP (SF HRP) architecture. Our architecture's goals are twofold 1) aid action selection to maintain believable behavior (e.g. transitions, plan aftereffect etc.) and 2) add semantic data to reactive plans on which reasoning can be performed. Our paper presents the SF HRP architecture and the possible use of additional data for object driven reasoning. We implemented a simple proof of concept prototype on which we base our conclusions.

Introduction

The term of intelligent virtual agent (IVA) denotes a software agent situated within a dynamic, complex, real–time, and unpredictable 3D virtual environment. The requirements on IVA's rationality [*Russel et al*, 2003] and believable behavior [*Loyall*, 1997] increases with ever growing realism of virtual environments they inhabit. The *Action Selection Mechanism* (ASM), which produces the IVA's behavior, is one of the dominating topics for agent Artificial Intelligence (AI) research.

The main goal of the ASM is to choose actions by which the agent can influence its environment and satisfy its goals, which are either predefined by the AI designer or a product of the agent's reasoning. The ASMs for IVAs can be divided into two basic groups – a) *deliberative* and b) *reactive*, based on the approach they choose the next action. Classical planning [*Ghallab et al*, 2004] is the typical representative of the deliberative paradigm. The main idea is to produce a sequence of actions creating a transition from the current to the goal state, based on the action's preconditions and effects.

On the other hand, the reactive approach, where Finite State Machines (FSM) [*Champandard*, 2003] and Hierarchical Reactive Planning (HRP) [*Bryson*, 2001] are key representatives, is build around the idea of selecting an action based on the current context the agent periodically perceives (i.e. at every AI engine update). The main benefits of this approach are the IVA's responsiveness within dynamic environments and the implementation's simplicity. However, the resulting behavioral patterns are repetitive and schematic.

Our paper's focus is on the topic of reactive planning, especially the Hierarchical Reactive Planning (HRP) variant. Based on the computation power available for AI engines within most simulation, it represents a very capable candidate for believably behaving IVAs. However, Brom [2005] presented several problems of the HRP approach which degrade the believability of the resulting behavior. The goal of this paper is to present our StateFull HRP architecture which is designed to overcome the observed issues and further add modularity to the HRP concept. As a proof of concept, we implemented a simple simulation of our concept [*Plch*, 2009] and ran several simple scenarios illustrating the observed issues.

The paper's structure is as follows. The following section describes the concept of HRP in more detail. The third section is concerned with problems we used as guidelines for our design. The fourth section describes our StateFull HRP architecture in detail. The fifth section is concerned with our prototype implementation. The sixth section focuses on future work. The last section is concerned with the conclusion.

Hierarchical Reactive Planning (HRP)

The Hierarchical Reactive Planning (HRP) is one of the basic representatives, besides Finite State Machines (FSM), for the reactive action selection paradigm [*Bryson*, 2001]. The HRP's key component is the Reactive Plan (RP), being a set of *If-Then rules* (Table 1) ordered by priority. The RPs recursively form a hierarchical tree-like structure – a behavioral tree called *BE-Tree*, where nodes are RP and plan links (**Goto**) are edges. Leafs of the BE-Tree are execution primitives (e.g. atomic actions, action sequences etc.) which impact the state of the world (e.g. "grab shovel", "throw axe").

Priority	Precondition	Action
1	LowHealth	**Goto(**Medikit)
2	LowAmmo	**Goto**(Reload)
3	seeEnemy	Shoot
4	seeFriend	**Goto**(Greet)

→

Priority	Precondition	Action
1	See(Enemy)	Run
2	See(Medkit)	Grab
3	Not(See(Medkit))	Search

Table 1. Simple example of a HRP's Reactive Plan "Fight" and a subplan "Medkit"

The *Priority* represents an ordering within the RP on how important a rule is. The *Precondition* represents a logical trigger, which when satisfied, makes a rule a candidate for execution. The *Action* may be twofold – either an execution primitive (e.g. single atomic action, action sequence etc.), or a *plan link* – presented as **Goto** statement in Table 1 which moves the search for an execution primitive deeper into the BE-tree's hierarchical structure to a lower level RP (i.e. a subplan).

Formal representation

Similar to [*Bryson*, 2001], the HRP can be seen as a system choosing an action changing the current world state, without utilizing any look-ahead. The system can be formally specified as a triplet $\Sigma = (S, A, \gamma)$, where S is a recursively enumerable set of agent states (agent's perceptions of the world), A is a recursively enumerable set of actions, and γ is a function $S \times A \rightarrow \mathcal{P}(S)$, where $\mathcal{P}(S)$ is the power set of S. A world's state is similar to the classical planning's representation [*Ghallab et al*, 2004], as a set of positive literals (grounded, function-less predicates) in first order logic.

The hierarchy only helps the AI design process by decomposing the activities into a hierarchical structure. Formally, the hierarchy can be translated into a single RP, where every execution primitive is at the same level and its releaser is a conjunction of all releasers along the BE-tree's path from the root to the execution primitive in a leaf node. The priority is a concatenation of all the priorities over the path (i.e. priorities $1 \rightarrow 2 \rightarrow 5$ translates into a single priority 125). Every priority is padded with 0 (at the end of the string, e.g. 12 translates into 120), so all the priorities have the same length (i.e. the length of the deepest execution primitive). Therefore, hierarchical view is irrelevant, both representations (one level RP and HRP) are equivalent.

The HRP action can be formalized as a triplet *(Priority, Precondition, Act)*, where *Priority* $\in \mathbb{N}$, *Precontion* is a boolean expression in first order logic describing agent states and *Act* is an execution primitive that results in executing an action from A.

Action Selection Mechanism

The ASM for HRP is show in Algorithm 1. The main idea of the algorithm is to recursively search the BE-Tree structure until an atomic action can be executed. The rules are chosen simply by considering two factors – the priority of the rule and a holding trigger. The algorithm follows the plan links and iterates at every level of the hierarchy. It is noteworthy that problems resulting from the simplicity of ASM were covered in [*Plch et al*, 2010]. This ASM algorithm is executed at every sensory cycle to determine which action is to be executed.

```
ASM( root ):
(1) plan ← root
(2) preactive rules ← get all rules from plan with holding trigger, order by priority
(2) while (preactive rules not empty )
(2.1) active rule  ← get and remove first of preactive rules
(2.2) case ( active rule )
(2.2.1) (is executable ) ret ← (execute (active rule) )
(2.2.2) (is plan link ) ret ← (ASM (active rule ))
(2.3) if ( ret is fail ) continue else return (ret)
(3) return fail
```

Algorithm 1. Action selection algorithm for HRP

Problems o f HRP

The main advantages of HRP are simplicity, ease of implementation and timely fashion of execution. These attributes made this approach a very popular one, used in computer games [*Isla*, 2005], military simulations [*Reidl et al*, 2006] and many others. The work presented in [*Brom*, 2005] and [*Plch*, 2009] studied the HRP techniques in further detail and discovered several key problems of the behavior resulting from HRP. These problems can be summarized into three classes: a) intentions, b) planning and deliberation, and c) transitional behavior.

Intentions

Based on the Belief-Desire-Intention [*Wooldridge*, 2002] model, an intention is a deliberative state, something the agent wants to achieve. An intention can be achieved by satisfying a set of goals, which are satisfied by reactive plans. This can be illustrated as follows: *intention →goals → plans*
Reactive planning provides only static structures with no capability to *add*, *remove* or *manage* intentions (with associated goals and plans), rendering the agent less receptive to newly introduced situations, able to solve only those who are hard-wired into his brain by the designer.

Planning and deliberation

In general, *planning* (in a classical sense) is what reactive planning tends to elude, keeping the agent responsive and reactive to its surroundings. However, a degree of planning is necessary to make the behavior of IVAs appear more intelligent and believable. Humans expect to see some deliberation and planning, to perceive intelligent behavior and see the IVA as more self-aware. We can divide the problems for reactive planning into categories: a) *pre-execution* b) *post-execution* c) *ordering*.

Pre-execution planning is tightly coupled with the ability to prepare the footing for successful plan execution (e.g. by timely acquisition of proper objects) or for the execution of a subplan (*deep preparation*) or a concurrent plan (*cross plan preparation*). Post-execution is concerned with smart cleaning up of objects. Effective ordering (*chaining*) of plan can even when breaking priority of execution, can lead to more effective execution patterns.

Transitional behavior

In dynamic environments, an executed reactive plan A could be interrupted and suspended by another plan B with higher priority (Figure 1). To be able to start plan B and later resume the plan A, it (plan A) has to be discontinued in a consistent way (e.g. hands of the agent are empty). In most cases, the observer awaits a smooth transition from one to the other behavior.

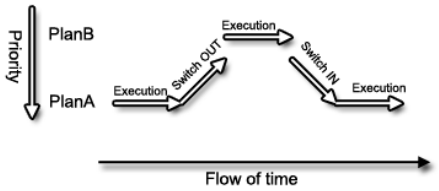


Figure 1. Simple example of PlanB with higher priority which suspends the PlanA with lower priority. Plan A is later resumed.

This topic was also addressed in [*Mikula*, 2006] and [*Bída et al*, 2011]. Expressing transitional behaviors can be almost impossible to achieve using plain HRP, mostly due to the fact, that plans do not have any knowledge of other plans that could succeed or surpass them.

A part of transitional behaviors is the *cleaning behavior* humans exhibit – when an activity is finished, items are put back to their respective places. The notion of cleaning up is tightly related to the knowledge of used objects by the current activity and other future activities that might make use of the objects in question. Creating a believable behavior using only HRP facilities could result in large complex plans and might not work at all.

StateFull Hierarchical Reactive Planning (SF HRP)

In this section, we present the main idea of the StateFull Hierarchical Reactive Planning (SF HRP) architecture. We extend the HRP's execution into a more structured form and base our design on the observation how actual Reactive Plans are designed by decomposing the task they address into phases by utilizing the priority of specific rule sets or creating subplans – *preparation*, *execution*, *cleanup*. We also adopt and extend the notion of *swiching* [*Mikula*, 2006], by introducing a specific set of phases addressing this issue. Thirdly, we add semantic data (*object list* and *object classes*) to RP on object needs and usage which can be utilized to perform reasoning and planning.

We introduce the StateFull Plan (SFP) which combines a Finite State Machine (FSM) integrating the RP's execution as a part of its workflow (Figure 2), thus maintaining backward compatibility with the HRP approach. We recognize the following phases: 1) initialization – used to prepare for plan execution (e.g. collects necessary objects), 2) execution – used to perform actions of carry out an activity (e.g. cut down trees), 3) termination – used after the plan terminates to perform necessary activities (e.g. empty agent's hands), 4) finish – used to evaluate the plan (e.g. choose objects to keep and for cleanup), 5) cleanup – used to perform cleanup (e.g. take

the axe back to a shed), 6) switch out – used for consistently suspend the plan (e.g. drop objects in hands), 7) switch in – used to resume the suspended plan (e.g. find the axe and grab it), 8) switched – representing the suspended state (e.g. agent whistling while doing nothing), and 9) emergency – used for special occasions (e.g. someone throws an axe at the agent, he dodges). It is noteworthy, that minor changes to the rule formalism presented in [*Plch et al, 2010*] further augment our architecture. The transitions between these phases can be seen in Figure 3. The actual execution of the FSM is triggered when the SFP is chosen based on its priority and holding releaser in the BE-Tree's layer one level above.

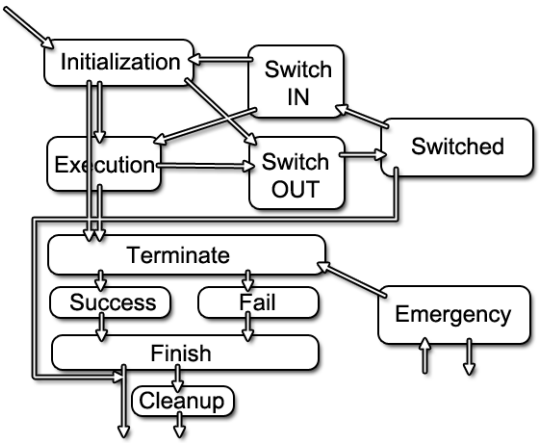


Figure 2. StateFull Plan's phases of execution. Boxes are phases and arrows are possible transitions

In most cases the SFP's execution is as follows – the SFP first enters the initialization phase, prepares for execution (e.g. collects objects) and afterwards enters the execution phase. During both of these phases, the workflow might be suspended by a higher priority SFP and the plan is switched out either in generic way (i.e. default switch out) or in respect to the suspending SFP (i.e. switch out based on a suspender ↔ suspended (plan) pair). After the suspended SFP (i.e. the switch in phase) is returned into execution, it continues and possibly terminates – either as success or fail. After performing some evaluation, the workflow finishes and possibly enters a Cleanup phase. This phase can be omitted or only partially executed, where the plan higher in the hierarchy takes responsibility for the cleanup. All mentioned phases can be realized either by a build-in code or script (e.g. C++ routines, LUA script) or by a simple SFP or RP.

These phases are distinct execution mechanisms and the basic idea of an implementation is to separate the above mentioned design phases into distinct entities (e.g. C++ classes) which have well defined transitions – i.e. the plan will not enter the execution phase, until all the objects necessary for successful execution are present. The plan also terminates in a consistent way (e.g. not needed objects are not in the agent's hands). If a transition behavior is required, the FSM above the RP allows for transparent (from the execution's point of view) suspending and resuming of the suspended activity, similarly to process switching in operating systems. The existence of a termination phase allows for additional execution reflecting the plan's success or fail (e.g. on success, the agent is happier, on fail it gets angry). Introducing the cleanup phase into the design, allows us to target the cleanup behavior not only of the currently executing plan, but also of plans below and above it in the hierarchy. The SFP at a given level is responsible for the cleanup of its own objects and of objects of plans below in the hierarchy. The SFPs above a given level can remove object from cleanup sets, based on requirements by other plans in other branches.

The notion of phases represented as entities (C++ classes) allows for extensive modularity - e.g. the IVA could change its plan's inner workings for the initialization phase in respect to its mood or personality (e.g. chaotic vs. methodic searches), but still maintaining the core execution (e.g. gardening task). This can result in more diversity of IVA's behaviors. The phases also allow tracking of the plan – e.g. the agent is only searching for items or the agent is already doing some cleanup. The overall chaotic and blackbox like behavior of a HRP plan is compensated in some degree, limiting it only to the execution phase. There is also the issue of separating processes that can be executed automatically (like object searches and cleanup) providing an "easier to design" concept.

The introduction of phases also allows adding of semantic information which can be evaluated during or before execution of the SFP. We consider the object relevancy as key information in respect to planning further activities and evaluating execution – the SFP might report fail, even before the execution starts, because it is aware of which objects it might need, and those are not available. The HRP concept is capable of such evaluation, but the plans might be too complex to design and maintain.

Object Relevancy

Problems of planning and deliberation require additional information introduced into the SFP to be able to perform some reasoning. We introduce the *Object Relevancy* (OR) into the SFP's structure. The idea behind OR is the observation, that most human and IVA activities are utilizing objects – the physical means to achieve goals. We introduce the *object list* into the SFP's semantic information set (i.e. specified for every plan by the AI designer). We provide a simple dining example to illustrate this.

(Fork[30 sec] & Knife[45 sec] & Spoon[60 sec]) || (Spoon[no limit] & Knife[600 sec]) || (Spoon[10 sec])

The semantic information is divided into sets of objects which at least one has to be satisfied – i.e. the IVA has to acquire or see these objects. The object list specification is also extended with a timeout information (per object), providing a guideline on how long the objects are to be searched for. Secondly, we introduce the *object classes* into the SFP. The object classes can be viewed as a description tag, describing properties and purpose of the objects. This concept was inspired by the academically unpublished game mechanics and structure for objects used in the game Sims©. The so-called *smart object* [*Champandard*, 2007] publish information to their surroundings – what *needs* of the in-game agents they satisfy [*Forbus*, 2002] – providing an *advertising concept*, allowing the agents to search for objects that suit them and their *needs.*

These two concepts provide necessary semantic information allowing us to perform some reasoning about the SFP and their purpose and their mutual relation.

Proof-of-concept prototype

We implemented a simple C++ based prototype, where we implemented all the above mentioned mechanisms [*Plch*, 2009]. The prototype is a simple 2D world home to various agents – a dog, a gardener, and a lubmerjack (Figure 3). We implemented 8 different scenarios to test the architecture in respect to the mentioned problems. For example, we tested the switching behavior with a scenario where the agent was a lumberjack/warrior cutting down trees to build a fort, but when an enemy disturbed him, he took out his shield and engaged him with his axe – the agent switched from woodcutter to warrior with a smooth transition. When utilizing HRP, the agent would put the axe in his bag (termination of "woodcutter") and take it out again (initialization of "warrior"). Our mechanism is smoother – the axe stays in hand. The deep preparation problem was tested by a scenario, where the lumberjack agent uses an axe which could get blunt, and acquires a sharpening tool when getting the axe, because it is required in the subplan ("sharpen axe") of the "woodcutter" plan used as his executed activity. This information is acquired during analyzing subplans during the initialization of the "woodcutter" plan.

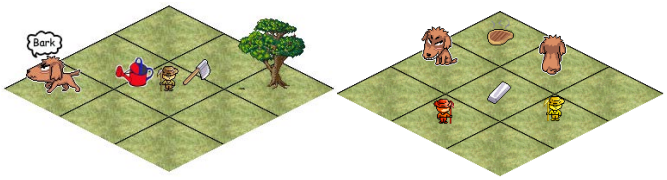


Figure 3. Prototype screenshot – dog and farmer agents are shown. Left image –dog is barking and farmer wants to water plants and cut down a tree on his field. Right image – two dogs are present, left dog is happy, right dog is ashamed. Two farmers are present, the left (red) one is angry the right (yellow) one is happy.

The scenarios ran in real-time, where the average AI engine's use of computational resources was negligible. The engine of the simulation took the majority of the computational resources, which was still tolerable (less than 10% CPU time and less then 100MB RAM). We used an Intel Pentium Dual Core system, with less than 2GB RAM operating a Linux operating system.

Future Work

The next step in our research is to create much more complex 3D testing environment utilizing a computer game (e.g. Half-Life 2 [*Valve*, 2004]) or a military simulation (e.g. Virtual Battlespace 2 [*Bohemia Interactive*, 2007]). We also want to further expand the SF-HRP concept into hybrid architecture – combining planning (e.g. Hierarchical Task Networks) and reactive paradigms.

Conclusion

We combined the FSM with the HRP concept introducing phases thus creating the SF HRP architecture capable of overcoming the presented issues of reactive planning. Our architecture still maintains HRP's core mechanisms (i.e. BE-Tree structure, the action selection algorithm, if-then rules). Therefore, we manage to

retain the important trait of timely execution and introduce more control over the plan's execution by extending the RP into SFP. The SF-HRP's modularity provides a more adaptable design, which leads to more complex and believable behavior.

Our architecture was designed to overcome the observed issues of the HRP concept. The statefullness of SFPs provides basic capability to overcome transitional behavior issues by providing explicit phases the SFP manages when transitioning from one behavior to another. The ability to have specific transitions (besides generic transitions) based on the mutual relationship between SFPs provides a more adaptable architecture.

The additional information on per SFP Object Relevancy provides information use full for planning, deliberation, cleanup behavior as well as information of online/offline analysis (e.g. future need for objects). This information can be used for more complex reasoning about object acquisition and more sophisticated choices for plan alternatives to satisfy a goal.

As a part of our testing, we implemented a proof-of-concept implementation of a 2D environment inhabited by various agents. We tested 8 scenarios simulating mentioned problems, where HRP was not able to manage producing believable behavior in timely fashion.

Acknowledgments This work was partially supported by a student grant GA UK No. 0449/2010/A-INF/MFF, and by project P103/10/1287 (GA ČR).

References

Brom, C.: Hierarchical Reactive Planning: Where is its limit? In: Proceedings of MNAS workshop. Edinburgh, Scotland, (2005)

Bída, M., Brom, C., Popelová, M.: To Date or Not to Date? A Minimalist Affect-Modulated Control Architecture for Dating Virtual Characters. In: Vilhjálmsson, H.H.,Kopp, S., Marsella, S., Thórisson, K.R. (eds.) IVA 2011. LNCS, vol. 6895, pp. 419–425. Springer, Heidelberg (2011)

Bryson, J.: Intelligence by Design: Principles of Modularity and Coordination for Engineering Complex Adaptive Agents. PhD thesis, Massachusetts Institute of Technology, p59-75, (2001)

Champandard, A.J.: AI Game Development: Synthetic Creatures with Learning and Reactive Behaviours (Chapter 8), New Rider, (2003)

Chamapandard, A.J.: Living with the Sims' AI: 21 Tricks to Adopt for your game (2007), URL: http://aigamedev.com/reviews/the-sims-ai [5.5.2011]

Forbus, D.F.: Under the Hood of the Sims, CS 395 Game Design (2002) URL: http://www.cs.northwestern.edu/~forbus/c95-gd/lectures/The_Sims_Under_the_Hood_files/v3_document.htm [5.5.2011]

Ghallab M., Nau D., S., Traverso P.: Automated Planning: Theory and Practice, Elsevier, (2004)

Isla, D.: Handling Complexity in the Halo2, In: Games Developer Conference (2005)

Loyall, A., B.: Believable Agents, Ph.D. thesis, Tech report CMU-CS-97-123, Carnegie. Mellon University, (1997)

Mikula, T.: Hierarchical reactive planning with transitions, Bachelor Thesis Matematicko-fyzikální fakulta, Univerzita Karlova, Praha (2006)

Plch, T.: Action selection for an animat, Diploma thesis, Faculty of Mathematics and Physics, Charles University, Prague, (2009)

Plch, T., Brom, C.: Enhancements for reactive planning - tricks and hacks, In: Proceedings of SOFSEM (2010)

Riedl, M.O. and Stern, A.: Believable Agents and Intelligent Scenario Direction for Social and Cultural Leadership Training. In: Behavior Representation in Modeling and Simulation Conference (2006)

Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach, Prentice Hall, (2003)

Wooldridge, N.: AN Introduction into MultiAgent Systems, John & Wiley & Sons (2002)

Valve Software: Half Life 2, Valve Software (2004)

Bohemia Interactive Australia: Virtual Battlespace 2, Bohemia Interactive Studio (2007)

WDS'11 Proceedings of Contributed Papers, Part I, 125–130, 2011. ISBN 978-80-7378-184-2 © MATFYZPRESS

Learning Automata and Grammars

P. Černo
Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. The problem of learning or inferring automata and grammars has been studied for decades and has connections to many disciplines, including bio-informatics, computational linguistics and pattern recognition. In this paper we present a short survey of basic models and techniques related to the grammatical inference and try to outline some new promising approaches which we expect to bring new light into this subject. For illustration, we introduce delimited string-rewriting systems as a sample model for grammatical inference and sketch the simplified version of the learning algorithm LARS for these systems.

1. Introduction

The central notion in the *formal language theory* is a (formal) *language*, which is a finite or infinite set of words. A *word* is a finite sequence consisting of zero or more letters, whereby the same letter may occur several times. The sequence of zero letters is called the *empty word*, written λ. In defining words (and languages) we usually restrict ourselves to some specific finite nonempty set of letters, called the *alphabet*. The set of all words (all nonempty words, respectively) over an alphabet Σ is denoted by Σ^* (Σ^+, respectively). If x and y are words over Σ, then so is their *concatenation* xy (or $x \cdot y$), obtained by juxtaposition, that is, writing x and y one after another. Concatenation is an associative operation and the empty word λ acts as an identity element: $w\lambda = \lambda w = w$ holds for all words w.

In formal language theory in general, there are two major types of mechanisms for defining languages: *acceptors* and *generators*. Acceptors are usually defined in terms of *automata*, which work as follows: they are given an input word and after some processing they either accept or reject this input word. For instance, the so-called *finite automata* consist of a finite set of internal states and a set of rules that govern the change of the current state when reading a given input symbol. The finite automaton reads a given input word from left to right starting in a specific *starting state*. After reading the input word it accepts only if it ends in the so-called *accepting state*, otherwise it rejects. Finite automata recognize the family of *regular languages*, which plays a central role in the whole formal language theory. Regular languages and finite automata have had a wide range of applications. Their most celebrated application has been lexical analysis in programming language compilation and user-interface translations. Other notable applications include circuit design, text editing, and pattern matching.

Generators, on the other hand, usually generate the language using some finite set of rules. Typically they are defined in terms of grammars. One of the most famous is the classical *Chomsky hierarchy* of grammars (and corresponding languages), which consists of *phrase-structure*, *context-sensitive*, *context-free*, and *regular* grammars (they are also called type 0, type 1, type 2, and type 3 grammars, respectively). Type 0 grammars and languages are equivalent to *computability*: what is in principle computable. Thus, their importance is beyond any question. The same or almost the same can be said about regular grammars and languages. They correspond to strictly finitary computing devices (finite automata). The remaining two classes lie in-between. The class of context-sensitive languages has turned out to be of smaller importance than the other classes. The particular type of context-sensitivity combined with linear workspace is perhaps not the essential type, it has been replaced by various complexity hierarchies. The Chomsky hierarchy still constitutes a testing ground often used: new classes are compared with those in the Chomsky hierarchy. However, it is not any more the only testing ground in

the language theory.

In this short survey we are interested mainly in the problem of learning automata and grammars under a suitable learning regime. This topic is associated with many different fields, and also under a number of names, such as *grammar learning*, *automata inference*, *grammar identification*, but principally *grammar induction* and *grammatical inference*. By grammar induction we mean finding a grammar (or automaton) that can explain the data, whereas grammatical inference relies on the fact that there is a (true) target grammar (or automaton), and that the quality of the learning process has to be measured relatively to this target.

2. Importance of learnability

Alexander Clark in Clark [2010] emphasized the importance of learnability of the representation classes of formal languages. He proposed that one way to build learnable representations is by making them objective or empiricist: the structure of the representation should be based on the structure of the language. He illustrated this approach with three classes corresponding to the lowest three levels of the Chomsky hierarchy. All these classes were efficiently learnable under suitable learning paradigms.

In defining these representation classes the author followed a simple slogan: "Put learnability first!" It means that we should design representations from the ground to be learnable. Rather than defining a representation, and then defining a function from the representation to the language, we should start by defining the map from the language to the representation. The basic elements of such formalism, whether they are states in an automaton, or non-terminals in a phrase-structure grammar, must have a clear definition in terms of sets of strings.

In the conclusive remarks the author suggested that the representations, which are both efficiently learnable and capable of representing mildly context-sensitive languages seem to be good candidates for models of human linguistic competence.

The fact that the human being should be able to discover the syntactic representations of language was the key motivation for introduction of many formalisms to define formal languages. However, as Alexander Clark pointed out, if we first define a representation and then a function from this representation to the language, we are likely to encounter difficult obstacle to going in the reverse direction. Imagine a context-free language L, and a grammar G such that $L(G) = L$. If N is a non-terminal in G, what constraints are there on the language $L(N)$? We can say literally nothing about this set, except that it is a context-free language.

In our opinion, the problems with efficient learnability arise whenever the representation makes use of any of the auxiliary elements such as states, non-terminals, auxiliary symbols, stack, working tapes etc. The complexity lies not only in the fact that the hidden structure between these elements in our target model can be very intricate and complex, but sometimes even if we know exactly the concrete structure of our model, there may still be some very hard (or even algorithmically undecidable) questions concerning the model. Although there is a number of methods and algorithms to learn regular languages, moving up the Chomsky hierarchy is proving to be a challenging task. Indeed, there are several theoretical barriers which make the class of context-free languages hard to learn. Alexander Clark has tried to avoid these obstacles by defining the basic elements of his formalisms in terms of sets of strings. Another promising way to tackle these barriers is to change the way we represent the languages. One approach is to consider models which do not use auxiliary elements at all. Typical representatives of this category are contextual grammars in Marcus [1969], pure grammars in Maurer et al. [1980], and of course locally testable languages in Salomaa [1987]. More recent associative language description model in Cherubini et al. [2002] was the main inspiration for introducing our own model of so-called clearing restarting automata in Černo and Mráz [2009, 2010]. Our model is also quite similar to the so-called delimited string-rewriting systems introduced in Eyraud et al. [2007]. Kutrib et al. [2009, 2010] introduced the so-called stateless restarting automata, which are in fact restarting automata (Jančar et al. [1995]; Otto [2006]) restricted to use only

one single state. However, they can still use auxiliary symbols. For brevity, we omit definitions of these models and refer the interested reader to the references.

In a typical grammatical inference scenario we are concerned with learning language representations based on some source of information, which can be text, examples and counter-examples, or anything that should provide us insight about the elements of the language being sought. In the following Section 3 we provide a discussion concerning the general setting for the problem of identifying languages. In Section 4 we introduce delimited string-rewriting systems as a sample model for grammatical inference. In Section 5 we sketch the simplified version of the algorithm LARS for delimited string-rewriting systems. We believe that the study of delimited string-rewriting systems will help us better understand our own model of clearing restarting automata, and perhaps we will find a similar algorithm to LARS adapted to our own model.

The main source for this whole article is de la Higuera [2010], which provides a nice and comprehensive survey of the techniques and results concerning the grammatical inference, and Rozenberg and Salomaa [1997], which rigorously covers the whole formal language theory.

3. General setting

There are some problems whose tractability is of great importance in grammatical inference. Let $\mathcal{L}$ be a language class, $\mathcal{G}$ be a class of representation of objects for $\mathcal{L}$ and $L : \mathcal{G} \to \mathcal{L}$ be the *naming function*, i.e. $L(G)$ is the language denoted, accepted, recognized or represented by $G \in \mathcal{G}$. In the following we fix the alphabet Σ.

The first problem concerns the fact whether the following *membership problem* is decidable: given $w \in \Sigma^*$ and $G \in \mathcal{G}$, is $w \in L(G)$? It is well known that, for instance, for context-free grammars the membership problem is decidable in polynomial time.

The second problem is the *equivalence problem*: given $G, G' \in \mathcal{G}$, do we have $L(G) = L(G')$? For context-free grammars, the equivalence problem is undecidable. It is decidable for finite automata, but the complexity depends on whether the automata are deterministic or not.

The aforementioned problems are well mathematically defined and formalized. However, considering machine learning as a field where good ideas can be tested and lead to algorithms, we encounter more difficult problems with no clear or established notion. How do we know that the method or algorithm in use is able to infer a reasonable model for our target language? The trick we will use is to consider that the problem we are really interested in is not about discovering the model that would explain the data, but about identifying the *hidden target model*. An alternative formalism for convergence may be that there is no target: the idea is just to induce a grammar from the data in such a way as to minimize some statistical criterion. But again, whether explicitly or implicitly, there is somewhere, hidden, an ideal solution that we can call a target. This discussion leads us to an important notion of the so-called *identification in the limit*. We will not delve much into the technical details of this notion, but only sketch informally the basic idea.

A *presentation* ϕ is an enumeration of elements, which represents a source of information about some specific language $L \in \mathcal{L}$. An example of presentation can be, for instance, the enumeration of all positive and negative samples of L (in some order). A *learning algorithm* $\mathbf{A}$ is a program that takes the first n elements of a presentation (denoted as ϕ_n) and returns some object $G \in \mathcal{G}$. We say that $\mathcal{G}$ is identifiable in the limit if there exists a learning algorithm $\mathbf{A}$ such that for any target object $G \in \mathcal{G}$ and any presentation ϕ of $L(G)$ there exists a rank n such that for all $m \geq n$, $\mathbf{A}(\phi_m)$ does not change and $L(\mathbf{A}(\phi_m)) = L(G)$. Notice that the above definition does not force us to learn the target object G, but only to learn an object equivalent to the target. However, there are some complexity issues with the identification in limit, since it neither tells us how we know when we have found what we are looking for nor how long it is going to take.

4. Rewritting systems

Languages can be defined by using string-rewriting systems, which are usually specified by some rewriting mechanism and a base of simple (accepted) words. In order to study a class containing all the regular languages, we use the so-called delimited string-rewriting systems.

Let us introduce two new symbols ¢ and \$, called the *sentinels*, that do not belong to the alphabet Σ. We will be concerned with languages that are subsets of $¢ \cdot \Sigma^* \cdot \$$. As for the rewrite rules, they will be made of pairs of *terms* partially marked; a term is a string from $T(\Sigma) = \{\lambda, ¢\} \cdot \Sigma^* \cdot \{\lambda, \$\}$.

Terms in $T(\Sigma)$ can be of the following *types*: type 1: $w \in \Sigma^*$, type 2: $w \in ¢ \cdot \Sigma^*$, type 3: $w \in \Sigma^* \cdot \$$, and type 4: $w \in ¢ \cdot \Sigma^* \cdot \$$. For $w \in T(\Sigma)$ the *root* of w is w without the sentinels ¢ and \$, e.g. $root(¢aab) = aab$. We define a specific order relation over $T(\Sigma)$: $u < v \Leftrightarrow root(u) <_{lex-length} root(v) \vee (root(u) = root(v) \wedge type(u) < type(v))$, where $w_1 <_{lex-length} w_2 \Leftrightarrow |w_1| < |w_2| \vee (|w_1| = |w_2| \wedge w_1 <_{lex} w_2)$. For instance, $ab < ¢ab < ab\$ < ¢ab\$ < ba$.

A *rewrite rule* ρ is an ordered pair of terms $\rho = (l, r)$, generally written as $\rho = l \vdash r$. The term l is called the *left-hand side* of ρ and r is *right-hand side* of ρ. We say that $\rho = l \vdash r$ is a *delimited rewrite rule* if l and r are of the same type. By a *delimited string-rewriting system* (DSRS), we mean any finite set $\mathcal{R}$ of delimited rewrite rules. The order relation extends to rules: $(l_1, r_1) < (l_2, r_2)$ if $l_1 < l_2$ or $(l_1 = l_2) \wedge (r_1 < r_2)$.

A system is *deterministic* if no two rules share a common left-hand side. Given a system $\mathcal{R}$ and string w, there may be several rules applicable upon w. Nevertheless, only one rule is eligible. This is the rule having the smallest left-hand side. The same rule might be eligible in different places, but we systematically privilege the leftmost position.

Given a DSRS $\mathcal{R}$ and two strings $w_1, w_2 \in T(\Sigma)$, we say that w_1 *rewrites in one step into* w_2, written $w_1 \vdash_{\mathcal{R}} w_2$ or simply $w_1 \vdash w_2$, if there exists an eligible rule $(l \vdash r) \in \mathcal{R}$ for w_1, and there are two strings $u, v \in T(\Sigma)$ such that $w_1 = ulv$ and $w_2 = urv$, and furthermore u is shortest for this rule. A string w is *reducible* if there exists w' such that $w \vdash w'$, and *irreducible* otherwise. Let $\vdash^*_{\mathcal{R}}$ (or simply $\vdash^*$) denote the reflexive and transitive closure of $\vdash_{\mathcal{R}}$. We say that w_1 *reduces to* w_2 or that w_2 is *derivable from* w_1 if $w_1 \vdash^*_{\mathcal{R}} w_2$.

Given a DSRS $\mathcal{R}$ and an irreducible string $e \in \Sigma^*$ we define the language $L(\mathcal{R}, e)$ as the set of strings that reduce to e using the rules of $\mathcal{R}$: $L(\mathcal{R}, e) = \{w \in \Sigma^* \mid ¢w\$ \vdash^*_{\mathcal{R}} ¢e\$\}$. Deciding whether a string w belongs to a language $L(\mathcal{R}, e)$ consists of trying to obtain e from w by a rewriting derivation. We will denote by $\mathsf{Apply}_{\mathcal{R}}(w)$ the string obtained by applying the different rules in $\mathcal{R}$ until no more rules can be applied. We extend the notation to a set of strings: $\mathsf{Apply}_{\mathcal{R}}(S) = \{\mathsf{Apply}_{\mathcal{R}}(w) \mid w \in S\}$.

Example 4.1 *Let* $\Sigma = \{a, b\}$.

1. $L(\{ab \vdash \lambda\}, \lambda)$ *is the* Dyck language. *The single rule erases substring ab, as is illustrated in the following example of derivation:*

$$¢aabb\underline{ab}\$ \vdash ¢a\underline{abb}\$ \vdash ¢\underline{ab}\$ \vdash ¢\lambda\$.$$

2. $L(\{ab \vdash \lambda; ba \vdash \lambda\}, \lambda)$ *is the language* $\{w \in \Sigma^* \mid |w|_a = |w|_b\}$, *because every rewriting step erases one a and one b.*

3. $L(\{aabb \vdash ab; ¢ab\$ \vdash ¢\$\}, \lambda)$ *is the language* $\{a^n b^n \mid n \geq 0\}$. *For instance,*

$$¢aa\underline{aabb}bb\$ \vdash ¢a\underline{aabb}b\$ \vdash ¢\underline{aabb}\$ \vdash ¢\underline{ab}\$ \vdash ¢\lambda\$.$$

4. $L(\{¢ab \vdash ¢\}, \lambda)$ *is the regular language* $(ab)^*$. *It can be shown that given any regular language* L *there is a system* $\mathcal{R}$ *such that* $L(\mathcal{R}, \lambda) = L$.

5. Algorithm LARS

Learning Algorithm for Rewriting Systems (LARS) introduced in Eyraud et al. [2007] generates the possible rules among those that can be applied over the positive samples S_+, tries using them and keeps them if they do not create inconsistency (using the negative samples S_- for that). Algorithm LARS calls the function NewRule, which generates the next possible rule to be checked.

For this, one should choose *useful* rules, i.e. those that can be applied on at least one string from S_+. One might also consider useful a rule that allows us to diminish the size of the set S_+: a rule which, when added, has the property that two different strings rewrite into an identical string. The goal of usefulness is to avoid an exponential explosion in the number of rules to be checked. The function Consistent checks that by adding the new rule to the system, one does not rewrite a positive example and a negative example into a same string.

Algorithm 5.1 *LARS*

Input: S_+, S_-.

Output: $\mathcal{R}$.

Description:

(1) $\mathcal{R} \leftarrow \emptyset$; $\rho \leftarrow (\lambda \vdash \lambda)$;
(2) **while** $|S_+| > 1$ **do**
(2.1) $\rho \leftarrow \mathsf{NewRule}(S_+, \rho)$;
(2.2) **if** $\mathsf{Consistent}(S_+, S_-, \mathcal{R} \cup \{\rho\})$ **then**
(2.2.1) $\mathcal{R} \leftarrow \mathcal{R} \cup \{\rho\}$;
(2.2.2) $S_+ \leftarrow \mathsf{Apply}_{\mathcal{R}}(S_+)$;
(2.2.3) $S_- \leftarrow \mathsf{Apply}_{\mathcal{R}}(S_-)$;
(2.3) **end**
(3) **end**
(4) **return** $\mathcal{R}$

The goal is to be able to learn any DSRS with LARS. The simplified version proposed here can be used as basis for that, and does identify in the limit any DSRS. But, a formal study of the qualities of the algorithm is beyond scope of this article. We refer the interested reader to the article Eyraud et al. [2007].

6. Conclusion

Our own model of clearing restarting automata is quite similar to the delimited string rewriting systems with the exception that we only allow rules of the form $\rho = (xzy \vdash xy)$. In the extended model, called the Δ-clearing restarting automata, we allow also rules $\rho = (xzy \vdash x\Delta y)$, where Δ is a special auxiliary symbol. However, algorithm LARS cannot be directly applied to clearing restarting automata, since we do not specify any order relation on terms or rules. This makes our model implicitly nondeterministic. It would be therefore interesting to investigate the deterministic version of clearing restarting automata obtained by using the same order relation as used in delimited string rewriting systems.

Acknowledgments. The author thanks RNDr. František Mráz, CSc. for his advices and guidance. The present work was partially supported by the Grant Agency of Charles University under Grant-No. 272111/MFF/A-INF and by the Czech Science Foundation under Grant-No. P103/10/0783 and Grant-No. P202/10/1333.

References

Černo, P. and Mráz, F., Clearing restarting automata, in *Workshop on Non-Classical Models for Automata and Applications (NCMA)*, edited by H. Bordinh, R. Freund, M. Holzer, M. Kutrib, and F. Otto, vol. 256 of *books@ocg.at*, pp. 77–90, Österreichisches Computer Gesellschaft, 2009.

Černo, P. and Mráz, F., Clearing restarting automata, *Fundamenta Informaticae*, *104*, 17–54, 2010.

Cherubini, A., Reghizzi, S. C., and San Pietro, P., Associative language descriptions, *Theor. Comput. Sci.*, *270*, 463–491, 2002.

Clark, A., Three learnable models for the description of language, in *Language and Automata Theory and Applications*, edited by A.-H. Dediu, H. Fernau, and C. Martn-Vide, vol. 6031 of *Lecture Notes in Computer Science*, pp. 16–31, Springer Berlin / Heidelberg, 2010.

de la Higuera, C., *Grammatical Inference: Learning Automata and Grammars*, Cambridge University Press, New York, NY, USA, 2010.

Eyraud, R., de la Higuera, C., and Janodet, J.-C., Lars: A learning algorithm for rewriting systems, *Machine Learning*, *66*, 7–31, 2007.

Jančar, P., Mráz, F., Plátek, M., and Vogel, J., Restarting automata, in *FCT'95*, edited by H. Reichel, vol. 965 of *LNCS*, pp. 283–292, Springer, Dresden, Germany, 1995.

Kutrib, M., Messerschmidt, H., and Otto, F., On stateless deterministic restarting automata, in *SOFSEM*, edited by M. Nielsen, A. Kučera, P. B. Miltersen, C. Palamidessi, P. Tůma, and F. D. Valencia, vol. 5404 of *LNCS*, pp. 353–364, Springer, 2009.

Kutrib, M., Messerschmidt, H., and Otto, F., On stateless deterministic restarting automata, *Acta Inf.*, *47*, 391–412, 2010.

Marcus, S., Contextual grammars, in *Proceedings of the 1969 conference on Computational linguistics*, COLING '69, pp. 1–18, Association for Computational Linguistics, Stroudsburg, PA, USA, 1969.

Maurer, H., Salomaa, A., and Wood, D., Pure grammars, *Information and Control*, *44*, 47 – 72, 1980.

Otto, F., Restarting automata, in *Recent Advances in Formal Languages and Applications*, edited by Z. Ésik, C. Martín-Vide, and V. Mitrana, vol. 25 of *Studies in Computational Intelligence*, pp. 269–303, Springer, Berlin, 2006.

Rozenberg, G. and Salomaa, A., eds., *Handbook of Formal Languages (3 volumes)*, Springer, 1997.

Salomaa, A., *Formal languages*, Academic Press Professional, Inc., San Diego, CA, USA, 1987.

WDS'11 Proceedings of Contributed Papers, Part I, 131–136, 2011. ISBN 978-80-7378-184-2 © MATFYZPRESS

Rules in Database Systems

J. Kozák

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. This paper deals with the principles of rule-based systems in databases and gives an overview of existing approaches. First, we introduce the concept of rules. The application of rule-based systems to databases is categorized. Deductive database systems are introduced as the first extension of classical database management systems. We briefly describe the concept of deductive databases and their language. The main part is dedicated to active database systems. Language and implementation of rules in active databases are discussed as well as rule processing and key issues connected with it. Also architecture of the active database systems is depicted. At the end, benchmark comparing different rule-based database technologies is presented.

Introduction

The very first thing coming into our minds might be a natural question: "What are the rules?" Generally, we can say that almost any commands, decision making or deductive reasoning can be understood as a set of rules. This brief description implies that we meet rules in real life all the time although usually we do not even realize it. The previous interpretation becomes obvious when we start to think about implementation of intelligence into a machine i.e., making machines think like people. If we want to do that, then we have to tell the machine what to do in various situations which means to teach it in the way of stating the rules.

These considerations lead to an idea of implementing such a rule system. In this article, we will continue in introduction of rule-based systems and rules in dabases specifically. Then we will describe the concepts of deductive databases and active databases. At the end, a benchmark comparing different rule-based technologies in databases will be presented.

Rule-based systems

The *rule-based systems* (RBSs) evolved from the work of researchers from artificial intelligence (AI). Already in 1985, [*Hayes-Roth*, 1985] stated that RBSs constitute the best currently available means for codifying the problem-solving know-how of human experts, because these experts often tend to express their problem-solving techniques in terms of a set of situation-action rules (i.e., in specific situation execute an action). According to [*Hayes-Roth*, 1985], RBSs can be used in very different areas such as quality assurance, process control, troubleshooting and so on. Situation-action behavior represents only one part of rules.

Referring to the informal description of rules at the beginning, we could separate another different type of rules which would represent deduction and inferring of something new from the known facts—deductive systems. The reason of separating these parts from each other will be seen in the following sections considering the extension of database technology.

Rules and databases

Databases were originally designed as repositories which should be able to store data for various purposes. Traditional database management systems (DBMSs) are passive in the sense that commands are executed by the database (e.g., query, update, delete) as and when requested by the user or application program [*Paton and Díaz*, 1999]. However, some situations do not have to match this pattern and need to be modeled in a different way. It should allow the DBMS to react automatically in such situations without the need of user access. This can be

solved by implementing a system called *active database system* which provides the ability of reactive behavior.

Large databases also offer extensive possibilities to deduce some new information from the existing facts. More or less, it can be done by stating a proper query in common query language (e.g., SQL). Unfortunately, not everything can be expressed using e.g., SQL. Therefore, new approaches were implemented under the name of *deductive database systems*.

The active and deductive database systems represent the extensions of database systems using rules. Next sections will deal with them in more detail and greater amount of space will be devoted to active databases.

Deductive database systems

Deductive database systems are DBMSs whose query language and (usually) storage structure are designed around a logical model of data. As relations are naturally thought of as the "value" of logical predicate, and relational languages such as SQL are syntactic sugarings of a limited form of logical expression, it is easy to see deductive database systems as an advanced form of relational systems [*Ramakrishnan and Ullman*, 1996]. The main goal of deductive systems is to provide the ability to express queries that form a superset of relational algebra. The main difference is represented in the possibility of formulation recursive queries.

Deductive database systems represent extensions of relational DBMSs. Their evolution was influenced by logical programming. Relational databases and logic programming have been found quite similar in their representation of data at the language level. They have also been found complementary in many aspects [*Liu*, 1999].

Deductive systems divide their data into two categories:

- *Data or facts* that are usually stored in relational DBMS and are represented by a predicate with constant arguments. For example, *parent(Frank,Anne)* represents that Frank is a parent of Anne. This category forms so called *extensional database*.

- *Rules* that are usually written in Prolog-style notation

$$p \text{ :- } q_1, q_2, \ldots, q_n$$

  This rule means: "If q_1 and q_2 and ... and q_n then p". Rules with terms being either constants or variables are often referred to as Datalog rules. Whole set of rules forms so called *intensional database*.

We can see that deductive languages are declarative and therefore allow the user to say what he or she wants but not how to do it. This is one of the greatest benefits of declarative languages. Standardization of syntax of the deductive languages was done in [RIF_1, 2010] and represents the recommendation of W3C.

The expressive power of deductive languages is generally greater than expressive power of relational algebra. But for nonrecursive range-restricted Datalog with negation can be proved that it is equivalent to relational algebra and domain-independent relational calculus (for details see e.g., [*Bry et al.*, 2007]).

Evaluation and optimization of rules in deductive databases usually use fixpoint techniques. These are extensively described in [*Bry et al.*, 2007] and go beyond the framework of this paper.

Many deductive languages and systems have been developed. Their historical overview and development, concrete implementations and their brief description of capabilities can be found in [*Ramakrishnan and Ullman*, 1996]. Some issues araising in deductive languages such as extensions of Datalog to support complex values and others are discussed in [*Liu*, 1999].

Management of larger sets of deductive rules becomes quite problematic but still it offers an interesting alternative to traditional ways of data analysis.

Active database systems

Active database systems are based on a different approach to rules in DBMSs than deductive database systems. They represent the active behavior originally implemented in AI and expert systems.

The rules in active databases are commonly made up from up to three parts: an event, a condition and an action. It perfectly fits the reactive behavior. When some event happens, the condition is evaluated and if it is true, the action is carried out. Such rules are known as *event-condition-action* or *ECA rules*. Thus, the semantics of active rules are procedural [*Ceri and Ramakrishnan*, 1996].

The active rules do not have to contain all three parts. The event or condition part can be omitted. Then we speak about *condition-action rules* (often referred as *production rules*) or *event-action rules*. Every type of active rule has its specific kind of use and production rules are quite similar to deductive rules.

In the article [*Paton and Díaz*, 1999] three categories of active database applications are distinguished:

- Database system extension—support for other parts of database such as integrity constraints and materialized views etc.
- Closed database application—rules directly support the semantics of the application e.g., repair actions in a modeling database
- Open database application—used in conjunction with monitoring devices, see e.g., [*Zoumboulakis et al.*, 2004].

All these types of applications have a general architecture of an active database system which comprises of two basic components—rule repository and rule engine. The rule engine can be further divided into more modules such as event detector, condition monitor, scheduler and query evaluator [*Paton and Díaz*, 1999].

Rule language and repository

A new language for rule formulation must be defined in order to allow users to operate the active database system. The idea is to keep it as close to the natural language as possible. This will make the system easy to use. On the other hand, there was no standard so each RBS developed its own language.

In the year 2005 Rule Interchange Format (RIF) Working Group of W3C was established. This group presented a document [RIF_2, 2010] which represents a recommendation of W3C for abstract syntax of production rules. They use "mathematical English" (a special form of English for communicating mathematical definitions, examples, etc.) as a presentation syntax and XML syntax as its concrete syntax. [RIF_2, 2010] describes the syntax which shares features with concrete production rule languages. Without any other comments, here is one example of production rule and its representation according to [RIF_2, 2010], which could be used for customer segmentation: A customer becomes a "Gold" customer when his cumulative purchases during the current year reach $5000.

```
Prefix(ex <http://example.com/2008/prd1#>)
(* ex:rule_1 *)
Forall ?customer ?purchasesYTD (
 If   And( ?customer#ex:Customer
           ?customer[ex:purchasesYTD->?purchasesYTD]
           External(pred:numeric-greater-than(?purchasesYTD 5000)) )
 Then Do( Modify(?customer[ex:status->"Gold"]) ) )
```

The language of rules is not the only issue when speaking about the rule repository. The rules in the repository can change quite frequently e.g., in order to meet the business needs and their management can become problematic. It follows that providing a powerful rule management environment should be important. Rule management is discussed e.g., in [*Viana et al.*, 2006] where a short overview about existing solutions is provided and graphical representation of rule anatomy (i.e., rule meta-model) is proposed.

Rule engine

Rule engine cooperates with the rule repository and the underlying database. This part of the whole system operates the main algorithms. The rule execution model can be divided into the sequence of tasks according to [*Paton and Díaz*, 1999]: signaling, triggering, evaluation, scheduling and execution. The schematic diagram can be seen in Figure 1.

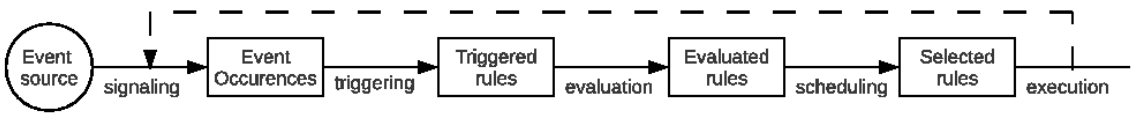


Figure 1. Rule execution diagram.

This schema implies possible parts of architecture of an active database system. First, an event must be detected. Then appropriate rules are selected and their conditions are evaluated. These processes select only applicable rules and a conflict set of rules is created. Conflicts are resolved and actions are executed.

Although the execution might look quite straightforward, there are many complicated issues hidden. At the very beginning the event monitor detecting complicated composite events can cause first problems (such as deciding for the right structure of composite event implementation) in an implementation of an active database system. But if we skip the event monitor part and concentrate rather on the production rules processing we get to the problem of condition evaluation at first.

Special algorithms for condition evaluation are often called *pattern matching algorithms*. They match facts against the rules. There is a number of pattern matching algorithms like TREAT, Rete etc [*Wang and Hanson*, 1992]. The Drools, production rule system developed by JBoss Community, uses an extended version of the Rete algorithm [*JBoss Community*, 2010].

Previous matching algorithm produces a conflict set of rules without specified order of execution. It means there has to be some kind of order of firing specified afterwards. This problem is often called a *conflict set resolution*. Some specific logic has to implemented to solve this problem dynamically or priority can be assigned to each rule in its definition. Dynamically prioritized conflict resolution is discussed in [*Dimitoglou and Rotenstreich*, 2007].

Benchmark

We presented two main types of rule systems which can be implemented in classical DBMS. But what kind of them performs better in real world situations? A unique benchmark was presented in [*Liang et al.*, 2009]. The authors compared several solutions from the area of rule based systems in databases. For our purposes the results of deductive and active databases are interesting. The authors chose DLV[1] and Ontobreaker[2] from the deductive databases and Drools[3] and Jess[4] from production and reactive rule systems.

[1] http://www.dbai.tuwien.ac.at/proj/dlv/

[2] http://www.ontoprise.de/en/products/ontobroker/

[3] http://www.jboss.org/drools

[4] http://www.jessrules.com

The benchmark was mainly designed to test the abilities which cannot be accomplished in tradional database languages such as relational algebra. The authors tested rule engines in large joins, datalog recursion and default negation. For illustration, we provide information about two tests performed in this benchmark.

One of the tests has a form of a non-recursive tree of binary joins which is expressed using the following inference rules written in Prolog syntax:

```
a(X,Y) :- b1(X,Z), b2(Z,Y)
b1(X,Y) :- c1(X,Z), c2(Z,Y)
b2(X,Y) :- c3(X,Z), c4(Z,Y)
c1(X,Y) :- d1(X,Z), d2(Z,Y)
```

The base relations, `c2`, `c3`, `c4`, `d1` and `d2` were randomly generated. Two data sets were used: one with 50000 facts and the other with 250000 facts.

Datalog recursion was tested e.g., on the same-generation problem i.e., finding all siblings in the same generation:

```
sg(X,Y) :- sib(X,Y)
sg(X,Y) :- par(X,X1), sg(X1,Y1), par(Y,Y1)
```

Tha base relations `par` and `sib` were randomly generated. Data were considered both cyclic and acyclic and for each type two data sizes were used: 6000 and 24000.

We can see results of these two chosen tests in the following tables which show the execution times. In these tables, size means the total size of all base relations; *error* means that the system gave an error during the evaluation; *timeout* indicates that evaluation did not finish within a set time limit of 30 minutes. All times are given in seconds and exclude the loading time.

Table 1. Large join test.

query	a(X,Y)		b1(X,Y)		b2(X,Y)	
size	50000	250000	50000	250000	50000	250000
ontobroker	4.089	28.385	0.213	4.806	0.019	0.168
dlv	85.459	838.781	7.177	60.239	0.820	9.392
drools	error	error	27.414	error	2.111	94.474
jess	310.000	timeout	12.000	317.000	1.000	26.000

Table 2. Same generation test.

size	6000	6000	24000	24000
cyclic data	no	yes	no	yes
ontobroker	1.402	1.926	5.424	5.309
dlv	20.274	31.346	365.136	438.008
drools	104.884	error	error	error
jess	64.000	error	1517.000	error

The results of two presented tests indicate that Ontobroker was the best performing system. Although Ontobroker was not the best system in all other tests, in the conclusion of the paper Ontobreaker (and partly DLV) were assigned to be the best performing systems. Results of other above enumerated systems were significantly worse.

Conclusion

We described the idea of rule-based systems and their implementation in databases. Two main branches, deductive database systems and active database systems, were introduced. Their main features and connected issues were discussed and results of a benchmark were presented. Of course, the description of RBSs was not exhaustive. The rules in databases is quite a vast area to be explored and there are many more topics to be discovered and problems to be solved.

For example, this paper did not deal with the static analysis of rules (termination problem etc.) at all.

Although the presented theory is interesting and forms the necessary background for all RBSs, the further research will rather concentrate on more recent problems of knowledge representation and reasoning such as ontological databases and reasoning on the Web which have become quite prominent now.

References

Bry F., Eisinger N., Eiter T., Furche T., Gottlob G., Ley C., Linse B, Pichler R., Wei F., Foundations of Rule-Based Query Answering, *Reasoning Web*, 1–153, 2007.

Ceri S., Ramakrishnan R., Rules in Database Systems, *ACM Computing Surveys*, 28, 109–111, 1996.

Dimitoglou G., Rotenstreich S., Architecture and Algorithms for Distributed Rule Management and Processing, *International Journal of Computer Science and Network Security*, 7, 397–404, 2007.

Hayes-Roth F., Rule-based Systems, *Communications of the ACM*, 28, 921–932, 1985.

JBoss Community, Drools Expert 5.1 Documentation, Available on *http://www.jboss.org/drools/documentation*, 2010.

Liang S., Fodor P., Wan H., Kifer M., OpenRuleBench: An Analysis of the Performance of Rule Engines, *Proceedings of the 18th international conference on World wide web*, 601–610, 2009.

Liu M., Deductive Database Languages: Problems and Solutions, *ACM Computing Surveys*, 31, 27–62, 1999.

Paton N. W., Díaz O., Active Database Systems, *ACM Computing Surveys*, 31, 63–103, 1999.

Ramakrishnan R., Ullman J. D., A Survey of Deductive Database Systems, *The Journal of Logic Programming*,23 ,125–149, 1995.

RIF_1 Working Group, RIF Basic Logic Dialect (BLD), Available on *http://www.w3.org/TR/rif-bld*, 2010.

RIF_2 Working Group, RIF Production Rule Dialect (PRD), Available on *http://www.w3.org/TR/rif-prd*, 2010.

Viana S., Pavón J., Rady de Almeida Junior J., Rule Management in Active Database Systems, *Proceedings of the 15th International Conference on Computing (CIC'06)*, 315–322, 2006.

Wang Y., Hanson, E. N., A performance comparison of the Rete and TREAT algorithms for testing database rule conditions, *Proceedings of the Eighth International Conference on Data Engineering*, 88–97, 1992.

Zoumboulakis M., Roussos G., Poulovassilis, Active Rules for Sensor Databases, *Proceedings of the First Workshop on Data Management for Sensor Networks (DMSN 2004)*, 98–103, 2004.

 ISBN 978-80-7378-184-2

Dependency Parsing

N. Green
Charles University in Prague, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics, Prague Czech Republic.

Abstract. Dependency parsing has been a prime focus of NLP research of late due to its ability to help parse languages with a free word order. Dependency parsing has been shown to improve NLP systems in certain languages and in many cases is considered the state of the art in the field. The use of dependency parsing has mostly been limited to free word order languages, however the usefulness of dependency structures may yield improvements in many of the word's 6,000+ languages.
I will give an overview of the field of dependency parsing while giving my aims for future research. Many NLP applications rely heavily on the quality of dependency parsing. For this reason, I will examine how different parsers and annotation schemes influence the overall NLP pipeline in regards to machine translation as well as the the baseline parsing accuracy.

Introduction

Dependency parsing has been shown to be an important part of many NLP applications. Contrary to it's counterpart constituency structure, dependency tree structure is considered state-of-the-art and more useful in free word order languages. A common problem parsers have in these languages is the phenomenon of non-projectivity. This is when a subtree of a dependency graph is not contiguous, or visually cannot be drawn without intersecting lines(Kuhlmann and Satta [2009]). Dependency structures are better at showing agreement whereas constituency, or phrase based, trees typically show neighboring node groupings better due to the divide and conquer approach that context free grammars impose on sentences. My research into dependency parsing will continue on a few different paths as described in the sections below.

Data

Much of the current progress in dependency parsing has been a result of the availability of common data sets in a variety of languages, made available through the CoNLL shared task. This data is in 13 languages and 7 language families. Later shared tasks also released data in other genres to allow for domain adaptation (Nivre et al. [2007a]). The availability of standard competition, gold level, data has been an important factor in dependency based research.

Metrics

As an artifact of the CoNLL shared tasks competition, two standard metrics for comparing dependency parsing systems emerged. *L*abeled *a*ttachment *s*core (LAS) and *u*nlabeled *a*ttachment *s*core (UAS). UAS studies the structure of a dependency tree and assesses whether the output has the correct head and dependency arcs. In addition to the structure score in UAS, LAS also measures the accuracy of the dependency labels on each arc. A third, but less common metric, is used to judge the percentage of sentences that are completely correct in regards to their LAS score. This score is better used to judge how a dependency parser will affect other NLP tools that make use of the dependency parser output (Buchholz and Marsi [2006]).

To evaluate machine translation results we will rely on the Bleu (*Bi*L*ingual *E*valuation *U*nderstudy) Metric. Bleu is an automatic scoring mechanism for machine translation that is quick and can be reused as a benchmark across machine translation tasks. BLEU is based on the geometric mean of n-gram precisions comparing a machine translation and a reference text (Papineni et al. [2002]).

Dependency Parsing Techniques

In Kübler et al. [2009] the authors confirm that two parsers, MST parser and Malt parser, give similar accuracy results but with very different errors. MST parser, a maximum spanning tree graph-based algorithm, has evenly distributed errors while MaltParser, a transition based parser, has errors on mainly longer sentences. This result comes from the approaches themselves. MST parser is globally trained so the best mean solution should be found, this is why errors on the longer sentences are about the same as the shorter sentences. Malt Parser on the other hand uses a greedy algorithm with a classifier that chooses a particular transition at each vertex. This leads to the possibility of the propagation of errors further in a sentence (McDonald and Nivre [2007]). Both these algorithms are discussed below along with a third technique, constituent transformation. It is important for all future empirical experiments to look at each kind of parser as the different types of errors may greatly change the resulting structures.

Graph-Based

A dependency tree is a special case of a dependency graph that spawns from an artificial root and is acyclic. Because of this we can look at a large history of work in graph theory to address finding the best spanning tree for each dependency graph. The most common form of this type of dependency parsing is called arc-factored parsing and deals with the parameterization of the edge weights. The main drawback of these methods is that for non-projective trees, the worst case scenario for most methods is a complexity of $O(n^3)$ (Eisner [1996]). However, for non-projective parsing Chu-Liu-Edmond's algorithm has a complexity of $O(n^2)$ (McDonald et al. [2005]). The most common tool for doing this is MST parser, which is also used in the noun-phrase bracketing experiments described later in this paper.

Transition-Based

Transition-based parsing creates a dependency structure that is parameterized over the transitions used to create a dependency tree. This is closely related to the the shift-reduce constituency parsing algorithms. Due to the notion of picking transitions in an abstract machine, the algorithms used for these systems tend to be greedy. The benefit of this is that the algorithms have a linear time complexity. However, due to the greedy algorithms, longer arc parses can cause error propagation across each transition (Kübler et al. [2009]). The standard tool for transition-based algorithms is Malt Parser (Nivre et al. [2007b]) which in the shared tasks was often tied with the best performing systems.

Constituent Transformation

While not a true dependency parser, one technique often applied is to take a state of the art constituent parser and transform its phrase based output into dependency relations. This has been shown to also be state-of-the-art in accuracy for dependency parsing in English. This method has also been applied to the Czech language with Collin's parser. One path of research should test how this process works in other languages and for treebanks specifically annotated for dependency relations. In most cases the models are built from the Penn Treebank, a constituent based treebank (Marcus et al. [1993]), using a phrase based parser. Then to parse a sentence into a dependency structure, the phrase based output is processed with a conversion tool e.g. Penn Converter (Johansson and Nugues [2007]) or Stanford Converter (Marneffe et al. [2006]). Versions of these converters were used in the CoNLL shared task to create dependency treebanks for a variety of the languages. For my experiments I will make use of Charniak and Johnson [2005] as well as other constituent parsers.

Improving Dependency Parsing Accuracy

Current methods are aimed at improving accuracy of state of the art parsers, not at increasing the breadth of languages. Although the shared competitions have always stressed multilingual application, the final results have always been concerned with the increase in LAS and UAS in particular languages. To address this issue there are two main approaches I plan to examine: domain adaptation and annotation structure.

Domain Adaptation

One major approach to improving parsing accuracy is to better model certain domains. Later shared tasks started testing this ability in models. The idea is that a parsing model is trained on one type of text, such as financial news, and must be applied on a different domain, such as blogs. This contains syntactical, style, and lexical changes that are very difficult for many models to adjust for. Three main techniques are being researched to address domain adaptation: self training, up-training, and model selection and combination. Evaluation of domain adaptation can be tricky. Initially I will look to converting constituency treebanks of different genres into dependency treebanks for testing purposes. Although when it comes to testing in domains such as social media, new dependency structures should be created.

Self Training. Self training is the idea of training a parser on its own output. This is particularly useful when a parser was trained on one domain and you are trying to extend it to another domain. While the parser has more errors than would be normally expected, the lower accuracy output improves future outputs on the same domain when used as training data in future iterations.

Up-training. Up-training takes a slightly different approach by training a particular parser on output of different parsers. Current research has applied this to make faster parsers more accurate by up-training with the slower but more accurate parser's output. This approach has been used to change the domain of a parser as well, using the slower model to parse the out of domain data. Similar techniques could be used, but instead of basing the up-training of a higher accuracy model it could be based on a classification of smaller more domain specific models (Petrov et al. [2010]).

Model Selection and Combination. As with most NLP tools a logical path of research is to apply ensemble methods and other techniques as a way to combine the outputs of many different parsers. This, in theory, combines the benefits of each parser. There are three main approaches to this technique. First, a classifier can be used to determine which model/parser should be used for a particular sentence. Second, the outputs of all parsers can be algorithmically combined for one final output (McClosky et al. [2010]). The third way is using one model as a feature, or input, to another model. This was seen to have a positive effect when using Malt Parser as a feature to MST Parser (McDonald and Nivre). Whether through a voting schema or an algorithm similar to maximum spanning tree, one can see the usefulness of using multiple parsers to select the correct edge. We should be able to minimize errors that would be caused by syntactic or style changes caused by a domain switch.

Annotation Structure

Annotation structure and style play an important role in dependency parsing. The specificity and the structure decided upon may change the dependency parsing accuracy along with the accuracy of NLP tools that use the dependency structure output.

Noun-Phrase Bracketing. Initially in the Penn Treebank (Marcus et al. [1993]) noun phrases were treated as flat structures. For this reason most parsing systems have looked at these structures as flat. However, recently, noun phrases were annotated with their correct structure in (Vadas and Curran [2007a] Vadas and Curran [2007b] Vadas and Curran [2008] Vadas and Curran [2007c]). It is a question whether these annotations help in the training of dependency parsers and more importantly whether these structures will aid in other NLP tasks that make use of dependency structures such as machine translation. TectoMT (Žabokrtský et al. [2008]) is used as our machine translation system since it makes direct use of dependency structure in its tree transformation stages. Due to increased specificity we find a slight decrease but statistically insignificant change in parsing accuracy when using noun-phrase bracketing. However, even with a possibly lower parsing accuracy scores, the system's

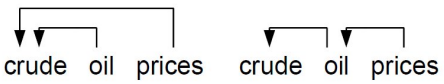


Figure 1. How noun-phrase structure can lead to ambiguities

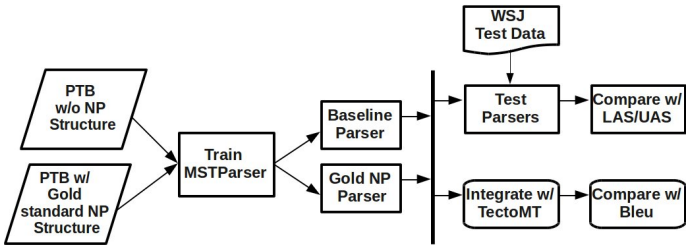


Figure 2. Flow of noun-phrase methodology and experimental decisions

Bleu scores improved with statistical significance over the baseline system, the one which didn't use noun-phrase bracketing. Both parsers were tested against their respective annotation standards.

Systems	Bleu
Baseline Parser	9.47
Gold Parser	**9.70**

Table 1. TectoMT results of a complete system run with both the Baseline Parser and Gold Parser

Both systems are tested on WMT08 data. Results are an average of 1,000 bootstrapped test sets with replacement using a pairwise comparison (Koehn [2004]), and the improvement in Bleu score is statistically significant with 95% confidence (Green [2011]).

Coordination structure. While noun phrase bracketing was a lack of structure, there are many situations where a structure is annotated but the annotators decided on a particular annotation style. For instance with coordination structure. There are standards that are left branching, right branching, coordination first, and coordination as the head of the dependency. No method is any more linguistically sound then any other method.

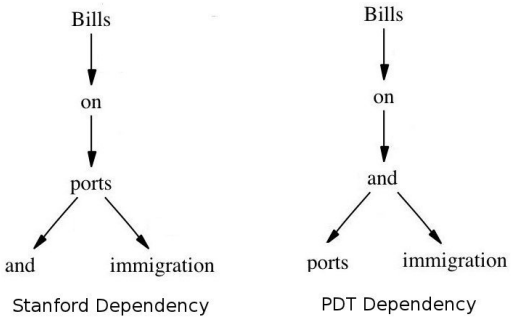


Figure 3. How Stanford dependency structure handles coordination (taken from the Stanford dependency web page)

Testing different annotation schemes for this and other structures that do not have linguistic backing behind their annotation schemes will be a good contribution to the field of dependency parsing and annotation. One style of annotation may work significantly better for a particular class of languages. Each annotation structure should be empirically tested to find the best combination for each language and machine translation pair.

Future Work and Conclusion

Increasing accuracy of the current parsers should not be the only goal going forward. There are 6,000+ languages in the world and very few of them have dependency trees available. This eliminates the possibility of any supervised training without a very heavy cost in creating the training data from scratch. For this reason, unsupervised methods should be researched and developed further. Current

techniques such as Klein and Manning [2004] only get in the 40% - 60% range in undirected and unlabeled accuracy, depending on language and treebank. This is a slight improvement over a baseline that characterizes a dependency as adjacency related. Work in this area is advantageous since the cost of manually creating dependency structures is a bottleneck for most languages in the world and unsupervised methods will give these languages a starting point into the field.

Initial research on the differences between a variety of parsers has been conducted in regards to their effect on machine translation (Popel et al. [2011]). This research should be expanded to include parser combination.

While dependency parsing has made many advancements with the shared task competitions, there is room for improvement to all current models. I hope to primarily focus on annotation standards, domain adaptation, and model combination to improve current performance. Furthermore, I hope to research and improve dependency parsing for under-resourced languages through unsupervised learning.

Acknowledgments. This research has received funding from the European Commission's 7th Framework Program (FP7) under grant agreement n° 238405 (CLARA), and from grant MSM 0021620838.

References

Buchholz, S. and Marsi, E., CoNLL-X shared task on multilingual dependency parsing, in *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X '06, pp. 149–164, Association for Computational Linguistics, Stroudsburg, PA, USA, URL http://portal.acm.org/citation.cfm?id=1596276.1596305, 2006.

Charniak, E. and Johnson, M., Coarse-to-fine n-best parsing and maxent discriminative reranking, in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pp. 173–180, Association for Computational Linguistics, Stroudsburg, PA, USA, URL http://dx.doi.org/10.3115/1219840.1219862, 2005.

Eisner, J., Three new probabilistic models for dependency parsing: An exploration, in *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, pp. 340–345, Copenhagen, URL http://cs.jhu.edu/~jason/papers/#coling96, 1996.

Green, N., Effects of noun phrase bracketing in dependency parsing and machine translation, in *Proceedings of the ACL 2011 Student Session*, pp. 69–74, Association for Computational Linguistics, Portland, OR, USA, URL http://www.aclweb.org/anthology/P11-3013, 2011.

Johansson, R. and Nugues, P., Extended constituent-to-dependency conversion for English, in *Proceedings of NODALIDA 2007*, pp. 105–112, Tartu, Estonia, 2007.

Klein, D. and Manning, C., Corpus-based induction of syntactic structure: Models of dependency and constituency, in *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pp. 478–485, Barcelona, Spain, 2004.

Koehn, P., Statistical significance tests for machine translation evaluation, in *Proceedings of EMNLP 2004*, edited by D. Lin and D. Wu, pp. 388–395, Association for Computational Linguistics, Barcelona, Spain, 2004.

Kübler, S., McDonald, R., and Nivre, J., *Dependency parsing*, Synthesis lectures on human language technologies, Morgan & Claypool, US, URL http://books.google.com/books?id=k3iiup7HB9UC, 2009.

Kuhlmann, M. and Satta, G., Treebank grammar techniques for non-projective dependency parsing, in *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pp. 478–486, Association for Computational Linguistics, Athens, Greece, URL http://www.aclweb.org/anthology/E09-1055, 2009.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B., Building a large annotated corpus of english: the Penn Treebank, *Comput. Linguist.*, *19*, 313–330, URL http://portal.acm.org/citation.cfm?id=972470.972475, 1993.

Marneffe, M.-C. D., Maccartney, B., and Manning, C. D., Generating typed dependency parses from phrase structure parses, in *In LREC 2006*, 2006.

McClosky, D., Charniak, E., and Johnson, M., Automatic domain adaptation for parsing, in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 28–36, Association for Computational Linguistics, Los Angeles, California, URL http://www.aclweb.org/anthology/N10-1004, 2010.

McDonald, R. and Nivre, J., Analyzing and integrating dependency parsers, *Comput. Linguist.*, *37*, 197–230, URL http://dx.doi.org/10.1162/coli_a_00039.

McDonald, R. and Nivre, J., Characterizing the errors of data-driven dependency parsing models, in *Proceedings*

of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 122–131, URL http://www.aclweb.org/anthology/D/D07/D07-1013, 2007.

McDonald, R., Pereira, F., Ribarov, K., and Hajic, J., Non-projective dependency parsing using spanning tree algorithms, in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pp. 523–530, Association for Computational Linguistics, Vancouver, British Columbia, Canada, URL http://www.aclweb.org/anthology/H/H05/H05-1066, 2005.

Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., and Yuret, D., The CoNLL 2007 shared task on dependency parsing, in *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pp. 915–932, Association for Computational Linguistics, Prague, Czech Republic, URL http://www.aclweb.org/anthology/D/D07/D07-1096, 2007a.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., and Marsi, E., MaltParser: A language-independent system for data-driven dependency parsing, *Natural Language Engineering*, *13*, 95–135, 2007b.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J., Bleu: a method for automatic evaluation of machine translation, in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pp. 311–318, Association for Computational Linguistics, Morristown, NJ, USA, URL http://dx.doi.org/10.3115/1073083.1073135, 2002.

Petrov, S., Chang, P.-C., Ringgaard, M., and Alshawi, H., Uptraining for accurate deterministic question parsing, in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 705–713, Association for Computational Linguistics, Cambridge, MA, URL http://www.aclweb.org/anthology/D10-1069, 2010.

Popel, M., Mareček, D., Green, N., and Žabokrtský, Z., Influence of Parser Choice on Dependency-Based MT, in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pp. 433–439, Association for Computational Linguistics, Edinburgh, Scotland, URL http://www.aclweb.org/anthology/W11-2153, 2011.

Vadas, D. and Curran, J., Adding noun phrase structure to the Penn Treebank, in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 240–247, Association for Computational Linguistics, Prague, Czech Republic, URL http://www.aclweb.org/anthology/P07-1031, 2007a.

Vadas, D. and Curran, J. R., Parsing internal noun phrase structure with Collins' models, in *Proceedings of the Australasian Language Technology Workshop 2007*, pp. 109–116, Melbourne, Australia, URL http://www.aclweb.org/anthology/U07-1016, 2007b.

Vadas, D. and Curran, J. R., Large-scale supervised models for noun phrase bracketing, in *Conference of the Pacific Association for Computational Linguistics (PACLING)*, pp. 104–112, Melbourne, Australia, URL http://www.it.usyd.edu.au/~james/pubs/pdf/pacling07bracket.pd%f, 2007c.

Vadas, D. and Curran, J. R., Parsing noun phrase structure with CCG, in *Proceedings of ACL-08: HLT*, pp. 335–343, Association for Computational Linguistics, Columbus, Ohio, URL http://www.aclweb.org/anthology/P/P08/P08-1039, 2008.

Žabokrtský, Z., Ptáček, J., and Pajas, P., TectoMT: Highly Modular MT System with Tectogrammatics Used as Transfer Layer, in *Proceedings of the 3rd Workshop on Statistical Machine Translation, ACL*, pp. 167–170, 2008.

WDS'11 Proceedings of Contributed Papers, Part I, 143–148, 2011. ISBN 978-80-7378-184-2 © MATFYZPRESS

TamilTB: An Effort Towards Building a Dependency Treebank for Tamil

L. Ramasamy

Charles University in Prague, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics (ÚFAL).

Abstract. Annotated corpora such as treebanks are important for the development of parsers, language applications as well as understanding of the language itself. Only very few languages possess these scarce resources. In this paper, we describe our effort in syntactically annotating a small corpora (600 sentences) of Tamil language. Our annotation is similar to Prague Dependency Treebank (PDT 2.0) and consists of 2 levels or layers: (i) morphological layer (m-layer) and (ii) analytical layer (a-layer). For both the layers, we introduce annotation schemes i.e. positional tagging for m-layer and dependency relations (and how dependency structures should be drawn) for a-layers. Finally, we evaluate our corpora in the tagging and parsing task using well known taggers and parsers and discuss some general issues in annotation for Tamil language.

Introduction

The most important thing in Natural Language Processing (NLP) research is data, importantly the data annotated with linguistic descriptions. Much of the success in NLP in the present decade can be attributed to data driven approaches to linguistic challenges, which discover rules from data as opposed to traditional rule based paradigms. The availability of annotated data such as Penn Treebank [*Mitchell et al.*, 1993] and parallel corpora such as Europarl [*Koehn*, 2005] had spurred the application of statistical techniques [*Ratnaparkhi*, 1996], [*Collins*, 2003], [*Koehn et al.*, 2003] to various tasks such as Part Of Speech (POS) tagging, syntactic parsing and Machine Translation (MT) and so on. They produced state of the art results compared to their rule based counterparts. Unfortunately, only English and very few other languages have the privilege of having such rich annotated data due to various factors.

In this paper, we take up the case of building a dependency treebank for Tamil language for which no annotated data is available. The broad objectives for the design of the Tamil dependency treebank (TamilTB) include: (i) annotate data at morphological level and syntactic level (ii) in each level of annotation, trying for maximum level of linguistic representation and (iii) building large annotated corpora using automatic tools. We have chosen dependency annotation over constituency representation for one obvious reason: that dependency annotation works well for free word order languages and the annotation is quite intuitive and easy to represent. One other reason is that, since treebanking for other Indian languages such as Hindi and Telugu [*Begum et al.*, 2008] too focuses on dependency annotation scheme, it would be easier in the future to compare or adopt features from those efforts. The focus of the paper is primarily on the annotation process at morphological level and syntactic level and evaluation of the annotated resources using publicly available taggers and parsers.

There is an active research on dependency parsing ([*Bharati*, 2009], [*Nivre*, 2009] and [*Zeman*, 2009]) and developing annotated treebanks for other Indian languages such as Hindi and Telugu. One such effort is, developing a large scale dependency treebank [*Begum et al.*, 2008] (aimed at 1 million words) for Telugu, as of now the development for which stands [*Vempaty*, 2010] at around 1500 annotated sentences. For Tamil, previous works which utilised Tamil dependency treebanks are: [*Dhanalakshmi et al.*, 2010] which developed dependency treebank (around 25000 words) as part of the grammar teaching tools, [*Selvam et al.*, 2009] which developed small dependency corpora (5000 words) as part of the parser development. Other works such as [*Janarthanam et al.*, 2007] focused on parsing the Tamil sentences. Those works did not make use of treebank to the parser development, rather they were based on linguistic rules. A somewhat detailed description of an effort to develop a TamilTB appeared in [*Ramasamy et al.*, 2011]. To our knowledge, this is the first attempt to develop a dependency treebank for Tamil with respect to the objectives defined earlier. This will also be the continuation of the work mentioned in [*Ramasamy et al.*, 2011].

The Section 2 will describe the general linguistic aspects of the Tamil language in brief, Section 3 will describe the annotation process in general and explain the preprocessing step in the annotation

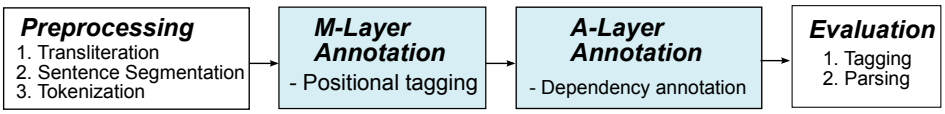


Figure 1. Annotation process.

process, Section 4 will introduce the morphological level annotation (m-layer annotation), Section 5 will introduce dependency annotation (a-layer annotation) and some of the issues involved and Section 6 will describe the evaluation on the developed resources.

General Aspects of the Tamil Language

Tamil is a south Indian language that belongs to the Dravidian family of languages. Other major languages in the Dravidian family include Telugu, Malayalam and Kannada. The main features of the Tamil language include agglutination, relatively free word order, head final and the subject-verb agreement. Below we touch briefly on these features.

Morphology. Tamil is an agglutinative language [*Lehmann*, 1989] and has a rich set of morphological suffixes which can be added one after another to noun and verb stems (mainly) as suffixes. Tamil morphology is mainly concatenative and derivations are also possible by means of *adjectivalization, adverbialization* and *nominalization*. In general, Tamil morphology can be represented [*Lehmann*, 1989] as [*stem (+affix)*^{n}]. Though there are only eight basic POS categories, with no such restrictions placed on as to how many words can be glued together, Tamil morphology pose significant challenges to POS tagging and parsing.

Head Final and Relatively Free Word Order. Tamil is a head final language, meaning the head of the phrasal categories always occur at the end of a phrase or constituent. Modifiers and other co-constituents always precede the phrasal head. For example, *postposition* is the head of the postpositional phrase, and will be modified by noun phrases. There are very few exceptions (identifiable) such as the subject of a sentence occuring after the finite verb (head). In most cases, head final rule is preserved.

Tamil is a Subject Object Verb (SOV) language and the word order is relatively free. Within a clause, phrases can be moved to almost any position except to the postion of clause head which should always be a verb. Besides the above features, subjects in Tamil agrees with verb in person, number and gender. Certain verbs will not code agreement with them, for ex: *illai, muti* etc.

Annotation Process

Our annotation scheme is based on Prague Dependency Treebank (PDT) [*Hajic*, 1998] and [*Böhmová et al.*, 2001]. PDT annotates the data in 3 levels or layers: (i) morphological layer (m-layer) (ii) surface syntax annotation (a-layer) and (iii) tectogrammatical annotation (t-layer). As we have mentioned earlier, our annotation process includes only the first 2 layers i.e. m-layer and a-layer. The Figure 1 shows the annotation process and the Table 1 shows the general information about the data used for annotation. This section will introduce in brief how preprocessing is done prior to the actual annotation process.

Table 1. General statistics of the data.

Description	value
Source	`www.dinamani.com`
Format	UTF-8
Transliterated	yes
Number of sentences	600
Number of words	9581

Preprocessing

The preprocessing stage consists of 3 steps. Once the raw corpus is downloaded from the web (`www.dinamani.com` in our case), the corpus in UTF-8 is transliterated into Latin for the ease of representation inside the programming components. Then the sentence segmentation is performed on the transliterated data to split the raw corpus into sentences. We used simple heuristics such as *fullstop, name initials, attribution* etc. to split into sentences. Wrong sentence splitting is corrected later during

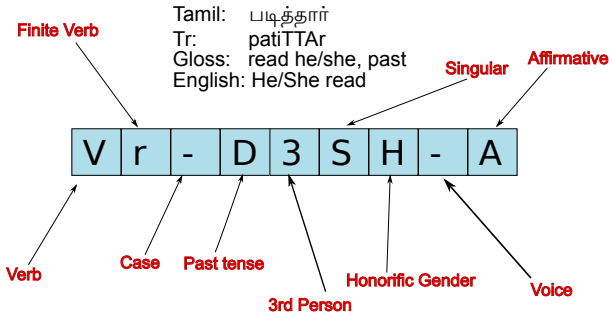


Figure 2. Positional tag.

the annotation. Tokenization is one of the important steps in preprocessing. The default delimiter in tokenization is *space*. However, Tamil is known to combine closed class words with general wordforms, which can be represented as separate words in languages such as English. For ex: Tamil, in certain situations combine *postpositions* with nouns, *clitics* with almost any wordforms and *auxiliary verbs* with verbs. We splitted those combination of words with the help of the list given in the Table 2. So given this list we will be able to split the agglutinative combinations such as *nouns + postpositions*, *verbs + auxiliaries* and etc. Initial splitting was done automatically using a few well known words from the list, and the remaining words or suffixes are found later when manually analyzing the data. This process will aid the m-layer annotation by reducing the tag complexity as well as data sparsity to some extent. We evaluated how much such combinations have been splitted from the original corpora. We found that 953 splits took place out of 9581 words. We can say that almost 10% of the additional corpus size is due to splitting some wordforms into separate tokens. The Table 3 shows an example sentence before and after applying the splitting.

Table 2. Closed class words for tokenization.

Category	**Word or suffix list**
Clitics	um, E, EyE, AvaTu
Postpositions	kUta, utan, pati, kuRiTTu, iliruwTu, anRu, uL, ARu, Tavira, pOTu, pOla, pinnar, pin, arukE, aRRa, inRi, illATa, mITu, kIz, mEl, munpE, otti, paRRi, paRRiya, pOnRa, mUlam, vaziyAka etc.
Auxiliary Verbs	patta, pattu, uLLa, pata, mAttATu, patuvArkaL, uLLAr, uLLanar, illai, iruwTAr, iruwTaTu, pattaTu, pattana, mutiyum, kUtATu, vENtum, kUtum, iruppin, uLLana, mutiyATu, patATu, koNtu, ceyTu etc.
Particles	Aka, Ana and their spelling variants such as Akak, Akac, AkaT
Demonstrative pronouns	ap, ac, ic, iw, aw

Table 3. An example splitting of word combinations.

Before splitting	puTiya cattaTTin**pati** , pATukAkkap**patta** winaivuc cinnaTT**iliruwTu** 1000 ati varai ewTa kattumAnam**um** katta anumaTi illai .
After splitting	puTiya cattaTTin **pati** , pATukAkkap **patta** winaivuc cinnaTT **iliruwTu** 1000 ati varai ewTa kattumAnam **um** katta anumaTi illai .

M-Layer Annotation

The m-layer annotation simply corresponds to POS tagging of the data. We decided to use *positional tagging* scheme to annotate our corpus. The main advantage of the positional tagging is that it can accommodate morphological features. We can easily train the POS taggers for *coarse grained* or *fine grained* tagsets. The main difference compared to ordinary POS tagging is that the *positional tag* for each word has a *fixed length* characters. Each character in the tag signifies a particular feature of that word. For our purpose, we have defined the length of the positional tag to be *9*.

The Figure 2 shows an example Tamil word and its *tag*. As you can see, the first letter of the tag is **V** which indicates that the word is a verb. The second position (**r**)[1] indicates that the verb is a finite verb and so on. The Figure 3 (a) & (b) shows the positional tagging system with a possible values for

[1]SUBPOS values are not given in this paper. We will release the data and the full annotation scheme soon.

Position	Feature	#Possible Values
1	POS	14
2	Sub POS	42
3	Case	10
4	Tense	05
5	Person	04
6	Number	03
7	Gender	06
8	Voice	02
9	Negation	02

(a) Each position & num of possible values

Value	Description
A	Adverbs
C	Conjunctions
D	Determiners
I	Interjections
J	Adjectives
N	Nouns
P	Postpositions
Q	Quantifiers
R	Pronouns
T	Particles
U	Numerals
V	Verbs
X	Unknown
Z	Punctuations

(b) POS values

Description	Value
Corpus size	9581 words
Vocabulary size	3583 words
# of tags for this corpus	217
# words received unique tags	3464
# words received 2 tags	109
# words received 3 tags	9
# words received 4 tags	1

(c) m-layer annotation statistics

Figure 3. Positional tag system.

No	Afun	Afun	Examples
1	AAdjn	Adverbial Adjunct	Optional adverbs, optional PP phrases attaching to verb
2	AComp	Adverbial Complement	Obligatory adverbs, obligatory PP phrases attaching to verb
3	AdjAtr	Adjectival Attribute	Adjectivalized verbs, or relative clauses
4	Apos	Apposition	Heads of the apposition clauses - clauses attaching to 'enRa'
5	Atr	Attribute	Noun modifiers
6	AuxA	Determiners	Demonstrative pronouns (iwTa-'this', awTa-'that')
7	AuxC	Subordinating Conjunctions	Subordinating Conjunctions (enRu, ena, Aka)
8	AuxG	Punctuations	-, ", ', $, rU., (,), [,]
9	AuxK	Terminal Punctuation	:, . , ?
10	AuxP	Postpositional head	mITu-'on', paRRi-'about', kIz-'under'
11	AuxS	Technical Root	Technical Root
12	AuxV	Auxiliary Verb	uL, koNtu, iru
13	AuxX	Comma (not coordination)	,
14	AuxZ	Emphatic particles (clitics)	TAn(emphasis), um-'also, even', E-'even'
15	CC	Part of a word	kiLarwTu ezuwTu - 'rise' as in *rising against*, written as 2 words
16	Comp	Complement (not adverbial)	Obligatory attachments to non verbs, "belongs to the batch **of 1977**"
17	Coord	Coordination node	maRRum - 'and', um
18	Obj	Object	Object
19	Pnom	Nominal Predicate	Nominal Predicate , nouns as predicates
20	Pred	Main Predicate	Main Predicate
21	Sb	Subject	Subject

Figure 4. Dependency relations (Analytical functions).

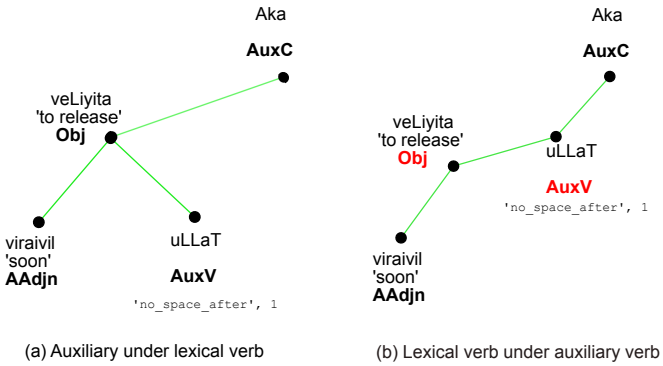
the first position i.e. main POS tag. Figure 3 (c) shows the basic statistics of the m-layer annotation. From the Figure, we observe that the entire corpus was tagged by 217 tags. The Figure also shows how many tags each word in the vocabulary can take. Over 96% of the wordforms are *unambiguous*. Only little over 3% of wordforms are ambiguous by having 2 tags. 3 tags and 4 tags are negligible. Lemmas for each wordform will also be stored as an attribute (`lemma` attribute) in m-layer annotation. At present, lemmas are identified partially through automation. Remaining are edited or added manually.

A-Layer Annotation

The a-layer annotation corresponds to dependency annotation. This step consists of two stages: (i) identifying the structure by attaching the *dependent* word as child to the *governing* word and (ii) labeling the relation with which the dependent and governing nodes (words) are related. Thus each sentence corresponds to a tree structure rooted at the predicate of the sentence or at the technical root. Each edge has a label and it signifies the relation between the parent and child nodes.

So far we have defined 21 dependency relations or analytical functions (afun) for labeling the edges. The Figure 4 shows the afun with some examples. After the *m-layer* annotation is performed, the structure and afun labels for the edges have been produced automatically by the rule based parser and edited manually. The *m-layer* and *a-layer* annotation have been performed for the dataset mentioned in Table 1.

In *a-layer* annotation, issues such as, handling of auxiliary verbs whether the auxiliaries should be hanged under the lexical verbs or the lexical verbs should be hanged under the auxiliary verbs, still remain. One reason for this dilemma is, that in Tamil, lexical verbs always precede auxiliary verbs but it is the auxiliary verb which codes the agreement and establishes morphological clues when there is an embedding of a clause into another clause. On the one hand, it is the lexical verb which is the head of a clause, so we can make the lexical verb as the head. On the other hand, it is the auxiliary verb which



(a) Auxiliary under lexical verb (b) Lexical verb under auxiliary verb

Figure 5. Auxiliary attachment dilemma.

Description	Value
# Sentences for training	479 (7715 words)
# Sentences for testing	121 (1866 words)
Morce Tagger	98.6 % (accuracy)
TnT Tagger	87.0 %

(a) Tagging performance

Description	Value
# Sentences for training	490 (7866 words)
# Sentences for testing	110 (1715 words)
MST (unlabeled)	77.8% (accuracy)
MST (labeled)	67.7%
Malt (unlabeled)	74.8%
Malt (labeled)	65.1%

(b) Parsing performance

Figure 6. Performance evaluation.

makes a connection between the embedded clause and the clause being embedded into, so the auxiliary is the head and the lexical verb will be the child. We chose to go by the first solution, i.e. auxiliary verb under the lexical verb. The Figure 5 shows an example for the auxiliary attachment problem. The `'no_space_after',1` indicates that the suffix "Aka" is part of the auxiliary verb "uLLaT".

As is common in the dependency approach, non-projective constructions can appear at the *a-layer*. They are observed at most in three situations: (i) when adverbs try to modify clauses by jumping the next immediate clause (ii) when arguments are shared between two clauses and when trying to attach some arguments to the first clause and some other to the second clause and (iii) when structures not belonging to Tamil occur.

Evaluation

This is actually not the evaluation of the TamilTB, rather, since the direct application of treebank is the parser development, we decided to evaluate how well the developed resource performs for the tagging and parsing tasks. We evaluated both *m-layer* and *a-layer* annotation independently with different training and test data. For *m-layer* annotation, we evaluated with Morce and TnT tagger. For *a-layer* annotation, we evaluated with Malt (projective) and MST parsers. The MST parser is trained with first order and projective algorithm settings. The data for training and testing are chosen randomly (around 80% for training and remaining for testing). The Figure 6 shows the evaluation results.

Conclusion and Future Work

In this paper, we described our ongoing efforts to develop a dependency treebank for Tamil language. As part of this development, we introduced our annotation scheme at word level and syntactic level. We also used our treebank resource to evaluate the performance in tagging and parsing tasks. The developed resource is still a small amount of data, and we are still trying to improve the annotation scheme and removing inconsistencies in the treebank data. As a future work, we will standardize the annotation scheme, optimize tools for low amount of data, and last but not the least, we will add more annotated data.

Acknowledgment. The research leading to these results has received funding from the European Commission's 7th Framework Program (FP7) under grant agreement n° 238405 (CLARA). We also would like to

thank reviewers for their useful comments.

References

Begum, R., Husain, S., Dhwaj, A., Sharma, D., Bai, L., Sangal, R., Dependency Annotation Scheme for Indian Languages, In: Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP), Hyderabad, India, 2008.

Bharati, A., Gupta, M., Yadav, V., Gali, K., Sharma, D.M., Simple Parser for Indian Languages in a Dependency Framework, In: Proceedings of the Third Linguistic Annotation Workshop (ACL-IJCNLP 2009), pp. 162–165, *Association for Computational Linguistics*, 2009.

Böhmová, A., Hajič, J., Hajičová, E., Hladká, B., The Prague Dependency Treebank: Three-Level Annotation Scenario, In: Anne Abeillé (Ed): Treebanks: Building and Using Syntactically Annotated Corpora, *Kluwer Academic Publishers*, 2001.

Collins, M., Head-Driven Statistical Models for Natural Language Parsing, *Comput. Linguist.* 29, 589–637, 2003.

Dhanalakshmi, V., Anand Kumar, M., Rekha, R.U., Soman, K.P., Rajendran, S., Grammar Teaching Tools for Tamil Language, In: Technology for Education Conference (T4E 2010), pp. 85–88, India, 2010.

Hajic, J., Building a Syntacticly Annotated Corpus: The Prague Dependency Treebank, In: Issues of Valency and Meaning, pp. 106-132, Karolinum, Prague, 1998.

Janarthanam, S., Nallasamy, U., Ramasamy, L., Santhoshkumar, C., Robust Dependency Parser for Natural Language Dialog Systems in Tamil, In *Proceedings of 5th Workshop on Knowledge and Reasoning in Practical Dialogue Systems(IJCAI KRPDS-2007)*, pp. 1–6, Hyderabad, India, 2007.

Koehn, P., Europarl: A Parallel Corpus for Statistical Machine Translation, In: MT Summit, 2005.

Koehn, P., Och, F.J., Marcu, D.: Statistical Phrase-Based Translation, In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, pp. 48–54, *Association for Computational Linguistics*, 2003.

Lehmann, T., A Grammar of Modern Tamil. Pondicherry Institute of Linguistics and Culture (PILC), Pondicherry, India, 1989.

Mitchell P.M., Mary Ann, M., Beatrice, S., Building a Large Annotated Corpus of English: the Penn Treebank. *Comput. Linguist.* 9, 313–330, 1993.

Nivre, J., Parsing Indian Languages with MaltParser, In: Proceedings of the ICON09 NLP Tools Contest: Indian Language Dependency Parsing, pp. 12–18, 2009.

Ramasamy, L., Žabokrtský, Z., Tamil dependency parsing: results using rule based and corpus based approaches, In: Proceedings of the 12th international conference on Computational linguistics and intelligent text processing - Volume Part I, CICLing'11, pp. 82–95, Tokyo, Japan, 2011.

Ratnaparkhi, A., A Maximum Entropy Model for Part-Of-Speech Tagging. In: Proceedings of the Empirical Methods in Natural Language Processing, pp. 133-142, 1996.

Selvam, M., Natarajan, A.M., Thangarajan, R., Structural Parsing of Natural Language Text in Tamil Language Using Dependency Model, *Int. J. Comput. Proc. Oriental Lang.*, Volume 22, 2009.

Vempaty, C., Naidu, V., Husain, S., Kiran, R., Bai, L., Sharma, D., Sangal, R., Issues in Analyzing Telugu Sentences towards Building a Telugu Treebank, In: Alexander F., G. (Ed): CICLing 2010, LNCS 6008, pp. 50–59, 2010.

Zeman, D., Maximum Spanning Malt: Hiring World's Leading Dependency Parsers to Plant Indian Trees, Proceedings of ICON09 NLP Tools Contest: Indian Language Dependency Parsing, *NLP Association of India*, Hyderabad, India, 2009.

Unstated Subject Identification in Czech

G. L. Nguy and M. Ševčíková
Charles University in Prague, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. In this paper we aim to automatically identify subjects, which are not expressed but nevertheless understood in Czech sentences. Our system uses the maximum entropy method to identify different types of unstated subjects and the system has been trained and tested on the Prague Dependency Treebank 2.0. The results of our experiments bring out further consideration over the suitability of the chosen corpus for our task.

Introduction

In Czech, the subject is often expressed in the surface shape of the sentence but can be omitted as well. When present, it is expressed by a sentence member (e.g. noun phrase or pronoun in nominative case, infinitive phrase) or by a dependent clause; we speak about an overt subject. Concerning sentences in which the subject position is not occupied, the subject can be understood from the respective verb form or from the context and/or situation (and added into the surface sentence structure) in most cases; these unstated (covert, null, zero, empty) subjects are the main concern of the present paper. However, there are several subjectless verbs such as *Prší* lit.: 'Rains' ('It rains'), which cannot be accompanied with a subject at all. These cases are taken into consideration in our work as well. In a subjectless finite verb clause we distinguish four following types of unstated subjects:

Implicit subject: The subject is omitted in the surface text but can be understood from the verb morphological information; most often it stands for an entity already mentioned in the text or can be deictic.

(1) Jana ráda peče. Dnes Ø upekla jablečný koláč.
Lit. Jane gladly bakes. Today [she] baked$_{3.SG.FEM}$ apple pie.
Jane likes to bake. Today she has baked an apple-pie.

General subject: The subject does not refer to any concrete entity; it has a general meaning, so it can be omitted in the surface structure.

(2) S rizikem se Ø počítá.
Lit. With risk RFLX [one] counts$_{3.SG}$.
Risk is counted in. (One counts risk in.)

Unspecified subject: The subject denotes an entity more or less known from the context which is however not explicitly referred to.

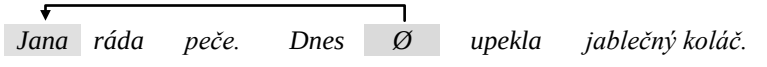
(3) Ø Hlásili to v rádiu.
Lit. [They] Announced$_{3.PL.ANIM}$ it on radio.
It was announced on radio. (They announced it on radio.)

Null subject[1]: The subject does not refer to any entity in the real world. It is neither phonetically realized, nor can be lexically retrieved. In this case the predicate is an impersonal (weather) verb.

(4) Zítra Ø bude oblačno.
Lit. Tomorrow [it] will$_{3.SG}$ cloudy.
Tomorrow it will be cloudy.

Rello & Ilisei [2009] consider another category of omitted subject, i.e. omitted subject in a non-finite verb clause. It is a case of control, but we do not study it in this paper.

In Czech, it is natural to drop out personal pronouns in subject position of the clause[2]. An overt subject pronoun indicates an emphasis of the speaker. In this paper we discuss the unstated subject identification problem, because an unstated implicit subject in third person form is often an anaphor that refers to an entity already mentioned in the text. The term 'zero pronoun identification' is used for these cases in computational approaches to anaphora resolution. An example of a zero pronominal anaphora and its possible usage in machine translation is illustrated in Fig. 1:



[1] It is a term we use in our work to be able to talk about it easily.

[2] Czech is so-called a pro-drop language. By active present and future tense verbs, the person and number can be recognized thanks to the inflectional suffix (excluding *-í* suffix which can be used to indicate both 3rd person singular and plural). By active past tense verbs or passive verbs, the gender can be also specified (excluding *-a* suffix indicating either active past tense 3rd feminine singular or neutrum plural).

Jane likes to bake. Today she has baked an apple-pie.

Figure 1. Zero pronominal anaphora: the implicit subject *Ø* refers to *Jane*.

We used the maximum entropy method to train a model for unstated subject classification and chose the data of the PDT 2.0 for the training and testing procedures. However, the corpus selection does not suit the task and we will discuss it later.

Motivation

In machine translation, the identification and resolution of zero pronouns play an important role if these pronouns are often omitted in the source language (e.g. Czech) but compulsory in the target language (e.g. English). There are systems for anaphora resolution in Czech [*e.g., Kučová & Žabokrtský,* 2005; *Nguy & Žabokrtský,* 2007; *Nguy et al.,* 2009], but none of them devoted sufficient space to zero pronoun resolution in particular as well as zero pronoun identification. Nevertheless, zero pronoun detection is an unseparatable part of anaphora resolution and has been widely investigated in other languages, e.g. in Japanese [*Seki et al.,* 2002], Chinese [*Zhao & Ng,* 2007], Korean [*Han et al.,* 2006], Spanish [*Ferrández & Peral,* 2000] etc.

Unstated Subject Identification

In this section, we introduce the method and the corpus we have used for unstated subject identification. We discuss the problems we have met during our experiments.

Resolution method

Maximum entropy was first introduced to Natural Language Processing (NLP) area by *Berger et al.* [1996]. Since then, the maximum entropy principle has been used widely in NLP, e.g. for tagging, parsing, named entity recognition and machine translation. Maximum entropy models have the following form:

$$p(y|x) = \frac{1}{Z(x)} exp \sum_i \lambda_i f_i(x, y)$$

where f_i is a feature function, λ_i is its weight, and $Z(x)$ is the normalizing factor.

For our task, we chose a maximum entropy classifier, an implementation of Laye Suen[3], a machine learning tool that takes data items and place them into one of *k* classes. In addition, it also gives probability distributions over classifications.

Our approach can be described in the following steps:

1. In a training set, extract features from each finite verb without an overt subject;
2. Train a MaxEnt classifier with them;
3. Test the MaxEnt model on a test set;

Data description

Our experiments make use of the PDT 2.0, which contains a large amount of Czech newspaper texts with complex and interlinked morphological (2 million words), syntactic (1.5 MW) and complex semantic annotation (0.8 MW); in addition, certain properties of sentence information structure and coreference relations are annotated at the semantic level [*Hajič et al.,* 2006].

At the so-called tectogrammatical (semantic) layer (t-layer), the meaning of the sentence is represented as a dependency tree structure. In addition to nodes corresponding to surface tokens, there are newly established nodes the tectogrammatical lemma of which is an artificial t-lemma substitute beginning with *#*. Our focused unstated subjects can be found at t-layer among nodes with t-lemma `#PersPron`, `#Gen` and `#Unsp`; except null subjects, which were not reconstructed at t-layer. These t-lemma substitutes have the following meanings:

`#PersPron` t-lemma substitutes are assigned to:

- personal and possessive pronouns present in the surface sentence;
- zero pronouns representing the implicit subject[4];
- textual ellipsis – obligatory arguments of a governing verb / noun[5];

[3] A Perl module `AI::MaxEntropy`, see http://search.cpan.org/perldoc?AI::MaxEntropy

[4] Dropped pronouns can be distinguished from the expressed ones by the additional node attribute is_generated

[5] A node with the t-lemma substitute `#PersPron` is used in cases of textual ellipsis no matter what the form of the omitted argument is; i.e. not only in the positions where it could be replaced by a personal or possessive pronoun.

#Gen t-lemma substitutes are used for:

- grammatical ellipsis of an obligatory argument – general argument;
- zero pronouns representing the general subject;

#Unsp t-lemma substitutes stand for:

- grammatical ellipsis of an obligatory argument – unspecified Actor;
- zero pronouns representing the unspecified subject;

Feature extraction

Our maximum entropy classifier was trained on the basis of feature vectors for each finite verb (predicate) having no overt subject depending on it. The following features were used:

Categorial features: t-lemma, form, tense, gender, number, person, and:

- adverbial form – an adverb in the case of an 'adverbial' predicate ('to be + an adverb')
- nominal form – a nominal part in the case of a nominal predicate

Binary features:

- `has_actor` – the considered predicate has an overt Actor
- `is_reflexive` – the predicate is reflexive
- `is_passive` – the predicate is a passive verb
- `has_o-ending` – the predicate is a finite verb ending with 'o'
- `is_to-be-infin` – the predicate is in the construction of 'to be + infinitive'
- `has_dep-clause` – there is a dependent clause hanging on the verb

Concatenated features:

- `reflexive_o-ending` – concatenation of the features is_reflexive and has_o-ending
- `passive_o-ending` – concatenation of the features is_passive and has_o-ending
- `reflexive_person_number_gender` – concatenation of the features is_reflexive, person, number and gender
- `passive_person_number_gender` – concatenation of the features is_passive, person, number and gender

The feature selection relies on characteristics of each unstated subject type. A general subject often comes along with a third person singular reflexive verb or a third person singular passive verb. A reflexive verb can be easily recognized by a reflexive particle. A third person singular passive verb and a past tense third person singular reflexive verb always end with 'o'. The case of a subject expressed by a dependent clause can be detected by the has_dep-clause feature. An adverbial form can indicate a null subject, e.g. *Je polojasno* ('It is somewhat cloudy').

Data problems

By the PDT 2.0 choice we have to face the problems related to it. The most crucial problem is the absence of the explicit annotation of unstated subjects we interest in. In Fig. 2 and Fig. 3, we illustrate ambiguous cases, in which two nodes with #PersPron and #Gen appear.

We tried to solve the problem of missing manual unstated subject annotation by proposing some rules:

if [the verb has a #Unsp among children] **then**
    It is the case of an unspecified subject
else if [the verb has a generated #PersPron and a #Gen.ACT among children] **then**
    if [the verb has_o-ending **or** is_to-be-infin **or** is_rflx_pass_by_active_present_3sg] **then**
        It is the case of a general subject
    else
        It is the case of an implicit subject
    end if
else if [the verb has a generated #PersPron.ACT among children] **then**
    It is the case of an implicit subject
else if [the verb has a #Gen.ACT among children] **then**
    It is the case of a general subject
else if [[the verb has a generated #PersPron.ACT among children] **and**
    [[it is_pass **or** is_rflx_pass_not_active_present_3sg] with no o-ending]] **then**
    It is the case of an implicit subject
else
    It is the case of a null subject

Another problem with the PDT 2.0 data is the absence of the manual annotation of person, number and gender. This information is very important for us because of its indication for a general / null subject by a third

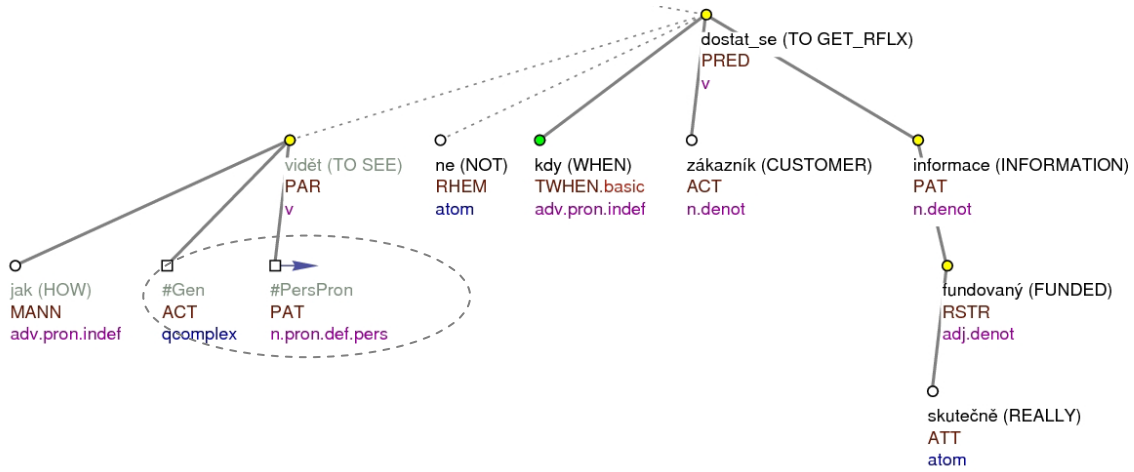


Figure 2. A simplified t-tree representing the sentence *Jak je vidět, ne vždy se zákazníkovi dostane skutečně fundovaných informací.* (Lit. How it's seen, not always RFLX customer gets really funded information.) In this case, the node with `#Gen` is considered to be the unstated general subject.

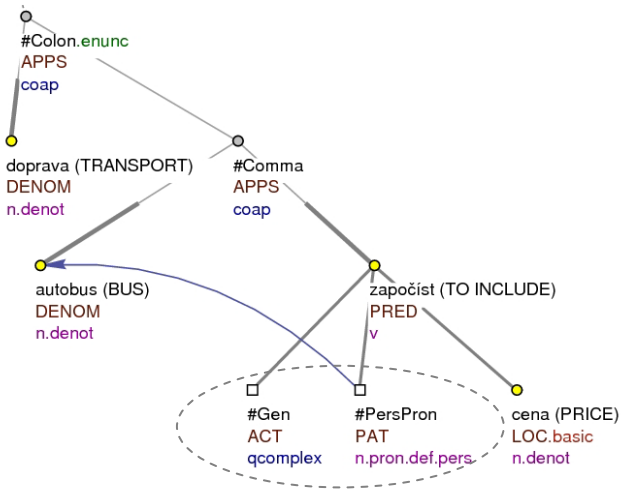


Figure 3. A simplified t-tree representing the sentence *Doprava: Autobus, je započten v ceně.* (Lit. Transport: Bus, is included in price.) In this case, the node with `#PersPron` is the unstated implicit subject.

person singular neuter / animate form or for an unspecified subject by a third person plural animate form.

We have no rules that guarantee a 100% correct resolution for the identification of unstated subjects on annotated data of the PDT 2.0. In addition, we rely on the genre of the corpus, where proverbs with general subjects do not often occur, and suppose all cases with third person singular animate active verb to be an implicit subject; whereas all cases with third person singular neutrum passive or reflexive verb to be a general subject. We expect that the occurrence of singular neuter implicit subject is sporadic as well.

Baselines

The following baselines were created for automatic identification of:

Implicit subject:

if [[a clause contains a finite verb] **and**
[the verb has neither overt subject nor Actor depending on it] **and**
[it has no o-ending] **and** [it is not a reflexive passive verb] **and**
[it is not a passive verb with o-ending] **and** [its t-lemma is not an impersonal verb[6]]
] **then**
There is an implicit subject.
end if

[6] We have manually created a test list of impersonal verbs consisting of verbs: *jednat se* (be about sth), *pršet* (rain), *zdát se* (seem), *dařit se* (do well), *oteplovat se* (get warmer), *ochladit se* (get colder)

General subject:
if [[a clause contains a finite verb] **and**
[the verb has neither overt subject nor Actor depending on it] **and**
[[it has o-ending] **or** [it has a 'to be + infinitive' construction] **or**
[it is a reflexive passive verb having an active present tense third person singular form]
]
] **then**
There is a general subject.
end if
Unspecified subject:
if [[a clause contains a finite verb] **and**
[the verb has no overt subject depending on it and a third person animate plural form] **and**
[[there is no preceding finite verb] **or**
[[there is a preceding finite verb] **and**
[[it has not a third person animate plural form] **or**
[it has not a dependent animate plural noun with functor ACT/PAT/ADDR]
]
]
]
] **then**
There is an unspecified subject.
end if
Null subject:
if [[a clause contains a finite verb] **and**
[the verb has neither overt subject nor Actor depending on it] **then**
There is a null subject.
end if

Evaluation

The PDT 2.0 is divided into three parts: 80% of data is used for training, 10% for development testing and 10% for evaluation testing. In the evaluation, we used the standard metrics with precision, recall and f-measure (Table 1) for unstated subject identification.

Precision = N_c / N_e	Recall = N_c / N_t	F-measure = 2 x P x R / (P + R)

N_c	Number of correctly identified controllees
N_e	Number of identified controllees
N_t	Number of all controllees

Table 1. Evaluation metrics for the unstated subject identification

If the problem of missing manual unstated subject annotation is considered to be 100% successfully resolved by proposed hand-written rules, then we obtain the following results (Table 2):

	P	R	F
Implicit Baseline	95.4%	98.4%	96.9%
Implicit MaxEnt	90.6%	99.4%	94.8%
General Baseline	24.9%	87.2%	38.7%
General MaxEnt	96.7%	74.4%	84.1%
Unspecified Baseline	4.55%	3.45%	3.92%
Unspecified MaxEnt	0%	0%	0%
Null Baseline	98%	85.7%	91.5%
Null MaxEnt	82.5%	29.7%	43.7%

Table 2. Results for the unstated subject identification

Such a poor result of unspecified subject identification can be explained for its rare occurrences in the data, the problem of missing manual person, gender and number annotation and the fact that it requires knowledge of a potential antecedent existence. If there is an antecedent to which the unstated subject can refer, then it is a case of an implicit subject; otherwise an unspecified subject. An anaphora resolution might help to improve this result.

The result of null subject identification might be higher by adding a sophisticated list of impersonal / weather verbs / constructions as well. In general a deeper error analysis should bring overall improvements and explain the doubt of better baseline results.

Conclusion

This paper introduces the linguistics phenomenon of unstated subjects in Czech and its automatic identification using a maximum entropy classifier trained on the PDT 2.0 data. Looking at the results and the data problems leads us to a question, whether we should continue on improving the approach to unstated subject identification or whether we should concern on the automatic identification of `#PersPron`, `#Gen` and `#Unsp` nodes as specified in the PDT 2.0 instead. Or should we try to find another more suitable corpus for the task?

Acknowledgments. The present work was supported by the Czech Grant Agency under Contracts GA CR P406/2010/0875, MSMT CR LC536 and by the Charles University Grant Agency under Contract GAUK 4383/2009.

References

Berge, A. L., V. J. Della Pietra, and S. A. Della Pietra, A Maximum Entropy Approach to Natural Language Processing, *Computational Linguistics*, 22(1), 39–71, 1996.

Ferrández, A. and J. Peral. A Computational Approach to Zero-pronouns in Spanish, in: *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (ACL'00)*, 166–172, Morristown, NJ, USA, Association for Computational Linguistics, 2000.

Hajič, J., J. Panevová, E. Hajičová, P. Sgall, P. Pajas, J. Štěpánek, J. Havelka, M. Mikulová, Z. Žabokrtský and M. Ševčíková-Razímová, *Prague Dependency Treebank 2.0*, Software prototype, Linguistic Data Consortium, Philadelphia, PA, USA, ISBN 1-58563-370-4, www.ldc.upenn.edu, 2006.

Han N. R., E. F. Prince, M. Palmer, and E. Buckley, *Korean Zero Pronouns: Analysis and Resolution*. Ph.D. thesis, University of Pennsylvania, 2006.

Kučová, L. and Z. Žabokrtský, Anaphora in Czech: Large Data and Experiments with Automatic Anaphora, *LNCS/Lecture Notes in Artificial Intelligence/Proceedings of Text, Speech and Dialogue*, 3658:93–98, 2005.

Nguy, G. L. and Z. Žabokrtský, Rule-based Approach to Pronominal Anaphora Resolution Applied on the Prague Dependency Treebank 2.0 Data, in: *Proceedings of the 6th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC 2007)*, 77–81, Lagos (Algarve), Portugal, CLUP-Center for Linguistics of the University of Oporto, 2007.

Nguy, G. L., V. Novák and Z. Žabokrtský, Comparison of Classification and Ranking Approaches to Pronominal Anaphora Resolution in Czech, in: *Proceedings of the SIGDIAL 2009 Conference*, 276–285, The Association for Computational Linguistics, London, UK, 2009.

Rello, L. and I. Ilisei, A Rule-Based Approach to the Identification of Spanish Zero Pronouns, in: *Proceedings of Student Workshop of Recent Advances in Natural Language Processing (RANLP 2009)*, 60–65, Borovets, Bulgaria, 14-15 September, 2009.

Seki, K., A. Fujii, and T. Ishikawa, A Probabilistic Method for Analyzing Japanese Anaphora Integrating Zero Pronoun Detection and Resolution, in: *Proceedings of the 19th International Conference on Computational Linguistics*, 1–7, Morristown, NJ, USA, Association for Computational Linguistics, 2002.

Zhao, S. and H. T. Ng, Identification and Resolution of Chinese Zero Pronouns: A Machine Learning Approach, in: *Proceedings of the 2007Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007.

WDS'11 Proceedings of Contributed Papers, Part I, 155–160, 2011. ISBN 978-80-7378-184-2 © MATFYZPRESS

Utilization of Anaphora in Machine Translation

M. Novák
Charles University in Prague, Faculty of Mathematics and Physics,
Malostranské nám. 25, Praha 1, CZ-118 00, Czech Republic.

Abstract. Majority of present machine translation systems do not address the retaining of text coherency, they translate just isolated sentences. On the other hand, the authors of anaphora resolvers rarely integrate these tools into more complex scenarios, e.g. the task of machine translation. We propose the ways how machine translation systems can utilize the knowledge of anaphoric relations both in the source as well as in the target language in order to improve the quality of translation. Specifically, we present how to incorporate anaphora resolution into the process of English to Czech translation using the TectoMT system.

Introduction

Machine translation (MT) is probably one of the most established tasks in the field of natural language processing. Therefore, it is surprising that nowadays so little attention is received by research on retaining the text coherency. On the contrary, current best systems based on statistical machine translations (SMT) make some strong independence assumptions.

Although very popular phrase-based SMT systems do not usually assume independence on segments smaller than a sentence, they make use of n-gram statistics,[1] which can capture only adjacent words and phrases. These systems face problems with handling long distance dependencies.

This shortcoming can be minimized by using SMT systems that include linguistic analysis up to the layer of deep syntax (e.g. TectoMT [Žabokrtský et al., 2008]). On this layer, a sentence is usually represented as a tree, keeping track of dependencies between the words, no matter how far from each other they appeared on the surface.

However, even for this deeper SMT approach, some problems still persist. The largest issue lies in that SMT systems translate individual sentences without taking the context of previous sentences into account. Another shortcoming, for example in TectoMT, is that even though it well describes dependency relations including the long distance ones, it insufficiently covers other types of relations, e.g. discourse and anaphoric relations.

Error analysis of TectoMT translation

In order to justify the claims on SMT shortcomings above, we carried out a survey on the English sentences translated by TectoMT system into Czech to find out how these problems reflect in real data. We made use of the data from the English-Czech test set for Shared Translation Task organized with EMNLP 2011 Sixth Workshop on Statistical Machine Translation.[2]

One of the problems that reduces the coherency of the translated text is the translation of personal pronouns. SMT systems usually do not investigate to what a pronoun refers. It results in an inconsistency between the translated pronoun and the translated referent.

To show the shortcomings of SMT in pronoun translation we conducted the following analysis (illustrated in Table). We randomly selected 100 English sentences that contained a pronoun "it", comprising 119 occurrences of this pronoun in total. We observed that more than half of them corresponds to either references to a bigger segment of the text (a clause, sentence or paragraph[3]) or pleonastic usages.[4] In the former case after their translation into the Czech variant in neutral almost never an error is incurred and in the latter case it is rather a matter of syntax how the pronoun's governing phrase looks like in the target language. Thus, we were not interested in these occurrences.

The rest of the occurrences referred to some entity mentioned in the previous text and we examined how the TectoMT system succeeded in their translation. Even though the English pronoun "it" can be translated into one of the three Czech pronouns, which differ in a gender, TectoMT outputs it always in neutral. We observed that the number of erroneously translated pronouns "it" account for 26% of

[1] With n usually smaller than 5.
[2] http://www.statmt.org/wmt11/test.tgz
[3] Boundaries of the segment the pronoun refers to does not even have to be precisely defined.
[4] We categorized them according to rules proposed in [Li et al., 2009].

	Deep	**Surface**	**Total**
	Erroneous translation		
Fem	9	18	27
Masc	4	13	17
Total	13	31	44
	Correct translation		
Neut	1	19	20
	Entity non-referring		
Pleo	—		31
Segm	—		24
Total	—		55

Table 1. The results of analysis of translating 119 pronouns "it". The section *Erroneous translation* contains pronouns incorrectly translated to neutral gender. The label Deep denotes those, which by chance did not produce a mistake on the surface but they were erroneous on the deep syntactic layer. The label Surface denotes those, which produced error also on the surface. The section *Correct translation* contains correctly translated pronouns. The label Deep denotes those, which were correctly translated on the deep syntactic layer, however, during the synthesis an error was incurred. The label Surface denotes those, which were correct also on the surface. The section *Entity non-referring* denotes pleonastic and segment-referring pronouns.

(a) **The leader of the PPC** ... present **her** conditions ... ***The president*** **of the popular Catalans** *went* to
Ref: Předsedkyně/fem ... se dostavila/fem
Tst: Prezident/masc ... šel/masc

(b) It is a combination of **location detection via GPS** (used by regular **car** ***navigation***) ...
Ref: navigace
Tst: plavba

(c) ... ***bunch*** of ripe **bananas** ...
Ref: svazek
Tst: parta

Figure 1. Illustrating examples from the English to Czech translation by TectoMT, where text coherency was not retained. Labels "Ref" and "Tst" denote a correct reference translation and a translation output by TectoMT, respectively.

all occurrences and together with errors that by chance did not appear on the surface[5] they form 37% (more than 2/3 of those referring to some entity).

Regarding the wrong choice of a gender we also came across the example in Figure 1a. The phrase "president ... went to" was mistakenly translated into masculine gender. If the system took into account that expressions in bold denote the same object and one of the expressions was a possessive pronoun "her", it would correctly output the subject with verb in feminine gender.

In the output of TectoMT system, we also noticed the incorrect choice of the translation for some word in the source text, caused by the lack of knowledge about the previous context. For instance, in the example in Figure 1b despite the evidence in the context that the sentences describe a car navigation system and GPS technology, the word "navigation" was erroneously translated into "plavba" ("cruise"). Similarly, in the example in Figure 1c the system failed in the translation of the word "bunch" into "parta" ("group" in the meaning of "group of people").

From the analysis and some particular examples above we can see that TectoMT system suffers from not taking a previous context into account and from insufficient handling of other than dependency relations, e.g. the anaphoric relations.

[5]For instance, a verb in the present tense has the same form no matter in what gender the subject is. However, it no longer holds for verbs in the past tense.

Anaphora as a means of coherence

In this paper we would like to address the mentioned deficiencies and suggest some solutions to avoid them. We believe that the awareness of anaphoric relations, i.e. relations between entities that appear in a text, in the process of machine translation could help in minimizing the errors.

Let us return to the examples found in the data. In the example in Figure 1a all the expressions in bold refer to the same entity – the leader of a Catalan popular party. They represent a special type of anaphora called coreference. In the example in Figure 1b the "location detection via GPS" is a function of a "car navigation", thus these expressions form a function–object bridging anaphora. Similarly, in the last example in Figure 1c "banana" is a part of the whole "bunch", what is in the theory of anaphora[6] denoted as a part–whole bridging anaphora. From these examples we see that if we had a tool to reveal such relations, we could use them in the process of MT to retain the coherency of the translated text, thus improving the quality of the translation.

Related work

A lot of research has been carried out on anaphora resolution (AR), especially on the coreference resolution (CR).[7] Many of hundreds of works on AR published so far declared MT as one of their main motivation. In light of that it seems peculiar that almost none of them conducted any experiments on integration of AR into the MT system.

The lack of interest in utilizing anaphora knowledge in MT was not present all the time. During 1990s some authors of rule-based MT systems attempted to handle inter-sentential relations. In 1999 this effort culminated with a special issue of Machine Translation journal concerning anaphora resolution in MT [Mitkov, 1999]. An example of such rule-based system using AR can be the work of Peral and Rodríguez [2002], who implemented translation from English to Spanish and vice versa using transfer via interlingua.

After the rise of SMT approach the research on this topic paused for 10 years. Just recently two new works have emerged, both applying CR on translation of personal pronouns (especially "it" and "they") from English into French and German, respectively. Le Nagard and Koehn [2010] tagged pronoun "it" with gender information by replacing it with one of the following surface forms: "it-neutral", "it-feminine" and "it-masculine". The corresponding form of corefering pronoun is determined by the gender of the antecedent's French translation. On the other hand Hardmeier and Federico [2010] introduced a robust system of translating the referred sentences into German in advance and integrated a so-called word dependency module comprising information on coreference links as an additional feature into a log-linear SMT model. Whereas results of the former suffered from usage of low-performance rule-based systems for CR, in the latter work they succeeded in increasing the quality of personal pronoun translation.[8]

Our suggestions

In the remaining part of this paper we present our suggestions on how to incorporate anaphora knowledge into the process of SMT, which might help in retaining coherency in the output of the translation. Our proposals are aimed to be integrated into TectoMT system, especially into the English to Czech translation via deep syntactic (tectogrammatical) transfer.

We opted for this language pair and direction because the tools and models neccessary for English to Czech translation are the most available for us. The majority of our suggestions requires the anaphora resolver for the source language. English seems to be the best choice, since a huge number of CR system are implemented. In addition, both languages are for us easy to understand, which facilitates revealing of errors.

The main reason for translation via a tectogrammatical layer, i.e. using the TectoMT framework, lies in a design of an available CR system for Czech. To our knowledge the only existing system presented by Nguy et al. [2009] requires several features, which are not present on linguistic layers lower than the deep syntactic layer.

Note, however, that the following suggestions were designed to be applied for any language pair. In addition, following ideas can be employed in a slightly modified way even for phrase-based SMT (for instance, in the system Moses).

[6] The typology we use is based on the theory described by Nědolužko [2009].
[7] The summary of statistical methods on CR can be found for instance in [Ng, 2010].
[8] Nonetheless, they did not achieve improvement in BLEU [Hardmeier and Federico, 2010].

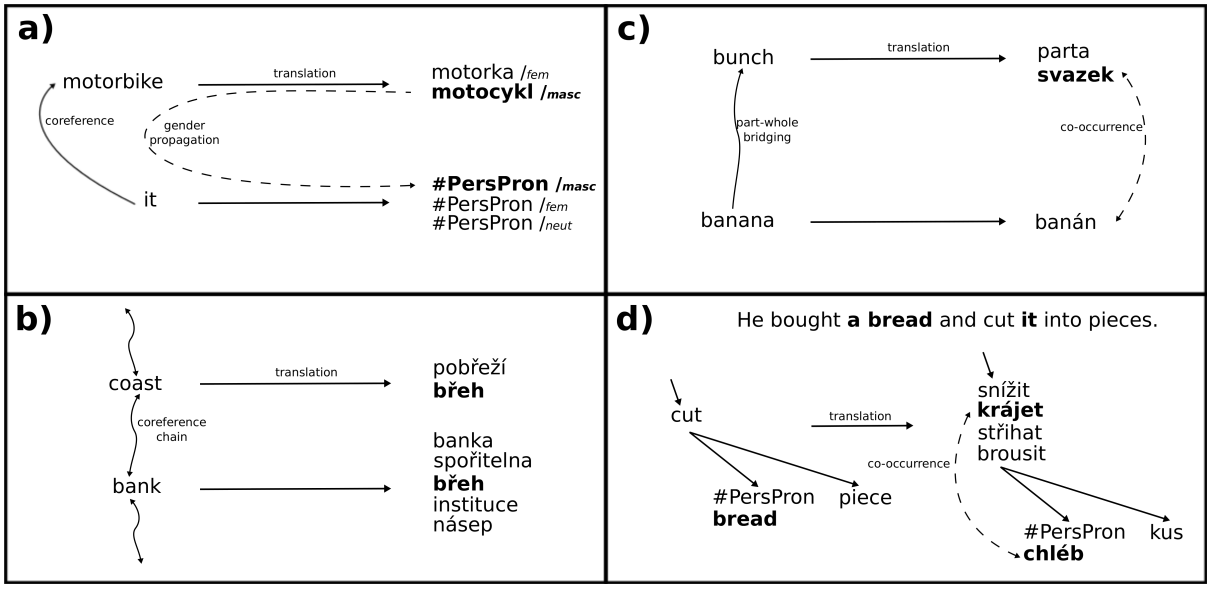


Figure 2. Suggestions of anaphora utilizing in MT on the side of the source language. The lemma #PersPron is an abstraction of all personal and possesive pronouns.

Anaphora resolver on the source side

AR resolver can be utilized on the side of the source language in several ways. One of them was already proposed in the works of Le Nagard and Koehn [2010] and Hardmeier and Federico [2010], i.e. helping to solve the issue of personal pronoun translation. To facilitate the correct translation of a personal pronoun (e.g. "it" in Figure 2a), the system has to be provided with the gender[9] of the closest antecedent's translation (e.g. masculine, if "motorbike" is translated to "motocykl"). This requires to translate referred sentences before the referring ones and to keep track of a gender of an antecedent's translation (propagated gender). Then we can determine the gender of the translated pronoun to agree with the propagated one or allow some value of uncertainty by introducing an additional feature, which describes this agreement, into a log-linear model.

Whereas in the utilization of anaphora mentioned above the system needs information just from the last coreferential expression preceding the pronoun, in the following idea we suppose we can obtain better results, if the available coreference chain is longer. We can use the knowledge of a whole coreference chain, i.e. all expressions referring to the same entity, to pick translation equivalents more confidently. When searching for the Czech translation of an English word e, instead of selecting the Czech word c_i that maximizes the probability $p(c_i|e)$, we choose the word c_j that maximizes the probability of translating e to c_j in the context of whole entity E that e belongs to. We calculate this probability as a weighted sum of probabilities that c_j is a correct translation of m over all expressions m that belongs to E. Written in a formula:

$$p_E(c_j|e) = \sum_{m \in E} \lambda_{e,m} p(c_j|m),$$

where the dependence of weights $\lambda_{e,m}$ also on the current translated word e allows for instance to prefer the translation probability $p(c_j|e)$ of the word e. Considering the example illustrated in Figure 2b, if the SMT system takes the whole coreference chain into account, it can favor "břeh" ("coast") over "banka" ("bank" as an institution) as a translation of English word "bank".

Similarly, we can use the knowledge of bridging anaphora in the same manner, as it is depicted in Figure 2c. Given the whole-part relation between the tectogrammatical nodes "bunch" and "banana", we can copy this relation into the target tree and infer the translations of nodes with respect to the constraint set by the relation (the system picks "svazek" rather than "parta" as a translation of "bunch"). More generally, it does not have to be strictly defined what kind of relation joins the words, it should be enough just to know whether such a relation between them exists. For setting these relations we can employ some of the association measures proposed for example in [Pecina, 2008].

The last proposal, how coreference knowledge on the source side could improve the SMT, is the help

[9] Sometimes the grammatical number can be required for correct translation as well.

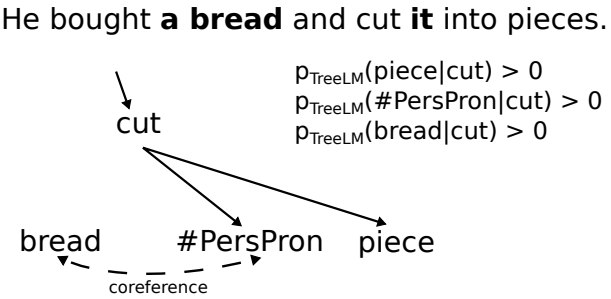


Figure 3. A suggestion of enrichment of target-language tree model, if taking the coreference relations into account. For clarity, we used an English example. In English – Czech translation that we focus on, the model would be nonetheless built from Czech data.

to decide the translation of a translated node's parent or children in a tectogrammatical tree. Considering again an example sentence "He bought a bread and cut it into pieces" (Figure 2d), the translated pronoun "it" alone does not give us sufficient information to predict the appropriate translation of the verb "cut". However, if we replace the node "#PersPron" of the pronoun expression by the node of the whole entity ("bread", "#PersPron"), its translation ("chléb", "#PersPron") then gives preference to "krájet" as the translation of the verb.

Anaphora resolver on the target side

While it is more obvious how to integrate anaphora resolution on the side of the source language, it is probably not so clear how to exploit the knowledge served by anaphora resolver on the side of the target language.

We suggest to make use of anaphora knowledge for in target-language tree model (TreeLM) as described in [Mareček et al., 2010]. TreeLM specifies the probabilities of nodes' word forms[10] given the word forms of their parents. Let us assume that the data, which these probabilities are estimated from, are enriched with coreference relations. If we enable replacement of individual personal pronoun expressions with the expressions they are coreferential with, we can build a tree model not only from those couples that co-occurred within the parent – child dependency relation, but it allows also for inclusion of the couples, which can potentially appear.

For instance, in Figure 3 the data for TreeLM consist solely of one sentence, in which the nodes "bread" and "#PersPron" are coreferential. Then, given that the parent is a node "cut", TreeLM can predict non-zero probabilities not only for the node "#PersPron" but also for the node "bread".

Such an enrichment of the TreeLM can both give rise to new dependency relations that did not appear in the corpus and lead to more reliable estimates of probabilities assigned to dependency relations.

Conclusion

In this paper we gave some proposals, how the knowledge of anaphora relations could help to improve the quality of MT. While our contribution so far lies in the stage of suggestions, we plan to continue this project and test the methods on English – Czech translation. Our proposals should be nevertheless appliable to all language pairs and some of the proposed ideas can be easily adapted to other SMT strategies than the deep-syntactic transfer we assumed here.

Acknowledgments. This work was supported by the grants GAUK 4226/2011 and Czech Science Foundation 201/09/H057. We thank two anonymous reviewers for their useful comments.

References

Hardmeier, C. and Federico, M., Modelling Pronominal Anaphora in Statistical Machine Translation, in *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, edited by M. Federico, I. Lane, M. Paul, and F. Yvon, pp. 283–289, 2010.

Le Nagard, R. and Koehn, P., Aiding Pronoun Translation with Co-Reference Resolution, in *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pp. 252–261, Association for Computational Linguistics, Uppsala, Sweden, 2010.

[10] In fact, rather than of a word form it specifies the probabilities of two attributes, the word form can be decomposed to: the lemma and formeme, which captures the surface morphosyntactic form of the node.

Li, Y., Musílek, P., Reformat, M., and Wyard-Scott, L., Identification of Pleonastic It Using the Web, *J. Artif. Intell. Res. (JAIR)*, *34*, 339–389, 2009.

Mareček, D., Popel, M., and Žabokrtský, Z., Maximum entropy translation model in dependency-based MT framework, in *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pp. 201–201, Association for Computational Linguistics, Uppsala, Sweden, 2010.

Mitkov, R., Introduction: Special Issue on Anaphora Resolution in Machine Translation and Multilingual NLP, *Machine Translation*, *14*, 159–161, 1999.

Ng, V., Supervised Noun Phrase Coreference Research: The First Fifteen Years, in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1396–1411, Association for Computational Linguistics, Uppsala, Sweden, 2010.

Nguy, G. L., Novák, V., and Žabokrtský, Z., Comparison of classification and ranking approaches to pronominal anaphora resolution in Czech, in *Proceedings of the SIGDIAL 2009 Conference*, pp. 276–285, Association for Computational Linguistics, London, UK, 2009.

Nědolužko, A., *Zpracování rozšířené textové koreference a asociační anafory na tektogramatické rovině v Pražském závislostním korpusu*, Ph.D. thesis, MFF UK, Praha, Czech Republic, in Czech, 2009.

Pecina, P., *Lexical Association Measures: Collocation Extraction*, Ph.D. thesis, Charles University in Prague, Prague, Czech Republic, 2008.

Peral, J. and Rodríguez, A. F., Pronominal Anaphora Generation in an English-Spanish MT Approach, in *CICLing*, pp. 187–196, 2002.

Žabokrtský, Z., Ptáček, J., and Pajas, P., TectoMT: Highly modular MT system with tectogrammatics used as transfer layer, in *ACL 2008 WMT: Proceedings of the Third Workshop on Statistical Machine Translation*, pp. 167–170, 2008.

WDS'11 Proceedings of Contributed Papers, Part I, 161–166, 2011. ISBN 978-80-7378-184-2 © MATFYZPRESS

Machine Translation with Significant Word Reordering and Rich Target-Side Morphology

B. Jawaid

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. This paper describes the integration of morpho-syntactic information in phrase-based and syntax-based Machine Translation systems. We mainly focus on translating in the hard direction which is translating from morphologically poor to morphologically richer languages and also between language pairs that have significant word order differences. We intend to use hierarchical or surface syntactic models for languages of large vocabulary size and improve the translation quality using two-step approach [Fraser, 2009]. The two-step scheme basically reduces the complexity of hypothesis construction and selection by separating the task of source-to-target reordering from the task of generating fully inflected target-side word forms. In the first step, reordering is performed on the source data to make it structurally similar to the target language and in the second step, lemmatized target words are mapped to fully inflected target words. We will first introduce the reader to the detailed architecture of the two-step translation setup and later its further proposed enhancements for dealing with the above mentioned issues. We plan to conduct experiments for two language pairs: English-Urdu and English-Czech.

1. Introduction

The task of a machine translation (MT) system is to translate the text from one language into text into another. MT approaches are roughly classified into rule-based and data-driven paradigms. In classical rule-based systems, linguists perform deep analyses of linguistic phenomena of the given language pair and capture them in hand-written transformation rules which is a very labor-intensive task. The rules are later applied by an MT engine. On the other hand, data-driven approaches use large text corpora to automatically learn translation equivalences based on the real examples that are extracted from the corpus. Modern statistical machine translation (SMT) [Koehn, 2010] systems extract the knowledge from large parallel corpora with added linguistic information.

SMT systems and in particular phrase-based SMT systems (PSMT) usually don't perform well for the language pairs that differ in sentence structure [Koehn et al., 2009] and when target language is rich in inflection. Our first language pair i.e. English-Urdu exhibits the same language characteristics which shows complexity of modeling this language pair for the translation task. English is SVO(subject-verb-object) language whereas Urdu follows SOV sentence structure which requires translation system to move verb to the end of the sentence when translating from English to Urdu. Urdu and Czech are morphologically rich languages. For instance, adjectives in Urdu are inflected according to the gender and number of the following noun. The morphological richness increases data sparseness and the differences in word order compel PSMT to learn long distance reordering. The author's Master thesis [Jawaid, 2010] already discussed translation issues in the direction from English to Urdu and proposed solutions to deal with the word order differences in the given language pair. This work is further extension of the author's previous research.

The rest of this paper is organized as follows: In Section 2 we briefly describe the tools and resources we will use to build the two-step model. Then we describe the basic two-step architecture in Section 3 and our proposed improvement techniques to exploit morpho-syntactic information for SMT systems in Section 4. Later, we briefly introduce the recent contributions

from other researchers in improving SMT quality (Section 5). Finally, in Section 6 we provide a brief summary of our proposed translation scheme.

2. Relevant Tools

In this section we provide a short overview of all the available tools and resources that will be used at each step of our two-step setup. All of the details concerning the specific use of each tool are explained in Section 4.

2.1. Moses

Moses [Koehn et al., 2007] is a statistical phrase-based MT system that automatically learns from the parallel corpus of any given language pair. It also combines the language model capabilities for producing fluent output translation. Moses offers two types of translation models: phrase-based and tree-based.

Phrase-based Translation Model (PBTM). Phrase-based translation model operates on sequences of words called phrases. It is based on the noisy channel model [Brown et al., 1990] approach which is well defined over Bayes decision rule. Bayes formula takes into consideration the language model probability and the probability of translating source phrase into best matching target phrase to obtain best output translation.

In phrase-based model source sentences are segmented into a number of phrases where each phrase gets translated into a target phrase. Target phrases might get reordered based on word order difference between source and target language. Moses by default uses *distance* reordering that allows movement of input phrases relative to previous phrase. The phrase movement over large distance means more expensive translation and it is thus seldom used.

Tree-based Translation Model (TBTM). Moses tree-based translation model [Hoang and Koehn, 2010] is formally known as *hierarchical phrase-based model* and *syntax-based model*, sometimes also referred as *moses-chart*. In PBTM, translation process is carried out from left-to-right of input whereas TBTM builds translation options recursively. The main motivation of TBTM is to introduce syntax using tree structures and for that it uses Synchronous Context-Free Grammar (SCFG) as the underlying formalism. SCFG represents sentence-pairs of source and target languages as pairs of constituency trees. Grammar rules are automatically learned during training from bitext and they consist of both linguistically motivated non-terminals (NP, VP, ...; making the syntax-based model) as well as generic non-terminal (X; making the hierarchical model). The hierarchical model can be trained similarly to phrase-based models but the training of the syntax-based model requires syntactically annotated input.

Instead of using simple phrases, hierarchical model of Moses uses hierarchical phrases i.e. phrases that contain sub-phrases. In hierarchical model all grammar rules consist of only non-terminal (X) with the exception of two special *gluing rules* that uses S to combines sequences of X for generating final output.

Sentence translation probability is calculated using language model probability and the product of weights of all grammar rules use to construct the output translation. Weight of each grammar rule is calculated using log-linear model. Beside these main components of sentence generation, other scoring functions are also used.

2.2. Joshua

Joshua [Li et al., 2010] is another SMT system that uses *hierarchical phrase-based model* introduced by [Chiang, 2005]. Joshua is also formally based on SCFG where rules are learnt from bitext during training. Joshua is more or less equivalent to hierarchical model of Moses and translations are also scored in similar fashion as described for Moses TBTM.

2.3. Maximum Entropy-Based Classifier

In this work, we plan to build a maximum-entropy-based classifier [McCallum et al., 2000] for inflection prediction task. The motivation behind introducing the classifier is to facilitate the use of features looking far away from the processed word. In the simple design of the two-step approach by [Bojar and Kos, 2010] and [Fraser, 2009] the prediction was performed using a simple n-gram model so only few previous words helped in the decision. Perhaps a more important flaw of the simple design is that the few previous words, if not relevant, increase the sparsity and thus make the inflection decision harder.

Our classifier approach is very similar to [Toutanova et al., 2008]. In their setup, the MT system generates only stems in the first step and produce an n-best list which is further sorted and augmented with the fully inflected word forms by the inflection prediction model in the second step. On the other hand, our setup generates augmented lemmatized output in the first step and outputs lattices which encode generally more translation candidates than n-best list. [Jeong et al., 2010] further extended work of [Toutanova et al., 2008] by integrating their discriminative lexicon model directly into the search within their tree-to-string-based SMT system.

A brief introduction to the proposed input features and target classification is provided in Section 4.3.

3. Two-Step Translation

Factored translation models [Koehn and Hoang, 2007] come into play when one of the source or target language is morphologically rich. Each token in the factored model consists of number of factors representing the surface form, lemma, POS tag, so on. Translation options are constructed in a sequence of mapping steps. Because each translation option needs to be fully constructed before the actual search takes place, there is a high risk of combinatorial explosion of the search space [Bojar and Kos, 2010].

The idea behind using two-step translation is to avoid the explosion of the search space by dealing with reordering and word inflections in separate steps. Target-specific morphological features are introduced in the second step only whereas morphological features common to both source and target together with word reordering are handled in the first step. This reduces the risk of the combinatorial explosion, because the target side of the first step is not cluttered with information not available and relevant for the source language and the transfer.

Our baseline system will be similar to the systems presented by [Bojar and Kos, 2010] and [Fraser, 2009]. They used Moses in the first step which produces augmented simple target output. Output of the first step is not fully inflected target instead it represents *middle language* consist of lemma and other morphological features. The second step translation is monotone where another Moses system is trained on augmented lemmatized target input and fully inflected target output.

Recently, [Fraser et al., 2011] has tried two-step setup by replacing Moses at the second step with 4 HMMs (Hidden Markov Models).

4. Proposed Configurations of Two-Step Translation

In this section we provide the details of further refinement techniques for two-step baseline system that will model reordering in more elegant way instead of relying only on Moses default reordering system. We will also try to deal with the inflection prediction task cleverly.

4.1. Reordering Techniques

We plan to use more sophisticated systems for dealing with word reordering issues. We will replace phrase-based Moses on the first step with either Joshua or Moses-chart. These SMT

systems allow block movements which could help in improving reordering. The output of the first step will consist of the series of *strings* representing 1-best reordering for each sentence.

To make the reordering task slightly easier for the first step's systems, we will first pre-reorder the input data and try to make the source and target word orders more similar to each other. For translating from English-to-Urdu, the data will be pre-reordered using the transformation system used in [Jawaid, 2010]. The transformation system will produce 1-best reordered output which will be used as input for Joshua and Moses-chart.

For overcoming the "hard decisions" that are encountered due to relying on one possible reordering of each sentence which cannot be undone during decoding phase, transformation system will produce multiple reorderings of each sentence that will be later fed into the Moses in the form of a *word lattice* [Dyer et al., 2008]. We leave the decision on Moses to pick the best reordering among several possible reorderings. [Niehues and Kolss, 2009] first used lattice-based pre-reordering approach where different possible reorderings of each sentence (collected by applying discontinuous non-deterministic POS rules learned from word-aligned corpus) encoded as weighted edges in lattice.

4.2. Exploring Middle Layer

In all the settings described above, the systems in the first step always produce the strings of 1-best reordered output that are later used by the second step. We further plan to extend the string-based output of the first step to the lattice-based output i.e. multiple reorderings of each input sentence will be produced, giving the second step systems the freedom to choose among reordered sentences the one that is the easiest to inflect.

4.3. Using Classifier

Phrase-based Moses in the second step will be replaced with the classifier previously introduced in Section 2.3. The classifier takes a string in the middle language as input and outputs the fully inflected target words. In Table 1, we provide a brief summary of relevant morphological features of our two target languages. The values of these features have to be predicted from the source or surrounding target-side context.

Table 1. Identified Morphological Features for Urdu and Czech

Features	Urdu	Czech	Both
POS categories	42	11 main or 67 detailed	
Gender		neuter, inanimate	masculine, feminine
Number		dual	singular, plural
Person			1,2,3
Tense			present, past, future
Aspect	subjunctive, continuous		perfective, imperfective
Case	ergative, oblique		nominative, accusative, dative, genitive, locative, vocative, instrumental
Grade			positive, comparative, superlative

5. Related Research

Significant research has been done in integrating linguistic information to the SMT systems including syntactically motivated translation models and introducing syntax in phrase-based SMT systems.

Many contributions have been made in the direction towards syntactic knowledge-oriented translation models. [Wu, 1997; Yamada and Knight, 2001] and many others proposed translation

systems similar to [Chiang, 2005]. Yamada and Knight [2001] used methods based on tree-to-string mappings where source language sentences are first parsed and later operations on each node such as reordering child nodes, inserting extra words at each node and translating leaf nodes are applied. In later research, [Eisner, 2003] presented issues of working with isomorphic trees and presented a new approach of non-isomorphic tree-to-tree mapping translation model using synchronous tree substitution grammar (STSG).

Different approaches have been adapted for applying syntactic knowledge to the corpus before passing it to the translation system. For instance syntactic pre-reordering, syntactic reranking (post-processing) and many others. Syntactic pre-reordering has been shown effective many times for introducing syntax in SMT. So far syntactic pre-processing is applied on a source language in two different ways, either by using hand-crafted transformation rules or by learning transformation rules automatically from bitext. Our transformation system [Jawaid and Zeman, 2011] is based on the former approach, this approach was previously successfully applied to other language pairs [Collins et al., 2005; Wang et al., 2007; Ramanathan et al., 2008] as well.

[Li et al., 2007] first gave idea of using maximum entropy model based on source language parse trees to get n-best syntactic reorderings of each sentence which was further extended to use of lattices. After [Niehues and Kolss, 2009], [Bisazza and Federico, 2010] further explored lattice-based reordering techniques for Arabic-English; they used shallow syntax chunking of the source language to move clause-initial verbs up to the maximum of 6 chunks where each verb's placement is encoded as separate path in lattice and each path is associated with a feature weight used by the decoder.

6. Conclusion

We have presented several techniques to deal with data sparsity and word reordering issues. We are trying to reduce the complexity of the search space and the risk of search errors that are mostly encountered due to modeling both reordering and morphology at the same step. We plan to split the two problems into separate steps. In the first step, only reordering and morphological features common to both languages are handled. In the second step, all remaining morphological features of the target language are decided based on monolingual information only. Although this is not the first time the two-step approach is presented, our work is still novel in terms of: the language pairs we are going to deal with and the integration of different reordering systems in the first step on top of classifier.

Acknowledgments. The work on this project was supported by the grant LC536 Centrum komputační lingvistiky of the Czech Ministry of Education.

References

Bisazza, A. and Federico, M., Chunk-based verb reordering in vso sentences for arabic-english statistical machine translation, in *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pp. 235–243, Stroudsburg, PA, USA, 2010.

Bojar, O. and Kos, K., 2010 failures in english-czech phrase-based mt, in *Proceedings of the WMT '10*, pp. 60–66, Stroudsburg, PA, USA, 2010.

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S., A statistical approach to machine translation, *Comput. Linguist.*, *16*, 79–85, 1990.

Chiang, D., A hierarchical phrase-based model for statistical machine translation, in *Proceedings of the ACL '05*, pp. 263–270, Stroudsburg, PA, USA, 2005.

Collins, M., Koehn, P., and Kučerová, I., Clause restructuring for statistical machine translation, in *Proceedings of the ACL '05*, pp. 531–540, Stroudsburg, PA, USA, 2005.

Dyer, C., Muresan, S., and Resnik, P., Generalizing word lattice translation, in *Proceedings of the ACL*, pp. 1012–1020, Columbus, Ohio, USA, 2008.

Eisner, J., Learning non-isomorphic tree mappings for machine translation, in *Proceedings of the ACL '03 - Volume 2*, pp. 205–208, Stroudsburg, PA, USA, 2003.

Fraser, A., Experiments in morphosyntactic processing for translating to and from german, in *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pp. 115–119, Stroudsburg, PA, USA, 2009.

Fraser, A., Weller, M., Cahill, A., and Fritzinger, F., Morphological generation of german for smt, in *Machine Translation and Morphologically-rich Languages. Research Workshop of the Israel Science Foundation*, University of Haifa, Israel, 2011.

Hoang, H. and Koehn, P., Improved translation with source syntax labels, in *Proceedings of the WMT '10*, pp. 409–417, Stroudsburg, PA, USA, 2010.

Jawaid, B., Statistical machine translation between languages with significant word order difference, in *Univerzita Karlova v Praze & University of Malta*, p. 99, Praha, Czechia, 2010.

Jawaid, B. and Zeman, D., Word-order issues in english-to-urdu statistical machine translation, *The Prague Bulletin of Mathematical Linguistics*, pp. 87–106, 2011.

Jeong, M., Toutanova, K., Suzuki, H., and Quirk, C., A discriminative lexicon model for complex morphology, in *Ninth Conference of the Association for Machine Translation in the Americas*, 2010.

Koehn, P., *Statistical Machine Translation*, Cambridge University Press, 2010.

Koehn, P. and Hoang, H., Factored translation models, in *Proceedings of the EMNLP-CoNLL*, pp. 868–876, 2007.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E., Moses: open source toolkit for statistical machine translation, in *Proceedings of ACL Demo and Poster Sessions*, pp. 177–180, Praha, Czechia, 2007.

Koehn, P., Birch, A., and Steinberger, R., 462 machine translation systems for europe, in *Proceedings of Machine Translation Summit XII*, 2009.

Li, C.-h., Zhang, D., Li, M., Zhou, M., Li, M., and Guan, Y., A probabilistic approach to syntax-based reordering for statistical machine translation, in *Proceedings of ACL*, pp. 720–727, 2007.

Li, Z., Callison-Burch, C., Dyer, C., Ganitkevitch, J., Irvine, A., Khudanpur, S., Schwartz, L., Thornton, W. N. G., Wang, Z., Weese, J., and Zaidan, O. F., Joshua 2.0: a toolkit for parsing-based machine translation with syntax, semirings, discriminative training and other goodies, in *Proceedings of the WMT '10*, pp. 133–137, Stroudsburg, PA, USA, 2010.

McCallum, A., Freitag, D., and Pereira, F. C. N., Maximum entropy markov models for information extraction and segmentation, in *Proceedings of the ICML '00*, pp. 591–598, San Francisco, CA, USA, 2000.

Niehues, J. and Kolss, M., A pos-based model for long-range reorderings in smt, in *Proceedings of the StatMT '09*, pp. 206–214, Stroudsburg, PA, USA, 2009.

Ramanathan, A., Bhattacharyya, P., Hegde, J., Shah, M., R., and M., S., Simple syntactic and morphological processing can help english-hindi statistical machine translation, in *IJCNLP*, 2008.

Toutanova, K., Suzuki, H., and Ruopp, A., Applying Morphology Generation Models to Machine Translation, in *Proceedings of ACL-08: HLT*, pp. 514–522, Columbus, Ohio, 2008.

Wang, C., Collins, M., and Koehn, P., Chinese syntactic reordering for statistical machine translation, in *EMNLP-CoNLL*, pp. 737–745, 2007.

Wu, D., Stochastic inversion transduction grammars and bilingual parsing of parallel corpora, *Comput. Linguist.*, *23*, 377–403, 1997.

Yamada, K. and Knight, K., A syntax-based statistical translation model, in *Proceedings of ACL '01*, pp. 523–530, Stroudsburg, PA, USA, 2001.

 ISBN 978-80-7378-184-2

Sentence-Level Polarity Detection in a Computer Corpus

K. Veselovská

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. The paper presents a preliminary research on possible relations between the syntactic structure and the polarity of a Czech sentence by means of the so-called sentiment analysis of a computer corpus. The main goal of sentiment analysis is the detection of a positive or negative polarity, or neutrality of a sentence (or, more broadly, a text). Most often this process takes place by looking for the polarity items, i.e. words or phrases inherently bearing positive or negative values. These words (phrases) are collected in the subjectivity lexicons and implemented into a computer corpus. However, when using sentences as the basic units to which sentiment analysis is applied, it is always important to look at their semantic and morphological analysis, since polarity items may be influenced by their morphological context. It is expected that some syntactic (and hypersyntactic) relations are useful for the identification of sentence polarity, such as negation, discourse relations or the level of embeddedness of the polarity item in the structure. Thus, we will propose such an analysis for a convenient source of data, the richly annotated Prague Dependency Treebank.

Introduction

Sentiment analysis (often referred to as opinion mining) tasks aim for the automatic extraction of subjective information from text and determination of speaker's attitude. The issue of subjective texts recognition has been discussed in linguistic works since early 80s and 90s, but a substantial progress in the area has started only recently with the rise of the semantically defined Web 2.0 which is based on user-generated content, e.g. social networks and weblogs [see *Ruppenhofer, Somasundaran, and Wiebe, 2008*].

There are two different types of text classification in opinion mining: subjectivity detection and polarity detection. In subjectivity detection the task is to determine whether a given text represents an opinion or a fact – or more precisely whether given information is factual or nonfactual, whereas the aim of polarity detection is to find whether the opinion expressed in a text is positive or negative.

Polarity is mostly indicated by subjective elements, i.e. single words or more complex expressions containing positive or negative polarity (e.g. *nice*, *awful* etc.). These elements are not only frequent content words. As *Wiebe et al.* [2004] states it: "Purely syntactic or morphological devices may also be subjective elements". This means that polarity items are subject to influences of sentence or larger text span context (e.g. negation or changes in aspect in both Czech and English) and thus can be profitably explored in a syntactic treebank.

Sentence-Level Polarity Classification

The main goal of sentence-level polarity detection is to decide whether a given sentence expresses either an overall positive or negative opinion. Thus, all sentences to be classified are assumed to be subjective and carrying either positive or negative overall polarity. There are several reasons why to investigate polarity detection at the sentence level. It is obvious that polarity classification at the sentence level is more fine-grained than document-level polarity classification, because every word has to be interpreted correctly (e.g. in English one needs to determine whether *like* is a verb and hence a positive polar expression or just a preposition; in Czech, we need to distinguish between particular senses of semantically ambiguous adjective *hrubá* etc.). Moreover, according to *Wiegand et al.* [2010] at the document level, text classification relies very much on redundancy and there are so many cues suggesting positive polarity more likely than negative polarity. Additionally, subjectivity is usually not uniformly distributed across a document, so the frequency analysis used e.g. in text summarization is not enough without knowledge of influence of particular polar expressions at the sentence level.

The most influential syntactic (and hypersyntactic) features useful for identification of sentence polarity ones are negation, sentential modality marking, discourse relations, intersentential coreferential relations and depth of the polarity item in the tree. The embeddedness of the polar node in a tree seems to be crucial for the polarity of a given sentence. In Figure 1, we can see an example of a sentence *Unfortunately, brother did a good job.* There are two polarity items in the structure, one positive and one negative, but its overall polarity is negative. Thus, we can assume that the higher the node is, the stronger influence it has.
It is also claimed that the main predicate is more predictive towards polarity than other words or that the main clause is more relevant than subordinate clause – see *Wiegand and Klakow* [2009].

The overall contribution of implementation of polarity items into a treebank is the inspection of such linguistic features and even polarity features derived from sentence structure and its usage in supervised machine learning.

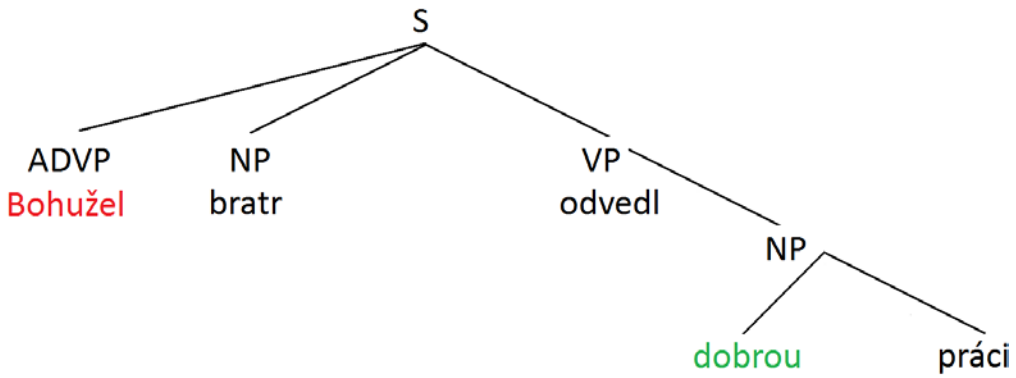


Figure 1. An illustration of the depth feature influence in the tree.

Application of Sentiment Analysis to PDT

Prague Dependency Treebank or PDT [*Hajič et al., 2006*] is a large-coverage treebank with a rich linguistic annotation (morphology, surface and deep syntax, topic-focus articulation, coreference, discourse relations etc.). Thanks to this rich annotation, it is well suited to tasks using different levels of linguistic features, like sentiment analysis. The subjective information is semantic in nature, therefore it should be embedded into the tectogrammatical layer of PDT. Despite the fact that the tectogrammatical layer seems already rather overburdened with linguistic annotation, it seems useful to keep the polarity detection at the same layer as the annotation of coreference and discourse relations, as these phenomena are closely related.

The process of application of sentiment analysis to the Prague Dependency Treebank is supposed to occur in three main phases. In the first phase of the project, it is necessary to compile a subjectivity lexicon, i.e. collection of polarity items, for Czech. The issue of building a subjectivity lexicon is concretely described e.g. in *Banea, Mihalcea and Wiebe* [2008]. The authors use a small set of subjectivity words and a bootstraping method of finding new candidates on the basis of a similarity measure. The authors get to the number of 4,000 top frequent entries for the final lexicon. The assumption is that for the purpose of PDT annotation, a sample of up to 1,000 entries should be sufficient. In case this number proves insufficient, a similar method as described in *Banea, Mihalcea and Wiebe* [2008] would be a suitable way of expanding the subjectivity lexicon further. Another method of establishing a subjectivity lexicon – translation of an existing foreign language subjectivity lexicon – is described in *Banea, Mihalcea, Wiebe and Hassan* [2008]. The authors use sentiment analysis for machine translation purposes. They are interested in how the information about polarity should be transferred from one language to another, if the polarity can differ in the corresponding text spans and if a subjectivity lexicon for the target language can be compiled during the translation. As far as Czech is concerned, the corpora can be simply based on the new Frequency Dictionary of Czech [*Čermák et al., 2004*] or the Czech Thesaurus [*Klégr, 2008*] or derived from a plain text annotation.

Secondly, the words and phrases from a subjectivity lexicon are expected to be automatically identified in the Prague Dependency Treebank and annotated using tags for “positive”, “negative”,

"neutral" or "undecidable" value. The manual and automatic identification of linguistic expressions of the private states (speaker's attitudes) is explored also in *Wilson* [2008]. Besides polarity, Wilson recognizes also intensity and attitude as important features of subjectivity expressions, with attitudes bearing two other important markers: source and target of sentiment. We believe that for the current purposes of the research on sentiment analysis in Czech, this is a too fine-grained distinction. If more tags are needed, they can be easily added during the manual control phase. The annotation should be automatic, but it will be required to make a series of manual controls of a random part of the data to ensure the reliability of annotation. Then, after tagging the data, the analysis of the annotation using statistical methods will be applied. The relationship between the number, the tag value and the position of the tagged nodes in the structure and the overall polarity of the sentence (or text) will stay in the centre of our linguistic interest. Moreover, the tagged data will thus be prepared as training data for future sentiment analysis and opinion mining experiments.

The results of the project should be applicable in many areas of Natural Language Processing, such as question answering, automatic summarization of a text, automatic dialogue systems etc.

Conclusion

Unlike some contemporary linguists [*Wiebe, Wilson, Bruce, Bell, and Martin, 2004*], we decided not to derive subjective language directly from the corpora. Though using primarily the non-contextual value of the word, the so-called prior polarity, we are aware of the possibility of context influence, therefore we include manual annotation controls into the research. We believe that the information about the amount of disagreement between a prior and contextual polarity (excluding irony) represents an important piece of information about the linguistic behaviour of subjectivity elements.

Concerning our future work, many studies [*e.g. Ruppenhofer, Somasundaran and Wiebe, 2008; Somasundaran, Namata, Wiebe and Getoor, 2009*] focus on the mutual dependency between opinion mining and discourse relations annotation. It has been pointed out that sentiment analysis is useful for the identification of discourse relations in the text, and vice versa. In this respect, our research is connected to a project aimed at the analysis and annotation of discourse relations in PDT, which is already under way.

We are not aware of any systematic research including sentiment analysis in Czech linguistic (or computational-linguistic) circles, though there are software experiments using it. Only scarcely a related study (like the one in *Smrž* [2006]) appears, but not primarily designed for Czech data. Moreover, although almost all studies of the topic mention the impact of syntactic structures, the actual research is devoted to the separate studies of individual syntactic phenomena (such as *Narayanan, Liu, and Choudhary* [2009]). Also, only a few projects use syntactically annotated corpora, although the idea is promising. Our assumption is that by using a treebank with rich linguistic annotation (including morphological, syntactical and semantic tagging, coreference, discourse and topic-focus articulation annotation) we will gain a general overview of the impact of syntactic phenomena on sentence polarity.

Acknowledgments. The present work was supported by the Charles University Grant Agency under Contract 3537/2011 and by the Czech Science Foundation under the project 201/09/H057.

References

Banea, C., R. Mihalcea and J. Wiebe, A Bootstrapping Method for Building SubjectivityLexicons for Languages with Scarce Resources. In The Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008), 2008.

Banea, C., R. Mihalcea, J. Wiebe and S. Hassan, Multilingual Subjectivity Analysis Using Machine Translation. Conference on Empirical Methods in Natural Language Processing (EMNLP 2008), 2008.

Čermák, F. *et al.*, Frekvenční slovník češtiny. NLN, 2004.

Hajič, J., J. Panevová, E. Hajičová, P. Sgall, J. Štěpánek, J. Havelka and M. Mikulová, Prague Dependency Treebank 2.0. CD ROM. CAT: LDC2006T01, 1-58563-370-4. Linguistic Data Consortium, Univ. of Pennsylvania, Philadelphia, USA, 2006.

Klégr, A., Tezaurus jazyka českého. NLN, 2008.

Narayanan, R., B. Liu and A. Choudhary, Sentiment Analysis of Conditional Sentences. In Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-09). August 6-7, 2009.

Ruppenhofer, J., S. Somasundaran and J. Wiebe, Finding the Sources and Targets of Subjective Expressions. In The Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008), 2008.

Smrž, P., Using WordNet for Opinion Mining. In: Proceedings of the Third International WordNet Conference, GWC 2006, Brno, CZ, MUNI, 333-335, 2006.

Somasundaran, S., G. Namata, J. Wiebe and L. Getoor, Supervised and Unsupervised Methods in Employing Discourse Relations for Improving Opinion Polarity Classification. In Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP 2009), 2009.

Wiebe, J., T. Wilson, R. Bruce, M. Bell and M. Martin, Learning subjective language. *Computational Linguistics*, *30*, 3, 2004.

Wiegand, M., A. Balahur, R. Benjamin, D. Klakow and A. Montoyo, A Survey on the Role of Negation in Sentiment Analysis. Proceedings of the Workshop on Negation and Speculation in Natural Language Processing . 60-68, 2010.

Wiegand, M. and D. Klakow, The Role of Knowledge-based Features in Polarity Classification at Sentence Level in Proceedings of the 22nd International FLAIRS Conference (FLAIRS-2009). 2009.

Wilson, T., Fine-Grained Subjectivity Analysis. PhD Dissertation, Intelligent Systems Program, University of Pittsburgh, 2008.

WDS'11 Proceedings of Contributed Papers, Part I, 171–176, 2011. ISBN 978-80-7378-184-2 © MATFYZPRESS

Nominal Valency in Lexicons

A. Vernerová

Abstract. The term valency refers to the number, type and form of arguments that are bound to a word. Valency is specific to any given lexical unit and therefore is covered by lexicons. This is a preliminary survey conducted with the creation of a valency lexicon of Czech nouns in mind. The authors of such a lexicon have to decide who will be the intended users, how the material will be presented and which aspects of valency behaviour will be covered; we present the choices made by the authors of several Czech, English and German resources that cover the valency of nouns, both machine readable [FrameNet 1.5, 2010; NomBank 1.0, 2008; PDT-Vallex, 2006] and printed [Herbst et al., 2004; Sommerfeldt and Schreiber, 1996; Svozilová et al., 2005].

Introduction

Valency plays a crucial role in the Czech linguistic tradition [Panevová, 1980; Daneš et al., 1987; Karlík, 2000; Sgall, 2006]. Lexicographic description of valency has been most extensive in the case of verbs: *Valenční slovník českých sloves* [Lopatková et al., 2008] gives rich linguistic information including valency patterns, division of verbs into semantic classes, information on control, reflexivity and reciprocity. In the last two years, two monographs concerning valency of Czech nouns were published [Čermáková, 2009; Kolářová, 2010]. However, although valency behaviour of nouns is covered by two existing lexical resources [PDT-Vallex, 2006; Svozilová et al., 2005], neither of them offers rich linguistic information. The aim of this survey is to present and compare existing English, Czech and German lexicons[1] which cover nominal valency and to identify some of the crucial decisions (both concerning the material and its presentation) that have to be made before any lexicographic work is begun.

In the first section, we discuss different kinds of uses and users of valency lexicons; in the second section, we touch upon the alternatives to the alphabetical ordering of the items; then we compare the presentation of the syntactic and semantic aspects of valency patterns in some of the available lexicons; finally, we mention the role of corpus evidence and the choice of example sentences.

Intended uses and users. Choice of entries

The creation of a valency lexicon is a lengthy and expensive process; therefore it should ideally produce a resource useful for a wide range of users. In this section, we have a look at how the intended group of users influences the choice of entries in the lexicon.

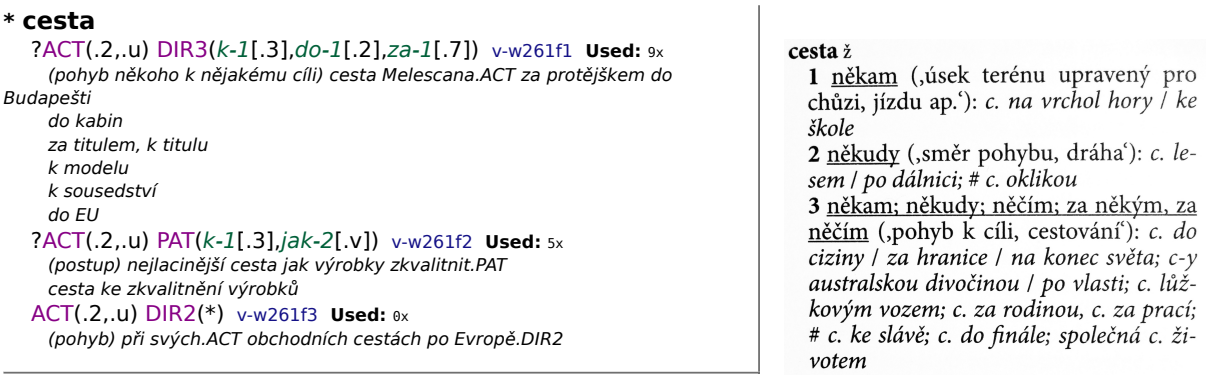
Second language learners

The term "valency," coined by Lucien Tesnière in his 1959 book *Eléments de syntaxe structurale*, was quickly adopted by researchers working in the area of foreign language education. Many valency lexicons are therefore primarily intended for non-native speakers, whether it is Helbig and Schenkel's lexicon of German verbs (published as early as 1969), its adjectival and nominal extensions [Sommerfeldt and Schreiber, 1974, 1977] or newer lexicons covering all three parts-of-speech in one volume [Sommerfeldt and Schreiber, 1996; Herbst et al., 2004].

These lexicons cover around 1500 words (counting the series of German lexicons from the 60's and 70's as one lexicon), of which 250–750 are nouns. The wordlist is based upon two criteria: frequency (only frequent words are important for learners) and the complexity of patterns (learners are more likely to struggle with such words).

However, research such as Bräunling's 1989 survey among the teachers of European Goethe Institutes shows that only very few teachers use valency lexicons in their classes. The rest either don't know valency lexicons at all or find them too theoretical, complex and specialized. Thus it seems that students are best served when valency information is included directly in learner's dictionaries.

[1] For lexical resources which are particularly concerned with valency information, we use the term *lexicon*; the term *dictionary* is reserved for general dictionaries. This may not coincide with the actual titles of the works discussed.



*** cesta**

?ACT(.2,.u) DIR3(*k-1*[.3],*do-1*[.2],*za-1*[.7]) v-w261f1 **Used:** 9x
(pohyb někoho k nějakému cíli) cesta Melescana.ACT za protějškem do Budapešti
do kabin
za titulem, k titulu
k modelu
k sousedství
do EU

?ACT(.2,.u) PAT(*k-1*[.3],*jak-2*[.v]) v-w261f2 **Used:** 5x
(postup) nejlacinější cesta jak výrobky zkvalitnit.PAT
cesta ke zkvalitnění výrobků

ACT(.2,.u) DIR2(*) v-w261f3 **Used:** 0x
(pohyb) při svých.ACT obchodních cestách po Evropě.DIR2

cesta ž

1 někam (‚úsek terénu upravený pro chůzi, jízdu ap.'): *c. na vrchol hory / ke škole*
2 někudy (‚směr pohybu, dráha'): *c. lesem / po dálnici; # c. oklikou*
3 někam; někudy; něčím; za někým, za něčím (‚pohyb k cíli, cestování'): *c. do ciziny / za hranice / na konec světa; c-y australskou divočinou / po vlasti; c. lůžkovým vozem; c. za rodinou, c. za prací; # c. ke slávě; c. do finále; společná c. životem*

Figure 1. Entries for the word *cesta* in PDT-Vallex and in *Slovník vazeb a spojení*

Native speakers and linguists

Dictionaries for native speakers differ from those previously mentioned mainly in size, as it is expected that native speakers tend to look up less frequent words. *Slovník vazeb a spojení* [Svozilová et al., 2005] has 16 000 entries, most of which are verbs. The entries are selected from *Slovník spisovné češtiny* [Filipec et al., 1994], the authoritative dictionary of current standard Czech. The selection includes all verbs which take valency arguments, but only a limited number of adjectives and nouns. Nouns are included only if they are deverbal or have similar valency patterns as deverbals (*hovor o Praze* "a talk about Prague" → *kniha o Praze* "a book about Prague"). Moreover, deverbal and deadjectival nouns are sometimes omitted if their patterns can be inferred from the patterns of the corresponding verb/adjective. Thus, the authors assume that the user is capable of making the inference from *potopit loď* "to sink a ship" to *potopení lodi* "the sinking of a ship," from *odolný vůči/proti/k něčemu* "resistant to something" to *odolnost vůči/proti/k něčemu* "resistance to something." In our opinion it would be necessary to conduct research among users to justify this assumption.

When lexicons are primarily intended as a resource for linguistic research, they may contain richer linguistic information, more complex notation and more elaborate search tools than what would be appropriate for general users. For example, in [PDT-Vallex, 2006] (Figure 1), the arguments are named by their tectogrammatical functors of the Functional Generative Description. A linguist may think that the meaning of these functors is fairly intuitive—*ACT* and *PAT* are the first and the second argument and as such are syntactically determined; the names of the other arguments are then determined by semantic criteria (in our example we have *DIR2* "which way" and *DIR3* "where to").[2] However, it is rather unlikely that general users would make the effort to understand this formalism.

Natural language processing

In Natural Language Processing, valency lexicons play two complementary roles: 1. during the creation of annotated data, valency lexicons enable consistency of annotation which could otherwise not be reached; this is important especially if the data is not large enough for statistical methods to filter out the "noise" [Hajič and Honetschläger, 2003]; 2. they are indispensable to many NLP applications that rely on accurate description of language phenomena, e.g. word sense disambiguation, data mining, language data visualisation and machine translation.

Lexicons created with NLP applications in mind include FrameNet 1.5 [2010]; NomBank 1.0 [2008] and PDT-Vallex [2006]. All three are connected with a project of corpus annotation: FrameNet is based on the *British National Corpus*[3]; NomBank uses the the Wall Street Journal Corpus of the *Penn Treebank*[4]; and PDT-Vallex was created in order to bring consistency into the tectogrammatical annotation of the *Prague Dependency Treebank*[5]. For the latter two, the aim of the project was to cover all nouns, resp. all words with valency behaviour in the corpus. On the other hand, FrameNet annotation does not progress by words but by semantic frames. Some semantic frame is declared to be

[2] For each argument, a list of possible surface forms is given in the brackets.

[3] `http://www.natcorp.ox.ac.uk/`

[4] `http://www.cis.upenn.edu/~treebank/`

[5] `http://ufal.mff.cuni.cz/pdt2.0/`

Travel

Definition:

In this frame a Traveler goes on a journey, an activity, generally planned in advance, in which the Traveler moves from a Source location to a Goal along a Path or within an Area. The journey can be accompanied by Co_participants and Baggage. The Duration or Distance of the journey, both generally long, may also be described as may be the Mode_of_transportation. Words in this frame emphasize the whole process of getting from one place to another, rather than profiling merely the beginning or the end of the journey.

FEs:

Area [Area]
Semantic Type: Location
This is the Area in which the traveling takes place. This frame element describes the enclosed area inside which travelling, of unspecified Source, Path or Goal takes place.
We TRAVELLED in Europe.

Direction [dir]
Excludes: Area
The direction in which the Traveler goes.
They began their ODYSSEY north.

Lexical Units:

commute.v, excursion.n, expedition.n, getaway.n, jaunt.n, journey.n, journey.v, junket.n, odyssey.n, peregrination.n, pilgrimage.n, safari.n, tour.n, tour.v, traveler.n, travel.n, travel.v, trip.n, voyage.n, voyage.v

Figure 2. Parts of the Travel frame in FrameNet: definition, first two core FEs, list of lexical units

finished if all lexical units that the lexicographers have assigned to it have been created and annotated. However, other senses of the same words may be left unannotated; moreover, only a small number of corpus occurrences of each lexical unit are annotated.

Multi-purpose lexicons

Sometimes, electronically available data of a lexicon originally intended for human users can be turned into a valuable resource for NLP applications [Boguraev et al., 1987; Herbst and Uhrig, 2009]. The availability of the data in clearly structured data formats is crucial for NLP usage.

On the other hand, human users benefit from tools that convert machine readable data into browsable form.[6] In an ideal world, flexible visualisation and search tools would serve various kinds of human users, each according to their needs. In particular, we believe that the presentation to general users should be so self-contained that no prior knowledge would be necessary.

Organisation of the lexicon: semantic frames, word fields and derived words

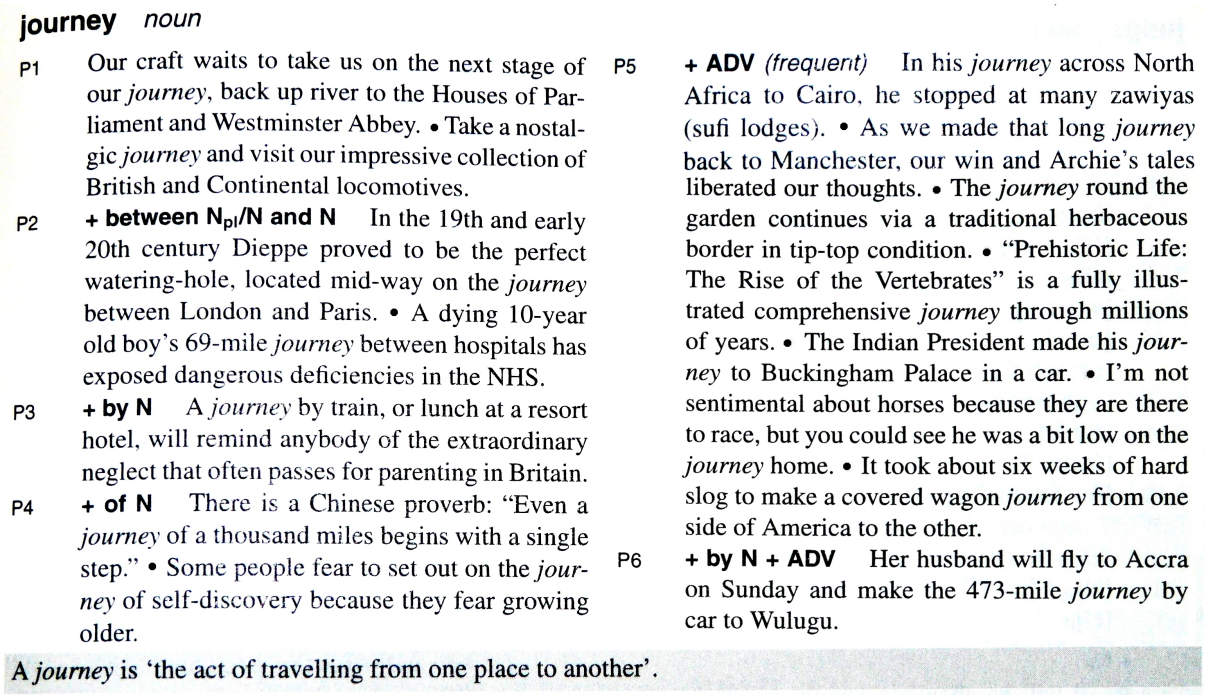
Most dictionaries, and valency lexicons are no exception, are organised so that entries are marked by their headwords and subdivided into "senses." The headwords are usually ordered alphabetically. However, valency is a syntacto-semantic phenomenon, and some regularities stand out more vividly when entries are grouped according to their semantic, or syntacto-semantic characteristics. In this section, we discuss examples of such groupings.

We have already mentioned that the creation of FrameNet proceeds from semantic frames to lexical units. A semantic frame is a schematic representation of a situation type (eating, spying, removing, classifying, etc.) together with a list of participants, propositions, and other conceptual roles that are seen as components of such situation. These participants are called frame elements. As we can see in Figure 2, the entry for each semantic frame contains a definition that characterizes the given semantic situation and the relationships between the most important frame elements, a list of frame elements with more detailed definition of each, and their relationships (e.g. if Direction is expressed, then Area is not expressed), and finally a list of lexical units that belong to this frame.[7]

Another lexicon which organises the entries into semantically based groups is the *Wörterbuch der Valenz etymologisch Vervandter Wörter* [Sommerfeldt and Schreiber, 1996]. The entries are divided into

[6] For example, FrameNet data is stored in XML files on the server; each XML file contains a link to its associated XSLT stylesheet which allows the client's browser to convert the data into a visually friendly report.

[7] There is one lexical entry for each lexical unit. We will discuss the structure of the lexical entries later.



journey *noun*

P1 Our craft waits to take us on the next stage of our *journey*, back up river to the Houses of Parliament and Westminster Abbey. • Take a nostalgic *journey* and visit our impressive collection of British and Continental locomotives.

P2 **+ between N_{pl}/N and N** In the 19th and early 20th century Dieppe proved to be the perfect watering-hole, located mid-way on the *journey* between London and Paris. • A dying 10-year old boy's 69-mile *journey* between hospitals has exposed dangerous deficiencies in the NHS.

P3 **+ by N** A *journey* by train, or lunch at a resort hotel, will remind anybody of the extraordinary neglect that often passes for parenting in Britain.

P4 **+ of N** There is a Chinese proverb: "Even a *journey* of a thousand miles begins with a single step." • Some people fear to set out on the *journey* of self-discovery because they fear growing older.

P5 **+ ADV** *(frequent)* In his *journey* across North Africa to Cairo, he stopped at many zawiyas (sufi lodges). • As we made that long *journey* back to Manchester, our win and Archie's tales liberated our thoughts. • The *journey* round the garden continues via a traditional herbaceous border in tip-top condition. • "Prehistoric Life: The Rise of the Vertebrates" is a fully illustrated comprehensive *journey* through millions of years. • The Indian President made his *journey* to Buckingham Palace in a car. • I'm not sentimental about horses because they are there to race, but you could see he was a bit low on the *journey* home. • It took about six weeks of hard slog to make a covered wagon *journey* from one side of America to the other.

P6 **+ by N + ADV** Her husband will fly to Accra on Sunday and make the 473-mile *journey* by car to Wulugu.

A *journey* is 'the act of travelling from one place to another'.

Figure 3. Valency Dictionary of English: entry for the noun *journey*

thirteen "word fields" such as "locomotion" (the mover and the moved thing are identical), "transport" (someone or something is causing someone or something else to move), "change of ownership," "feelings" etc. In a short introductory passage about each field, the common characteristics such as the prevalent number of arguments or most common syntactic structures are described, then the words are classified into smaller groups according to further semantic criteria (eg. locomotion is divided into "general," "without auxiliary means: slow/quick," "with auxiliary means," "through water," "through air"). Finally, detailed entries of all the words in the word field are listed alphabetically; each entry comprises several etymologically related words. See Figure 4 for the entry of the words *reisen / einreisen / verreisen - Reisen / Reise* "to travel / to enter / to go on a journey - (the) travelling / (a) journey."

We consider this combined approach particularly fruitful: the division of words into word fields or semantic frames brings attention to the differences between the surface form of elements with the same or similar semantic role. On the other hand, the simultaneous presentation of etymologically related words shows the changes in argument structure (both as to the number, form and semantics) that take place during derivation.

Valency patterns and arguments

Obviously the most important part of a valency lexicon are the valency patterns.

Sometimes, the patterns are characterized purely by their surface form, as in the *Valency Dictionary of English* [Herbst et al., 2004] (see Figure 3). In this case, the different surface forms in patterns 2–5 in fact imply different semantic roles (the argument with preposition *by* is the means, with preposition *of* is the attribute, and the arguments with the preposition *between* as well as the adverbial expressions denote the direction or location in which the journey takes place). However, the user is expected to infer the information about the semantic roles of the arguments from the examples.

On the other hand, the NomBank lexicon presents the patterns as rolesets, which means they are purely semantically defined. For example, the roleset for the noun *journey* consists of two roles, the "traveller" and the "destination or path." How the roles are expressed in the surface structure of the sentences can only be seen from the annotated data.

We have already seen that in FrameNet, the arguments (here called the frame elements) are characterized by their semantic roles. The information about the surface form is, similarly as in NomBank, a result of the annotation process: the user may look up all combinations of frame elements that were found within the span of a single sentence during the annotation, together with their syntactic realizations such as "a prepositional phrase with preposition *in*," "definite null instantiation" (the argument

reisen / einreisen / verreisen – Reisen / Reise

Die Familie (a) reist an die Ostsee (b). Viele Polen (a) reisen nach Deutschland (b) ein. In diesem Jahr verreisen wir (a) ins Gebirge (b). Das Reisen in ferne Länder (b) ist meine liebste Freizeitbeschäftigung. Die Reise der Expedition (a) zum Basislager (b) verlief ohne Störungen.

1. 'allgemeine Fortbewegung auf ein Ziel', 'über eine größere Entfernung hinweg', 'mit einem Instrument (Verkehrsmittel)', 'von einem Ort an einen anderen', 'für eine längere Zeit'
2. a – Täter / Mensch /
   V: Sn; S: Sg/Sp (von)
   b – Richtung / Ding /
   V: Sp (von – über – nach, zu, in. . .); S: Sp (von – über – nach, zu. . .)
3. Die Verwandten / Nachbarn reisen / verreisen ans Meer. Sie reisen in den Süden. Immer mehr Touristen reisen nach Deutschland ein. Das Reisen mit dem Flugzeug nimmt zu. Die Reise zum Nordkap war ein einmaliges Erlebnis.

Figure 4. [Sommerfeldt and Schreiber, 1996], entry for *reisen* "to travel" and its etymologically related words

did not appear in the sentence, but its value was clear from the context).

However, there is more to the semantics of the arguments than just semantic roles. This can be seen in part 2 of the entry in the *Wörterbuch der Valenz etymologisch Vervandter Wörter* [Sommerfeldt and Schreiber, 1996] (Figure 4). In this case, the semantic roles are *Täter* "actor" and *Richtung* "direction," which corresponds to the frame elements Traveller and Direction/Goal/Source/Area in FrameNet or to the roles of traveller and destination-or-path in the NomBank roleset. Besides that, there is the semantic requirement on the argument: the actor has to be a human, the direction is an object. Another example of a lexicon which lists the semantic requirements is the *Slovník vazeb a spojení* [Svozilová et al., 2005] (Figure 1), where the indefinite pronouns *někdo* "someone" and *něco* "something" mark the difference between animate and inanimate nominal arguments.

Corpus evidence for patterns. Examples

In FrameNet as well as in the *Valency Dictionary of English* [Herbst et al., 2004], only patterns that were actually found during corpus annotation are listed. This has the disadvantage that some more complex patterns may be left out not because they are ungrammatical, but because of lack of corpus evidence.

Example sentences and sentence fragments play an integral part of any valency lexicon. The *Wörterbuch der Valenz etymologisch Vervandter Wörter* [Sommerfeldt and Schreiber, 1996] (Figure 4) offers an interesting solution of the dilemma between illustrative examples made up by the lexicographers and corpus evidence: the first set of examples directly under the headword are made up so that each word appears with its full valency potential (all arguments are expressed in the same sentence). Section 3 of the entry then gives natural examples.

The use of made up examples may also reflect the findings of Opavská [2002] that two thirds of general users prefer examples in the form of short phrases to full sentences taken from the corpus.

Conclusion

Among the approaches to the creation of valency lexicons, we find the following ideas and strategies particularly useful:

- the availability of the data in an electronic form, with tools which can be adjusted to the needs of various kinds of users;
- the organisation of the entries into semantically and linguistically motivated groups,
- the inclusion of both semantic roles and semantic requirements,
- the listing of the surface forms that the arguments may take,
- the availability of real-life examples to end users and of simplified or made up examples for the needs of foreign learners and users who prefer short, compact entries.

References

Boguraev, B., Briscoe, T., Carroll, J., Carter, D., and Grover, C., The derivation of a grammatically indexed lexicon from the Longman Dictionary of Contemporary English, in *Proceedings of the 25th annual meeting on Association for Computational Linguistics*, ACL '87, pp. 193–200, Association for Computational Linguistics, Stroudsburg, PA, USA, 1987.

Bräunling, P., Umfrage zum Thema Valenzwörterbücher, *Lexikographica*, *5*, 168–177, 1989.

Daneš, F., Hlavsa, Z., Jirsová, A., Macháčková, E., Prouzová, H., and Svozilová, N., *Větné vzorce v češtině, 2. opravené vydání*, no. 23 in Studie a práce lingvistické, Academia, 1987.

Filipec, J., Daneš, F., Machač, J., and Mejstřík, V., *Slovník spisovné češtiny pro školu a veřejnost, 2. upravené a doplněné vydání*, Academia, Praha, 1994.

FrameNet 1.5, URL `http://framenet.icsi.berkeley.edu/`, 2010.

Hajič, J. and Honetschläger, V., Annotation lexicons: Using the valency lexicon for tectogrammatical annotation, *Prague Bulletin of Mathematical Linguistics (PBML)*, 2003.

Helbig, G. and Schenkel, W., *Wörterbuch zur Valenz und Distribution deutscher Verben*, VEB Bibliographisches Institut, Leipzig, 1969.

Herbst, T. and Uhrig, P., Erlangen Valency Patternbank, URL `http://www.patternbank.uni-erlangen.de/cgi-bin/patternbank.cgi`, 2009.

Herbst, T., Heath, D., Roe, I. F., and Götz, D., *A valency dictionary of English: a corpus-based analysis of the complementation patterns of English verbs, nouns, and adjectives*, vol. 40 of *Topics in English Linguistics*, Walter de Gruyter, Berlin, New York, URL `http://books.google.com/books?id=HC6wUJeq6MUC&printsec=frontcover#v=onepage&q&f=false`, 2004.

Karlík, P., Hypotéza modifikované valenční teorie, *Slovo a slovesnost*, *61*, 170–189, 2000.

Kolářová, V., *Valence deverbativních substantiv v češtině (na materiálu substantiv s dativní valencí)*, Karolinum, Praha, 2010.

Lopatková, M., Žabokrtský, Z., and Kettnerová, V., *Valenční slovník českých sloves*, Karolinum, Praha, 2008.

NomBank 1.0, URL `http://nlp.cs.nyu.edu/meyers/NomBank.html`, 2008.

Opavská, Z., Postoje a preference uživatelů slovníku. K jednomu aspektu dotazníkovho průzkumu, in *Varia IX. Zborník materiálov zo IX. kolokvia mladých jazykovedcov*, pp. 87–96, Slovenská jazykovedná spoločnost pri SAV, Bratislava, URL `http://lexiko.ujc.cas.cz/index.php?page=14&idStudie=8`, 2002.

Panevová, J., Formy a funkce ve stavbě české věty, *Academia, Praha*, 1980.

PDT-Vallex, URL `http://ufal.mff.cuni.cz/pdt2.0/browse/visual-data/pdt-vallex/`, 2006.

Sgall, P., Valence jako jádro jazykového systému, *Slovo a slovesnost*, *67*, 163–179, 2006.

Sommerfeldt, K. E. and Schreiber, H., *Wörterbuch zur Valenz und Distribution deutscher Adjektive*, VEB Bibliographisches Institut, Leipzig, 1974.

Sommerfeldt, K. E. and Schreiber, H., *Wörterbuch zur Valenz und Distribution der Substantive*, VEB Bibliographisches Institut, Leipzig, 1977.

Sommerfeldt, K. E. and Schreiber, H., *Wörterbuch der Valenz etymologisch Vervandter Wörter: Verben, Adjektive, Substantive*, Max Niemeyer Verlag, Tübingen, 1996.

Svozilová, N., Prouzová, H., and Jirsová, A., *Slovník slovesných, substantivních a adjektivních vazeb a spojení*, Academia, Praha, 2005.

Tesnière, L., *Eléments de syntaxe structurale*, Libraire C. Klincksieck, 1959.

Čermáková, A., *Valence českých substantiv*, no. 9 in Studie z korpusové lingvistiky, Nakladatelství Lidové noviny, Praha, 2009.

J. Šafránková and J. Pavlů (editors)

WDS'11
Proceedings of Contributed Papers

PART I

Vydal
MATFYZPRESS
vydavatelství
Matematicko-fyzikální fakulty
Univerzity Karlovy
Ke Karlovu 3,
121 16 Praha 2
jako svou 378. publikaci

Z připravených předloh
vytisklo Reprostředisko UK MFF
Sokolovská 83, 186 75 Praha 8

Vydání první

Praha 2011

ISBN 978-80-7378-184-2