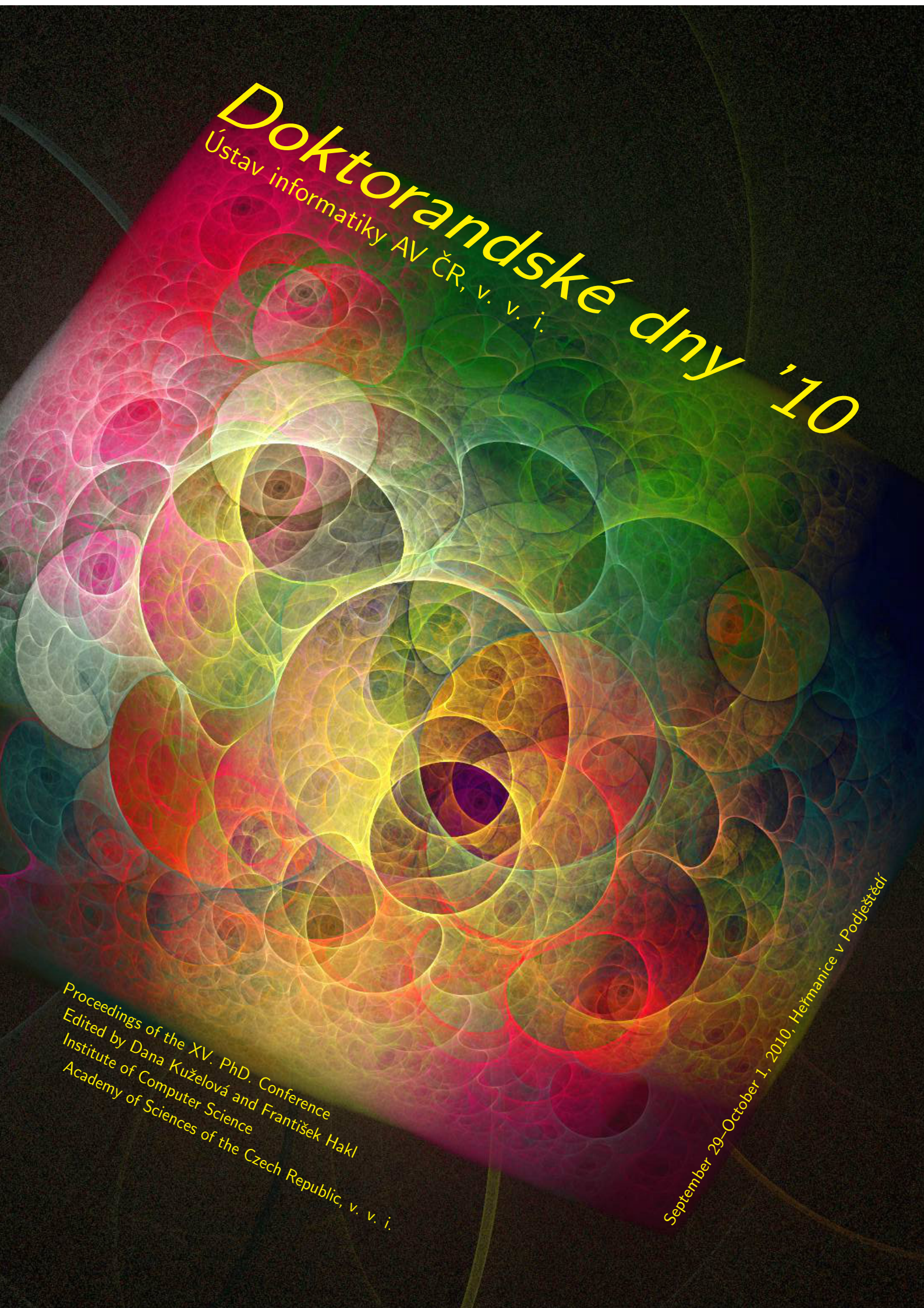
Doktorandské dny ’10

Kuželová, Dana
2010


Doktorandské dny '10
Ústav informatiky AV ČR, v. v. i.
Proceedings of the XV. PhD. Conference
Edited by Dana Kuželová and František Hakl
Institute of Computer Science
Academy of Sciences of the Czech Republic, v. v. i.
September 29–October 1, 2010, Heřmanice v Podještědí

Doktorandské dny ’10

Ústav informatiky AV ČR, v. v. i.

Heřmanice v Podještědí

29. září – 1. říjen 2010

vydavatelství Matematicko-fyzikální fakulty
Univerzity Karlovy v Praze

Ústav informatiky AV ČR, v. v. i., Pod Vodárenskou věží 2, 182 07 Praha 8

ISBN *– not yet –*

Doktorandské dny Ústavu informatiky AV ČR, v. v. i., se konají nepřetržitě od roku 1996 a poskytují doktorandům, podílejícím se na odborných aktivitách Ústavu informatiky, možnost prezentovat výsledky jejich odborného studia. Současně poskytuje prostor pro oponentní připomínky k přednášené tematice a použité metodologii práce ze strany přítomné odborné komunity.

Z jiného úhlu pohledu, toto setkání doktorandů podává průřezovou informaci o odborném rozsahu pedagogických aktivit, které jsou realizovány na pracovištích či za spoluúčasti Ústavu informatiky.

Jednotlivé příspěvky sborníku jsou uspořádány podle jmen autorů. Uspořádání podle tematického zaměření nepovažujeme vzhledem k rozmanitosti jednotlivých témat za účelné.

Vedení Ústavu informatiky jakožto organizátor doktorandských dnů věří, že toto setkání mladých doktorandů, jejich školitelů a ostatní odborné veřejnosti povede ke zkvalitnění celého procesu doktorandského studia zajišťovaného v součinnosti s Ústavem informatiky a v neposlední řadě k navázání a vyhledání nových odborných kontaktů.

1. září 2010

Obsah

Survey on Estimation of Distribution Algorithms with Emphasis on Copula-based Models

Post-Graduate Student:
MGR. LUKÁŠ BAJER

Faculty of Mathematics and Physics
Charles University in Prague
Malostranské náměstí 25
118 00 Prague 1, CZ

bajer@cs.cas.cz

Supervisor:
DOC. ING. RNDR. MARTIN HOLEŇA, CSC.

Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ

martin@cs.cas.cz

Field of Study:
Theoretical Computer Science

This work was supported by the Czech Science Foundation (GAČR), grant No. 201/09/H057.

Abstract

Evolutionary algorithms (EAs) have become one of the successful optimization methods during the last few decades, especially for problems where smooth optimization cannot be used. This work summarises present trends in Estimation of distribution algorithms (EDAs), a recent kind of EAs, and suggests a further research direction. These algorithms evolve a population of partial solutions, and instead of modifying the individuals by genetic operations, they construct the probability distribution of their variables and use it for sampling new population. Besides overview of the current EDAs for discrete and continuous parameters, different approaches to construction of probability distributions including the very recent usage of copulas are presented.

1. Introduction

Evolutionary algorithms (EAs) [1] have attained intense attention since 1980's. They belong to a large family of stochastic optimization methods called metaheuristics. These methods do not need any information about shape or smoothness of the function they optimize (functions being optimized by these algorithms are also called fitness or objective functions). Therefore, they are successfully applied especially in areas where smooth optimizations cannot be used. However, they are not guaranteed to find the optimum: usually, they only converge to some „near-optimal" solution.

This paper focuses on a novel kind of EAs: estimation of distribution algorithms (EDAs) [2]. They have many common aspects with the most popular EAs: genetic algorithms. Similarly to them, they evolve a set of promising candidate solutions, a population of individuals. During each step, which is also called generation, a new set of individuals is generated and a part or the whole former population is replaced according to some selection criterion.

Nevertheless, the new individuals are in EDAs generated differently. Instead of genetic operators like crossover and mutation, EDAs estimate the probability distribution of the most promising solutions, and new populations are obtained by random sampling from this distribution. The current paper summarizes different kinds of EDAs and models for estimating the probability distributions as well as possible future research directions.

The paper is divided in five sections. In the next section, the general concept of EDAs is presented. The third and fourth sections give a brief overview of the different variants of EDAs for discrete and continuous domains respectively, and the last section suggests future development in this field with a special emphasis on using copulas for expressing the joint probability distribution.

2. Basic principles of EDAs

As was already stated above, the rough structure of both the EAs and EDAs are similar. The general pseudo-code of the estimation of distribution algorithms is outlined in Fig. 1. Here, steps (1), (2) and (3) are the same as in many evolutionary algorithms while steps (4) and (5) are typical particularly for EDAs.

The main difference between EDAs and EAs lies in the method how they generate new individuals according to the previous generation. Whereas traditional EAs, for example genetic algorithms, try to implicitly combine

building blocks representing promising parts of genetic code of already found good solutions by genetic operations (crossover, mutation) [3], EDAs try to find correlations among variables in an explicit way.

(1) $P_0 \leftarrow$ randomly generate m individuals
(2) **for** $k = 1, 2, \ldots$ until a stopping criterion is met **do**
(3) $\quad pool \leftarrow$ select $n \leq m$ individuals from P_{k-1}
. $\quad$ according to the selection method
(4) $\quad p_l(\mathbf{x}) = p(\mathbf{x} \mid pool) \leftarrow$ estimate the probability
. $\quad$ distribution of an individual based on
. $\quad$ the selected individuals (in $pool$)
(5) $\quad P_l \leftarrow$ sample new population from $p_l(\mathbf{x})$
(6) **end for**

Figure 1: Estimation of distribution algorithm

These algorithms estimate the probabilistic distribution of the input variables of solutions. In the following text, the term *model* will represent a formal framework for estimating the joint probability distribution of individuals. Having this model, generating of individuals is relatively easy. However, estimating of the distribution with the model is often a bottleneck of EDAs; especially when the problem being solved is hard and complex dependencies among variables have to be determined.

2.1. Probabilistic graphical models

The majority of present EDAs estimate the probability distribution with probabilistic graphical models [2,4,5]. These models make use of a directed acyclic graph (DAG) S (see Fig. 2 for an example). Each node corresponds to one input variable X_i, and the arcs define dependencies between variables: for each node X_i and the set of its parents $\mathbf{Pa}_i^S = \{X_j \mid X_j \to X_i \text{ is an arc}\}$, the variable X_i and its non-descendants $nde(X_i) = \{X_j : \neg\exists \text{ oriented path } X_i \to X_j\}$ are conditionally independent given the parents $\mathbf{Pa}_i^S$.

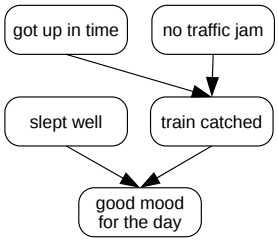


Figure 2: Structure of a Bayesian net

Further, the models consist of a set of unconditional probabilities for all root nodes of S

$$\rho(X_i = x_i \mid \emptyset, \boldsymbol{\theta}_i) \tag{1}$$

and a set of conditional probabilities for other nodes

$$\rho(X_i = x_i \mid \mathbf{pa}_i^S, \boldsymbol{\theta}_i). \tag{2}$$

Here, ρ denotes *generalized probability distribution* which stands for *mass probability* $p(X_i = x_i^k)$ for discrete random variables $X_i \in \{x_i^k \mid k = 1, \ldots, m\}$, and *density function* $f(x_i)$ for continuous X_i. The distributions in (2) are given by a set of values of X_i's parents $\mathbf{pa}_i^S$, and $\boldsymbol{\theta}_i$ is a finite set of parameters of the generalized probability distribution.

From the conditional (in)dependence defined by the structre S, the factorization of the joint probability distribution of the variables can be expressed as

$$\rho(x_1, \ldots, x_n \mid \boldsymbol{\theta}_S) = \prod_{i=1}^{n} \rho(x_i \mid \mathbf{pa}_i^S, \boldsymbol{\theta}_i). \tag{3}$$

The most frequent representatives of probabilistic graphical models are Bayesian networks for discrete variables and Gaussian networks for continuous variables. While in case of Bayesian networks the joint probability distribution can be written almost identically as in (3): $p(x_1, \ldots, x_n \mid \boldsymbol{\theta}_S) = \prod_{i=1}^{n} p(x_i \mid \mathbf{pa}_i^S, \boldsymbol{\theta}_i)$, Gaussian networks use the density function of normal distribution with nontrivial parameters

$$\begin{gathered} f(x_1, \ldots, x_n \mid \boldsymbol{\theta}_S) = \textstyle\prod_{i=1}^{n} \phi(x_i) \\ \phi(x_i) \sim N(m_i + \textstyle\sum_{x_j \in \mathbf{Pa}_i} b_{ji}(x_j - m_j), v_i). \end{gathered} \tag{4}$$

The parameter m_i denotes unconditional mean of X_i, b_{ji} is a linear coefficient reflecting the strength of relationship between variables X_j and X_i, and v_i is the variance of X_i given $\mathbf{Pa}_i$.

3. EDAs in discrete domain

The work of Mühlenbein [6] should be considered as one of the founders of the estimation of distribution algorithms. In their work, later improved in [7], they introduced *Univariate Marginal Distribution Algorithm* (**UMDA**) which conforms the outline of EDAs in Fig. 1. This algorithm uses the simplest estimation of the joint probability distribution: it does not consider any dependencies between variables. The estimates of univariate marginal distributions are taken as

$$p(x_i) = \frac{\#\text{ individuals in } pool \text{ with } X_i = x_i}{\#\text{ all individuals in } pool},$$

and thus the joint distribution is

$$p(x_1, \ldots, x_n) = p(x_1, \ldots, x_n \mid pool) = \prod_{i=1}^{n} p(x_i). \tag{5}$$

De Bonet [8] proposed *Mutual-Information-Maximizing Input Clustering* (**MIMIC**) algorithm. Taking into account how costly building of arbitrarily complex dependency graph is, he consider only dependencies between pairs of variables. His algorithm uses greedy-search for the best permutation of input variables and builds a path-like dependency graph. The joint probability distribution estimate is factorized as

$$p(x_1,\dots,x_n)_\pi = \\ = p(x_{i_1}|x_{i_2})\cdot p(x_{i_2}|x_{i_3})\cdots p(x_{i_{n-1}}|x_{i_n})\cdot p(x_{i_n})$$

where $\pi = (i_1,\dots,i_n)$ is the permutation of variables found by the algorithm.

As a measure for choosing the most appropriate single-conditioned probability, Kullback–Leibler divergence [9] between two probabilistic distributions was used in [8].

The previous algorithms are able to sufficiently optimize problems with relatively simple relations between variables. However, even the example in Fig. 2 cannot be precisely described by the proposed models. Therefore, algorithms considering multiple dependencies were proposed. The *Estimation of Bayesian Networks Algorithm* (**EBNA**), developed in [10], starts with an arc–less DAG. Similarly to the MIMIC, this algorithm uses greedy search and adds arcs to the graph as long as a given measure of quality increases: the authors suggest Bayesian Information Criterion (BIC) [11] and K2 algorithm [12]. In addition, independence–detection PC algorithm [13] can be used, too.

Among other algorithms, one of the most interesting and most actively developed is *Bayesian Optimization Algorithm* (**BOA**) [14]. Similarly to the previous methods, this approach uses greedy search, but it employs other interesting features such as incorporating prior knowledge of the problem (if there is any) or restricting the number of possible parents of nodes to the given number k. In the work [15], the algorithm is extended for continuous domains (**rBOA**), and in [16] incremental version (**iBOA**) is described.

4. EDAs in continuous domain

Most of the algorithms for discrete domains have their continuous counterparts: *Univariate Marginal Distribution Algorithm for continuous domains* [17] (**UMDA**$_c$), continuous *Mutual-Information-Maximizing Input Clustering* algorithm (**MIMIC**$_c$), and a set of algorithms considering multiple dependencies (**EGNA**).

In the **UMDA**$_c$, the variables $X_1,\dots,X_n$ are considered independent. Finding of the best density estimation for each variable consists of two phases: first, the appropriate density function is chosen via a hypothesis test (normal or Student's distributions are taken most frequently). Next, the parameters of the densities are learnt as a maximum likelihood estimates. Authors concentrate on a variant which uses only Gaussian distribution (**UMDA**$_c^G$). In this case, the joint probability density function can be expressed as

$$f(x_1,\dots,x_n;\boldsymbol{\theta}_S) = \textstyle\prod_{i=1}^n f(x_i;\boldsymbol{\theta}_S) = \\ = \textstyle\prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma_i^2}}\exp\left(-\frac{1}{2}\frac{(x_i-\mu_i)^2}{\sigma_i^2}\right). \qquad (6)$$

Bivariate densities with Gaussian distribution are taken into account by **MIMIC**$_c^G$. As its discrete analogy, the variables are assumed conditionally dependent on at most one predecessor, and the appropriate permutation of variables $\pi = (i_1,\dots,i_n)$ is found by greedy search.

Correspondingly to the discrete EBNA algorithms, *Estimation of Gaussian Networks Algorithms* (**EGNA**) take into account multiple dependencies among variables. The network structure of the probabilistic model can be learnt by different methods including edge–exclusion tests or penalized maximum likelihood.

5. Current issues in EDAs

A recent development in the field of estimation of distribution algorithms is performed especially in the following areas:

- *Mixture of discrete and continuous variables*. Most of the present models consider either discrete, or continuous dimensions of individuals, but not both. One of the few exceptions is work of Očenášek et al. [18] with their *Mixed Bayesian Optimization Algorithm* (**MBOA**) based on decision trees.
- *Combining different models*. One of the recent strategies in metaheuristic optimization is combining different approaches. Platel et al. [19] introduced *Quantum-inspired Evolutionary Algorithm* (**QEA**) which uses the whole population of probabilistic models of promising solutions.
- *Diversity maintenance*. Similarly to other evolutionary algorithms, preserving enough diversity within populations is crucial in order to avoid premature convergence to local optima. In the work [20], the concept of „rebels“ is suggested: individuals purposely being generated outside of the major part of the population.

- *Constraints satisfaction*. Although different techniques of constrained optimization was successfully developed in the area of genetic algorithms, we have not found any literature concerning satisfying constrains in the field of EDAs.
- *Copulas as a „probabilistic model"*. The concept of copulas is well-known among statisticians. A copula is a function which combines more univariate marginal distribution functions and forms one joint multivariate distribution. Wang et al. [21] use 2-dimensional copulas as a „probabilistic model" for EDAs, however, this integration is at the very beginning of the development and our aim is to develop this approach further.

6. Copulas as a probabilistic model for EDAs

More formally, the copula is a function $C : [0,1]^n \to [0,1]$ satisfying following conditions:

- $C(x_1, \ldots, x_n) = 0$ whenever $\exists i : x_i = 0$,
- $C(x_1, \ldots, x_n) = x_j$ whenever $\forall i \neq j : x_i = 1$, and
- $C(x_1, \ldots, x_n)$ is n-increasing (see [22] for details).

Especially from the second condition follows that all the copula function have uniformly distributed marginals. Typical example of copula is represented on Fig. 3.

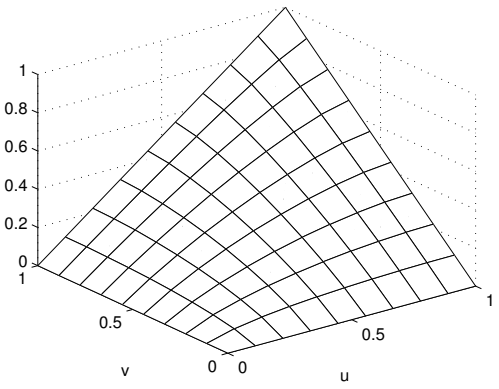


Figure 3: Gaussian copula function

The important result of the Sklar's theorem [22] is that for any given joint distribution function $H(x_1, \ldots, x_n)$ with marginals $F_1(x_1), \ldots, F_n(x_n)$, there exists an n-copula C such that for all $(x_1, \ldots, x_n) \in [\mathbb{R} \cup \{-\infty, \infty\}]^n$

$$H(x_1, \ldots, x_n) = C(F_1(x_1), \ldots, F_n(x_n)). \quad (7)$$

Expressing or estimating marginal distributions $F_1(x_1), \ldots, F_n(x_n)$ from data is easy. However, as the true distribution function H is usually unknown and the Sklar's theorem gives only existence of the copula C, the correct variant of the copula function and its parameters have to be estimated.

6.1. 2-dimensional Gaussian copulas

Gaussian copulas belong to the most well-known kinds of copulas. They attained their attention, for example, in financial sector as a mean of modelling risks [23], although the true contribution in this area is disputable [24].

2-dimensional versions of these copulas combine two (univariate) inverse normal cumulative distribution functions (CDF) Φ^{-1} according to Sklar's theorem using bivariate normal CDF Φ_ρ with Pearson's product moment correlation coefficient ρ

$$C(u, v; \rho) = \Phi_\rho \left(\Phi^{-1}(u), \Phi^{-1}(v) \right). \quad (8)$$

Using copulas as a probabilistic model for EDAs requires a method for generating individuals from copula function which represents the true joint distribution function. Nelsen [22] describes the conditional distribution method which is also used in [21].

First, a conditional distribution function for the second variable V given the first variable U, denoted $c_u(v)$, is needed

$$c_u(v) = P[V \leq v \mid U = u] = \frac{\partial C(u, v)}{\partial u}. \quad (9)$$

Then, the process of generating individuals is as follows [22]

1. generate two independent numbers from uniform distribution $u, t \sim U(0, 1)$
2. $v \leftarrow c_u^{(-1)}(t)$, $c_u^{(-1)}(t)$ is a quasi-inverse of c_u
3. take the desired pair (x, y): use quasi-inverses $x = F^{(-1)}(u)$, $y = G^{(-1)}(v)$ of the marginal distribution functions F, G

The very preliminary tests were performed in Matlab environment using Mateda toolbox [25] implementing

EDAs. However, implementation of the copula-based model appeared not to be as straightforward as was supposed; therefore, the concrete results will appear in the following works.

7. Conclusion

Overview of the main types of estimation of distribution algorithms were provided in this paper. A special emphasis was given on different kinds of probabilistic models, which are the crucial part of these algorithms. Several particular EDAs were briefly described, both for the discrete and continuous problems, and the final part of the paper discusses the latest issues from the field of EDAs as well as possible focus of the future research: using copulas in connection with EDAs.

References

[1] K. Deb, *Multi-objective optimization using evolutionary algorithms*. Wiley, 2001.

[2] P. Larrañaga and J.A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*. Kluwer, 2002.

[3] D.E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Jan. 1989.

[4] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[5] S.L. Lauritzen, *Graphical models*. Oxford University Press, 1996.

[6] H. Mühlenbein and G. Paass, "From recombination of genes to the estimation of distributions I. binary parameters," in *Parallel Problem Solving from Nature IV*, pp. 178–187, 1996.

[7] H. Mühlenbein, "The equation for response to selection and its use for prediction," *Evolutionary Computation*, vol. 5, no. 3, pp. 303–346, 1997.

[8] J.S. De Bonet, C.L. Isbell, and P. Viola, "MIMIC: finding optima by estimating probability densities," *Advances in neural information processing systems*, pp. 424–430, 1997.

[9] S. Kullback and R.A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, pp. 79–86, 1951.

[10] R. Etxeberria and P. Larrañaga, "Global optimization with bayesian networks," in *Second Symposium on Artificial Intelligence (CIMAF-99)*, pp. 332–339, 1999.

[11] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[12] G.F. Cooper and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, pp. 309–347, Oct. 1992.

[13] P. Spirtes and C. Glymour, "An algorithm for fast recovery of sparse causal graphs," *Social Science Computer Review*, vol. 9, no. 1, p. 62, 1991.

[14] M. Pelikan, D.E. Goldberg, and E. Cauntú-Paz, "BOA: the bayesian optimization algorithm," in *Proc. of the Genetic and Evolutionary Computation Conference (GECCO–99)*, pp. 525–532, 1999.

[15] C.W. Ahn, R. Ramakrishna, and D.E. Goldberg, "Real-Coded bayesian optimization algorithm: Bringing the strength of BOA into the continuous world," in *Genetic and Evolutionary Computation (GECCO–2004)*, pp. 840–851, 2004.

[16] M. Pelikan, K. Sastry, and D.E. Goldberg, "iBOA: the incremental bayesian optimization algorithm," *Proceedings of the 10th annual Genetic and Evolutionary Computation Conference (GECCO–2008)*, 2008.

[17] P. Larrañaga, R. Etxeberria, J.A. Lozano, and J.M. Pena, "Optimization by learning and simulation of bayesian and gaussian networks," Tech. Rep. EHU-KZAAIK-4, University of the Basque Country, 1999.

[18] J. Očenášek and J. Schwarz, "Estimation distribution algorithm for mixed continuous-discrete optimization problems," in *Intelligent technologies – theory and applications* (P. Sinčák, ed.), pp. 227–232, IOS Press, Slovakia, 2002.

[19] M.D. Platel, S. Schliebs, and N. Kasabov, "Quantum-inspired evolutionary algorithm: A multimodel EDA," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 6, pp. 1218–1232, 2009.

[20] L. delaOssa, J.A. Gamez, J.L. Mateo, and J.M. Puerta, "Avoiding premature convergence in estimation of distribution algorithms," in *2009 IEEE Congress on Evolutionary Computation*, (Trondheim, Norway), pp. 455–462, 2009.

[21] L.F. Wang, J.C. Zeng, and Y. Hong, "Estimation of distribution algorithm based on copula theory," in *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, pp. 1057–1063, 2009.

[22] R.B. Nelsen, *An Introduction to Copulas*. Springer Verlag, 2006.

[23] D. Li, "On Default Correlation: A Copula Function Approach," *Journal of Fixed Income*, vol. 9, no. 4, pp. 43–54, 2000.

[24] A. Lipton and A. Rennie, *Credit correlation: life after copulas*. World Scientific, 2007.

[25] R. Santana, C. Echegoyen, A. Mendiburu, C. Bielza, J.A. Lozano, P. Larranaga, R. Armananzas, and S. Shakya, "MATEDA: a suite of EDA programs in matlab," Tech. Rep. EHU-KZAA-IK-2/09, University of the Basque Country, Feb. 2009.

Reliability of Predictions in Regression Models

Post-Graduate Student:
ING. RADIM DEMUT

Department of Mathematics
Faculty of Nuclear Science and Physical Engineering
Czech Technical University
Trojanova 13

120 00 Prague 2, CZ

demut@seznam.cz

Supervisor:
DOC. ING. RNDR. MARTIN HOLEŇA, CSC.

Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2

182 07 Prague 8, CZ

martin@cs.cas.cz

Field of Study:
Mathematical Engineering

Abstract

This paper is concerned with the reliability of individual predictions in regression. For this purpose we describe conformal predictors and some methods for estimating the reliability of individual predictions such as sensitivity analysis or local modeling of prediction error. Finally we introduce confidence intervals in regression models.

1. Introduction

This paper is concerned with the reliability of predictions in regression models. In the first section we are interested in conformal predictors, which for every confidence level $1-\varepsilon$ output a prediction set. The conformal predictors should be valid in the sense that in the long run the frequency of error does not exceed ε at each confidence level $1-\varepsilon$ and the prediction set is as small as possible. The second chapter describes different approaches to estimate the reliability of individual predictions in regression such as sensitivity analysis or local modeling of prediction error. The third chapter deals with confidence intervals in regression models.

2. Conformal prediction

We assume that we have successive pairs

$$(x_1, y_1), (x_2, y_2), \ldots, \tag{1}$$

called examples. Each example (x_i, y_i) consists of an object x_i and its label y_i. The objects are elements of a measurable space $\mathbf{X}$ called the object space and the labels are elements of a measurable space $\mathbf{Y}$ called the label space. Moreover we assume that $\mathbf{X}$ is non-empty and that the σ-algebra on $\mathbf{Y}$ is different from $\{\emptyset, \mathbf{Y}\}$. We denote $z_i := (x_i, y_i)$ and we set

$$\mathbf{Z} := \mathbf{X} \times \mathbf{Y} \tag{2}$$

and call $\mathbf{Z}$ the example space. Thus the infinite data sequence (1) is an element of the measurable space $\mathbf{Z}^\infty$.

Our standard assumption is that the examples are chosen independently from some probability distribution Q on $\mathbf{Z}$, that means the infinite data sequence (1) is drawn from the power probability distribution Q^∞ on $\mathbf{Z}^\infty$. Usually we need only slightly weaker assumption that the infinite data sequence (1) is drawn from a distribution P on $\mathbf{Z}^\infty$ that is exchangeable, that means that for every $n \in \mathbb{N}$, every permutation π of $\{1, \ldots, n\}$, and every measurable set $E \subseteq \mathbf{Z}^\infty$ hold

$$P\{(z_1, z_2, \ldots) \in \mathbf{Z}^\infty : (z_1, \ldots, z_n) \in E\} =$$
$$P\{(z_1, z_2, \ldots) \in \mathbf{Z}^\infty : (z_{\pi(1)}, \ldots, z_{\pi(n)}) \in E\}$$

We denote $\mathbf{Z}^*$ the set of all finite sequences of elements of $\mathbf{Z}$, $\mathbf{Z}^n$ the set of all sequences of elements of $\mathbf{Z}$ of length n. The order in which old examples appear should not make any difference. In order to formalize this point we need the concept of a bag. A bag of size $n \in \mathbb{N}$ is a collection of n elements some of which may be identical. To identify a bag we must say what elements it contains and how many times each of these elements is repeated. We write $\backslash z_1, \ldots, z_n /$ for the bag consisting of elements $z_1, \ldots, z_n$, some of which may be identical with each other. We write $\mathbf{Z}^{(n)}$ for the set of all bags of size n of elements of a measurable space $\mathbf{Z}$. The set $\mathbf{Z}^{(n)}$ is itself a measurable space. It can be defined formally as the power space $\mathbf{Z}^n$ with a nonstandard σ-algebra, consisting of measurable subsets of $\mathbf{Z}^n$ that contain all permutations of their elements. We write $\mathbf{Z}^{(*)}$ for the set of all bags of elements of $\mathbf{Z}$.

2.1. Confidence predictors

We assume that at the nth trial we have firstly only the object x_n and only later we get the label y_n. If we simply want to predict y_n, then we need a function

$$D : \mathbf{Z}^* \times \mathbf{X} \to \mathbf{Y}. \tag{3}$$

We call such a function a simple predictor, always assuming it is measurable. For any sequence of old examples $x_1, y_1, \ldots, x_{n-1}, y_{n-1} \in \mathbf{Z}^*$ and any new object x_n, it gives $D(x_1, y_1, \ldots, x_{n-1}, y_{n-1}, x_n) \in \mathbf{Y}$ as its prediction for the new label y_n.

Instead of merely choosing a single element of $\mathbf{Y}$ as our prediction for y_n, we want to give subsets of $\mathbf{Y}$ large enough that we can be confident that y_n will fall in them, while also giving smaller subsets in which we are less confident. An algorithm that predicts in this sense requires additional input $\varepsilon \in (0, 1)$, which we call significance level, the complementary value $1 - \varepsilon$ is called confidence level. Given all these inputs

$$x_1, y_1, \ldots, x_{n-1}, y_{n-1}, x_n, \varepsilon \tag{4}$$

an algorithm Γ that interests us outputs a subset

$$\Gamma^\varepsilon(x_1, y_1, \ldots, x_{n-1}, y_{n-1}, x_n) \tag{5}$$

of $\mathbf{Y}$. We require this subset to shrink as ε is increased that means it holds

$$\begin{aligned}\Gamma^{\varepsilon_1}(x_1, y_1, \ldots, x_{n-1}, y_{n-1}, x_n) \subseteq \\ \Gamma^{\varepsilon_2}(x_1, y_1, \ldots, x_{n-1}, y_{n-1}, x_n)\end{aligned} \tag{6}$$

whenever $\varepsilon_1 \geq \varepsilon_2$.

Formally, we call a measurable function

$$\Gamma : \mathbf{Z}^* \times \mathbf{X} \times (0, 1) \to 2^{\mathbf{Y}} \tag{7}$$

that satisfies (6) for all $n \in \mathbb{N}$, all incomplete data sequences $x_1, y_1, \ldots, x_{n-1}, y_{n-1}, x_n$ and all significance levels $\varepsilon_1 \geq \varepsilon_2$ a confidence predictor.

We now introduce a formal notation for the errors Γ makes when it processes the data sequence

$$\omega = (x_1, y_1, x_2, y_2, \ldots) \tag{8}$$

at significance level ε. Whether Γ makes an error on the nth trial can be represented by a number that is one in case of an error and zero in case of no error

$$\mathrm{err}_n^\varepsilon(\Gamma, \omega) := \begin{cases} 1 & \text{if } y_n \notin \Gamma^\varepsilon(x_1, y_1, \ldots, \\ & x_{n-1}, y_{n-1}, x_n), \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

and the number of errors during the first n trials is

$$\mathrm{Err}_n^\varepsilon(\Gamma, \omega) := \sum_{i=1}^{n} \mathrm{err}_i^\varepsilon(\Gamma, \omega) \tag{10}$$

If ω is drawn from an exchangeable probability distribution P, the number $\mathrm{err}_n^\varepsilon(\Gamma, \omega)$ is the realized value of a random variable , which we may designate $\mathrm{err}_n^\varepsilon(\Gamma, P)$. We say that confidence predictor is exactly valid if for each ε

$$\mathrm{err}_1^\varepsilon(\Gamma, P), \mathrm{err}_2^\varepsilon(\Gamma, P), \ldots \tag{11}$$

is a sequence of independent Bernoulli random variables with parameter ε.

The confidence predictor Γ is conservatively valid if for any exchangeable probability distribution P on $\mathbf{Z}^\infty$ there exists a probability space with two families

$$(\xi_n^{(\varepsilon)} : \varepsilon \in (0, 1), n = 1, 2, \ldots) \tag{12}$$

and

$$(\eta_n^{(\varepsilon)} : \varepsilon \in (0, 1), n = 1, 2, \ldots) \tag{13}$$

of $\{0, 1\}$-valued variables such that

- for a fixed $\varepsilon, \xi_1^{(\varepsilon)}, \xi_2^{(\varepsilon)}, \ldots$ is a sequence of independent Bernoulli random variables with parameter ε;
- for all n and ε, $\eta_n^{(\varepsilon)} \leq \xi_n^{(\varepsilon)}$;
- the joint distribution of $\mathrm{err}_n^\varepsilon(\Gamma, P)$, $\varepsilon \in (0, 1)$, $n = 1, 2, \ldots$, coincides with the joint distribution of $\eta_n^{(\varepsilon)}$, $\varepsilon \in (0, 1)$, $n = 1, 2, \ldots$.

Randomized confidence predictor is a measurable function

$$\Gamma : (\mathbf{X} \times [0, 1] \times \mathbf{Y})^* \times (\mathbf{X} \times [0, 1]) \times (0, 1) \to 2^{\mathbf{Y}} \tag{14}$$

which, for all significance levels $\varepsilon_1 \geq \varepsilon_2$, all positive integer n, and all incomplete data sequences

$$x_1, \tau_1, y_1, \ldots, x_{n-1}, \tau_{n-1}, y_{n-1}, x_n, \tau_n, \tag{15}$$

where $x_i \in \mathbf{X}$, $\tau_i \in [0, 1]$ and $y_i \in \mathbf{Y}$ for all i satisfies

$$\begin{aligned}\Gamma^{\varepsilon_1}(x_1, \tau_1, y_1, \ldots, x_{n-1}, \tau_{n-1}, y_{n-1}, x_n, \tau_n) \subseteq \\ \Gamma^{\varepsilon_2}(x_1, \tau_1, y_1, \ldots, x_{n-1}, \tau_{n-1}, y_{n-1}, x_n, \tau_n).\end{aligned} \tag{16}$$

We will always assume that $\tau_1, \tau_2, \ldots$ are random numbers independently drawn from uniform distribution on $[0, 1]$. We define $\mathrm{err}_n^\varepsilon(\Gamma, \omega)$ by (9) with x_i now being extended objects $x_i \in \mathbf{X} \times [0, 1]$ and $\mathrm{Err}_n^\varepsilon(\Gamma, \omega)$ is defined by (10) as before.

2.2. Conformal predictors

A nonconformity measure is a measurable mapping

$$A : \mathbf{Z}^{(*)} \times \mathbf{Z} \to \overline{\mathbb{R}}. \tag{17}$$

To each possible bag of old examples and each possible new example, A assigns a numerical score indicating how different the new example is from the old ones. It is sometimes convenient to consider separately how a nonconformity measure deals with bags of different sizes. If A is a nonconformity measure, for each $n = 1, 2, \ldots$ we define the function

$$A_n : \mathbf{Z}^{(n-1)} \times \mathbf{Z} \to \overline{\mathbb{R}} \tag{18}$$

as the restriction of A to $\mathbf{Z}^{(n-1)} \times \mathbf{Z}$. The sequence $(A_n : n \in \mathbb{N})$, which we abbreviate to (A_n) will also be called a nonconformity measure.

Given a nonconformity measure (A_n) and a bag $\setminus z_1, \ldots, z_n/$ we can compute the nonconformity score

$$\alpha_i := A_n(\setminus z_1, \ldots, z_{i-1}, z_{i+1}, \ldots z_n/, z_i) \tag{19}$$

for each example z_i in the bag. Because a nonconformity measure (A_n) may be scaled however we like, the numerical value of α_i does not, by itself, tell us how unusual (A_n) finds z_i to be. For that we define p-value for z_i as

$$\frac{|\{j = 1, \ldots, n : \alpha_j \geq \alpha_i\}|}{n}. \tag{20}$$

The conformal predictor defined by a nonconformity measure (A_n) is the confidence predictor Γ obtained by setting

$$\Gamma^\varepsilon(x_1, y_1, \ldots, x_{n-1}, y_{n-1}, x_n) \tag{21}$$

equal to the set of all labels $y \in \mathbf{Y}$ such that

$$\frac{|\{i = 1, \ldots, n : \alpha_i \geq \alpha_n\}|}{n} > \varepsilon, \tag{22}$$

where

$$\begin{aligned} \alpha_i &:= A_n(\setminus(x_1, y_1), \ldots, (x_{i-1}, y_{i-1}), \\ &\quad (x_{i+1}, y_{i+1}), \ldots, (x_{n-1}, y_{n-1}), (x_n, y)/, \\ &\quad (x_i, y_i)), \quad \forall i = 1, \ldots, n-1, \\ \alpha_n &:= A_n(\setminus(x_1, y_1), \ldots, (x_{n-1}, y_{n-1})/, (x_n, y)) \end{aligned}$$

Proofs of the next two theorems can be found in [2].

Theorem 2.1 *All conformal predictors are conservative.*

The smoothed conformal predictor determined by the nonconformity measure (A_n) is a randomized confidence predictor Γ obtained by setting

$$\Gamma^\varepsilon(x_1, \tau_1, y_1, \ldots, x_{n-1}, \tau_{n-1} y_{n-1}, x_n, \tau_n) \tag{23}$$

equal to the set of all labels $y \in \mathbf{Y}$ such that

$$\frac{|\{i=1,\ldots,n:\alpha_i>\alpha_n\}|}{n} + \frac{\tau_n|\{i=1,\ldots,n:\alpha_i=\alpha_n\}|}{n} > \varepsilon, \tag{24}$$

where α_i are defined by (23). The left-hand side of (24) is called the smoothed p-value.

Theorem 2.2 *Any smoothed conformal predictor is exactly valid.*

If we are given a simple predictor (3) whose output does not depend on the order in which the old examples are presented, than the simple predictor D defines a prediction rule $D_{\setminus z_1,\ldots,z_n/} : \mathbf{X} \to \mathbf{Y}$ by the formula

$$D_{\setminus z_1,\ldots,z_n/}(x) := D(z_1, \ldots, z_n, x). \tag{25}$$

A natural measure of nonconformity of z_i is the deviation of the predicted label

$$\widehat{y}_i := D_{\setminus z_1,\ldots,z_n/}(x_i) \tag{26}$$

from the true label y_i. We can also use the deleted prediction defined as

$$\widehat{y}_{(i)} := D_{\setminus z_1,\ldots,z_{i-1},z_{i+1},\ldots,z_n/}(x_i). \tag{27}$$

More generally, the prediction rule $D_{\setminus z_1,\ldots,z_n/}$ may map $\mathbf{X}$ to some prediction space $\hat{\mathbf{Y}}$ not necessarily coinciding with $\mathbf{Y}$. An invariant simple predictor is a function D that maps each bag $\setminus z_1, \ldots, z_n/$ of each size n to a prediction rule $D_{\setminus z_1,\ldots,z_n/} : \mathbf{X} \to \hat{\mathbf{Y}}$ and such that the function

$$(\setminus z_1, \ldots, z_n/, x) \mapsto D_{\setminus z_1,\ldots,z_n/}(x) \tag{28}$$

of the type $Z^{(n)} \times \mathbf{X} \to \hat{\mathbf{Y}}$ is measurable for all n. A discrepancy measure is a measurable function $\Delta : \mathbf{Y} \times \hat{\mathbf{Y}} \to \mathbb{R}$. Given an invariant simple predictor D and discrepancy measure Δ we define functions A_n as follows: for any $((x_1, y_1), \ldots, (x_n, y_n)) \in \mathbf{Z}^*$, the values

$$\begin{aligned} \alpha_i = A_n(&\setminus(x_1, y_1), \ldots, (x_{i-1}, y_{i-1}), \\ &(x_{i+1}, y_{i+1}), \ldots, (x_n, y_n)/, (x_i, y_i)) \end{aligned} \tag{29}$$

are defined by the formula

$$\alpha_i := \Delta(y_i, D_{\setminus z_1,\ldots,z_n/}(x_i)) \tag{30}$$

or the formula

$$\alpha_i := \Delta(y_i, D_{\setminus z_1,\ldots,z_{i-1},z_{i+1},\ldots,z_n/}(x_i)). \tag{31}$$

It can be easily checked that in both cases A_n form a nonconformity measure.

3. Reliability estimates

In this chapter we are interested in different approaches to estimate the reliability of individual predictions in regression.

3.1. Sensitivity analysis

To estimate the reliability for a given example x, we compute the initial prediction K of the example x. Then we label x with $K + \varepsilon(l_{max} - l_{min})$, where ε is a sensitivity parameter, and l_{min} and l_{max} denote the lower and the upper label bounds of the learning examples, respectively. We add the new labeled x in the learning set. In the next step, a new sensitivity model is induced on the modified learning set and this model is used to compute a sensitivity prediction K_ε for the same particular example x. After computing different sensitivity predictions using different values of parameter $\varepsilon \in E$, where E is some set of positive real parameters, the predictions are combined into different reliability estimates. Sensitivity analysis - variance is defined as

$$\text{SEvar}(x) := \frac{\sum_{\varepsilon \in E}(K_\varepsilon - K_{-\varepsilon})}{|E|} \tag{32}$$

and sensitivity analysis - bias is defined as

$$\text{SEbias}(x) := \frac{\sum_{\varepsilon \in E}(K_\varepsilon - K) + (K_{-\varepsilon} - K)}{2|E|}. \tag{33}$$

3.2. Variance of a bagged model

We are given a learning set $L = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. We take repeated bootstrap samples $L^{(i)}, i = 1, \ldots, m$ from the learning set and induce a model on each of these samples. Each of the models yields a prediction $K_i, i = 1, \ldots, m$ for an example x. The label of the example x is predicted by averaging the individual predictions

$$K := \frac{\sum_{i=1}^{m} K_i}{m}. \tag{34}$$

We call this procedure bootstrap aggregating or bagging. The reliability estimate of a bagged model is defined as the prediction variance

$$\text{BAGV}(x) := \frac{1}{m}\sum_{i=1}^{m}(K_i - K)^2. \tag{35}$$

3.3. Local cross-validation reliability estimate

Suppose we are given an unlabeled example x for which we wish to compute the prediction and the local cross-validation (LCV) reliability estimate. We define the set of k nearest neighbors of x: $N = \{(x_1, C_1), \ldots, (x_k, C_k)\}$, where k is selected in advance. For each $(x_i, C_i) \in N$ we generate a model M_i on $N \setminus \{(x_i, C_i)\}$. Then we compute local leave-one-out (LOO) prediction K_i for example x_i using model M_i and we compute LOO error $E_i = |C_i - K_i|$. The LCV reliability estimate is computed as the weighted average of the nearest neighbors' local errors

$$\text{LCV}(x) := \frac{\sum_{(x_i,C_i)\in N} \frac{1}{d(x_i,x)} E_i}{\sum_{(x_i,C_i)\in N} \frac{1}{d(x_i,x)}}, \tag{36}$$

where d is some distance on the object space.

3.4. Local modeling of prediction error

Given a set of k nearest neighbors $N = \{(x_1, C_1), \ldots, (x_k, C_k)\}$, we define the estimate CNK ($C_{\text{Neighbors}} - K$) for an unlabeled example x as the difference between the average label of the nearest neighbors and the example's prediction K (using the model that was generated on all learning examples)

$$\text{CNK}(x) := \frac{\sum_{i=1}^{k} C_i}{k} - K. \tag{37}$$

3.5. Density-based reliability estimate

The density-based estimation of prediction error assumes that error is lower for predictions which are made in denser problem subspaces (a portion of the input space with a more learning examples), and higher for predictions which are made in sparser problem subspaces. But it has the disadvantage that it does not take into account the learning examples' labels. This causes the method to perform poorly with noisy data and in cases when distinct examples are not clearly separable. Given the learning set $L = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, the density estimate for unlabeled example x is defined as

$$p(x) := \frac{\sum_{i=1}^{n} \kappa(d(x, x_i))}{n} \tag{38}$$

where d denotes some distance on the object space and κ is a kernel function (for example the Gaussian). Since we expect the prediction error to be higher in cases when the density is lower, it means that $p(x)$ correlates negatively with the prediction error. To establish the positive correlation we define the reliability estimate as

$$\text{DENS}(x) := \max_{i=1,\ldots,n}(p(x_i)) - p(x). \tag{39}$$

4. Confidence intervals in regression models

4.1. General linear model

We define general linear model (GLM) as a regression model

$$y = \boldsymbol{X}\beta + \varepsilon, \tag{40}$$

where y is an n-dimensional response vector and ε is an n-dimensional error vector, whose components have zero means and common variance σ^2 and they are uncorrelated. The matrix $\boldsymbol{X}$ is an $n \times d$ model matrix and β is a d-dimensional vector of unknown regression coefficients. We can assume without restriction that the model matrix is nonsingular and that we have more than d observations.

4.1.1 Least squares estimate: The least squares estimate (LSE) of the regression parameter vector β is the value of β, which minimizes the expression

$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - x_i\beta)^2 = (y - \boldsymbol{X}\beta)^T (y - \boldsymbol{X}\beta), \quad (41)$$

where x_i is the i^{th} row of the model matrix. The LSE b can be obtained as the solution of the Gauss normal equation

$$\boldsymbol{X}^T\boldsymbol{X}\beta = \boldsymbol{X}^T y. \quad (42)$$

We assume that the columns of the model matrix are linearly independent, therefore we get the solution

$$b = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T y. \quad (43)$$

This solution minimizes the sum of squared errors, the mean vector of b is β and the variance matrix is $\text{var}(b) = \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}$. Moreover the LSE and the vector of residuals $e = y - \boldsymbol{X}b$ are uncorrelated.

Under the above-made assumptions $c^T b$ is the best linear unbiased estimator of the linear combination $c^T\beta$ of the regression coefficients.

4.1.2 Maximum likelihood estimate: Now we will assume that the components of the error vector are statistically independent and normally distributed with zero means and common unknown standard deviation σ. The log-likelihood function of the model is

$$l(\beta, \sigma; y) = -\frac{n}{2}\ln(2\pi) - n\ln(\sigma) - \frac{(y - \boldsymbol{X}\beta)^T(y - \boldsymbol{X}\beta)}{2\sigma^2}. \quad (44)$$

For a fixed β the minimum of the log-likelihood function with respect to σ is achieved at

$$\widetilde{\sigma}_\beta = \sqrt{\frac{(y - \boldsymbol{X}\beta)^T(y - \boldsymbol{X}\beta)}{n}}. \quad (45)$$

Inserting this into equation (44) we get

$$l(\beta, \widetilde{\sigma}_\beta; y) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln\left(\frac{(y - \boldsymbol{X}\beta)^T(y - \boldsymbol{X}\beta)}{n}\right) - \frac{n}{2}, \quad (46)$$

which is maximized at that value of β that minimizes the sum of squared errors.

We denote the maximum likelihood estimate (MLE) of the vector of regression coefficients $\widehat{\beta}$ and the MLE of the variance $\widehat{\sigma}$. The MLE $\widehat{\beta}$ has the multivariate normal distribution $N_d(\beta, \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1})$. The statistic $n\widehat{\sigma}^2/\sigma^2$ as a function of $y - \boldsymbol{X}\widehat{\beta}$ is statistically independent of $\widehat{\beta}$ and the distribution of $n\widehat{\sigma}^2/\sigma^2$ is χ^2-distribution with $n - d$ degrees of freedom.

In a GLM, the uniformly best α-level critical region for the linear hypothesis $H : c^T\beta = a$ with respect to the set of alternative hypothesis $H_A : c^T\beta > a$ is

$$t_H(y) = \frac{c^T\widehat{\beta} - a}{s\sqrt{c^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}c}} > t_{1-\alpha}[n - d], \quad (47)$$

where

$$s^2 := n\widehat{\sigma}^2/(n - d) \quad (48)$$

is the minimum variance unbiased estimate of the scale parameter (variance) and $t_\alpha[n - d]$ is the α^{th} quantile of the Student t-distribution with $n - d$ degrees of freedom. This test is also the likelihood ratio test. The likelihood ratio test of the hypothesis $H : c^T\beta = a$ with respect to the set of alternative hypothesis $H_A : c^T\beta \neq a$ is

$$|t_H(y)| = \frac{|c^T\widehat{\beta} - a|}{s\sqrt{c^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}c}} > t_{1-\alpha/2}[n - d]. \quad (49)$$

Let us assume that we are given a $q \times d$ matrix $\boldsymbol{A}$ with linearly independent rows and we want to compute the likelihood ratio test statistic of the hypothesis $H : \boldsymbol{A}\beta = a$ with respect to the set of alternative hypothesis $H : \boldsymbol{A}\beta \neq a$. It can be shown using the Lagrangean function that the maximum of the log-likelihood function (44) under the constraint $H : \boldsymbol{A}\beta = a$ is achieved in points

$$\widehat{\beta}_H = \widehat{\beta} - (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{A}^T(\boldsymbol{A}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{A}^T)^{-1}(\boldsymbol{A}\widehat{\beta} - a) \quad (50)$$

and

$$\widehat{\sigma}_H^2 = \frac{(y - \boldsymbol{X}\widehat{\beta}_H)^T(y - \boldsymbol{X}\widehat{\beta}_H)}{n} = \frac{SS_E}{n} + \frac{SS_H}{n}, \quad (51)$$

where SS_E is the residual sum of squares

$$SS_E := (y - \boldsymbol{X}\widehat{\beta})^T(y - \boldsymbol{X}\widehat{\beta}) = (n - d)s^2 \quad (52)$$

and SS_H is the hypothesis sum of squares

$$SS_H := (\boldsymbol{A}\widehat{\beta} - a)^T(\boldsymbol{A}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{A}^T)^{-1}(\boldsymbol{A}\widehat{\beta} - a). \quad (53)$$

Thus the maximum value of the log-likelihood function under the constraint $H : \boldsymbol{A}\beta = a$ is

$$l(\widehat{\beta}_H, \widehat{\sigma}_H; y) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln\left(\frac{SS_E}{n} + \frac{SS_H}{n}\right) - \frac{n}{2}. \quad (54)$$

The global maximum value of the log-likelihood function is

$$l(\widehat{\beta}, \widehat{\sigma}; y) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln\left(\frac{SS_E}{n}\right) - \frac{n}{2}. \quad (55)$$

Therefore the likelihood ratio statistic of the hypothesis $H : \boldsymbol{A}\beta = a$ which is defined as

$$w_H(y) = 2(l(\widehat{\beta}, \widehat{\sigma}; y) - l(\widehat{\beta}_H, \widehat{\sigma}_H; y)) \quad (56)$$

has the following form

$$w_H(y) = n\ln\left(1 + \frac{SS_H}{SS_E}\right). \quad (57)$$

We can write

$$w_H(y) = n\ln\left(1 + \frac{q}{n-d}F_H(y)\right), \quad (58)$$

where

$$F_H(y) = \frac{SS_H/q}{SS_E/(n-d)} = \frac{\left(\frac{SS_H}{\sigma^2}\right)/q}{\left(n\frac{\widehat{\sigma}^2}{\sigma^2}\right)/(n-d)}. \quad (59)$$

The sums of squares in (59) are statistically independent because the hypothesis sum of squares is a function of the MLE and the residual sum of squares is a function of residuals. It can be shown that SS_H/σ^2 has χ^2-distribution with q degrees of freedom. Moreover we know that the distribution of $n\widehat{\sigma}^2/\sigma^2$ is χ^2-distribution with $n-d$ degrees of freedom. Thus, under hypothesis $H : \boldsymbol{A}\beta = a$ the distribution of F_H is F-distribution with q numerator and $n-d$ denominator degrees of freedom.

Because the likelihood ratio statistic of the hypothesis $H : \boldsymbol{A}\beta = a$ is a monotone function of the F-statistic, the observed significance level for the hypothesis $H : \boldsymbol{A}\beta = a$ calculated from the observation y_{obs} and the model is equal to

$$\Pr(w_H(y) \geq w_H(y_{\text{obs}}); \beta, \sigma) = \Pr(F_H(y) \geq F_H(y_{\text{obs}}); \beta, \sigma) = 1 - F(F_H(y_{\text{obs}})), \quad (60)$$

where F is a distribution function of F-distribution with q numerator and $n-d$ denominator degrees of freedom.

From this follows that in general linear model the $(1-\alpha)$-level confidence region for a finite set of linear combinations $\psi = \boldsymbol{A}\beta$ of regression coefficients has the form

$$\{\psi : (\psi - \boldsymbol{A}\widehat{\beta})^T(\boldsymbol{A}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{A}^T)^{-1}(\psi - \boldsymbol{A}\widehat{\beta}) \leq qF_{1-\alpha}[q, n-d]s^2\}, \quad (61)$$

where $F_{1-\alpha}[q, n-d]$ is $(1-\alpha)^{\text{th}}$ quantile of F-distribution with q numerator and $n-d$ denominator degrees of freedom.

The $(1-\alpha)$-level confidence region for the whole regression coefficient vector has the form

$$\{\beta : (\beta - \widehat{\beta})^T\boldsymbol{X}^T\boldsymbol{X}(\beta - \widehat{\beta}) \leq dF_{1-\alpha}[d, n-d]s^2\} \quad (62)$$

and the $(1-\alpha)$-level confidence interval for the linear combination $\psi = c^T\beta$ has the form

$$\left\{\psi : \frac{(\psi - c^T\widehat{\beta})^2}{c^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}c} \leq F_{1-\alpha}[1, n-d]s^2\right\}. \quad (63)$$

4.2. Nonlinear regression model

We define nonlinear regression model as a regression model

$$y_i = f(x_i; \beta) + \varepsilon_i, \qquad \text{for all } i = 1, \ldots, n, \quad (64)$$

where the x_is contain values of explaining variables, f is a known function of explaining variables and vector β of the unknown regression parameters, and the ε_is have zero means and common variance σ^2 and they are uncorrelated. The unknown regression parameter vector β is assumed to belong to a given open subset B of $\mathbb{R}^d$.

4.2.1 Least squares estimate: The least squares estimate (LSE) of the regression parameter vector β is the value of β, which minimizes the expression

$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n}(y_i - f(x_i; \beta))^2 = (y - f(\boldsymbol{X}; \beta))^T(y - f(\boldsymbol{X}; \beta)), \quad (65)$$

where x_i is the i^{th} row of the model matrix and

$$f(\boldsymbol{X}; \beta) := (f(x_1; \beta), \ldots, f(x_n; \beta))^T. \quad (66)$$

The LSE b can be obtained as the solution of the Gauss normal equations

$$\widetilde{\boldsymbol{X}}(\beta)^T f(\boldsymbol{X}; \beta) = \widetilde{\boldsymbol{X}}(\beta)^T y, \quad (67)$$

where

$$\widetilde{\boldsymbol{X}}(\beta) := \frac{\partial f(\boldsymbol{X}; \beta)}{\partial \beta^T} = \left(\frac{\partial f(x_i; \beta)}{\partial \beta_j}\right) \quad (68)$$

is an $n \times d$ matrix with the i^{th} row containing partial derivatives of $f(x_i;\beta)$ with respect to the components of the regression parameter vector.

The normal equations are nonlinear with respect to the components of the parameter vector and normally need to be solved by some iterative method. One such method is the Gauss-Newton method, which is based on approximating the regression surface by the linear surface at the current iteration point β^0 that is

$$f(\boldsymbol{X};\beta) \approx f(\boldsymbol{X};\beta^0) + \widetilde{\boldsymbol{X}}(\beta^0)(\beta - \beta^0). \tag{69}$$

Normal equations have then the form

$$\widetilde{\boldsymbol{X}}(\beta^0)^T(f(\boldsymbol{X};\beta^0) + \widetilde{\boldsymbol{X}}(\beta^0)(\beta - \beta^0)) = \widetilde{\boldsymbol{X}}(\beta^0)^T y. \tag{70}$$

This gives the solution

$$\beta^1 = \beta^0 + (\widetilde{\boldsymbol{X}}(\beta^0)^T\widetilde{\boldsymbol{X}}(\beta^0))^{-1}\widetilde{\boldsymbol{X}}(\beta^0)^T(y - f(\boldsymbol{X};\beta^0)) \tag{71}$$

for the next iteration point.

Using the Gauss-Newton iteration scheme starting from the "true" value of the regression parameter vector β and taking only one step we get

$$b \approx \beta + \boldsymbol{W}(\beta)^{-1}\widetilde{\boldsymbol{X}}(\beta)^T\varepsilon, \tag{72}$$

where

$$\boldsymbol{W}(\beta) := \widetilde{\boldsymbol{X}}(\beta)^T\widetilde{\boldsymbol{X}}(\beta). \tag{73}$$

The mean vector of b is approximately β and the variance matrix is approximately $\sigma^2\boldsymbol{W}(\beta)^{-1}$. Moreover the LSE and the vector of residuals $e = y - f(\boldsymbol{X};\beta)$ are approximately uncorrelated.

4.2.2 Maximum likelihood estimate: Now we will assume that the components of the error vector are statistically independent and normally distributed with zero means and common unknown standard deviation σ. The log-likelihood function of the model is

$$l(\beta,\sigma;y) = -\frac{n}{2}\ln(2\pi) - n\ln(\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - f(x_i;\beta))^2. \tag{74}$$

For a fixed β the minimum of the log-likelihood function with respect to σ is achieved at

$$\widetilde{\sigma}_\beta = \sqrt{\frac{S(\beta)}{n}}, \tag{75}$$

where we denoted

$$S(\beta) := \sum_{i=1}^{n}(y_i - f(x_i;\beta))^2. \tag{76}$$

Inserting this into the log-likelihood function we get

$$l(\beta,\widetilde{\sigma}_\beta;y) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln\left(\frac{S(\beta)}{n}\right) - \frac{n}{2} \tag{77}$$

which is maximized at that value of β which minimizes the sum of squared errors. We denote the MLE of the vector of regression parameters $\widehat{\beta}$. Then the MLE of σ is

$$\widehat{\sigma} = \sqrt{\frac{S(\widehat{\beta})}{n}}. \tag{78}$$

The MLE $\widehat{\beta}$ has approximately d-dimensional normal distribution $N_d(\beta,\sigma^2\boldsymbol{W}(\beta)^{-1})$. The MLE $\widehat{\beta}$ and the vector of residuals $y - f(\boldsymbol{X};\widehat{\beta})$ are approximately statistically independent. The statistic $n\widehat{\sigma}^2/\sigma$ as a function of $y - f(\boldsymbol{X};\widehat{\beta})$ is approximately statistically independent of $\widehat{\beta}$ and has approximately χ^2-distribution with $n - d$ degrees of freedom.

Let us consider a submodel of a nonlinear regression model such that the regression parameter vector is constrained to belong to a subset of B defined by the condition $g(\beta) = 0$, where g is a given smooth q-dimensional vector valued function ($1 \le q \le d$). The likelihood ratio statistic of the statistical hypothesis $H : g(\beta) = 0$ with respect to the set of alternative hypothesis $H_A : g(\beta) \neq 0$ is

$$\begin{aligned} &2(l(\widehat{\beta},\widehat{\sigma};y_{\text{obs}}) - l(\widetilde{\beta},\widetilde{\sigma};y_{\text{obs}})) = \\ &-n\ln\left(\frac{S(\widehat{\beta})}{n}\right) + n\ln\left(\frac{S(\widetilde{\beta})}{n}\right) = \\ &n\ln\left(1 + \frac{S(\widetilde{\beta}) - S(\widehat{\beta})}{S(\widehat{\beta})}\right), \end{aligned} \tag{79}$$

where $\widetilde{\beta}$ and $\widetilde{\sigma}$ are the points in which is achieved the maximum of the log-likelihood function under the constraint $H : g(\beta) = 0$.

Let β^* denote the "true" value of the parameter vector satisfying $g(\beta^*) = 0$. Now if the approximation

$$g(\beta) \approx g(\beta^*) + \frac{\partial g(\beta^*)}{\partial \beta^T}(\beta - \beta^*) \tag{80}$$

is valid then the condition $g(\beta) = 0$ is approximately linear hypothesis on β. Also if the regression surface is well approximated by the tangent "plane" at β^* it follows that the distribution of

$$\frac{n-d}{q}\frac{S(\widetilde{\beta}) - S(\widehat{\beta})}{S(\widehat{\beta})} \tag{81}$$

is approximately F-distribution with q numerator and $n-d$ denominator degrees of freedom. That is the distribution of the likelihood ratio statistic (79) is approximately equal to the distribution of

$$n \ln \left(1 + \frac{q}{n-d} F(y)\right), \tag{82}$$

where $F(y)$ has the F-distribution with q numerator and $n-d$ denominator degrees of freedom.

In a nonlinear regression model the approximate $(1-\alpha)$-level profile likelihood-based confidence region for the regression parameters has the form

$$\left\{\beta : S(\beta) \leq n \left(1 + \frac{d}{n-d} F_{1-\alpha}[d, n-d]\right) \widehat{\sigma}^2\right\}, \tag{83}$$

where $F_{1-\alpha}[d, n-d]$ is $(1-\alpha)^{\text{th}}$ quantile of F-distribution with d numerator and $n-d$ denominator degrees of freedom.

The approximate $(1-\alpha)$-level profile likelihood-based confidence interval for the real-valued function $\psi = g(\beta)$ has the form

$$\left\{\psi : S(\widetilde{\beta}_\psi) \leq n \left(1 + \frac{1}{n-d} F_{1-\alpha}[1, n-d]\right) \widehat{\sigma}^2\right\}, \tag{84}$$

where $\widetilde{\beta}_\psi$ is the MLE of the regression parameters under the hypothesis $H : g(\beta) = \psi$.

5. Conclusion

We described some approaches to estimating the reliability of individual predictions in regression. In our future work we will compare these methods in a simulation study and we will try to use them on some real data.

References

[1] Z. Bosnic and I. Kononenko, "Comparison of Approaches for Estimating Reliability of Individual Regression Predictions", *Data & Knowledge Engineering*, 504-516, 2008.

[2] A. Gammerman, G. Shafer, and V. Vovk, "Algorithmic Learning in a Random World", *Springer Science+Business Media*, 2005.

[3] E. Uusipaikka, "Confidence Intervals in Generalized Regression Models", *Chapman & Hall*, 2009.

Verification of Hybrid Systems by Incremental Abstract Forward/Backward Computation

Post-Graduate Student:
MGR. TOMÁŠ DZETKULIČ
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
dzetkulic@cs.cas.cz

Supervisor:
ING. STEFAN RATSCHAN, PH.D.
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
ratschan@cs.cas.cz

Field of Study:
Verification of Hybrid Systems

This work has been supported by Czech Science Foundation grants 201/08/J020 and 201/09/H057, and by the institutional research plan AV0Z100300504 of the Czech Republic.
This paper will be presented at the International Workshop on Reachability Problems 2010 and will be published in the internal workshop proceedings.

Abstract

In this paper, we introduce a new approach to unbounded safety verification of hybrid systems with non-linear ordinary differential equations. It incrementally refines an abstraction of the system, but avoids increases in the size of the abstraction as much as possible, in order to avoid the usual blow-up problem of applications of counter-example guided abstraction refinement in a hybrid systems context.

1. Introduction

In this paper, we study hybrid (dynamical) systems, that is systems with both discrete and continuous state and evolution. We address the problem of unbounded safety verification of hybrid systems, that is, the verification that a given hybrid system does not have a trajectory (of unbounded length) from an initial state to a set that is considered unsafe. The traditional approach to solving this problem computes the set of reachable states of the system. If the intersection of this reach set with the set of unsafe states is empty, the safety property holds. This has several disadvantages: (1) When computing the reach set, information about the topology of the set of unsafe states is ignored. (2) Even over bounded time, exact reachability computation is possible only for very special cases, and hence (unlike for discrete systems) one has to use over-approximation. It is a-priori not clear, how much to over-approximate in order to prove a given property.

Hence, it is necessary, to compute several, incrementally tighter reach set over-approximations. However, the current approaches do not exploit information from one over-approximation to the next. Some approaches do exploit dual (forward/backward) or incrementally tighter reachability analyses [6, 8]. But, reuse of analyses only concerns dropping initial/unsafe states that have been shown not to lie on any error trajectory—no reuse is done concerning the analysis itself.

In order to avoid these problems, the hybrid systems community has tried to employ counter-example guided abstraction refinement techniques in the hybrid systems context [4, 5]. However, unlike in the discrete case, this has had only limited success in the hybrid case, since here the removal of one single counter-example at a time often already very early blows up the size of the abstraction.

Our previous approach [1] employs local reachability checking techniques on the hybrid system abstraction. This can be simulated in the method described in this paper by an extremely aggressive widening strategies which does not behave well for hybrid systems with cyclic behavior.

Computational experiments show the efficiency of the approach.

2. Hybrid Systems

In this section, we briefly recall our formalism for modeling hybrid systems. It captures many relevant classes of hybrid systems, and many other formalisms for hybrid systems in the literature are special cases of it. We use a set S to denote the discrete modes of a hybrid system, where S is finite and nonempty. $I_1, \ldots, I_k \subseteq \mathbb{R}$ are compact intervals over which the continuous variables

of a hybrid system range. Φ denotes the state space of a hybrid system, i.e., $\Phi = S \times I_1 \times \cdots \times I_k$.

Definition 1 *A* hybrid system H *is a tuple* $(\texttt{Flow}, \texttt{Jump}, \texttt{Init}, \texttt{Unsafe})$*, where* $\texttt{Flow} \subseteq \Phi \times \mathbb{R}^k$*,* $\texttt{Jump} \subseteq \Phi \times \Phi$*,* $\texttt{Init} \subseteq \Phi$*, and* $\texttt{Unsafe} \subseteq \Phi$*.*

Informally speaking, the predicate `Init` specifies the initial states of a hybrid system and `Unsafe` the set of unsafe states that should not be reachable from an initial state. The relation `Flow` specifies the possible continuous flow of the system by relating states with corresponding derivatives, and `Jump` specifies the possible discontinuous jumps by relating each state to a successor state. Formally, the behavior of H is defined as follows:

Definition 2 *A* flow *of length* $l \geq 0$ *in a mode* $s \in S$ *is a function* $r : [0, l] \to \Phi$ *such that the projection of* r *to its continuous part is differentiable and for all* $t \in [0, l]$*, the mode of* $r(t)$ *is* s*. A* trajectory *of* H *is a sequence of flows* $r_0, \ldots, r_p$ *of lengths* $l_0, \ldots, l_p$ *such that for all* $i \in \{0, \ldots, p\}$*,*

1. *if* $i > 0$ *then* $(r_{i-1}(l_{i-1}), r_i(0)) \in \texttt{Jump}$*, and*
2. *if* $l_i > 0$ *then* $(r_i(t), \dot{r}_i(t)) \in \texttt{Flow}$*, for all* $t \in [0, l_i]$*, where* $\dot{r}_i$ *is the derivative of the projection of* r_i *to its continuous component.*

A (concrete) error trajectory *of a hybrid system* H *is a trajectory* $r_0, \ldots, r_p$ *of* H *such that* $r_0(0) \in \texttt{Init}$ *and* $r_p(l) \in \texttt{Unsafe}$*, where* l *is the length of* r_p*.* H *is* safe *if it does not have an error trajectory.*

In the rest of the paper we will assume an arbitrary, but fixed hybrid system H. We will denote the set of its error trajectories by $\mathcal{E}$.

In practice one would also have to define some concrete syntax in which hybrid systems are described. However, this paper will be independent of concrete syntax. Instead, we will later require some operations that will provide information on the hybrid system at hand.

3. Incremental Abstract Forward/Backward Computation

The main shortcoming of the usual hybrid systems reachability algorithms is its lack of incrementality which is an especially pressing problem for systems with complex dynamics, because in that case even bounded time reach set computation necessarily involves over-approximation. In such cases we would like to first compute approximate information using high over-approximation, and incrementally refine this.

Our approach will be based on an incremental refinement of a covering of the hybrid systems state space by connected sets that we will call *regions*. In our case, the regions will be formed by pairs consisting of a mode and a Cartesian product of intervals (i.e., a *box*). Moreover, we will form the regions in such a way that no pair of regions with the same mode will have overlapping boxes. In theory the approach is also applicable to regions that have a different form.

The operations that we require on regions are the following:

- $\uplus$ s.t. $a_1 \cup a_2 \subseteq a_1 \uplus_b a_2$
- $\sqsubseteq$ s.t. $a_1 \sqsubseteq a_2$ implies $a_1 \subseteq a_2$

In our case of boxes, $a_1 \uplus a_2$ is the smallest boxes that includes both argument boxes a_1 and a_2 (i.e., box union), and $\sqsubseteq$ is the subset operation on boxes.

Definition 3 *An* abstraction *is a graph whose vertices (which we call* abstract states*) may be labeled with labels* `Init` *or* `Unsafe`*. Moreover, to each abstract state, we assign a region. We call the edges of an abstraction* abstract transitions*.*

By abuse of notation, we will usually use the same notation for an abstract state and the region assigned to it.

A given abstraction $\mathcal{A}$ represents the set of trajectories that start in abstract states marked as `Init`, end in abstract states marked as `Unsafe`, never leave the abstraction, and move from one abstract state to the next only if there is a corresponding abstract transition. We denote this set by $[\![\mathcal{A}]\!]$.

The intuition is that, the abstraction is an over-approximation of the set of error trajectories $\mathcal{E}$ of a given system during the computation. We say that an abstraction $\mathcal{A}^*$ is *tighter* than an abstraction $\mathcal{A}$ iff

- the abstraction $\mathcal{A}^*$ represents less trajectories than $\mathcal{A}$, that is, $[\![\mathcal{A}^*]\!] \subseteq [\![\mathcal{A}]\!]$, and
- the abstraction $\mathcal{A}^*$ does not lose error trajectories from $\mathcal{A}$, that is $[\![\mathcal{A}^*]\!] \supseteq [\![\mathcal{A}]\!] \cap \mathcal{E}$.

Now we will come up with an algorithm that will incrementally improve an abstraction by making it tighter.

Note that, in particular, $\mathcal{A}$ is tighter than $\mathcal{A}$ itself, but in practice we will try to remove as many trajectories from the abstraction as possible.

Given abstract states a and a', we will assume a procedure $InitReach(a)$ that computes an over-approximation of the set of points in a that are reachable from an initial point in a, and a procedure $Reach(a, a')$ that computes an over-approximation of the set of points in a' reachable from a according to the system dynamics. In our case, we implemented both procedures based on interval constraint propagation [1, 7]. We assume that smaller inputs improve the precision of these operations, that is:

- $a_1 \subseteq a_2$ implies $InitReach(a_1) \subseteq InitReach(a_2)$
- $a_1 \subseteq a_2$ and $a'_1 \subseteq a'_2$ implies $Reach(a_1, a'_1) \subseteq Reach(a_2, a'_2)$

Furthermore, we assume that these procedures exploit information about empty inputs, that is:

- $a = \emptyset$ implies $InitReach(a) = \emptyset$
- $a = \emptyset$ implies $Reach(a, a') = \emptyset$
- $a' = \emptyset$ implies $Reach(a, a') = \emptyset$

Now, the following algorithm (which we will call *pruning algorithm*) computes a tighter abstraction for a given abstraction $\mathcal{A}$.

$\mathcal{A}^* \leftarrow$ copy of $\mathcal{A}$
in $\mathcal{A}^*$: set all regions to $\emptyset$, delete initial labels and edges
// from now on, for every abstract state a of $\mathcal{A}$,
// we denote by a^* the corresponding abstract state of $\mathcal{A}^*$
for all $a \in A$, a is initial
  $a^* \leftarrow InitReach(a)$
  if $a^* \neq \emptyset$ **then**
    mark a^* as initial
let $update(a_1, a_2) =$ // defines a function update
  if $a_1^* \not\rightarrow a_2^*$ and $Reach(a_1^*, a_2) \neq \emptyset$ **then**
    introduce an edge $a_1^* \rightarrow a_2^*$
  if $Reach(a_1^*, a_2) \not\subseteq a_2^*$ **then**
    $a_2^* \leftarrow a_2^* \uplus Reach(a_1^*, a_2)$
    return true
  else
    return false in
while $\exists(a_1, a_2)$ such that $a_1 \rightarrow a_2$, $update(a_1, a_2)$
return $\mathcal{A}^*$

Unlike approaches based on counter-example guided abstraction refinement, the pruning algorithm does *not* increase the size (i.e., the number of nodes) of the abstraction. Still it deduces some some interesting information:

Theorem 1 *The result of the pruning algorithm is tighter than the input abstraction $\mathcal{A}$.*

Proof: We have to prove two items:

- $[\![\mathcal{A}^*]\!] \subseteq [\![\mathcal{A}]\!]$: This follows from the following:
  - the set of initial/unsafe marks of $\mathcal{A}^*$ is a subset of the set of marks of $\mathcal{A}$
  - the set of edges of $\mathcal{A}^*$ is a subset of the set of edges of $\mathcal{A}$
  - the abstract states of $\mathcal{A}^*$ are subsets of the corresponding abstract states of $\mathcal{A}$ since $InitReach(a) \subseteq a$, and $Reach(a^*, a) \subseteq a$.
- $[\![\mathcal{A}^*]\!] \supseteq [\![\mathcal{A}]\!] \cap \mathcal{E}$: Let T be an error trajectory in $[\![\mathcal{A}]\!] \cap \mathcal{E}$. We prove that T is an element of $[\![\mathcal{A}^*]\!]$. Let $a_1 \rightarrow a_2 \rightarrow \cdots \rightarrow a_n$ be the abstract error trajectory corresponding to T in $\mathcal{A}$. We prove that the corresponding abstract trajectory $a_1^* \rightarrow a_2^* \rightarrow \cdots \rightarrow a_n^*$ in $\mathcal{A}^*$ is an abstract error trajectory containing T.
  - a_1^* is initial in $\mathcal{A}^*$ and contains the initial point of T
  - We assume that $a_1^* \rightarrow a_2^* \rightarrow \cdots \rightarrow a_i^*$, with $i < n$ forms an abstract trajectory containing T in $\mathcal{A}^*$, and prove that also $a_1^* \rightarrow a_2^* \rightarrow \cdots \rightarrow a_i^* \rightarrow a_{i+1}^*$ forms an abstract trajectory containing T in $\mathcal{A}^*$.

    To prove that $a_i^* \rightarrow a_{i+1}^*$ in $\mathcal{A}^*$ we observe that T leads from a_i^* to a_{i+1}. Hence $Reach(a_i^*, a_{i+1})$ is non-empty and the abstract transition $a_i^* \rightarrow a_{i+1}^*$ exists. Moreover, $Reach(a_i^*, a_{i+1})$ contains all points of T in a_{i+1}, and since the while loop terminated, $Reach(a_i^*, a_{i+1}) \subseteq a_{i+1}^*$ and hence a_{i+1}^* also contains these points.

■

Note however, that it is a-priori not clear, that the pruning algorithm terminates. In our case, we ensure this by simply doing all computation on *finite* set of floating point numbers (cf. the notion of "widening"). Hence

there are only finitely many possibilities of changing boxes with $\uplus$, until a fixpoint is reached.

Moreover, by using an implementation of $Reach$ that is idempotent we also can avoid stuttering (i.e., many small improvements by close floating point numbers), in most cases.

As already mentioned, the pruning algorithm tries to deduce information about a given system without increasing the size of the abstraction. In cases, where it can deduce no more information, we have to fall back to some increase of the size of the abstraction (cf. to a similar approach in constraint programming where one falls back to exponential-time splitting, when polynomial-time deduction does not succeed any more).

We do this by the *Split* operation that chooses an abstract state and splits it into two, copying all the involved edges and introducing edges between the two new states. All the labels and abstract transitions to other abstract states are copied as well. Moreover, two new abstract transitions that connect the original abstract state with its copy are added. The region assigned to the abstract state is equally split among two abstract states. To do this we pick a splitting dimension of the box assigned to the region and we split the box into halves using this dimension. For picking the splitting dimension, a round-robin strategy has proved to be the useful heuristics [1]. Such an refinement decreases the amount of over-approximation in subsequent calls to the pruning algorithm due to the properties of the $Reach$ and $InitReach$.

It is clear that the pruning algorithm can also be done backward in time (i.e., removing parts of the abstraction not leading to an unsafe state). We will denote the resulting algorithm by $Prune^{-}(\mathcal{A})$. Now we have to following overall algorithm for safety verification:

while $\mathcal{A}$ contains an abstract error path
  $A \leftarrow Prune(\mathcal{A})$
  $A \leftarrow Prune^{-}(\mathcal{A})$
  $A \leftarrow Split(\mathcal{A})$
return "safe"

Since neither pruning nor splitting removes an error trajectory, the absence of an abstract error path at the termination of the while loop implies the absence of an error trajectory of the original system. This implies the correctness of the algorithm.

4. Improvements

4.1. Avoided Redundant Edge Checks

One disadvantage of the pruning algorithm is that it may do redundant tests for the condition $Reach(a_1^*, a_2) \not\subseteq a_2^*$ in the update function. Whenever such a test has been made, this can be remembered until the information is not valid any more.

To this purpose we add additional edges to the abstraction that we label with $\subseteq$ (and which we call *consistency edges*). We keep the invariant (that we will call *consistency invariant*) that whenever $a_1^* \rightarrow_{\subseteq} a_2^*$, then $Reach(a_1^*, a_2) \subseteq a_2^*$.

Moreover we use a procedure `propChange`(a) that, for every a' with $a \rightarrow a'$ deletes every edge $a \rightarrow_{\subseteq} a'$. This allows us to change the while loop in the pruning algorithm as follows:

$\mathcal{A}^* \leftarrow$ copy of $\mathcal{A}$
in $\mathcal{A}^*$: set all regions to $\emptyset$, delete initial labels and edges
// from now on, for every abstract state a of $\mathcal{A}$,
// we denote by a^* the corresponding abstract state of $\mathcal{A}^*$
for all $a \in A$, a is initial
  $a^* \leftarrow InitReach(a)$
  if $a^* \neq \emptyset$ then
    mark a^* as initial
    propChange(a^*)
let $update(a_1, a_2) =$
  if $a_1^* \rightarrow_{\subseteq} a_2^*$ **then return false**
  introduce an edge $a_1^* \rightarrow_{\subseteq} a_2^*$
  if $a_1^* \not\rightarrow a_2^*$ and $Reach(a_1^*, a_2) \neq \emptyset$ **then**
    introduce an edge $a_1^* \rightarrow a_2^*$
  if $Reach(a_1^*, a_2) \not\subseteq a_2^*$ **then**
    $a_2^* \leftarrow a_2^* \uplus Reach(a_1^*, a_2)$
    propChange(a_2^*)
    return true
  else
    return false in
while $\exists (a_1, a_2)$ such that $a_1 \rightarrow a_2$, $update(a_1, a_2)$
return $\mathcal{A}^*$

Theorem 2 *Independent of the consistency edges of the input $\mathcal{A}$, the improved pruning algorithm computes the same result as the original one.*

Proof: Clearly, at the beginnning of the while loop, in both algorithms, $\mathcal{A}^*$ is the same. We prove that every time the termination condition of the while loop is tes-

ted, the consistency invariant holds, and hence the algorithm produce the same result.

The first time, the termination condition of the while loop is tested, the consistency invariant holds due to the following reasoning: Let a_1^*, a_2^* be such that $a_1^* \rightarrow_{\subseteq} a_2^*$, then $a_1^* = \emptyset$, since otherwise the operation update(a_1^*) would have deleted the consistency edge. Hence $Reach(a_1^*, a_2) = Reach(\emptyset, a_2) \subseteq a_2^*$.

■

4.2. Incremental Refinement of Abstraction

Now observe that splitting, or dual pruning, only changes a part of the abstraction. Still, the pruning algorithms do a complete re-computation. This is not necessary, and in order to avoid it:

- We mark all abstract states for which we know, that a re-computation will not improve, with the mark `Cons` (the *consistency mark*).
- Whenever splitting or dual pruning changes an abstract state, we delete this consistency mark, and all consistency marks of states reachable from it.
- At the beginning of the pruning algorithm for all abstract states we reset the abstract state with the result of $InitReach$ only if the consistency mark is not set. Abstract states with he consistency mark, retain the value from the input abstraction $\mathcal{A}$.

Since we do separate forward and backward pruning, we also need separate consistency marks for both cases. Splitting removes both consistency marks at the same time.

5. Conclusion

In this paper, we have introduced a new approach to unbounded safety verification of hybrid systems with non-linear ordinary differential equations. Currently we are doing detailed computational experiments comparing the algorithm with alternatives and studying various heuristics and implementation choices.

References

[1] S. Ratschan and Z. She, "Safety Verification of Hybrid Systems by Constraint Propagation Based Abstraction Refinement", ACM TECS 2007.

[2] T. Dzetkulič and S. Ratschan, "How to Capture Hybrid Systems Evolution Into Slices of Parallel Hyperplanes", 3rd IFAC Conference on Analysis and Design of Hybrid Systems 2009.

[3] S. Ratschan and T. Dzetkulič, "Verification of Hybrid Systems by Incremental Abstract Forward/Backward Computation" to appear on International Workshop on Reachability Problems 2010.

[4] E. Clarke, A. Fehnker, Z. Han, B. Krogh, J. Ouaknine,O. Stursberg, and M. Theobald, "Abstraction and Counterexample-Guided Refinement" in Model Checking of Hybrid Systems Int. Journal of Foundations of Comp. Science, 2003.

[5] R. Alur, T. Dang, and F. Ivančić, "Predicate abstraction for reachability analysis of hybrid systems", ACM TECS 2006.

[6] G. Frehse, B.H. Krogh, and R.A. Rutenbar, "Verifying Analog Oscillator Circuits Using Forward/Backward Abstraction Refinement" DATE 2006: Design, Automation and Test in Europe 2006.

[7] S. Ratschan and Z. She, "Constraints for Continuous Reachability in the Verification of Hybrid Systems", Proc. 8th Int. Conf. on Artif. Intell. and Symb. Comp., 2006.

[8] T.A. Henzinger, "Hybrid Automata with Finite Bisimulations", Proceedings of the 22nd International Colloquium on Automata, Languages, and Programming (ICALP) 1995.

Multivariate Methods of Survival Analysis

Post-Graduate Student:
MGR. JANA FÜRSTOVÁ
First Faculty of Medicine
Charles University in Prague
Kateřinská 32
121 08 Prague 2, CZ
jana.furstova@email.cz

Supervisor:
DOC. MGR. ZDENĚK VALENTA, PH.D.
Department of Medical Informatics
Instutite of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
valenta@euromise.cz

Field of Study:
Biomedical Informatics

Abstract

In this paper, an overview of statistical methods used in survival analysis is presented. Standard univariate concepts in survival analysis and their non-parametric and semi-parametric estimates are collected. The univariate models are then extended to multivariate setting. Several methods of treating multi-state and correlated survival data are presented.

1. Introduction

Survival analysis is a collection of statistical methods for analyzing time-to-event data. The commencement of survival analysis dates back to the 18th century when analyses of mortality experience of human populations started. During the World War II, survival analysis focused on engineering – reliability of military equipment was being analyzed. After the World War II the interest turned towards economics and medicine. In 1960s, after the fundamental article of E. L. Kaplan and P. Meier [8] had been published, medical application of survival analysis shifted to the center of statistical focus.

2. Basic concepts in survival analysis

2.1. The survival and hazard function

Let X be the time until some specified event occurs, i.e. X is a non-negative real valued random variable having continuous distribution with finite expectation. There are several functions characterizing the distribution of X :

- The probability density of $X : f(x), x \geq 0$.
- The survival function:

$$\begin{aligned} S(x) &= P(X > x) = \int_x^\infty f(u)du \\ &= 1 - F(x), \end{aligned}$$

  where $F(x)$ is the cumulative distribution function. The survival function describes the probability of an individual surviving beyond time x (experiencing the event after time x).
- The hazard function:

$$\lambda(x) = \lim_{\Delta x \to 0^+} \frac{P(x \leq X < x + \Delta x | X \geq x)}{\Delta x},$$

  for all $x > 0$. The hazard function represents a conditional probability rate at which an individual alive at time x will experience an event in the next instant. There is a close relationship between the hazard and the survival functions:

$$\lambda(x) = -\frac{d}{dx} \ln S(x).$$

- The cumulative hazard function:

$$\Lambda(x) = \int_0^x \lambda(u)du = -\ln S(x).$$

  Thus

$$S(x) = \exp(-\Lambda(x)) = \exp\left(-\int_0^x \lambda(u)du\right).$$

2.2. Censoring

Survival data possess a special feature of censoring, compared to other statistical data. Censoring is used when the survival time is not known exactly, the event is only known to have occurred within some time interval. There are several types of censoring: right, left and interval. In biomedical applications, right censoring

is the most common type of censoring. It occurs when survival time is incomplete on the right-hand side of the follow-up period, i.e. the study ends before all patients experience the event or a patient is lost to follow-up (dies due to reasons other than the event of interest, withdraws from the study, moves to another city, etc.).

Let $X_1, X_2, \ldots, X_n$ be independent and identically distributed (i.i.d.) survival times and $C_1, C_2, \ldots, C_n$ be i.i.d. censoring times. The lifetime X_i of the $i-$th individual will be known if, and only if, $X_i < C_i$. If $C_i > X_i$, the event time will be censored at C_i. Thus it is convenient to represent the survival experience of the group of patients by pairs of random variables (T_i, δ_i), where $T_i = \min(X_i, C_i)$ and $\delta_i = \mathrm{I}(X_i < C_i)$, where I is an indicator of the event's occurring, having value one if the event occurs, and zero otherwise.

2.3. Counting processes and martingales

An alternative approach to develop inference procedures for censored data involves counting processes. A counting process $N = \{N(t), t \geq 0\}$ is a stochastic process with $N(0) = 0$, whose value at time t counts the number of events that have occurred in the interval $(0, t]$. The sample paths (realizations) of N are non-decreasing, right-continuous step functions that jump whenever an event (or events) occur. In the counting process formulation, the pair of variables (T_i, δ_i) introduced in Section 2.2 is replaced with the pair of functions $N_i(t), Y_i(t), i = 1, \ldots, n$, where

$$\begin{aligned} N_i(t) &= \text{no. of events observed in } [0,t] \text{ for unit } i \\ Y_i(t) &= \begin{cases} 1 & \text{unit } i \text{ is at risk at time } t, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

$N_i(t)$ is a counting process, while $Y_i(t)$ is a predictable process, i.e. a process whose value at time t is known infinitesimally before t, at time t^-. This process has left-continuous sample paths. Right-censored survival data are included in this formulation as a special case: $N_i(t) = \mathrm{I}(T_i \leq t, \delta_i = 1)$ and $Y_i(t) = \mathrm{I}(T_i \geq t)$.

To deal with all on-study information of each patient, a term history (or filtration) is used. A history, denoted $\{\mathcal{F}_t, t \geq 0\}$, is a σ-algebra generated by N_i and Y_i :

$$\mathcal{F}_t = \sigma(N_i(s), Y_i(s^+), i = 1, \ldots, n; 0 \leq s \leq t),$$

where $Y_i(s^+) = \lim_{u \to s^+} Y_i(u)$. Thus $\mathcal{F}_t$ contains the information up to and including time t. The information in $\mathcal{F}_t$ increases with increasing time on study, i.e. $\mathcal{F}_s \subseteq \mathcal{F}_t$ for $s \leq t$ [4]. Let $dN_i(t)$ denote the increment of N_i over the time interval $[t, t + dt)$:

$$dN_i(t) = N_i((t + dt)^-) - N_i(t^-).$$

For each $t > 0$, let

$$\mathcal{F}_{t^-} = \sigma(N_i(s), Y_i(s), i = 1, \ldots, n; 0 \leq s < t)$$

denote the full history of the processes $N_i(s), Y_i(s), i = 1, \ldots, n$, up to but not including t. Then (see [4]):

$$\mathrm{E}(dN_i(t)|\mathcal{F}_{t^-}) = Y_i(t)\lambda_i(t)dt,$$

where $\lambda_i(t)$ is the hazard function. The process

$$\Lambda_i(t) = \int_0^t Y_i(s)\lambda_i(s)ds, \quad t \geq 0,$$

is called the intensity process. At each fixed t, this process is a random variable which approximates the number of jumps by N_i over $(0, t]$. In fact, $\mathrm{E}N_i(t) = \mathrm{E}\Lambda_i(t)$ and thus $\mathrm{E}(N_i(t)|\mathcal{F}_{t^-}) = \mathrm{E}(\Lambda_i(t)|\mathcal{F}_{t^-}) = \Lambda_i(t)$ [4].

For any given i, define the process

$$M_i(t) = N_i(t) - \int_0^t Y_i(s)\lambda_i(s)ds, \quad t \geq 0. \qquad (1)$$

Equivalently, the process can be defined $M_i(t) = \int_0^t dM_i(s)$, where

$$dM_i(t) = dN_i(t) - Y_i(t)\lambda_i(t)dt.$$

It can be seen that $\mathrm{E}(dM_i(t)|\mathcal{F}_{t^-}) = 0$ for all t and $\mathrm{E}(M_i(t)|\mathcal{F}_s) = M_i(s)$, for all $s \leq t$ [4]. A process that satisfies these (equivalent) conditions is a martingale. According to a Doob-Meier decomposition theorem (see [4]), any counting process may be uniquely decomposed as the sum of a martingale and a compensator C, which is a predictable, right-continuous process with $C(0) = 0$. As an example, according to (1)

$$\begin{aligned} N_i(t) &= M_i(t) + \int_0^t Y_i(s)\lambda_i(s)ds \qquad (2) \\ &= M_i(t) + \Lambda_i(t), \end{aligned}$$

where $M_i(t)$ is the counting process martingale corresponding to $N_i(t)$, and $\Lambda_i(t)$ is the compensator of the counting process N_i with respect to the filtration $\mathcal{F}_{t^-}$. In terms of differential increments, the process (2) can be equivalently written as

$$dN_i(t) = dM_i(t) + Y_i(t)\lambda_i(t)dt.$$

The approach using martingale methods is very useful in yielding results for censored data, especially for calculating and verifying asymptotic properties of test statistics and estimators.

3. Non-parametric and semi-parametric models

A principle objective of survival analysis focuses on estimation of basic quantities (the survival and hazard function) based on censored data. To analyze survival data parametrically, assumptions about the distribution of the failure times would have to be made. To avoid such assumptions, it is common to use non-parametric models. The simplest non-parametric estimate of a distribution function is the empirical distribution function – a continuous distribution is estimated by a discrete one. The only problem with this approach is the censoring – it is not taken into account in standard statistical methods. Important steps in the development of appropriate methods were done by Kaplan and Meier [8] and Cox [3].

3.1. The Kaplan-Meier and the Nelson-Aalen estimators

The Kaplan-Meier estimator (called also the product-limit estimator) estimates the survival function by

$$\hat{S}(t) = \prod_{i:t_i \leq t} \left(1 - \frac{d_i}{R_i}\right),$$

where there are d_i events observed at time t_i and R_i is the number of individuals still at risk at time t_i (uncensored survivors just before t_i). The variance of the estimator can be estimated using Greenwood's formula (see [10]):

$$\hat{V}(\hat{S}(t)) = \hat{S}(t)^2 \sum_{i:t_i \leq t} \frac{d_i}{R_i(R_i - d_i)}.$$

The product-limit estimator can also be used to estimate the cumulative hazard function: $\hat{\Lambda}(t) = -\ln(\hat{S}(t))$.

An alternative estimator of the cumulative hazard function was proposed by Nelson in 1972 [11] and rediscovered by Aalen in 1978 [1]:

$$\tilde{\Lambda}(t) = \sum_{i:t_i \leq t} \frac{d_i}{R_i}.$$

The variance of the Nelson-Aalen estimator was estimated by Aalen using counting process techniques and is given by

$$\tilde{V}(\tilde{\Lambda}(t)) = \sum_{i:t_i \leq t} \frac{d_i}{R_i^2}.$$

Based on the Nelson-Aalen estimator of the cumulative hazard function, an alternative estimator of the survival function becomes $\tilde{S}(t) = \exp(-\tilde{\Lambda}(t))$.

Suppose now that n individuals from a homogeneous population are put on a study at time 0. Let N_i be the counting process and Y_i be the at-risk process of the i-th individual, as described in Section 2.3. Let $N.(t) = \sum_{i=1}^{n} N_i(t)$ and $Y.(t) = \sum_{i=1}^{n} Y_i(t)$, $0 < t < \infty$. $N.(t)$ denotes the total number of observed failures in the interval, while $Y.(t)$ is the number of individuals in the entire study group that are at risk at time t. The Nelson-Aalen estimator of the cumulative hazard can be written in the counting process notation as

$$\tilde{\Lambda}(t) = \int_0^t \frac{J(u)}{Y.(u)} dN.(u),$$

where $J(u) = \mathrm{I}(Y.(u) > 0)$ with the convention that $0/0$ is interpreted as 0 [7]. The Kaplan-Meier estimator of the survival function is then

$$\hat{S}(t) = \prod_{u \leq t} (1 - d\tilde{\Lambda}(u)).$$

3.2. The Cox model

In many clinical studies we need to assess risk factors for the event of interest. The patients have various additional characteristics which may affect their survival experience (e.g. age, sex, blood pressure, .). The Cox proportional hazards model has become a popular approach to modeling covariate effects on survival. In this model the intensity process (hazard) for the i-th subject is

$$\lambda_i(t) = Y_i(t)\lambda_0(t)\exp(\beta^T X_i),$$

where $Y_i(t)$ is the at-risk process, λ_0 is the baseline hazard (common to all individuals in the study population), X_i is the vector of covariates of individual i, and β is a vector of unknown regression parameters. In this model, the ratio of hazard functions of two individuals is constant (the baseline hazard $\lambda_0(t)$ is canceled out), thus the temporal effect is separated from the effect of the covariates. Estimation of the regression coefficients is based on maximizing of the partial likelihood function, which was introduced by Cox in 1972 [3]. The parameters estimates can then be obtained numerically using the Newton-Raphson algorithm (see [9]).

4. Multivariate survival analysis

In most clinical applications univariate survival analysis assumes that the observed survival times are mutually independent (i.i.d. failure times). In practice, however, dependence can occur for very different kinds of data, e.g. survival of twins or other several individuals, similar organs, recurrent events or multi-state events. Multivariate survival analysis covers the field where independence between survival times cannot be assumed. According to [6], the various approaches to analyzing

multivariate survival data fall into four main categories: multi-state models, frailty models, marginal modeling and non-parametric methods. The data structure should be considered as well. The data can be parallel (where the number of failures is fixed by the design of the study) or longitudinal (where the number of failures is random for each object under study). The data sets are classified into six types: several individuals, similar organs, recurrent events, repeated measurements, different events and competing risks. Relation of the data types to the main approaches of analysis is described in Table 1. The multivariate non-parametric estimation methods are not included in Table 1 and are not presented in this paper, because this approach is not our major focus at present and there are several fundamental problems to be solved. For more details, see e.g. [6].

Type of data	Multi-state	Frailty	Marginal
Several individuals	x	x	x
Similar organs	x	x	x
Recurrent events	x	x	x
Repeated measurements		x	x
Different events	x		
Competing risks	x		

Table 1: Overview of data types and approaches; x means relevant, blank not relevant. Adopted from [6].

4.1. Competing risks and multi-state models

Multi-state models are commonly used for describing the development of longitudinal data. They model stochastic processes, which at any time point occupy one of a set of discrete states. In medicine, the states can be e.g. healthy, diseased, and dead. A change of state is called a transition. The competing risk model is an example of multi-state modeling. In competing risks, various causes of death "compete" in the life of patient, and occurrence of one event precludes occurrence of the other events. There are generally three areas of interest in the analysis of competing risks [7]:

1. Studying the relationship between a vector of covariates and the rate of occurrence of specific types of failure.

2. Analyzing whether patients at high risk of one type of failure are also at high risk for others.

3. Estimating the risk of one type of failure after removing others.

Suppose that individuals under study can experience any one of m distinct failure types. For each individual, the underlying failure time T and a covariate vector X are known. The overall hazard function at time t is

$$\lambda(t, X) = \lim_{\Delta t \to 0^+} \frac{P(t \leq T < t + \Delta t | T \geq t, X)}{\Delta t}.$$

To model competing risks, a cause-specific hazard function is considered:

$$\lambda_j(t, X) = \\ = \lim_{\Delta t \to 0^+} \frac{P(t \leq T < t + \Delta t, J = j | T \geq t, X)}{\Delta t}$$

for $j = 1, \ldots, m$, J is a random variable representing the type of failure, and $t > 0$. In words, $\lambda_j(t, X)$ specifies the rate of type j failures, given X and in the presence of all other failure types [7]. If only one of the m failure types can occur, then

$$\lambda(t, X) = \sum_{j=1}^{m} \lambda_j(t, X)$$

due to the law of total probability.

It is possible to calculate the Kaplan-Meier estimator for each type of failure separately, but it is difficult to give this a survival function interpretation and therefore this is not recommended [6]. Instead, generalizations of the Kaplan-Meier and Nelson-Aalen estimators can be made (see e.g. [7]). The generalized estimator includes all causes of failure and is usually denoted the Aalen-Johanson estimator.

Cox model for the cause-specific hazard functions can be considered:

$$\lambda_j(t, X) = \lambda_{0j}(t) \exp(\beta_j^T X), \quad j = 1, \ldots, m.$$

Both the baseline hazards λ_{0j} and the regression coefficients β_j vary arbitrarily over the m failure types. Estimation and comparison of the coefficients β_j can be conducted by applying asymptotic likelihood techniques individually to the m factors.

A traditional approach to multi-state models is based on the Markov models. Consider first a homogeneous population with no covariates. Let $A(t)$ be the state occupied at time $t, t \geq 0$, with probability model of $A(t)$ being the Markov process. The individuals under study move among $m > 1$ discrete states. If a randomly chosen individual is in state i at time t^-, the transition rate (or intensity) from i to j at time t is given by

$$d\Lambda_{ij}(t) = \\ = P[A(t^- + dt) = j | A(u), 0 \leq u < t, A(t^-) = i] \\ = P[A(t^- + dt) = j | A(t^-) = i], \quad t > 0,$$

which holds for all $A(u), 0 \leq u < t$ with $A(t^-) = i$, and $i, j \in \{1, \ldots, m\}, j \neq i$. The process is memoryless in that only the current state occupied is relevant in specifying the transition rates [7]. In the continuous

case, $d\Lambda_{ij}(t) = \lambda_{ij}(t)dt$ for all $i, j = 1, \ldots, m$, so that $\lambda_{ij}(t), i \neq j$ is the continuous-time intensity function for i-to-j transitions. Estimation of the cumulative intensity functions $\Lambda_{ij}(t)$ proceeds as follows [7]: consider a possibly right-censored sample of n individuals. For $k = 1, \ldots, n$, let $N_{ijk}(t)$ be the right continuous process that counts the number of observed direct i-to-j transitions for k-th individual, $i, j = 1, \ldots, m, i \neq j$. Let $Y_{ik}(t)$ be the corresponding at-risk process. Define the filtration process as

$$\mathcal{F}_t = (N_{ijk}(t), Y_{ik}(u^+), 0 \leq u \leq t),$$

for $k = 1, \ldots, n$; $i, j = 1, \ldots, m$; and suppose that censoring is independent, so that

$$P(dN_{ijk}(t) = 1|\mathcal{F}_{t^-}) = Y_{ik}(t)d\Lambda_{ij}(t),$$

which must hold for all i, j, k and $t > 0$. The Nelson-Aalen estimator of $\Lambda_{ij}(t)$ is then given by

$$d\hat{\Lambda}_{ij}(t) = \frac{dN_{ij.}(t)}{Y_{i.}(t)}$$

for all $i = j$.

When the vector of covariates X is present, the continuous-time modulated Markov model can be specified for the underlying intensity function

$$\lambda_{ijk}(t) = \lim_{dt \to 0^+} \frac{P(A_k(t^- + dt) = j|A_k(t^-) = i, X)}{dt}.$$

Parametric and semi-parametric models for λ_{ijk} are obtained analogously as earlier and may be found in [7].

4.2. Frailty models

Frailty models represent an extension of the Cox proportional hazards model. The concept of frailty provides a way to introduce random effects into the model to account for association (correlation) and unobserved heterogeneity. This heterogeneity may be difficult to assess but is nevertheless of great importance. The frailty is an unobserved random factor that modifies multiplicatively the hazard function of an individual or a group of individuals. The key idea of these models is that individuals most "frail" die earlier than the others [15]. The frailty models are relevant to lifetimes of several individuals, similar organs and repeated measurements. They are not generally relevant for the case of different events [6].

First, bivariate models will be considered. Let

$$S_{12}(t_1, t_2) = P(T_1 \geq t_1, T_2 \geq t_2)$$

be the joint survival function for the two survival times T_1 and T_2, where $S_{12}(t, t)$ is the probability that both subjects under study are alive at time t. The marginal survival functions are then

$$\begin{aligned} S_1(t_1) &= P(T_1 \geq t_1) = S_{12}(t_1, 0) \\ S_2(t_2) &= P(T_2 \geq t_2) = S_{12}(0, t_2). \end{aligned}$$

If T_1 and T_2 are independent, $S_{12}(t_1, t_2) = S_1(t_1)S_2(t_2)$. The joint hazard function is

$$\lambda_{12}(t_1, t_2) = \lim_{\Delta t \to 0^+} \frac{P(T_1 \in [t_1, t_1 + \Delta t), T_2 \in [t_2, t_2 + \Delta t)|T_i \geq t_i)}{\Delta t^2},$$

and the marginal hazards are

$$\lambda_i(t_i) = \lim_{\Delta t \to 0^+} \frac{P(T_i \in [t_i, t_i + \Delta t)|T_i \geq t_i)}{\Delta t},$$

for $i = 1, 2$. To address heterogeneity in the survival times it is assumed that the lifetimes are conditionally independent, i.e. T_1 and T_2 are independent given the random effect Z called frailty:

$$S_{12}(t_1, t_2|Z) = S_1(t_1|Z)S_2(t_2|Z).$$

Usually, the frailty is assumed to act multiplicatively on the hazard, so that

$$\lambda_i(t_i) = Z\lambda_{0i}(t_i) \quad \text{and} \quad S_i(t_i|Z) = S_{0i}(t_i)^Z$$

for some baseline hazard $\lambda_{0i}(t)$ and baseline survival function $S_{0i}(t)$ (when known covariates X_i are present, the hazard may be expressed as $\lambda_{0i}(t_i) = \lambda_0(t_i)\exp(\beta^T X_i)$ through the Cox regression model). Under the assumption of multiplicative frailty, the cumulative hazards are $\Lambda_i(t_i) = Z\Lambda_{0i}(t_i)$. The conditional joint survival function is then

$$\begin{aligned} S_{12}(t_1, t_2|Z) &= S_{01}(t_1)^Z S_{02}(t_2)^Z \\ &= \exp(-Z\Lambda_{01}(t_1))\exp(-Z\Lambda_{02}(t_2)) \\ &= \exp(-Z(\Lambda_{01}(t_1) + \Lambda_{02}(t_2))). \end{aligned}$$

As the frailty Z is an unobserved effect, it needs to be 'integrated out' of the survival function. This is done by the Laplace transform, which is defined for a random variable Z as

$$L_g(s) = \int \exp(-sz)g(z)dz = \mathrm{E}(\exp(-sZ)),$$

where $g(z)$ is the probability density of Z. For the bivariate survival function thus

$$\begin{aligned} &S_{12}(t_1, t_2) = \\ &= \int_0^\infty S_{12}(t_1, t_2|Z)g(z)dz \\ &= \int_0^\infty \exp(-Z(\Lambda_{01}(t_1) + \Lambda_{02}(t_2)))g(z)dz \\ &= L_g(\Lambda_{01}(t_1) + \Lambda_{02}(t_2)), \end{aligned}$$

where the Laplace transform of $g(z)$ is evaluated at $s = \Lambda_{01}(t_1) + \Lambda_{02}(t_2)$.

In many applications, the frailty Z is assumed to follow some distribution with explicit Laplace transform. A standard (and most widely used) distribution for frailty is the gamma distribution. The random variable Z is gamma distributed with parameters k and θ ($Z \sim \Gamma(k, \theta)$), if its probability density function is

$$g(z) = \frac{\theta^k z^{k-1} \exp(-\theta z)}{\Gamma(k)}, \quad k, \theta > 0 \text{ and } z > 0,$$

with $\mathrm{E}Z = \frac{k}{\theta}$, $\mathrm{var}Z = \frac{k}{\theta^2}$. The gamma function in the denominator of the probability density function is defined as

$$\Gamma(k) = \int_0^\infty u^{k-1} \exp(-u) du, \quad \text{for } k > 0.$$

It satisfies $\Gamma(k+1) = k\Gamma(k)$. The gamma distribution fits very well to failure data and is also convenient from computational and analytical point of view [18].

Suppose the common frailty component Z has a gamma distribution with parameters $k = \theta = 1/\sigma^2$. The Laplace transform of the gamma density is then $L(s) = \left(\frac{\theta}{\theta+s}\right)^k$, which leads to (see [6])

$$S_{12}(t_1, t_2) = \left(\frac{1}{1 + \sigma^2\Lambda_{01}(t_1) + \sigma^2\Lambda_{02}(t_2)}\right)^{1/\sigma^2}.$$

To extend the bivariate model to a multivariate one, consider a set of clustered data where for the j-th individual in the i-th group (or cluster) there are the observation times t_{ij} and the vector of covariates X_{ij}. The assumption is, again, that given X_{ij} and a random effect Z_i, the m_i lifetimes in group i are independent. Thus the joint distribution of these lifetimes given Z_i is the product of the marginal distributions given Z_i. The marginal hazards then satisfy

$$\lambda_{ij}(t_{ij}|X_{ij}, Z_i) = Z_i \lambda_{0ij}(t_{ij}|X_{ij}).$$

When the hazards are modeled using the Cox proportional hazards,

$$\lambda_{0ij}(t_{ij}|X_{ij}) = \lambda_0(t_{ij}) \exp(\beta^T X_{ij}).$$

If the cluster-specific random effects Z_i have independent gamma distributions, then the unconditional survival for the m_i lifetimes in cluster i is

$$S_i(t_i, X_i) = \int_0^\infty \prod_j S_{ij}(t_{ij}|X_{ij}, Z_i) g(z_i) dz_i,$$

where $t_i = (t_{i1}, t_{i2}, \ldots, t_{im_i})^T$, $X_i = (X_{ij})_{m_i \times n}$. This can be solved using the Laplace transform (see [13])

$$S(t_i, X_i) = \left(\frac{1}{\psi}\right)^{1/\sigma^2},$$

where

$$\psi = \quad 1 + \sigma^2\Lambda_0(t_{i1}) \exp(\beta^T X_{i1}) + \cdots + \\ + \sigma^2\Lambda_0(t_{im_i}) \exp(\beta^T X_{im_i}).$$

Different choices of distribution for the frailty Z are possible, e.g. the family of positive stable distributions or the PVF (power variance function) family. For more information about these, see [6]. The frailty Z may also be treated non-parametrically. Although it is desirable to have completely non-parametric estimate of the survival function, the estimates are mathematically complicated and are not of major importance [6].

Statistical models that use counting process notation and are convenient for these types of analyses are slightly different from those used until now. In the previously used models, the intensity process $\lambda(t)$ at the follow-up time t, given the covariates X was

$$\lambda(t)dt = P(dN(t) = 1|N(s), 0 \le s < t, X).$$

In this expression it is assumed that jumps in N are of unit size only. However, recurrent and correlated failure time data include jumps of size greater than one (more than one event can be recorded for an individual at a specific follow-up time). Thus it is natural to model the mean jump in N across time:

$$d\Lambda(t) = \mathrm{E}(dN(t)|N(s), 0 \le s < t, X)$$

in the cumulative intensity process. The Cox-type model for the intensity process is then

$$d\Lambda(t) = d\Lambda_0(t) \exp(\beta^T X).$$

For more details, see [7].

4.3. Marginal models

Marginal model approach is based on the marginal distributions. The dependence is not considered to be the main aspect here. The regression parameters are estimated from generalized estimating equations and the corresponding variance-covariance estimators are properly corrected to account for the dependence structure. For parallel data, there are in practice two versions of the general idea [6]: the coordinate-wise approach considers each marginal separately, estimates the regression coefficients in each marginal and then combines the estimates by means of a weighted average, whereas the independence working model approach makes the estimates under the (incorrect) assumption of independence

between the coordinates, which directly yields the final estimate of the regression coefficients.

For ordered multiple events, i.e. multiple events of the same type, there are three common approaches: the independent increment model (Andersen-Gill model), the marginal model (WLW) and the conditional model (PWP). The model of Andersen and Gill (see [2]) is well suited for data where all events are identical and observations within a subject are mutually independent. This assumption is equivalent to each individual counting process possessing independent increments, i.e. the number of events in non-overlapping time intervals are independent given the covariates [15]. The Andersen-Gill approach is in spirit close to Poisson regression because the independent increment processes are modeled as time-varying Poisson processes. In the WLW model (Wei, Lin, Weissfeld [17]), the ordered outcome dataset is treated as though it were an unordered competing risks problem [15]. Since all time intervals start at zero, the model allows a separate underlying hazard for each event. In the PWP model (Prentice, Williams, Peterson [12]), it is assumed that a subject is not at risk for the k-th event until the $k-1$st event has occurred for this subject. This model is thus called the conditional model. The underlying intensity functions in this model may vary from event to event [15]. The marginal models are generally suitable for making inferences on the average effect of risk factors on failure time. However, it provides no insight into the multivariate relationship among failure times [18].

Another approach of the marginal modeling is the concept of copulas. The purpose of this approach is purely to study the dependence, removing all effects of the distribution as function of time and covariates [6]. This is done by assumption that the marginal distributions are uniform or can be transformed to the uniform one. The dependence is then evaluated by specifying the distribution for the bivariate observations. The two approaches – marginal and copula – can be mixed as well. This way the advantages of both procedures are combined: the effect of the covariates and the time scale is modeled by means of the marginal hazard functions and the dependence is modeled by the copulas [6].

5. Conclusion

Survival analysis is a collection of specific statistical methods. In this paper, a short overview of these methods was presented. The standard univariate models were extended to multivariate models dealing with parallel and longitudinal data. The three major multivariate concepts were introduced: multi-state, frailty and marginal models.

In the future, I plan to apply the multivariate concepts to real data coming from the University Hospital in Olomouc. Building on recent leading ideas in survival analysis, I will especially focus on copula modeling and stochastic processes.

References

[1] O.O. Aalen, *Nonparametric Inference for a Family of Counting Processes.* Annals of Statistics 6 (1978), 701–726.

[2] P.K. Andersen and R.D. Gill, *Cox's regression model for counting processes: a large sample study.* The Annals of Statistics 10 (1982), 1100–1120.

[3] D.R. Cox, *Regression Models and Life-Tables.* Journal of the Royal Statistical Society B 34 (1972), 187–220.

[4] T.R. Fleming and D.P. Harrington, *Counting Processes and Survival Analysis.* John Wiley & Sons, New York, 1991.

[5] R.D. Gill, *Understanding Cox's regression model: a martingale approach.* Journal of the American Statistical Association 79 (1984), 441–447.

[6] P. Hougaard, *Analysis of Multivariate Survival Data.* Springer, New York, 2000.

[7] J.D. Kalbfleisch and R.L. Prentice, *The Statistical Analysis of Failure Time Data.* John Wiley & Sons, New York, 2002.

[8] E.L. Kaplan and P. Meier, *Nonparametric Estimation from Incomplete Observations.* Journal of the American Statistical Association 53 (1958), 457–481.

[9] J.P. Klein and M.L. Moeschberger, *Survival Analysis. Techniques for Censored and Truncated Data.* Springer, New York, 2003.

[10] R.G. Miller, G. Gong, and A. Muñoz, *Survival Analysis.* John Wiley & Sons, New York, 1998.

[11] W. Nelson, *Theory and Applications of Hazard Plotting for Censored Failure Data.* Technometrics 14 (1972), 945–965.

[12] R.L. Prentice, B.J. Williams, and A.V. Peterson, *On the regression analysis of multivariate failure time data.* Biometrika 68 (1981), 373–379.

[13] G. Rodríguez, *Multivariate Survival Models.* available at http://data.princeton.edu/ cited on April 10, 2010.

[14] S.G. Self and R.L. Prentice, *Commentary on Andersen and Gill's "Cox's regression model for counting processes: a large sample study".* The Annals of Statistics 10 (1982), 1121–1124.

[15] T.M. Therneau and P.M. Grambsch, *Modeling Survival Data. Extending the Cox Model.* Springer, New York, 2000.

[16] J.W. Vaupel, K.G. Manton, and E. Stallard, *The impact of heterogeneity in individual frailty on the dynamics of mortality.* Demography 16 (1979), 439–454.

[17] L.J. Wei, D.Y. Lin, and L. Weissfeld, *Regression analysis of multivariate incomplete failure time data by modeling marginal distributions.* Journal of the American Statistical Association 84 (1989), 1065–1073.

[18] A. Wienke, *Frailty Models in Survival Analysis.* Habilitation. Martin-Luther-Universität Halle-Wittenberg, 2007. available at http://sundoc.bibliothek.uni-halle.de/habil-online/ cited on September 30, 2009.

Structural Sensitivity of Shortest Path Betweenness Centrality

Post-Graduate Student:
ING. DAVID HARTMAN, PH.D.

Faculty of Mathematics and Physics
Charles University in Prague
Malostranské náměstí 25
118 00 Prague 1, CZ

hartman@kam.mff.cuni.cz

Supervisor:
PROF. RNDR. JAROSLAV NEŠETŘIL, DRSC.

Faculty of Mathematics and Physics
Charles University in Prague
Malostranské náměstí 25
118 00 Prague 1, CZ

nesetril@kam.mff.cuni.cz

Field of Study:
Informatics

Supported by the EC FP7 project BrainSync (EC: HEALTH-F2-2008-200728, CR: MSM/7E08027) and ITI supported as project 1M0545 by the Ministry of Education of the Czech Republic, thanks to M. Corbetta and D. Mantini for data.

Abstract

Complex network analysis is a widely used method in a study of multidimensional dynamical systems. An important part of its methodology is based on various network measures. One of these measures is betweenness centrality based on shortest paths. In this paper an analysis of its specific sensitivity in weighted graphs is studied. This measure is based on a number of shortest paths between pairs of vertices and described sensitivity is based on small differences between lengths of used shortest paths. Results are shown on brain connectivity.

1. Introduction

Many complex systems under study show behavior which can be analyzed using graph theory namely its special part complex network analysis [1–3]. Usually these systems can be represented as a set of coupled subsystems for which structural properties are studied. Usual representatives are coupled biological and chemical systems represented for example by metabolic pathways [4, 5], a system of social interacting species studied by social networks analysis [6–9], World Wide Web [10, 11] and others. A behavior of such systems is analyzed from a perspective of structural properties using graph theory. Sections 2, 3, 4 and 5 describe an introduction into a studied problem and description of data processing and sections 6, 7 and 8 provide our contribution to a given problem.

2. Notations from graph theory

A *graph* G is a pair $G = (V, E)$, where V represents a set of *nodes* (or vertices) and E is a set of *edges* where each one represents a connection between two nodes. There are two possible types of edges (connections). The first one is a directed edge which leads from a vertex v_1 to a vertex v_2 and which is different from another one leading from v_2 to v_1. A graph using such types of edges is called a *directed graph*. If moreover edges between similar vertices, called *loops*, are allowed the resulting graph is called a *directed graph with loops allowed*. The second type of edges is an undirected edge for which a pair of vertices v_1 and v_2 can be connected only by a single edge without direction. This graph can be considered as a directed graph where each pair of nodes which is not disconnected (i.e. there exists at least one edge) is connected by edges in both directions. A graph that has all edges undirected is called an *undirected graph*. Even in this case loops can be used and such graph is then called an *undirected graph with loops allowed*.

Moreover there is another extension of any type of graph that is important especially for complex networks and it is called a *weighted graph* that is derived from a chosen type of graphs by adding a new function assigning to each edge a specific value $w_{i,j}$ called a *weight*. Such an assignment is called a *weight function* $\omega : E \to \mathbb{R}$. This function can be given as a *weight matrix* $W = (w_{i,j})_{i,j=1}^{n}$ with elements representing weights for all edges. Then a weighted graph G_w is a triple $G_w = (V, E, \omega)$ with the similar meaning of the first two elements as in a case of graphs and with a weight function ω. These weights are usually source of information provided by a graph as a representation of a real system. A graph that is not weighted is called an *unweighted graph* or simply a graph. Unweighted graphs can be considered as weighted with all weights equal to 1.

Nodes are usually denoted as v_i or only by their index i when there is an enumeration of nodes defined or simply by small letters (e.g. v) if indices are not important.

The similar situation is in a case of edges for which $e_{i,j}$ or $\{i,j\}$ is used. Number of vertices will be denoted by $n = |V|$ and number of edges by $m = |E|$. From an algebraic point of view the most important notion for us will be the *adjacency matrix* A that is a matrix $A = (a_{i,j})_{i,j=1}^{n}$ whose elements are $a_{i,j} = 1 \iff e_{i,j} \in E$ and 0 otherwise. For weighted graph adjacency information is already stored in weight matrix. Usually it is important to denote for a vertex v a set of vertices that are connected with v by an edge for which $\Gamma(v)$ is used.

There are other important notions dealing with distances in a graph. A *walk* L in a graph G is an alternating sequence of vertices and edges $v_0, e_1, v_1, e_2, \ldots, e_k, v_k$ where $e_i = \{x_{i-1}x_i\}, 0 < i \leq k$. The length of walk is k for unweighted and $\omega(L) = \sum_{i=1}^{k} w_i$ for weighted graphs. A walk is called a *trail* if all edges are distinct and a *path* if also all vertices are distinct [12]. A *shortest path* connecting two vertices is a path with length ℓ such that no other path between the same vertices has sharply smaller length. A length of shortest path between vertices i and j is called a *distance* between vertices and denoted as $d_G(i,j)$. Shortest paths between any pair of vertices are usually stored in a *distance matrix* $D = \{d_G(i,j)\}_{i,j}$. There can be more shortest paths.

3. Complex network application

When a specific system has a structure that enables an analysis by a complex network the processes of creating such a network have few more of less similar common steps. A considered system usually has several interesting locations that are studied from a perspective of mutual connections. Therefore the first step is to measure time series from all these locations. With the resulting set of series an analysis of a targeted type of mutual connection (usually correlation or mutual information) between all pairs is done which results in a *matrix of connectivity* (a correlation matrix or a matrix of mutual information). This matrix is then considered as a weight matrix of a graph generally considered as directed (for symmetric connectivity measure undirected). If an unweighted graphs are considered the values of weights are binarized by a chosen threshold.

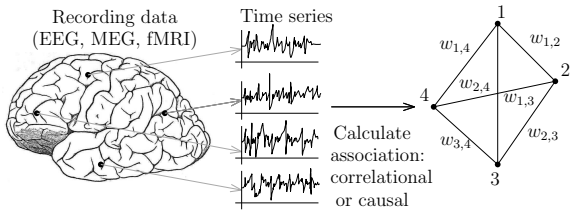


Figure 1: Steps in creating weighted graph from neural data

Within this work neural data from brain are used as an example. A chain of preprocessing for this system looks like in Fig. 1. In this work resting state BOLD-fMRI data from 12 healthy volunteers (5 males and 7 females, age range 20 – 31 years) are used. Specific properties of data sets and preprocessing can be described in [13]. From a graph theoretical perspective an important information is that there are 90 regions from which fMRI data were loaded, i.e. the resulting graph has 90 vertices.

4. Centrality measures

Constructed networks are usually characterized by several graph theoretical measures that can indicate some specific behavior of an underlying network and therefore a studied system [1]. For our purpose a weighted undirected graph is always used unless stated otherwise. The studied measure here is in fact a set of various measures called centralities [14] that are very important to network analysis. These characteristics are connected with vertices by capturing information about an importance of an actual vertex. There are simple measures like a *degree centrality* defined for a vertex simply as a number of its adjacent vertices possibly normalized [15] and a more complex like a *closeness centrality* based on distances between vertices, namely given by an equation

$$C_B(v) = \frac{n-1}{\sum_{t \in V} d_G(v,t)} \tag{1}$$

originally defined in [16] and in this refined form in [17]. As the most descriptive form of centrality can be considered betweenness-like centrality [15, 18, 19].

A general way to define *betweenness centrality* is by using general type W_T of walks with specified properties and use a definition:

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\phi_{st}(v)}{\phi_{st}}, \tag{2}$$

where $\phi_{s,t}$ is a number of walks of a type W_T that goes from a vertex s to a vertex t and $\phi_{s,t}(v)$ is the same number but with a condition that each such a walk has to cross the vertex v. This definition of betweenness centrality expresses relatively how many times a vertex v is crossed by a walk of chosen type when other vertices "communicates" between each others in a specific way determined by W_T. This gives an importance of a vertex from perspective of bridging possible communications.

A character of communication is usually expressed by a length of walk of type W_T between two vertices. For unweighted graphs simply a number of edges is used,

however for weighted graphs usually a sum of reciprocal values of weights is used, because it represents better a close connection of two vertices when they are highly correlated.

5. Using shortest path

Several approaches to W_T are used, but the most common and the oldest one is to use a shortest path as W_T, i.e. to search what fraction of shortest paths between any pair of vertices crosses an analyzed vertex. This leads to a very similar definition to the above which uses different sub-measures:

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}, \tag{3}$$

where $\sigma_{s,t}$ is a number of shortest paths that lead from a vertex s to a vertex t and $\sigma_{s,t}(v)$ is the same number but with a condition that each shortest path has to cross the vertex v. Calculations for weighted and unweighted graphs are similar except determination of a shortest path. Where it is needed to stress a betweenness centrality for a weighted graph is denoted by $C_B^w(v)$.

A subproblem of this problem is called *all-pairs shortest path problem* (APSP) for which there are several classical algorithms [20–22]. Classical ways for betweenness calculation uses breadth-first search (BSF) or Dijkstra algorithm [18] or for slightly better *Floyd-Warshall algorithm* that all result in a time complexity $\Theta(n^3)$. Using Brandes algorithm [23] based on BSF for unweighted and Dijkstra for weighted graphs this complexity can be reduced to $\mathcal{O}(nm)$ in the former case and $\mathcal{O}(nm + n^2 \log n)$ in the second.

Let denote a shortest path between vertices i and j according to weights as $P_{i,j}^S$. For any algorithm variable $\sigma_{i,j}$ has to be enumerated and used for $C_B(v)$ determination. These algorithms handle this enumeration similarly to following simple algorithm computing $\sigma_{i,j}$:

Algorithm 1 Algorithm solving shortest path betweenness centrality

1. Find the shortest path $P_{i,j}^S$ between i and j 2. Set $\sigma_{i,j} = 1$ 3. Iterate through all possible paths $P_{i,j}$ between these nodes and check if $\omega(P_{i,j}) = \omega(P_{i,j}^S)$ 4. If condition 3 is true then $\sigma_{i,j} = \sigma_{i,j} + 1$

This procedure is well suited for both weighted and unweighted graphs and because this paper deals in most cases with weighted graphs let $\sigma_{i,j}$ and $\sigma_{i,j}(v)$ denotes corresponding numbers of shortest paths for weighted graphs and moreover all associated characteristics like distance of two vertices are denoted like defined generally (i.e. $d_G(i, j)$). From this place let all graphs in a following text are considered as weighted unless stated otherwise.

6. Weakness of shortest paths

One problem of a shortest path approach to betweenness centrality that is studied by this paper deals with the equality of paths' lengths used in a comparison within step 3 in algorithm 1. Imagine a network shown in Fig. 2 for which you want to calculate a betweenness centrality.

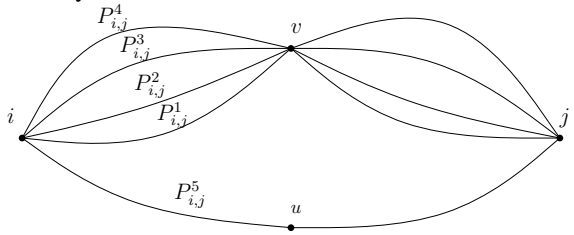


Figure 2: A network representing five possible shortest paths between i and j all with similar number of edges but with various weights on them.

Let for unweighted case (or all weights are equal 1), where all edges "alive" thresholding (i.e. they are "strong enough" in weights), be all presented paths with the same number of edges. In this case the resulting betweenness centralities are $C_B(v) = \frac{4}{5}$ and $C_B(u) = \frac{1}{5}$.

For weighted case, let a graph provides following values of paths' lengths $\omega(P_{i,j}^i) = 1.0 \quad \forall i = 1, 2, 3, 4$ and $\omega(P_{i,j}^5) = 1.01$. This leads to the betweenness values $C_B(v) = 1$ and $C_B(u) = 0$. Nevertheless for the slightly modified values $\omega(P_{i,j}^i) = 1.01 \quad \forall i = 1, 2, 3, 4$ and $\omega(P_{i,j}^5) = 1.0$ the resulting betweenness centralities are $C_B(v) = 0$ and $C_B(u) = 1$. This strong jump is even more evident when paths' lengths differ very small $|\omega(P_{i,j}^i) - \omega(P_{i,j}^5)| < \epsilon \quad \forall i = 1, 2, 3, 4$.

The question arises whether such a small difference should be still predominant in changing a shortest path preference or whether there should be some adjustment for this situation. An immediate idea is to allow shortest paths to differ by some constant δ_P from the real shortest path to still be considered as valuable for computation. However this can lead to a problem when applying this constant to various shortest paths (note even a situation for which $\delta_P \gg \omega(P^S)$). For such reason δ_P should be a function of a shortest path length. Let ϵ_P be a constant and $\delta_P(i, j) = \epsilon_P \omega(P_{i,j}^S)$. This means that there are several values that should be determined to even make our

new algorithm possible. These values can be easily computed by Floyd-Warshall algorithm and we can consider them to be known at first sight. Altogether this gives a following definition:

Definition 6.1 *Let $\epsilon_P > 0$, a path $P_{i,j}$ is called $\delta(i,j)$-*almost shortest *(or simply δ-*almost shortest*) if a condition $\omega(P_{i,j}) - \omega(P^S_{i,j}) < \delta(i,j) = \epsilon_P \omega(P^S_{i,j})$ holds.*

Let all associated characteristics are denoted with a comma (e.g. $\sigma'_{i,j}$). Using this definition as W_T can solve mentioned problem, however this approach has also one big problem dealing with calculation. Most of the algorithms using shortest paths are based on lemma 6.1.

Lemma 6.1 (Bellman criterion) *A vertex $v \in V$ lies on a shortest path between vertices $s,t \in V$, if and only if $d_G(s,t) = d_G(s,v) + d_G(v,t)$.*

To see better the use of this lemma one can define, recalling the computation of Brandes [23], a *pair-dependency* value as

$$\delta_{st}(v) = \frac{\sigma_{st}(v)}{\sigma_{st}}, \tag{4}$$

in which due to Bellman criterion following substitution can be done:

$$\sigma_{st}(v) = \begin{cases} 0 & \text{if } d_G(s,t) < d_G(s,v) + d_G(v,t) \\ \sigma_{s,v} \cdot \sigma_{v,t} & \text{otherwise.} \end{cases} \tag{5}$$

Results of a calculation directly dependent only on determination of $\sigma_{st}, \forall s \neq t$. Finally another lemma can be used to simplify whole process. Let for $P_s(v) = \{u \in V : \{u,v\} \in E, d_G(s,v) = d_G(s,u) + \omega(u,v)\}$ holds:

Lemma 6.2 (Combinatorial paths counting [23]) *For $s \neq v \in V$*

$$\sigma_{sv} = \sum_{u \in P_S(v)} \sigma_{su}$$

Set $P_s(v)$ represents all predecessors of a given vertex v that lie on at least one shortest path. Using this lemma one can compute resulting $\sigma_{s,t}$ for fixed s by simply looking iteratively on shortest paths of predecessors [23]. However both approaches are not possible when using δ-almost shortest paths.

Let start with described lemma 6.2. Probably the least restrictive alternative of $P_s(v)$ for weighted graphs is

$$P'_s(v) = \{u \in V : \{u,v\} \in E, \\ d_G(s,u) + \{u,v\} - d_G(s,v) < \delta(s,v)\}. \tag{6}$$

The condition means that there exists at least one path $P_{s,v}$ crossing the vertex u that is $\delta(s,v)$-almost shortest. However problems appear when iterative calculation upgrades quantity $\sigma'_{s,v}$ from several quantities σ'_{s,u_i} for all $u_i \in P'_s(v)$, see Fig. 3.

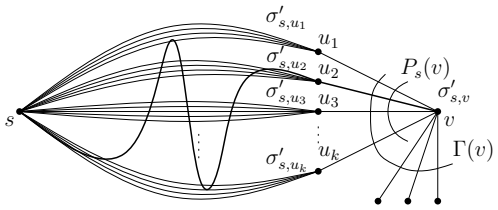


Figure 3: Computing a number of $\delta(s,v)$-almost shortest paths iteratively from $P'_s(v)$

The problem why values σ'_{s,u_i} cannot be used is formulated in a following proposition:

Proposition 6.1 *There exists $\epsilon_P > 0$ such that a path $P_{s,u}$ that is no longer $\delta(s,u)$-almost shortest can be prolonged in a way that resulting path $P_{s,v} = P_{s,u} + \{u,v\}$ is $\delta(s,v)$-almost shortest.*

Proof: Imagine a situation shown in Fig. 6, where vertices s and u are connected by several $\delta(s,u)$-almost shortest paths indicated by narrow belt. One of that paths is shortest and one has maximal length still bounded by $(1+\epsilon_P)\omega(P^S_{s,u})$. Let moreover an edge $\{u,v\}$ is a part of a shortest path $P^S_{s,v}$.

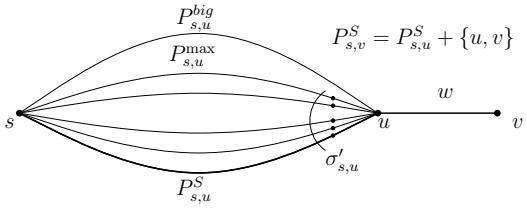


Figure 4: An extension of $\delta(s,u)$-almost shortest path by an edge contained in a shortest path

One can easily see that each path that is $\delta(s,u)$-almost shortest extended by an edge $\{u,v\}$ is also $\delta(s,v)$-almost shortest. This can be seen by considering that even for maximal path holds

$$\omega(P^{\max}_{s,u}) - \omega(P^S_{s,u}) < \delta(s,u) = \epsilon_P \omega(P^S_{s,u}),$$

which can be extended by an edge $e_{u,v}$ such that

$$\omega(P^{\max}_{s,u}) - \omega(P^S_{s,u}) + \omega(e_{u,v}) - \omega(e_{u,v}) < \\ \epsilon_P \omega(P^S_{s,u}) + \omega(e_{u,v}) - \omega(e_{u,v})$$

that can be rewritten in

$$\omega(P^{\max}_{s,v}) - \omega(P^S_{s,v}) < \epsilon_P \omega(P^S_{s,u}) < \epsilon_P \omega(P^S_{s,v}),$$

where $P^{\max}_{s,v}$ is a path that is extended from the maximal path $P^{\max}_{s,u}$ by an edge $\{u, v\}$. So even such maximal path remains $\delta(s, v)$-almost shortest. However when we consider other path $P^{big}_{s,u}$ which is not $\delta(s, u)$-almost shortest we can derive a following statement:

$$\omega(P^{big}_{s,u}) - \omega(P^S_{s,u}) = d > \delta(s, u) = \epsilon_P \omega(P^S_{s,u}).$$

Let $d = \epsilon_P \omega(P^S_{s,u}) + \epsilon'_P \omega(e_{u,v})$, where $0 < \epsilon'_P < \epsilon_P$ then the above condition still holds:

$$\omega(P^{big}_{s,u}) - \omega(P^S_{s,u}) = d = \epsilon_P \omega(P^S_{s,u}) + \epsilon'_P \omega(e_{u,v}) > \delta(s, u) = \epsilon_P \omega(P^S_{s,u})$$

and in the same time it follows for a definition of an extended path $P^{big}_{s,v} = P^{big}_{s,u} + e_{u,v}$ that

$$\begin{aligned} \omega(P^{big}_{s,v}) - \omega(P^S_{s,v}) &= \epsilon_P \omega(P^S_{s,u}) + \epsilon'_P \omega(e_{u,v}) < \delta(s, v) \\ &= \epsilon_P \omega(P^S_{s,v}) \\ &= \epsilon_P \omega(P^S_{s,u}) + \epsilon_P \omega(e_{u,v}). \end{aligned}$$

Therefore one can have $\delta(s, v)$-almost shortest path crossing vertex u and in the same time its subpath $P_{s,u}$ is not necessarily $\delta(s, u)$-almost shortest. ∎

The second problem arises when trying to use an equation 5 for which Bellman criterion 6.1 should be valid. For such purpose defined Bellman criterion is too strong. An alternative to an equation 5 can be defined as:

$$\sigma'_{st}(v) = \begin{cases} 0 & \nexists P_{s,t} \ni v : P_{s,t} \text{ is} \\ & \delta(s, t)\text{-almost shortest} \\ \sigma'_{s,v} \cdot \sigma'_{v,t} & \text{otherwise.} \end{cases} \quad (7)$$

To enable a use of this equation another criterion can be defined shown in equation 6.2 in its negative form.

Proposition 6.2 *The number of $\delta(s, t)$-almost shortest path between vertices s and t that cross vertex v cannot be always calculated using equation 7.*

Proof: The described situation is show in Fig. 5.

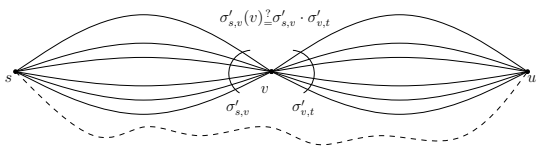


Figure 5: A number of $\delta(s, v)$-almost shortest paths crossing vertex i as multiplication of partial quantities.

Simple way to show assertion is by using a little generalization of proposition 6.1 which shows that $\delta(s, t)$-almost shortest path crossing vertex v can emerge even from a combination of non $\delta(s, v)$-almost shortest and $\delta(v, t)$-almost shortest paths. The zero case in equation 7 is not a problem itself, but it is not the only situation for which this quantity equals zero as was shown above. ∎

Propositions 6.1 and 6.2 show that an algorithm to calculate a betweenness centrality based on a $\delta(s, t)$-almost shortest path cannot be simply constructed from already used algorithms for a shortest path version. For this reason a new exhaustive algorithm is presented:

Data: σ', σ_T, s and t are handled as global and $\Delta_{i,j} = (1 + \epsilon_P) d_{s,t}$ are pre-computed for distance matrix $(d_{i,j})_{i,j}$
$C^\delta_B(v) \leftarrow 0, v \in V$;
foreach *pair* $s, t \in V(G)$ **do**
  $\sigma'(t) \leftarrow 0, \nu(t) \leftarrow 0; \forall t \in V$;
  $\sigma_T \leftarrow 0$; $P \leftarrow$ empty stack;
  ASP$(s, 0)$;
  foreach $v \in V(G) \setminus \{s, t\}$ **do**
    $C^\delta_B(v) = C^\delta_B(v) + \frac{\sigma'(v)}{\sigma_T}$;
  end
end

Procedure δ-`Betweenness`

$\nu(u) \leftarrow 1$; **if** $u \neq s$ **then** $u \xrightarrow{push} P$;
foreach $v \in \Gamma(u)$ **do**
  if $\nu(v) = 0$ **then**
    $w' \leftarrow w + \omega(u, v)$;
    if $v = t$ **then**
      if $w' < \Delta_{s,t}$ **then**
        $\sigma'(v) = \sigma'(v) + 1 \; \forall v \in P$;
        $\sigma_T = \sigma_T + 1$;
      end
    end
    else if $w' + d_{v,t} < \Delta_{s,t}$ **then** ASP(v, w')
  end
end
$\nu(v) = 0$; **if** $u \neq s$ **then** $P \xrightarrow{pop}$;

Procedure `ASP (u,w)`

For each pair of vertices s and t the algorithm exhaustively searches through all paths between them and help itself by cutting off some impossible cases for which

$$w' + d_{v,t} < \Delta_{s,t}, \quad (8)$$

where w' is already reached length with an actual path and $d_{v,t}$ is a shortest path from an actual position to the end t, is not true. At the very first sight it seems to be

super-polynomial. To see even more than that let have following graph as in Fig. 6.

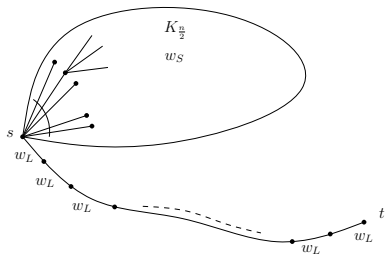


Figure 6: A graph generated from one path $P^L_{s,t}$ of $n/2+1$ vertices having large weights w_L and one complete graph $K_{\frac{n}{2}}$ on $n/2$ vertices with small weights w_S connected in a common vertex.

When searching in this graph for all δ-almost shortest paths one can for a given ϵ_P define a ratio of weights w_S/w_L such that an algorithm started at a vertex s has to search a whole subgraph $K_{\frac{n}{2}}$ which takes $(\frac{n}{2}-1)!$ steps. When a length of the search path $P_{s,i} \in K_{\frac{n}{2}}$ is summed up with a length of a shortest path from i to t as

$$\omega(P_{s,i}) + \omega(P^S_{i,t}) = (i+1)w_S + \frac{n}{2}w_L,$$

combined with a shortest path from s to t having a length of $\omega(P^S_{s,t}) = n/2 \cdot w_L$ and substituted in a condition 8 it gives a following inequality:

$$(i+1)w_S + \frac{n}{2}w_L < (1+\epsilon_P)\frac{n}{2}w_L.$$

When $i = n/2$ this inequality turns into condition:

$$w_S < \frac{\epsilon_P}{2}w_L. \tag{9}$$

So if weights are chosen in such way it ends in $\mathcal{O}(n!)$ steps and therefore a time complexity of this algorithm is at least $\mathcal{O}(n!)$, so the next step is to decide:

Conjecture 6.1 *There exists a polynomial algorithm for betweenness based on $\delta(s,t)$-almost shortest path.*

7. Results from computations

The results of computations are values $C_B(v)$ and $C^\delta_B(v)\ \forall v \in V$. For each $v \in V$ define a coefficient

$$\psi_B = \frac{(C^{delta}_B(v) - C_B(v))}{\max\{C(v)^{delta}_B - C_B(v)\}} \tag{10}$$

This coefficient represents normalized differences between both measures. One of the results from computations on data specified in section 3 are shown in Fig. 7.

Because there are usually a lot of small differences the results are thresholded by ± 0.3. And for those that has characteristic from equation 10 out of small differences region the values for C^δ_B and C_B are plotted.

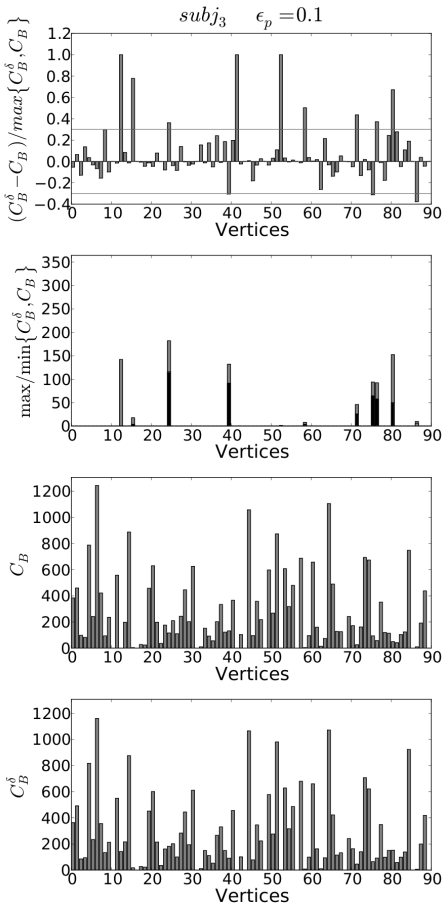


Figure 7: Results of computation of betweenness centrality

There are not very significant changes in distribution of values, but there are some differences which can even get previously zero element to high nonzero value. Similar results were obtained on other subjects.

8. Conclusion and discussion

For this specific neural measurements on a data set with its relatively small number of interacting elements and for small values of ϵ_P the results seem to be very similar for both measures. Some of the vertices can start to be higher valued when other paths are permitted by using δ-almost shortest path version of an algorithm, but this phenomenon in not significant in this case from a perspective of whole betweenness centrality distribution. There should be some additional analysis of data on a larger scale (more vertices) or of a different type (e.g. electroencephalography data EEG or climatic data).

However results can be also considered as promising when talking about δ-betweenness centrality as a comparative approach to a random walk betweenness [24] which includes in a computation possibly all walks between two vertices (based on probability) or a distance scaled betweenness [14] that scales each fraction from equation 3 by a distance of actually computed end vertices. With this approach even larger values of ϵ_P can be used, because even more non-shortest paths are accepted, and it is believed that results start to be more different. The problem with this approach can be a fact that walks respectively paths in mentioned approaches are somehow scaled by its weights where δ-betweenness just accept more and more paths. However such a comparison can be valuable if it can be proven that for a relatively small ϵ_P some problems arising when only shortest path definition is used can be solved.

Except these analysis a more detailed study can be done by testing a behavior of an algorithm on various classes of networks, for example on scale-free networks [25]. For this purpose some tests can be done on suitably chosen random graphs (e.g. Erdös-Renyi or scale-free random graphs [26]).

References

[1] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Physics Reports*, vol. 424, no. 4-5, pp. 175 – 308, 2006.

[2] M.E.J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[3] S. Strogatz, "Exploring complex networks," *Nature*, vol. 410, pp. 268 – 276, 2001.

[4] R. Tanaka, "Scale-rich metabolic networks," *Phys. Rev. Lett.*, vol. 94, p. 168101, Apr 2005.

[5] M. Ma and A. Zeng, "The connectivity structure, giant strong component and centrality of metabolic networks," *BIOINFORMATICS*, vol. 19, no. 11, pp. 1423–1430, 2003.

[6] J. Scott, *Social Network Analysis: A Handbook*. Sage Publications, London, 2nd ed., 2000.

[7] G. Palla, A.-L. Barabasi, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, pp. 664–667, Apr. 2007.

[8] M.E.J. Newman, "Detecting community structure in networks," *The European Physical Journal B*, vol. 38, no. 2, pp. 321–330, 2004.

[9] M.E.J. Newman and J. Park, "Why social networks are different from other types of networks," *Phys. Rev. E*, vol. 68, p. 036122, Sep 2003.

[10] J. Park and M.E.J. Newman, "Origin of degree correlations in the internet and other networks," *Phys. Rev. E*, vol. 68, p. 026112, Aug 2003.

[11] R. Pastor-Satorras and A. Vespignani, *Evolution and Structure of the Internet: A Statistical Physics Approach*. Cambridge University Press, 2004.

[12] B. Bollobás, *Modern graph theory*. Springer, 1998.

[13] J. Hlinka, M. Palus, M. Vejmelka, D. Mantini, and M. Corbetta, "Nongaussian dependence and neglected information in resting fmri functional connectivity," *Neuroimage*, no. 53, 2010. in submission.

[14] S.P. Borgatti and M.G. Everett, "A graph-theoretic perspective on centrality," *Social Networks*, vol. 28, no. 4, pp. 466 – 484, 2006.

[15] L.C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215 – 239, 1978-1979.

[16] G. Sabidussi, "The centrality index of a graph," *Psychometrika*, vol. 31, pp. 581–603, Dec. 1966.

[17] M.A. Beauchamp, "An improved index of centrality," *Behavioral Science*, vol. 10, no. 2, pp. 161–163, 1965.

[18] M.E.J. Newman, "Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality," *Physical Review E 64*, no. 64, p. 016132, 2001.

[19] H. Wang, J.M. Hernandez, and P. Van Mieghem, "Betweenness centrality in a weighted network," *Phys. Rev. E*, vol. 77, p. 046105, Apr 2008.

[20] T.M. Chan, "More algorithms for all-pairs shortest paths in weighted graphs," in *STOC*, 2007.

[21] N. Alon, Z. Galil, and O. Margalit, "On the exponent of the all-pairs shortest path problem," *J. Compu. Syst. Sci.*, no. 54, pp. 255–262, 1997.

[22] M.L. Fredman, "New bound on the complexity of shortest path problem," *J. Comput.*, no. 5, pp. 49–60, 1976.

[23] U. Brandes, "A faster algorithm for betweenness centrality," *Journal of Mathematical Sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[24] M.J. Newman, "A measure of betweenness centrality based on random walks," *Social Networks*, vol. 27, no. 1, pp. 39 – 54, 2005.

[25] A.-L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[26] M.E.J. Newman, S.H. Strogatz, and D.J. Watts, "Random graphs with arbitrary degree distributions and their applications," *Phys. Rev. E*, vol. 64, p. 026118, Jul 2001.

Modelling and Analysis of Spontaneous Brain Activity

Post-Graduate Student:
ING. MGR. JAROSLAV HLINKA

Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ

hlinka@cs.cas.cz

Supervisor:
PROF. DOROTHEE AUER, PHD [1]
PROF. STEPHEN COOMBES, PHD [2]

[1] School of Clinical Sciences
[2] School of Mathematical Sciences
University of Nottingham
Nottingham, NG7 2RD, UK

stephen.coombes@nottingham.ac.uk

Field of Study:
Neuroimaging and Mathematical Neuroscience

The work was supported by the EC FP6 Marie Curie Action Programme (MEST-CT-2005-021170) and by the EC FP7 project BrainSync (EC: HEALTH-F2-2008-200728, CR: MSM/7E08027). I thank both my supervisors for long-term support in my work, Milan Paluš and Martin Vejmelka for valuable discussions of the nonlinearity/nongaussianity topic and Dante Mantini for providing fMRI data for the analysis reported in the second part of this paper.

Abstract

In brain imaging and neuroscience in general, the scientific and application interest in spontaneous brain activity has rapidly increased in the recent years. The key role of formalised models and effective data analysis methods for further development of the field is becoming widely accepted. In this paper we describe two applications of mathematics in the field. The first concerns modelling of the temporal properties of spontaneous brain activity fluctuations while the second assesses the potential of nonlinear analysis methods in the study of the commonly observed synchronisation patterns.

1. Introduction

The last two decades have witnessed great progress in mapping neural networks associated with task-induced brain activation using neuroimaging techniques – with functional magnetic resonance imaging (fMRI) playing a crucial role. More recently, identification of resting state networks (RSN) paved the way to investigation of spontaneous task-unrelated brain activity [1]. The first cardinal feature characterising RSN is low-frequency fluctuation (LFF, $0.01 - 0.1$Hz) of blood oxygenation level dependent (BOLD) fMRI signals synchronised between spatially distinct, but functionally connected brain areas. Specific patterns of this functional connectivity (FC) in terms of temporal synchronisation between remote neurophysiologic events are the second key feature of spontaneous brain activity.

In the first part of this paper we focus on modelling of LFF, following our study recently published in Physical Review Letters [2]. The second part deals with the potential of application of nonlinear analysis methods for FC – this work has recently been accepted to Neuroimage [3].

2. Candidate model for low-frequency fluctuation

The neuroscientific relevance of this fluctuation has been repeatedly confirmed by reports of its relation to electrophysiological measurements of brain activity [1]. Nevertheless, the role of these fluctuations as well as the underlying mechanism is still unclear. Current models of spontaneous brain activity have not yet fully addressed the question of the origin of low-frequency fluctuations. Typically, the recent modelling papers do mention LFF property of some version of the signal [4, 5], but the relation to and relevance for the neuroimaging signals is often vague. Further, all of the proposed mechanisms rely on long-range inter-regional interactions or advocate the necessary role of transmission delays and noise. In contrary, below we propose a local model of emergence of LFF not relying on particular delays and noise. For slightly more detail and colored version of the illustrations we refer the reader to our full paper [2], **available online** under Open Access.

2.1. Introduction

The principle of the proposed model lies in postulating a local feedback loop regulating the activity level based on previous memory of the localised system. As an example of such a regulatory process we have implemented a simple phenomenological model of the action of endogenous cannabinoids on synaptic activity. Indeed, other known regulatory mechanism could be also considered.

We document that the local network activity can show slow to ultra-slow fluctuations that do not have to match the timescale of the memory mechanism. Rather, they can exhibit arbitrarily slow frequencies dependent on other parameters of the model.

Endogenous cannabinoids (CBs) represent a fundamentally new class of *retrograde* messengers [6], which are released postsynaptically and bind to presynaptic CB receptors. One function of endogenous CBs is to regulate neurotransmitter release via activation of presynaptic CB receptors, allowing neurones to regulate, via feedback, their upstream neuronal inputs [7]. This suppression of upstream presynaptic release of GABA or glutamate is termed *depolarisation-induced suppression of inhibition* (DISI) or *depolarisation-induced suppression of excitation* (DISE) respectively [8, 9].

After introducing the full model, we first analyse the fast-scale behaviour of a synaptically coupled network of Morris-Lecar neurons and subsequently describe the emergence of nested fast and ultra-slow oscillations in the network when endowed with a phenomenological form of retrograde second messenger signalling that can support DISE. We hypothesise that when linked to other modules in a larger network the latter would be reflected as an ultra-slow component of the macroscopic network dynamics and could therefore underlie those seen in spontaneous brain activity (SBA).

2.2. Model description

2.2.1 Synaptically coupled network of Morris-Lecar neurons: For the single neuron we have chosen the Morris-Lecar (ML) [10] neuron model. This is a classical two dimensional conductance based model, often used as an idealized fast-spiking pyramidal neuron, written in the form

$$\dot{v} = f(v,w) + I + s(t), \qquad \dot{w} = g(v,w). \quad (1)$$

Here v plays the role of a voltage variable, w that of a gating variable, I is a fixed input and $s(t)$ represents a time varying synaptic input. The details of the functional forms for $f(v,w)$ and $g(v,w)$ can be found e.g. in [11] (with time measured in ms). The structure of the phase-plane and nullclines is recapitulated in Figure 1 for $s = 0$.

Indexing each neuron in the network with $i = 1, \ldots, N$ the synaptic drive to the i-th neuron is given by

$$s_i(t) = g_s(v_s - v(t)) \sum_{j=1}^{N} W_{ij} \sum_{m\in\mathbb{Z}} \eta(t - T_j^m), \quad (2)$$

where T_j^m is the m-th firing time of the j-th neuron, v_s the synaptic reversal potential and W_{ij} the connection strength between neurons i and j with a global conductance scaling g_s. The function $\eta(t)$ captures the shape of a conductance change in response to the arrival of an action potential. Here we choose an alpha function and write $\eta(t) = \alpha^2 t e^{-\alpha t} H(t)$, where H is a Heaviside step function. The firing times are specified in terms of a threshold h according to

$$T_i^m = \inf\{t \mid v_i(t) > h, \dot{v}_i > 0, t > T_i^{m-1}\}. \quad (3)$$

We focus on the case of an excitatory globally coupled network with homogeneous connectivity and therefore set $W_{ij} = 1/N$ and $v_s = 2 > 0$ with respect to the resting state.

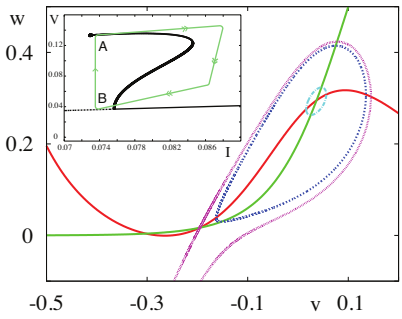


Figure 1: Phase-plane portrait for the Morris-Lecar model with constant external drive $I = 0.0761$. The voltage (gating) nullcline is in red (green). A large amplitude stable limit cycle (blue, dashed) coexists with a stable fixed point at $v \sim 0.04$. A small amplitude unstable orbit also exists (light blue, dash-dot). The separatrix (pink, stable manifold of saddle at $v \sim -0.2$) delimits the basin of attraction for the stable fixed point at $v \sim -0.3$. The associated bifurcation diagram illustrating bistability of the large amplitude limit cycle and the fixed point at $v \sim 0.04$ is shown in the inset. Here unstable orbits emerge in a Hopf bifurcation.

2.2.2 DISE mechanism: The endocannabinoid level is directly linked to the effective depolarisation, which we define by:

$$v_e(t) = \frac{1}{N} \sum_{j=1}^{N} \int_0^\infty K(t-s) v_j(s) ds, \quad (4)$$

where K is a temporal kernel reflecting the cannabinoid dynamics, $K(t) = 0$ for $t < 0$. Here we choose $K(t) = \lambda e^{-\lambda t} H(t)$, where λ^{-1} is an indirect measure of the long time-scale for cannabinoid dynamics, which is on the order of tens of seconds to minutes [9]. As a minimal model of DISE we assume that if the global CB level is sufficiently high then all excitatory synapses are blocked. In this case the network becomes uncoupled in the sense that excitatory synaptic currents drop to zero.

We implement this model of DISE by letting the firing threshold adjust in response to $v_e(t)$ according to

$$h = \begin{cases} v^{\text{th}} & v_e \leq v_e^{\text{th}} \\ \infty & v_e > v_e^{\text{th}} \end{cases}. \qquad (5)$$

The threshold v_e^{th} controls when the level of CB is sufficient to trigger DISE. In essence the model (5) means that synaptic interaction is curtailed if the effective level of depolarisation becomes too large.

2.3. Results

2.3.1 Model properties: Before we focus on the effects of DISE on the network dynamics, we analyse the fast-scale dynamics of the network. We first focus on the most symmetric oscillatory states expected to exist in a globally coupled system – namely the fully synchronous and asynchronous 'splay' (evenly distributed) solution. These are guaranteed by symmetry arguments [12].

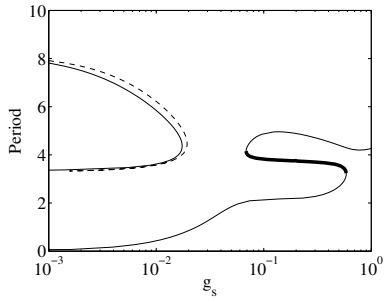


Figure 2: The period of the synchronous (solid line) and splay (dashed line) solutions as function of the coupling strength g_s. The numerically stable section of the synchronous solution branch for strong coupling is shown in thick line.

Synchronous solutions: In the synchronous state all neurons have identical T-periodic trajectories with firing times given by $T_i^m = mT$ for all i. In this case the synaptic drive to every neuron takes the identical form $s(t) = g_s(v_s - v(t))P(t)$, where $P(t) = \sum_{m\in\mathbb{Z}} \eta(t - mT)$ can be shown to equal to

$$P(t) = \frac{\alpha^2 e^{-\alpha t}}{1 - e^{-\alpha T}} \left[t + \frac{T e^{-\alpha T}}{1 - e^{-\alpha T}} \right], \qquad t \in [0, T), \qquad (6)$$

with $P(t)$ periodically extended outside $[0, T)$. Equation (1) may then be solved as a periodic boundary value problem (PBVP) for the periodic orbit $(v(t), w(t)) = (v(t + T), w(t + T))$ with $v(0) = v^{\text{th}}$. This describes the synchronous orbit given that the corresponding mean depolarisation does not trigger the DISE mechanism.

We solve this PBVP numerically, using XPPAUT [13]. The period of these solutions as a function of the coupling g_s is shown in Figure 2.

While these synchronous solutions must exist for small enough g_s, using a weakly coupled oscillator description with standard techniques reviewed in [14], we established that such solutions are unstable.

Splay solutions: For an asynchronous splay state the firing times are given by $T_j^m = mT + jT/N$. In the limit $N \to \infty$ network averages may be replaced by time averages due to:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{j=1}^{N} F(jT/N) = \frac{1}{T} \int_0^T F(t)dt, \qquad (7)$$

for any T-periodic function $F(t) = F(t + T)$. Hence a splay state in which all neurons fire is given by $v_i(t) = v(t + iT/N)$, where $v(t)$ is a T-periodic solution of (1) with $s(t) = g_s(v_s - v(t))P_0$ and $P_0 = \int_0^T P(t)dt/T = 1/T$.

Solving the PBVP, we find that for small g_s the splay state has a similar period to that of the synchronous solution (see Figure 2). Note that for the splay state $v_e(t)$ takes on the constant value $v_0 = \int_0^T v(t)dt/T$. This is lower than the DISE threshold and therefore the DISE mechanism is not triggered. A weak-coupling analysis shows that the splay solution is also unstable.

Clustered solutions: While the instability of both the synchronous and splay solution for weak coupling can be determined semi-analytically, direct numerical simulations of the network suggest that another specific stable oscillatory solution exists even for the weak coupling. This has a hybrid form where the network splits into several clusters of fully synchronised neurons. These clusters then form a splay with evenly distributed phases.

Interestingly this type of solution typically further combines with a special type of clustered solution that can also occur for a wide range of g_s. This type of solution can be predicted purely from the theory. Consider two clusters of neurons; one in a (clustered) splay state, populating the orbit corresponding to the stable limit cycle of a single neuron; the other cluster consistsing of neurons sitting at rest at the central fixed point, which is stable for a sufficient level of synaptic input from the oscillating cluster of neurons. This can be described using

the differential-algebraic system

$$\dot{v} = f(v,w) + I + r\frac{g_s}{T}(v_s - v), \quad \dot{w} = g(v,w),$$
$$0 = f(\overline{v},\overline{w}) + I + r\frac{g_s}{T}(v_s - \overline{v}), \quad 0 = g(\overline{v},\overline{w}), \quad (8)$$

where r is the fraction of firing neurons and $(v(t + iT/M), w(t + iT/M))$ with $M = Nr$, and $(\overline{v},\overline{w})$ describe neurons in the splay and resting cluster respectively. In this case $v_e = rv_0 + (1-r)\overline{v}$.

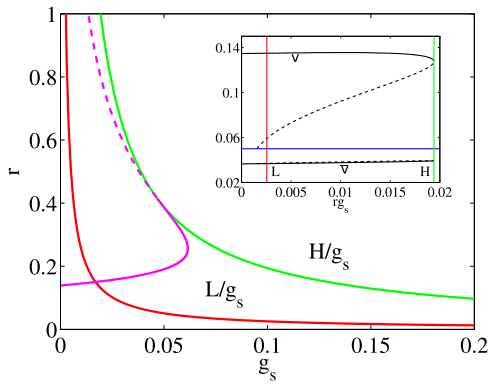


Figure 3: Fraction of firing neurons r as a function of the synaptic coupling strength g_s. See text for details.

For $v_e < v_e^{\text{th}}$ the parameter region of existence for such a solution is illustrated in the inset of Figure 3, where a pair of splay states (with $r \neq 1$) only coexists with a rest state for $rg_s \in [L, H]$. Here the splay state is annihilated in a saddle-node bifurcation at $rg_s = H$, while below $rg_s = L$ the central fixed point becomes unstable (assuming the oscillating cluster sitting at the upper branch of the limit cycle solution). For fixed rg_s, as g_s is increased, v_e grows until it reaches v_e^{th} and activates the DISE mechanism. The border in the (r, g_s) parameter plane where $v_e = v_e^{\text{th}}$ for a cluster state is shown in Figure 3 (magenta line), and we see that it defines a critical curve marking the onset of DISE which we can write in the form $g_s = g_c(r)$. In the absence of DISE, cluster states with limit cycle corresponding to the upper branch of the limit cycle solution would exist for a greater area of parameter space defined by the right-infinite strip between the lines L/g_s and H/G_s.

For $g_s < g_c(r)$ direct numerical simulations do indeed show cluster states with properties in excellent agreement with the solution of (8) (with $v(0) = v^{\text{th}}$) up to small fluctuations. An example is shown in Figure 4.

For a given value of g_s the fraction of neurons r in the firing state is a function of initial data, as expected. Importantly, after transients, the mean depolarisation signal is flat (no oscillations) and the period of oscillation of a firing neuron is of the same order of magnitude as a single isolated neuron.

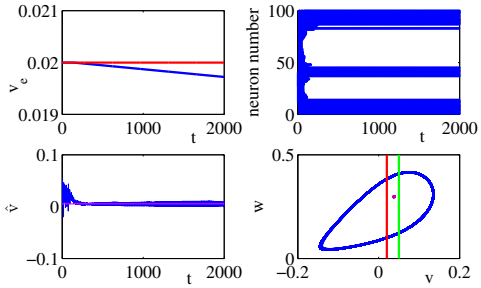


Figure 4: A cluster state for $N = 100$, $g = 0.03$, $v_e^{\text{th}} = 0.02$, $\lambda = 10^{-5}$. Top left: a plot of the average signal $v_e(t)$, showing that after transients the emergent state lies below the threshold to activate DISE (red line). Top right: A raster plot of spike times, illustrating the drop-out of some neurons and the emergence of a splay state with the fraction of firing neurons $r = 0.38$. Bottom left: The average network potential $\widehat{v} = \sum_{i=1}^{N} v_i/N$ oscillates around the predicted value $rv_0 + (1-r)\overline{v}$ (magenta line) for $r = 0.38$. Bottom right: Phase plane dynamics for the network (dropping transients) showing that the network has split into two clusters (one with a common periodic orbit shown in blue with a period $T \sim 6$ and a rest state in purple). $v^{\text{th}} = 0.05$ (green line), $v_e^{\text{th}} = 0.02$ (red line).

2.3.2 Emergence of LFF: In the region where $g_s > g_c(r)$ and DISE precludes the existence of the above discussed cluster state we expect more exotic non-periodic network states to emerge. Notably, while stable synchronous oscillations are possible with increasing g_s, the average depolarisation for these rhythms is relatively high and also an increasing function of g_s. Hence there is also a critical value of g_s at which the DISE mechanism will also preclude the existence of this periodic *synchronous* state.

The mechanism for LFF emergence for this stronger coupling is as follows. A synchronous (or near synchronous) solution can lead to a strong level of average depolarisation for which $v_e(t) > v_e^{\text{th}}$. This activates the DISE mechanism, precluding further synaptic input and subsequently leading to a drop in network firing activity and hence a drop in $v_e(t)$. Once $v_e(t)$ drops sufficiently to cross v_e^{th} from above then excitatory synaptic currents can once again drive the network leading to an increase in $v_e(t)$ so that the process may repeat over. In this case the emergent time scale of the network rhythm is set by the duration of $v_e(t)$ above v_e^{th}. Even for a synchronous solution this will depend on initial data, so that network oscillations would not generically be periodic.

To quantify the value of possible inter-spike intervals (ISIs) we focus on synchronous rhythms with

$(v(0), w(0)) = (v^{\text{th}}, w_0)$ for some given w_0 and solve the BVP $v_e(0) = v_e^{\text{th}} = v_e(\Delta)$ with $s(t) = g_s(v_s - v(t))P(t)$.

The growth of the ISI, Δ, as a function of g_s is shown in Figure 5, together with results from direct simulations. The numerical spread of ISIs for low g_s can be ascribed to fast multi-spike bursts. With higher g_s a single spike response is more common and the period of the network state is accurately predicted by the theory.

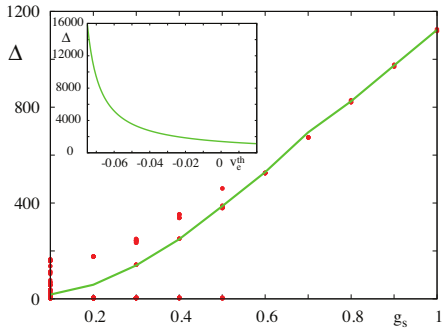


Figure 5: The predicted synchronous population ISI (in ms) as a function of g_s, for $w_0 = 0.121$ (green line), fits the ISIs seen in direct numerical simulations with $N = 100$ (red dots). Other parameters as in Figure 4. The inset shows the increase in ISI with decreasing v_e^{th} for $g_s = 1$.

Note that the spike times considered here are only those that contribute to synaptic currents, while the neurons do in fact spike on a fast time scale during the synaptically silent period. Hence, the network as a whole shows *nested* oscillations with a slow variation of synaptic currents superimposed on fast oscillations of the instantaneous average network voltage (see Figure 6 bottom left).

To understand how decreasing v_e^{th} can lead to a rapidly increasing Δ, as shown in the inset of Figure 5, it is useful to develop the correspondence of the evolution of the network (fixed parameters) with that of a single neuron with varying background drive I. Referring to the inset of Figure 1 the network can leave point A, corresponding to a synchronous firing state with average voltage v_2, when $v_e(t)$ drops below v_e^{th}. The subsequent large increase in synaptic drive causes a transition to the right of the saddle-node of periodics, where firing is not possible, and so synaptic currents fall which causes the transition to point B. This unstable fixed point, with voltage v_1, generates orbits which spiral outward for a time $T_1 = T_1(g_s)$ generating a signal with $v_e(t) > v_e^{\text{th}}$ (so that synaptic currents are suppressed). These transition to full blown nonlinear oscillations, with average voltage v_2 and $\dot{v}_e(t) < 0$, and complete the path to point A so that the process may repeat over. Making the convenient (and obviously not accurate) assumption that $v_{1,2}$ are constant then the BVP may be solved by hand for $\lambda = 0$ to give $\Delta = T_1(g_s)(v_1 - v_2)/(v_e^{\text{th}} - v_2)$, explaining the dependence of Δ on v_e^{th} seen in Figure 5.

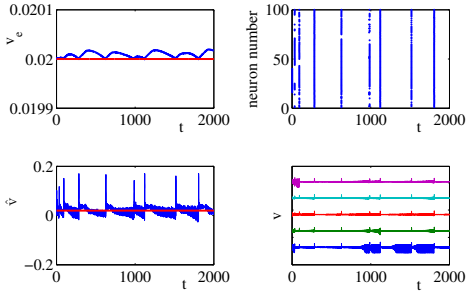


Figure 6: A similar plot to Figure 4 showing the emergence of slow synchronized firing patterns in the strong coupling regime with $g_s = 0.5$. Other parameters as in Figure 4. Bottom right shows voltage traces of 5 neurons (arbitrary offset for better display).

2.4. Discussion

Importantly, without any parameter fine-tuning, we see the emergence of very large ISIs for large values of g_s, which are largely independent of the network size. Moreover, in contrast to other network models of slow oscillations (< 1 Hz) [15] we do not require a mixture of excitation and inhibition. As shown in the inset of Figure 5 with decreasing choices of v_e^{th} can easily achieve ISIs on the order of tens of seconds. Thus DISE in the strong coupling regime is a candidate mechanism for the generation of ultra-slow rhythms.

3. Functional connectivity - analysing dependence patterns

3.1. Introduction

In functional neuroimaging, the most widely spread method of measuring functional connectivity between a pair of regions is computing a linear correlation of activity time series derived from these regions by e.g. simple spatial averaging across all the voxels in the regions. Linear correlation is also widely used to obtain so-called correlation maps by correlating the seed voxel or seed region signal with signal from all the other voxels in the brain, or constrained to gray matter area. From all possible bivariate measures of association, linear correlation is clearly a method of first choice, reflecting the assumption that the relationship between the fMRI time series can be suitably approximated by a multivariate Gaussian white noise process. Additionally, linear correlation is a well-known statistical concept, sufficiently simple to allow wide use and easy communication of results between researchers of diverse backgrounds.

On the other hand, from the mid-1980's, nonlinear approaches to analysis of brain signals are getting increased interest of researches who consider nonlinearity as an intrinsic property of brain dynamics, see e.g. [16] for a review. Hemodynamic nonlinearities are known to affect the blood oxygenation level-dependent (BOLD) fMRI signal [17]. More specifically, non-linearity of dependence between fMRI time series during resting state has been reported [18]. Use of non-linear measures of functional connectivity for the analysis of resting state data has been proposed [19], particularly including measures based on analysis of chaotic non-linear dynamical systems to analyze resting state data.

The question arises, to what extent and in what context is it justified and beneficial to use non-linear measures of functional connectivity. When linear correlation is used as a measure of functional connectivity, there are some implicit assumptions made. The first is that the information in the temporal order of the samples can be ignored (both within each timeseries and the mutual interaction). While the extent of justifiability of this assumption deserves exploration of its own, we keep this interim assumption for the purposes of this paper, not least in order to keep the comparison of linear correlation to nonlinear measures fair. Nonetheless, we ask if the instantaneous (zero-lag) dependence between the time series, expressed in the probability distribution p(X, Y), is fully captured by the linear correlation r(X,Y). We answer that this is true under the assumption of bivariate Gaussianity of the distribution. Bivariate normal distribution is fully characterised by its mean $\mu = (\mu_x, \mu_y)$ and its 2×2 covariance matrix $\mathrm{Cov}(X, Y)$ – if we allow for linear shifting and scaling, the remaining invariant parameter characterizing fully the distribution is indeed the correlation $r(X, Y)$. For a bivariate Gaussian distribution, the correlation also uniquely defines the mutual information shared between the two variables X, Y which can be computed as $I(X, Y) = I_{Gauss}(r) \equiv -\frac{1}{2}\log(1 - r^2)$.

On the other side, when the Gaussianity assumption does not hold, the distribution cannot be fully described by the mean and covariance. Interestingly, we can use the prominent properties of normal distribution to derive a useful lower bound on mutual information valid for a broad class of probability distributions. In particular, for a bivariate distribution $p(X, Y)$ with standard normal marginals $p(X)$, $p(Y)$, it holds that $I(X, Y) \geq I_{Gauss}(r) = -\frac{1}{2}\log(1 - r^2)$, where the equality holds exactly for bivariate Gaussian distributions. This allows us to quantify the deviation from Gaussianity as the difference between the total mutual information of the two variables $I(X, Y)$ and the mutual information $I_{Gauss}(r) = -\frac{1}{2}\log(1-r^2)$ that correspond to bivariate Gaussian distribution with the observed correlation r.

While there are many potential nonlinear FC measure candidates, mutual information holds a specific position among these for its generality. In theory, it is general enough to capture an arbitrary form of dependence relation between the variables without any apriori model restrictions on its form. The properties of mutual information allow us not only to test the suitability of linear correlation through probing the Gaussianity of the fMRI time series, but also to construct a quantitative estimate of connectivity information neglected by the use of linear correlation. This gives the amount of additional information available and bounds the potential contribution of non-linear alternatives over the Pearson correlation coefficient.

We implement the outlined ideas by comparing the total mutual information between the signals with the mutual information between the signals in surrogate datasets. These surrogates are generated in a way that preserves the linear correlation, but cancels any nonlinear information by enforcing bivariate Gaussian distribution on the surrogate signal-pair. This approach allows us to both test and quantify the deviation from Gaussianity, providing a principled guide in judging the suitability of linear correlation as a measure of FC. The focus on bivariate Gaussianity as the crucial condition of suitability of use of linear correlation as FC index, along with the illustrative quantitative estimation of the deviation from Gaussianity by means of the mutual information neglected by linear correlation, are the two main contributions of this study to the discussion of fMRI functional connectivity methods. We apply the presented method to parcel-average time series obtained from resting state fMRI BOLD signal of healthy subjects, testing and quantifying the deviation from bivariate Gaussianity.

3.2. Material and Methods

3.2.1 Data: Twelve right-handed healthy young volunteers (5 males and 7 females, age range 20 – 31 years) participated in the study. Each volunteer underwent two scanning runs of 10 minutes in a resting-state condition.

Scanning was performed with a 3T MR scanner (Achieva; Philips Medical Systems). Three hundred 3D-volumes with repetition time of 2 seconds, whole brain coverage and spatial resolution of $3 \times 3 \times 3$ mm^3 were used for the analysis. Standard preprocessing steps were applied, see our full paper [3] for details.

Ninety parcels from the Automated Anatomical Labeling (AAL) atlas were used to extract mean BOLD time series after masking out non-gray matter voxels. The anatomical positions of the parcels are described in [20]. Every parcel time series was orthogonalized with respect to motion parameters and global mean signal and high-pass filtered at 1/120Hz.

3.2.2 Analysis: As already mentioned in the Introduction, for a bivariate distribution $p(X,Y)$ with standard normal marginals $p(X)$, $p(Y)$, it holds that

$$I(X,Y) \geq I_{Gauss} = -\frac{1}{2}\log(1-r^2), \qquad (9)$$

where the equality holds exactly for bivariate Gaussian distributions. The inequality (9) stems from the fact, that normal distribution is the *maximum* entropy distribution for a given covariance matrix (or for a given correlation, as we assume without loss of generality that $\sigma(X) = \sigma(Y) = 1$). From the relation between mutual information and entropy ($I(X,Y) = H(X) + H(Y) - H(X,Y)$) it follows that mutual information of Gaussian distribution $I_{Gauss}(r)$ is then *minimal* from all distributions of given correlation r, under the assumption of fixed marginal entropies, which is true when the marginals have standard normal distribution. Note that the assumption of normality of the marginals is far less restrictive than it might seem. First, approximate data normality is commonly assumed in areas not restricted to fMRI FC analysis. More importantly, even if we find particular data deviating strongly from normality, any sample distribution can be monotonously transformed to match normal distribution.

To assure precise non-Gaussianity estimates, we have indeed carried out this 'normalization' step. It consists in assigning the appropriate percentile to each value of a given variable and then replacing the original values of the variable by values corresponding to these percentiles in a standard normal distribution. Note that this normalization step does not affect mutual information between the time series.

For two discrete random variables X_1, X_2 with sets of values Ξ_1 and Ξ_2, the mutual information is defined as

$$I(X_1,X_2) = \sum_{x_1\in\Xi_1}\sum_{x_2\in\Xi_2} p(x_1,x_2)\log\frac{p(x_1,x_2)}{p(x_1)p(x_2)},$$

where the probability distribution function is defined by $p(x_i) = Pr\{X_i = x_i\}, x_i \in \Xi_i$ and the joint probability distribution function is $p(x_1,x_2)$ is defined analogously. When the discrete variables X_1, X_2 are obtained from continuous variables on a continuous probability space, then the mutual information depends on a partition ξ chosen to discretize the space. Here a simple box-counting algorithm based on marginal equiquantization method [21] was used, i.e., a partition was generated adaptively in one dimension (for each variable) so that the marginal bins become equiprobable. This means that there is approximately the same number of data points in each marginal bin. In this paper we used a simple pragmatic choice of $Q = 8$ bins for each marginal variable.

For each session, we have computed the mutual information (MI) for each pair of parcels, yielding a symmetric 90-by-90 matrix of MI values. To minimize bias of the MI estimates due to inevitable discretization and finite sample estimation, the MI values were further monotonously transformed to correct for these effects. This transformation map was generated using random samples from normal distributions with correlation ranging from 0 to 1 in 200 steps of 0.005. For each correlation value, 50000 such random bivariate samples with N=300 independent observations each were generated and the mean of their MI as computed by the equiquantization method was tabulated. As for bivariate Gaussian random distribution with correlation r the true MI is $I_{Gauss} = -\frac{1}{2}\log(1-r^2)$, this tabulation allows approximate transformation of estimated MI to true bivariate MI.

To compare the (total) mutual information to the portion of information conveyed in the linear correlation, for each dataset, 99 random realizations of multivariate time series preserving the linear structure but canceling the nonlinear structure were constructed, and MI was computed for these surrogates. If the original time series dependence structure was Gaussian (and therefore fully captured by the linear correlation), the MI in the surrogates should not differ from the original MI, up to some random error.

The surrogates were constructed as multivariate Fourier transform (FT) surrogates [22]: realizations of multivariate linear stochastic process which mimic individual spectra of the original time series as well as their cross-spectrum. The multivariate FT surrogates are obtained by computing the Fourier transform of the series, keeping unchanged the magnitudes of the Fourier coefficients (the amplitude spectrum), but adding the same random number to the phases of coefficients of the same frequency bin; the inverse FT into the time domain is then performed.

The idea of comparing the MI of data to MI of 'linear' surrogates rather than directly to linear correlation of data has two aspects. First, it allows a direct quantitative comparison of the nonlinear and linear connecti-

vity, while correlation and mutual information estimators have generally different properties. Second, generation of the surrogates allows direct statistical testing of the difference. However, this procedure generates 99 estimates of the linear MI for each parcel pair; one for each surrogate. While these are useful for hypothesis testing, for general presentation of the difference we use the mean value of these 99 values. In the following we refer to this as 'Gaussian' MI, and it actually closely estimates the MI of a bivariate Gaussian distribution $I_{Gauss}(r) = -\frac{1}{2}\log(1-r^2)$, where r stands for the correlation of the two variables. The 'neglected' MI is estimated by the difference between data MI and the Gaussian MI: $I_{neglected}(X,Y) = I(X,Y) - I_{Gauss}(r)$.

3.2.3 Statistical tests: For each session and each parcel pair, non-Gaussianity was tested at $p = 0.05$ by comparing data MI against MI distribution of multivariate FT surrogates. To correct for mutual comparisons, the number of significant pairs in given session was than tested against the null hypothesis that the number of individual significant entries has a binomial distribution $B(n = 4005, p = 0.05)$, where $n = 4005 = \frac{90(90-1)}{2}$ is the number of all parcel pairs and $p = 0.05$ is the single entry false positive rate under condition of pure Gaussianity of the bivariate distributions.

As it may be argued that the assumption of pair independence is too lenient, but the exact level of dependence is difficult to establish, we also carried out group level tests. The percentages of significant pairs were compared by means of a paired t-test to the percentages of significant pairs obtained from 'shadow' datasets. Each shadow dataset was created as a multivariate FT surrogate of normalized data of a given session, preserving only the linear structure of the dataset after normalisation of univariate marginals. Subsequently, each shadow dataset has undergone the same procedure as original data, including the initial normalization, generation of multivariate surrogates, computation of MI and statistical testing of pair-wise MI against surrogates. In this way, we have mimicked the full procedure using the shadow dataset, accounting for any potential bias in the detection rate introduced by numerical properties of the algorithm. Apart from the percentages, we have also tested the mean neglected information from data versus shadow datasets by mean of a paired t-test.

3.3. Results

3.3.1 Descriptive assessment: In descriptive terms, the data MI has proved very similar to the Gaussian MI (see Figure 7). In particular, averaging across all parcel pairs, the data MI ranged between 0.04-0.10 bits for different sessions, while the neglected MI was more than an order of magnitude smaller (0.0005-0.0068 bits). Nevertheless, the neglected MI was consistently positive, which was not the case for shadow datasets (ranging from -0.0007 to 0.0016 bits).

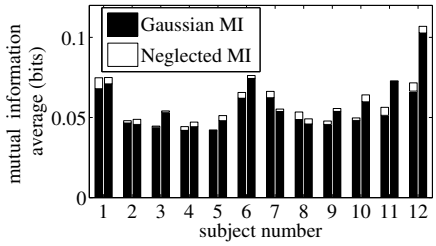


Figure 7: Comparison of the average Gaussian and neglected information. Each stackbar represents values for one session, averaged across all parcel pairs.

Independently of the strength of coupling, the data MI was moreover typically within the range of surrogate MI, as illustrated on Figure 8 top. Although the session

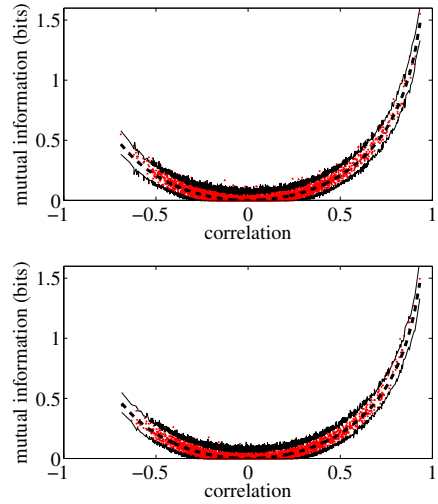


Figure 8: Mutual information as function of correlation in an example dataset (top) and the corresponding linearized 'shadow' dataset. Each dot corresponds to MI of one parcel pair; full lines show the 1st and 99th percentile of the surrogate distribution; dashed line shows the theoretical prediction for Gaussian data. The session with the most non-Gaussianity is depicted.

with the most non-Gaussianity is depicted here, the distribution of computed MI for data and the corresponding shadow dataset (Figure 9 bottom) are almost indiscernible. Also, apart from the random error due to MI estimation from short time series, which is shared by data and shadow data, both scatters follow well the theoretical prediction of dependence of MI on linear correlation ($I_{Gauss} = -\frac{1}{2}\log(1-r^2)$, valid exactly under Gaussianity).

3.3.2 Statistical tests: The percentage of parcel pairs with significant non-Gaussianity was slightly elevated in all sessions above the 5% expected under the null hypothesis (ranging from 5.3 to 10.0% of significant pairs in different sessions). If all the parcel pairs were considered independent this would constitute significant percentage for all but 5 of the sessions considered. Group level tests confirmed the statistical deviation from Gaussianity – when compared on group level by means of a paired t-test, the counts of pairs with significant nonlinearity were significantly higher than similar counts obtained from shadow datasets ($t = 6.26, \mathrm{df} = 23, p < 0.00001$). Also, the neglected information in data averaged over parcel pairs was positive for all sessions and on average had value 0.0029 bits. On the other side, the neglected information in the shadow datasets fluctuated around zero with mean of 0.0006 bits. This difference was also clearly statistically significant ($t = 6.51, \mathrm{df} = 23, p < 0.00001$).

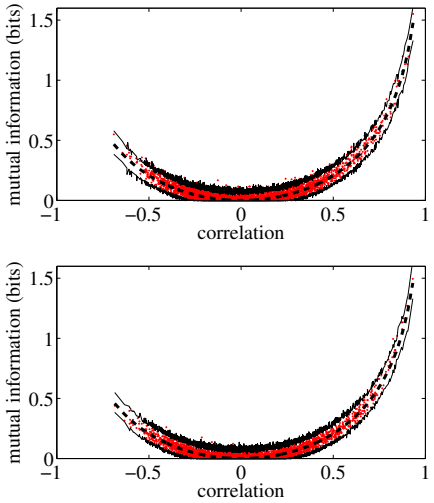


Figure 9: Mutual information as function of correlation in an example dataset (top) and the corresponding linearized 'shadow' dataset. Each dot corresponds to MI of one parcel pair; full lines show the 1st and 99th percentile of the surrogate distribution; dashed line shows the theoretical prediction for Gaussian data. The session with the most non-Gaussianity is depicted.

3.4. Discussion

The presented study confirms the suitability of linear correlation as functional connectivity measure for fMRI time series by testing and quantifying the deviation from bivariate Gaussianity using mutual information. The quantitative assessment revealed that the portion of mutual information neglected by using linear correlation instead of considering an arbitrary non-linear form of instantaneous dependence is minor. Nevertheless, formal group-level test revealed that the percentage of parcel-pairs with significant non-Gaussian dependence contribution is indeed above random. Overall we conclude that practical relevance of nonlinear methods trying to improve over linear correlation might be limited by the fact that the data are indeed almost Gaussian.

It is important to keep in mind that the observed deviations from Gaussianity might not reflect only a stationary non-Gaussianity in neuronal connectivity. In the presented framework, deviation from the null hypothesis could be caused also by nonstationarity of the signal.

For completeness, we note that linearity is also often discussed as an alternative to nonlinear, potentially chaotic deterministic dynamical systems. In this context caution is warranted with the interpretation of many 'chaotic' characteristics such as fractional correlation dimension or Lyapunov exponents when the underlying system might be of stochastic (non)linear nature rather than deterministic (non)linear dynamical system, and particularly when short time series such as those acquired from fMRI are being analyzed.

4. Summary

In this paper, we have described two exemplar contributions of mathematical modelling and analysis to the study of large-scale spatiotemporal patterns of spontaneous brain activity. This investigation inevitably led to new questions.

Challenges in spontaneous brain activity modelling

Further investigation is needed to reveal the role of regulatory feedbacks in the slow brain activity dynamics. The search should not be limited to the action of endocannabinoids that served as an example carrier of this mechanism in this paper - the role of other neuromodulators or metabolic fatigue should be investigated.

Nongaussianity in fMRI functional connectivity

The confirmation of suitability of linear correlation as a functional connectivity measure for resting state fMRI provides important support to the common practice in neuroimaging. Nevertheless, the scope of possible generalisation of the results remains an open question. Particularly interesting is the generalisation to time-lagged dependence structures and other preprocessing and time series extraction methods. Explicit modelling might have something to say regarding the emergence of largely 'linear' or 'gaussian' dependencies on the macroscopic level of regional fMRI signals from the highly nonlinear nature of the local neuronal dynamics.

Most of the questions discussed above are not only interesting intellectual challenges – the answers are crucial for further development and applications of the study of spontaneous brain activity.

References

[1] M.D. Fox and M.E. Raichle, "Spontaneous fluctuations in brain activity observed with functional magnetic resonance imaging," *Nature Reviews Neuroscience*, vol. 8, no. 9, pp. 700–711, 2007.

[2] J. Hlinka and S. Coombes, "Depolarization Induced Suppression of Excitation and the Emergence of Ultraslow Rhythms in Neural Networks," *Physical Review Letters*, vol. 104, no. 6, 2010.

[3] J. Hlinka, M. Palus, M. Vejmelka, D. Mantini, and M. Corbetta, "Non-gaussian dependence and neglected information in resting fmri functional connectivity," *NeuroImage*, In Press.

[4] C.J. Honey, R. Kotter, M. Breakspear, and O. Sporns, "Network structure of cerebral cortex shapes functional connectivity on multiple time scales," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 24, pp. 10240–10245, 2007.

[5] G. Deco, V. Jirsa, A.R. McIntosh, O. Sporns, and R. Koetter, "Key role of coupling, delay, and noise in resting brain fluctuations," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 25, pp. 10302–10307, 2009.

[6] T. Freund, I. Katona, and D. Piomelli, "Role of endogenous cannabinoids in synaptic signaling," *Physiological Reviews*, vol. 83, no. 3, pp. 1017–1066, 2003.

[7] R. Wilson and R. Nicoll, "Neuroscience - Endocannabinoid signaling in the brain," *Science*, vol. 296, no. 5568, pp. 678–682, 2002.

[8] T. Ohno-Shosaku, T. Maejima, and M. Kano, "Endogenous cannabinoids mediate retrograde signals from depolarized postsynaptic neurons to presynaptic terminals," *Neuron*, vol. 29, no. 3, pp. 729–738, 2001.

[9] A. Straiker and K. Mackie, "Depolarization-induced suppression of excitation in murine autaptic hippocampal neurones," *Journal of Physiology*, vol. 569, no. 2, pp. 501–517, 2005.

[10] C. Morris and H. Lecar, "Voltage oscillations in the barnacle giant muscle-fiber," *Biophysical Journal*, vol. 35, no. 1, pp. 193–213, 1981.

[11] S. Han, C. Kurrer, and Y. Kuramoto, "Dephasing and bursting in coupled neural oscillators," *Physical Review Letters*, vol. 75, no. 17, pp. 3190–3193, 1995.

[12] P. Ashwin and J.W. Swift, "The dynamics of n-weakly coupled identical oscillators," *Journal of Nonlinear Science*, vol. 2, no. 1, pp. 69–108, 1992.

[13] B. Ermentrout, *Simulating, Analyzing, and Animating Dynamical Systems: A Guide To Xppaut for Researchers and Students*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2002.

[14] F.C. Hoppensteadt and E.M. Izhikevich, *Weakly connected neural networks*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997.

[15] A. Compte, M. Sanchez-Vives, D. McCormick, and X. Wang, "Cellular and network mechanisms of slow oscillatory activity (¡ 1 Hz) and wave propagations in a cortical network model," *Journal of Neurophysiology*, vol. 89, no. 5, pp. 2707–2725, 2003.

[16] C. Stam, "Nonlinear dynamical analysis of EEG and MEG: Review of an emerging field," *Clinical Neurophysiology*, vol. 116, no. 10, pp. 2266–2301, 2005.

[17] J.A. de Zwart, P. van Gelderen, J.M. Jansma, M. Fukunaga, M. Bianciardi, and J.H. Duyn, "Hemodynamic nonlinearities affect BOLD fMRI response timing and amplitude," *NeuroImage*, vol. 47, no. 4, pp. 1649 – 1658, 2009.

[18] P.J. Lahaye, J.B. Poline, G. Flandin, S. Dodel, and L. Garnero, "Functional connectivity: studying nonlinear, delayed interactions between BOLD signals," *NeuroImage*, vol. 20, no. 2, pp. 962–974, 2003.

[19] X.P. Xie, Z.T. Cao, and X.C. Weng, "Spatiotemporal nonlinearity in resting-state fMRI of the human brain," *NeuroImage*, vol. 40, no. 4, pp. 1672–1685, 2008.

[20] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot, "Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain," *NeuroImage*, vol. 15, no. 1, pp. 273–289, 2002.

[21] M. Palus, V. Albrecht, and I. Dvorak, "Information theoretic test for nonlinearity in time series," *Physics Letters A*, vol. 175, no. 3-4, pp. 203–209, 1993.

[22] D. Prichard and J. Theiler, "Generating surrogate data for time series with several simultaneously measured variables," *Physical Review Letters*, vol. 73, no. 7, p. 951, 1994.

Non-Equivalence of Some Implicational Deduction Theorems

Post-Graduate Student:
MGR. KAREL CHVALOVSKÝ
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ

Department of Logic, Charles University
Celetná 20
116 42, Prague 1, CZ

chvalovsky@cs.cas.cz

Supervisor:
MGR. MARTA BÍLKOVÁ, PH.D.
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ

bilkova@cs.cas.cz

Field of Study:
Logic

This work was supported by GAAV project IAA900090703, GAUK project 73109/2009 and GAČR project 401/09/H007.
The paper is substantially based on the forthcoming paper by Petr Cintula (principal author) and the author.

Abstract

We show that some classes of logics in the hierarchy of Implicational Deduction Theorems, defined in the forthcoming paper [1], are not equal. This completes the picture of hierarchy of these logics.

1. Introduction

One of the most important theorems of classical propositional logic is the Deduction Theorem, independently discovered by Herbrand [2] and Tarski [3], which connects provability and implication. In its most popular form it says

$$\Gamma, \varphi \vdash \psi \text{ iff } \Gamma \vdash \varphi \to \psi.$$

It enables us to find some proofs much easier. However, this theorem does not hold in all logics. For example in logics without contraction, we usually have so called Local Deduction Theorem, which says that there exists some natural k such that

$$\Gamma, \varphi \vdash \psi \text{ iff } \Gamma \vdash \underbrace{\varphi \to (\dots (\varphi}_{k\text{-times}} \to \psi) \dots).$$

The problem is that generally we do not have any (reasonable) upper bound on k.

We can try to estimate k somehow. The immediate idea is to count how many times the assumption φ is used in the proof of ψ. This idea is captured in the forthcoming paper [1] where the situation is shown not to be so easy. Authors define some hierarchy of logics with Implicational Deduction Theorems and investigate relations between its members. It is shown that this hierarchy collapses on some level. In this paper we show in full details that some of its members are not the same.

This is shown by presenting, for the remaining case, a counter-example. It is worth to note that this counter-example was found with the help of computer. For further details, proofs and references we refer the reader to the forthcoming paper [1] mentioned already.

2. Preliminaries

We use some standard terminology from the theory of logical calculi (see e.g. [4])—a *propositional language* $\mathcal{L}$ (a set of logical connectives with some finite arity, in this paper we have just one binary connective called implication $\to$ and we use the following convention: $\varphi \to^0 \psi = \psi$ and $\varphi \to^{i+1} \psi = \varphi \to (\varphi \to^i \psi)$), the set of *$\mathcal{L}$-formulae* $Fle_{\mathcal{L}}$ over some fixed countably infinite set of propositional variables and *$\mathcal{L}$-substitutions*. An *$\mathcal{L}$-theory* Γ is a set of $\mathcal{L}$-formulae. An *$\mathcal{L}$-consecution* $\Gamma \rhd \varphi$ is a pair consisting of a theory Γ and a formula φ.

A *logic* $\mathbf{L}$ in the language $\mathcal{L}$ is a structural consequence relation (in the sense of Tarski) on $Fle_{\mathcal{L}}$. That is, $\mathbf{L}$ is a set of relations between theories and formulae (writing $\Gamma \vdash_{\mathbf{L}} \varphi$, and $\Gamma \vdash_{\mathbf{L}} \Gamma'$ as an abbreviation for $\Gamma \vdash_{\mathbf{L}} \varphi$ for each $\varphi \in \Gamma'$) satisfying the following conditions:

(i) If $\varphi \in \Gamma$, then $\Gamma \vdash_{\mathbf{L}} \varphi$.
(ii) If $\Gamma \vdash_{\mathbf{L}} \Gamma'$ and $\Gamma' \vdash_{\mathbf{L}} \varphi$, then $\Gamma \vdash_{\mathbf{L}} \varphi$.
(iii) If $\Gamma \vdash_{\mathbf{L}} \varphi$, then there is a finite set $\Gamma' \subseteq \Gamma$ s.t. $\Gamma' \vdash_{\mathbf{L}} \varphi$.
(iv) If $\Gamma \vdash_{\mathbf{L}} \varphi$, then $\sigma(\Gamma) \vdash_{\mathbf{L}} \sigma(\varphi)$ for any $\mathcal{L}$-substitution σ.

The previous conditions are called reflexivity, cut, finitarity and structurality.

Definition 1 *An* axiomatic system $\mathcal{AX}$ *is a set of finitary consecutions closed under substitutions. The members of* $\mathcal{AX}$ *with non-empty theories are called* deductive rules*, these with empty theories are called* axioms*. We say that* $\mathcal{AX}$ *is* MP-based *if modus ponens is its only deduction rule.*

Note that we have only finitary rules, and axioms as well as rules are presented by schemata.

Definition 2 *Let* $\mathcal{AX}$ *be an axiomatic system. An* $\mathcal{AX}$-proof *of the formula* φ *in theory* Γ *is a finite tree labelled by formulae satisfying*

(i) *the root is labelled by* φ,
(ii) *leaves by either axioms or elements of* Γ,
(iii) *if a node is labelled by* ψ *and its preceding nodes are labelled by* $\psi_1, \ldots, \psi_n$ *then* $\{\psi_1, \ldots, \psi_n\} \rhd \varphi \in \mathcal{AX}$.

If such a proof exists we write $\Gamma \vdash^p_{\mathcal{AX}} \varphi$.

We say that $\mathcal{AX}$ is an *axiomatic system* for (a presentation of) a logic $\mathbf{L}$ iff $\mathbf{L} = \vdash^p_{\mathcal{AX}}$. A logic $\mathbf{L}$ is *MP-based* if it has some MP-based presentation.

2.1. Matrix models and semantics

A *matrix* $\mathbf{M}$ for $\mathcal{L}$ is a pair $\langle A, D\rangle$, where A is an $\mathcal{L}$-*algebra* and $D \subseteq A$ is the *set of designated elements* of $\mathbf{M}$. An $\mathbf{M}$-*evaluation* for matrix $\mathbf{M} = \langle A, D\rangle$ is a mapping $e : Fle_{\mathcal{L}} \to A$ which commutes with all connectives in $\mathcal{L}$.

Logics can be defined semantically through logical matrices. Any class of $\mathcal{L}$-matrices $\mathcal{C}$ is called *matrix semantics* for $\mathcal{L}$. We say $\Gamma \models_{\mathcal{C}} \varphi$ iff for each $\mathbf{M} \in \mathcal{C}$, $\mathbf{M} = \langle A, D\rangle$, and each evaluation e in $\mathbf{M}$, $e(\varphi) \in D$ whenever $e[\Gamma] \subseteq D$.

Notice that $\models_{\mathcal{C}}$ is a logic for $\mathcal{C}$ being a finite set of finite $\mathcal{L}$-matrices. By relaxing either of the finiteness conditions we obtain a consequence relation, but not necessarily a finitary one.

3. Implicational Deduction Theorems

In this section we define some basic notions concerning the study of Implicational Deduction Theorems. First, we define an analog to a Local Deduction Theorem.

Definition 3 *A logic* $\mathbf{L}$ *has* Simple Implicational Deduction Theorem *(IDT$_0$) if for each theory* Γ *and formulae* φ, ψ*:*

$$\Gamma, \varphi \vdash_{\mathbf{L}} \psi \text{ iff there is } n \text{ such that } \Gamma \vdash_{\mathbf{L}} \varphi \to^n \psi.$$

We immediately obtain the following important property of logics with IDT$_0$, which is a consequence of our assumptions concerning finitarity.

Lemma 1 *A logic* $\mathbf{L}$ *with IDT$_0$ is MP-based.*

Now we present a finer analysis of Local Deduction Theorems arising from the idea of counting number of occurrences of φ in the leaves of some proof of ψ in Γ and φ.

Definition 4 *Let* $n > 0$*. A logic* $\mathbf{L}$ *has* n-Implicational Deduction Theorem *(IDT$_n$) if*

(i) $\mathbf{L}$ *has an MP-based presentation* $\mathcal{AX}$,
(ii) *for each theory* Γ*, formula* ψ*, mutually different formulae* φ_i*,* $1 \leq i \leq n$*, and for each* $\mathcal{AX}$*-proof* $\mathcal{P}$ *of* ψ *in* $\Gamma \cup \{\varphi_i \mid 1 \leq i \leq n\}$*:*

$$\Gamma \vdash \varphi_1 \to^{j_1} (\varphi_2 \to^{j_2} \ldots (\varphi_n \to^{j_n} \psi) \ldots),$$

where j_i *is the number of occurrences of* φ_i *in the leaves of* $\mathcal{P}$*.*

It may seem that eg. IDT$_2$ can be obtained just by double application of IDT$_1$, but it is not true.

Example 1 *Let we assume that*

$$\varphi, \psi, \varphi \to (\psi \to \chi) \vdash \chi. \tag{1}$$

IDT$_2$ gives

$$\varphi \to (\psi \to \chi) \vdash \psi \to (\varphi \to \chi) \tag{2}$$

and IDT$_1$ gives

$$\psi, \varphi \to (\psi \to \chi) \vdash \varphi \to \chi, \tag{3}$$

but now we cannot use IDT$_1$ once again to obtain (2)*. We only know that* $\varphi \to \chi$ *is provable from* ψ *and* $\varphi \to (\psi \to \chi)$*, but we do not know how many times* ψ *has to be used.*

From now on, we shall use IDT$_n$ also for the class of all logics satisfying IDT$_n$. The meaning will be obvious from context.

In the paper [1] we prove

Theorem 1

(i) If a logic $\mathbf{L}$ *has IDT*$_n$ *then* $\mathbf{L}$ *has IDT*$_m$ *for any* $m \leq n$.
(ii) If a logic $\mathbf{L}$ *has IDT*$_3$ *then* $\mathbf{L}$ *has IDT*$_m$ *for any* $m \geq 3$.
(iii) IDT$_0 \neq$ IDT$_1$ *and* IDT$_1 \neq$ IDT$_2$.

The previous theorem shows that the hierarchy of logics with Implicational Deduction Theorems has the following properties

$$\mathrm{IDT}_0 \supsetneq \mathrm{IDT}_1 \supsetneq \mathrm{IDT}_2 \supseteq \mathrm{IDT}_3 = \mathrm{IDT}_4 = \dots$$

The remaining problem is whether $\mathrm{IDT}_2 = \mathrm{IDT}_3$ or not, which is the result of this paper. We solve this problem negatively in the next section. Before we proceed we recall an important characterisation lemma:

Lemma 2 *Let* $\mathbf{L}$ *be a logic and* $n > 0$ *then* $\mathbf{L}$ *has IDT*$_n$ *iff*

(i) $\mathbf{L}$ *is MP-based,*
(ii) $\vdash_{\mathbf{L}} \varphi \to \varphi$,
(iii) for each natural a_i, b_i, *for* $1 \leq i \leq n$, *holds*

$$\varphi_1 \to^{a_1} (\dots(\varphi_n \to^{a_n} (\chi \to \psi))\dots),$$
$$\varphi_1 \to^{b_1} (\dots(\varphi_n \to^{b_n} \chi)\dots)$$
$$\vdash_{\mathbf{L}} \varphi \to^{a_1+b_1} \dots(\varphi_n \to^{a_n+b_n} \psi)\dots).$$

4. IDT$_2 \neq$ IDT$_3$

In this section we prove that there is a logic with IDT$_2$ but without IDT$_3$. The proof is based on the matrix $\mathbf{M}$ in Table 1. The only denoted element of $\mathbf{M}$ is $\mathbf{1}$. Let us note that the matrix was found with the help of computer.

$\to$	$\mathbf{1}$	$\mathbf{a}$	$\mathbf{b}$	$\mathbf{c}$	$\mathbf{d}$	$\mathbf{e}$	$\mathbf{f}$	$\mathbf{g}$
$\mathbf{1}$	$\mathbf{1}$	$\mathbf{a}$	$\mathbf{b}$	$\mathbf{c}$	$\mathbf{d}$	$\mathbf{e}$	$\mathbf{f}$	$\mathbf{g}$
$\mathbf{a}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{a}$	$\mathbf{a}$	$\mathbf{a}$	$\mathbf{c}$	$\mathbf{d}$	$\mathbf{f}$
$\mathbf{b}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{a}$	$\mathbf{a}$	$\mathbf{b}$	$\mathbf{d}$	$\mathbf{e}$
$\mathbf{c}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{a}$	$\mathbf{a}$	$\mathbf{c}$	$\mathbf{e}$
$\mathbf{d}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{a}$	$\mathbf{a}$	$\mathbf{d}$
$\mathbf{e}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{a}$	$\mathbf{b}$
$\mathbf{f}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{a}$
$\mathbf{g}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$

Table 1: Model $\mathbf{M}$.

From now on, we use the following notation. We abbreviate $e(\varphi) =_{\mathbf{M}} \mathbf{1}$ by $\varphi = \mathbf{1}$, because model $\mathbf{M}$ is fixed and the evaluation is obvious from the context. We shall also abbreviate it simply saying φ is $\mathbf{1}$.

Lemma 3 *The logic* $\mathbf{L}$ *given by model* $\mathbf{M}$ *does not have IDT*$_3$.

Proof: There is a proof of

$$\varphi, \psi, \varphi \to (\psi \to \chi) \vdash \chi,$$

where φ, ψ and $\varphi \to (\psi \to \chi)$ are used only once. IDT$_3$ would give

$$\vdash (\varphi \to (\psi \to \chi)) \to (\psi \to (\varphi \to \chi)),$$

which is not true. Consider an evaluation $e(\varphi) = \mathbf{a}, e(\psi) = \mathbf{b}$ and $e(\chi) = \mathbf{g}$. ∎

Now we shall show that $\mathbf{L}$ has IDT$_2$. First, we establish some useful properties of $\mathbf{M}$.

Observation 1 *The following statements are true in* $\mathbf{M}$*:*

(i) $\varphi \to \varphi = \mathbf{1}$,
(ii) $\mathbf{1} \to \varphi = \varphi$,
(iii) if $\varphi = \mathbf{1}$ *and* $\varphi \to \psi = \mathbf{1}$ *then* $\psi = \mathbf{1}$ *(MP holds in* $\mathbf{M}$*).*

Definition 5 *We define the* ordering $<$ *on the elements of* $\mathbf{M}$ *by*

$$\mathbf{g} < \mathbf{f} < \mathbf{e} < \mathbf{d} < \mathbf{c} < \mathbf{b} < \mathbf{a} < \mathbf{1}.$$

We use $x \leq y$ with the standard meaning $x = y$ or $x < y$. In the very same way as for $=$ we use $\varphi \leq \psi$ which means that for given evaluation e it holds that $e(\varphi) \leq e(\psi)$.

Lemma 4 *The model* $\mathbf{M}$ *has the following properties:*

(i) $\psi \leq \varphi \to \psi$,
(ii) $\varphi_0 \leq \varphi$ *implies* $\varphi \to \psi \leq \varphi_0 \to \psi$,
(iii) $\psi_0 \leq \psi$ *implies* $\varphi \to \psi_0 \leq \varphi \to \psi$.

These properties of implication in $\mathbf{M}$ play a very important role in the rest of the section.

Lemma 5 *For any evaluation such that* $\varphi_1, \varphi_2, \varphi_3, \varphi_4$ *are different from* $\mathbf{1}$ *holds*

$$\varphi_1 \to (\varphi_2 \to (\varphi_3 \to (\varphi_4 \to \psi))) = \mathbf{1}.$$

Proof: By Lemma 4 the worst case is $\varphi_1 = \varphi_2 = \varphi_3 = \varphi_4 = \mathbf{a}$ and $\psi = \mathbf{g}$, but even for such evaluation lemma holds. ∎

Observation 1 and Lemma 5 are very important. From Lemma 2 we immediately obtain that we only need to check finitely many cases to show that logic $\mathbf{L}$ given by model $\mathbf{M}$ has IDT_2.

Corollary 1 *The logic* $\mathbf{L}$ *given by model* $\mathbf{M}$ *has IDT*$_2$ *iff*

$$\varphi_1 \to^{a_1} (\varphi_2 \to^{a_2} \to (\chi \to \psi)) = \mathbf{1} \tag{4}$$

and

$$\varphi_1 \to^{b_1} (\varphi_2 \to^{b_2} \chi) = \mathbf{1} \tag{5}$$

imply

$$\varphi_1 \to^{a_1+b_1} (\varphi_2 \to^{a_2+b_2} \psi) = \mathbf{1} \tag{6}$$

for any $a_1 + a_2 + b_1 + b_2 < 4$.

Now we can proceed by exhaustive checking of all possible variants, or we can simplify our work significantly as shown by following three lemmata.

Lemma 6 *Given* (4) *and* (5)*, and if*

(i) $\psi = \mathbf{1}$,
(ii) $\chi = \mathbf{1}$,
(iii) $\chi = \mathbf{g}$,
(iv) $\varphi_1 = \mathbf{g}$ *(for* $a_1 + b_1 > 0$*)*,
(v) $\varphi_2 = \mathbf{g}$ *(for* $a_1 + b_1 > 0$*)*,

then the condition (6) *holds.*

Proof: Cases (i), (iv) and (v) are evident. If $\chi = \mathbf{1}$ then

$$\begin{aligned} \mathbf{1} &= \varphi_1 \to^{a_1} (\varphi_2 \to^{a_2} (\mathbf{1} \to \psi)) \\ &= \varphi_1 \to^{a_1} (\varphi_2 \to^{a_2} \psi) \\ &\leq \varphi_1 \to^{a_1+b_1} (\varphi_2 \to^{a_2+b_2} \psi). \end{aligned}$$

If $\chi = \mathbf{g}$ then $\chi \leq \psi$ and hence

$$\begin{aligned} \mathbf{1} &= \varphi_1 \to^{b_1} (\varphi_2 \to^{b_2} \chi) \\ &\leq \varphi_1 \to^{b_1} (\varphi_2 \to^{b_2} \psi) \\ &\leq \varphi_1 \to^{a_1+b_1} (\varphi_2 \to^{a_2+b_2} \psi). \end{aligned}$$

∎

Lemma 7 *Given* (4) *and* (5)*, and if*

(i) $a_1 + a_2 = 0$,
(ii) $b_1 + b_2 = 0$,

then the condition (6) *holds.*

Proof: Case (i) gives $\chi \to \psi = \mathbf{1}$ hence $\chi \leq \psi$ and then (5) implies (6). Case (ii) is even easier, because $\chi = \mathbf{1}$ and then (4) implies (6). ∎

Lemma 8 *Given* (4) *and* (5)*, and if*

(i) $a_1 = 1$, $a_2 = 0$, $b_1 = 1$, $b_2 = 0$,
(ii) $a_1 = 1$, $a_2 = 0$, $b_1 = 0$, $b_2 = 1$,
(iii) $a_1 = 0$, $a_2 = 1$, $b_1 = 0$, $b_2 = 1$,
(iv) $a_1 = 2$, $a_2 = 0$, $b_1 = 1$, $b_2 = 0$,
(v) $a_1 = 2$, $a_2 = 0$, $b_1 = 0$, $b_2 = 1$,
(vi) $a_1 = 0$, $a_2 = 2$, $b_1 = 0$, $b_2 = 1$,
(vii) $a_1 = 1$, $a_2 = 1$, $b_1 = 0$, $b_2 = 1$,

then the condition (6) *holds.*

Proof: Let us show case (i). Other cases are more or less similar. We have

$$\varphi_1 \to (\chi \to \psi) = \mathbf{1}, \tag{7}$$
$$\varphi_1 \to \chi = \mathbf{1}. \tag{8}$$

Now we have $\varphi_1 \leq \chi$ from (8) and hence $\chi \to \psi \leq \varphi_1 \to \psi$. From (7) we have $\varphi_1 \leq \chi \to \psi$. So we have $\varphi_1 \leq \varphi_1 \to \psi$. ∎

We can now on assume that $\varphi_1 \neq \mathbf{1}$ and $\varphi_2 \neq \mathbf{1}$, because $\mathbf{1} \to \varphi = \varphi$ and consequently $\varphi_1 = \mathbf{1}$ and $\varphi_2 = \mathbf{1}$ is the same as $a_1 = b_1 = 0$ and $a_2 = b_2 = 0$, respectively. In such case all the following instances lead to the cases solved already.

The remaining cases have to be checked separately and in more details. We analyse all possible evaluations and show that (4) and (5) imply (6).

Lemma 9 *Given* (4) *and* (5)*, and if* $a_1 = 0$, $a_2 = 1$, $b_1 = 1$, $b_2 = 0$*, then the condition* (6) *holds.*

Proof: We need to show that

$$\varphi_2 \to (\chi \to \psi) = \mathbf{1}, \tag{9}$$
$$\varphi_1 \to \chi = \mathbf{1} \tag{10}$$

imply

$$\varphi_1 \to (\varphi_2 \to \psi) = \mathbf{1}. \tag{11}$$

We prove it by cases. First, if $\psi = \mathbf{a}, \ldots, \mathbf{d}$ then it is easy to show that $\varphi_1 \to (\varphi_2 \to \psi) = \mathbf{1}$ (we assume $\varphi_1 \neq \mathbf{1}$ and $\varphi_2 \neq \mathbf{1}$). The only interesting cases are the following:

$\cdot\, \psi = \mathbf{e}$

- $\cdot\, \varphi_2 = \mathbf{a}$
  - $\cdot\, \varphi_1 = \mathbf{a}, \mathbf{b}$ then $\varphi_1 \leq \chi$ from (10), hence $\chi \geq \mathbf{b}$ and therefore $\varphi_2 \to (\chi \to \psi) < \mathbf{1}$.
  - $\cdot\, \varphi_1 = \mathbf{c}, \dots$ then $\varphi_1 \to (\varphi_2 \to \psi) = \mathbf{1}$.
- $\cdot\, \varphi_2 = \mathbf{b}$
  - $\cdot\, \varphi_1 = \mathbf{a}$ then $\chi \geq \mathbf{a}$ and therefore $\varphi_2 \to (\chi \to \psi) < \mathbf{1}$.
  - $\cdot\, \varphi_1 = \mathbf{b}, \dots$ then $\varphi_1 \to (\varphi_2 \to \psi) = \mathbf{1}$.
- $\cdot\, \varphi_2 = \mathbf{c}, \dots$ then $\varphi_1 \to (\varphi_2 \to \psi) = \mathbf{1}$.

$\cdot\, \psi = \mathbf{f}$

- $\cdot\, \varphi_2 = \mathbf{a}, \mathbf{b}$
  - $\cdot\, \varphi_1 = \mathbf{a}, \mathbf{b}, \mathbf{c}$ then $\chi \geq \mathbf{c}$ and therefore $\varphi_2 \to (\chi \to \psi) < \mathbf{1}$.
  - $\cdot\, \varphi_1 = \mathbf{d}, \dots$ then $\varphi_1 \to (\varphi_2 \to \psi) = \mathbf{1}$.
- $\cdot\, \varphi_2 = \mathbf{c}$
  - $\cdot\, \varphi_1 = \mathbf{a}, \mathbf{b}$ $\chi \geq \mathbf{b}$ then $\varphi_2 \to (\chi \to \psi) < \mathbf{1}$.
  - $\cdot\, \varphi_1 = \mathbf{c}, \dots$ then $\varphi_1 \to (\varphi_2 \to \psi) = \mathbf{1}$.
- $\cdot\, \varphi_2 = \mathbf{d}, \dots$ then $\varphi_1 \to (\varphi_2 \to \psi) = \mathbf{1}$.

$\cdot\, \psi = \mathbf{g}$

- $\cdot\, \varphi_2 = \mathbf{a}$
  - $\cdot\, \varphi_1 = \mathbf{a}, \dots, \mathbf{e}$ then $\chi \geq \mathbf{e}$ and therefore $\varphi_2 \to (\chi \to \psi) < \mathbf{1}$.
  - $\cdot\, \varphi_1 = \mathbf{f}, \dots$ then $\varphi_1 \to (\varphi_2 \to \psi) = \mathbf{1}$.
- $\cdot\, \varphi_2 = \mathbf{b}, \mathbf{c}$
  - $\cdot\, \varphi_1 = \mathbf{a}, \dots, \mathbf{d}$ then $\chi \geq \mathbf{d}$ and therefore $\varphi_2 \to (\chi \to \psi) < \mathbf{1}$.
  - $\cdot\, \varphi_1 = \mathbf{e}, \dots$ then $\varphi_1 \to (\varphi_2 \to \psi) = \mathbf{1}$.
- $\cdot\, \varphi_2 = \mathbf{d}$
  - $\cdot\, \varphi_1 = \mathbf{a}, \mathbf{b}, \mathbf{c}$ then $\chi \geq \mathbf{c}$ and therefore $\varphi_2 \to (\chi \to \psi) < \mathbf{1}$.
  - $\cdot\, \varphi_1 = \mathbf{d}, \dots$ then $\varphi_1 \to (\varphi_2 \to \psi) = \mathbf{1}$.
- $\cdot\, \varphi_2 = \mathbf{e}$
  - $\cdot\, \varphi_1 = \mathbf{a}$ then $\chi \geq \mathbf{a}$ and therefore $\varphi_2 \to (\chi \to \psi) < \mathbf{1}$.
  - $\cdot\, \varphi_1 = \mathbf{b}, \dots$ then $\varphi_1 \to (\varphi_2 \to \psi) = \mathbf{1}$.
- $\cdot\, \varphi_2 = \mathbf{f}, \dots$ then $\varphi_1 \to (\varphi_2 \to \psi) = \mathbf{1}$.

■

Let us point out that if $\mathbf{f} \leq \psi$ and φ_0, φ_1 and φ_2 are different from $\mathbf{1}$, then

$$\varphi_0 \to (\varphi_1 \to (\varphi_2 \to \psi)) = \mathbf{1}.$$

So if $a_1 + a_2 + b_1 + b_2 = 3$ then we need to check only $\psi = \mathbf{g}$ case.

Lemma 10 *Given* (4) *and* (5)*, and if* $a_1 = 0$*,* $a_2 = 2$*,* $b_1 = 1$*,* $b_2 = 0$*, then the condition* (6) *holds.*

Proof: We need to show that

$$\varphi_2 \to (\varphi_2 \to (\chi \to \psi)) = \mathbf{1}, \tag{12}$$
$$\varphi_1 \to \chi = \mathbf{1} \tag{13}$$

imply

$$\varphi_1 \to (\varphi_2 \to (\varphi_2 \to \psi)) = \mathbf{1}. \tag{14}$$

We prove it by cases:

$\cdot\, \psi = \mathbf{g}$

- $\cdot\, \varphi_2 = \mathbf{a}$
  - $\cdot\, \varphi_1 = \mathbf{a}, \mathbf{b}, \mathbf{c}$ then $\varphi_1 \leq \chi$ from (13), hence $\chi \geq \mathbf{c}$ and therefore (12) does not hold.
- $\cdot\, \varphi_2 = \mathbf{b}$
  - $\cdot\, \varphi_1 = \mathbf{a}$ then $\chi \geq \mathbf{a}$ and therefore (12) does not hold.

In all other cases (14) holds. ■

Lemma 11 *Given* (4) *and* (5)*, and if* $a_1 = 1$*,* $a_2 = 0$*,* $b_1 = 2$*,* $b_2 = 0$*, then the condition* (6) *holds.*

Proof: We need to show that

$$\varphi_1 \to (\chi \to \psi) = \mathbf{1}, \tag{15}$$
$$\varphi_1 \to (\varphi_1 \to \chi) = \mathbf{1} \tag{16}$$

imply

$$\varphi_1 \to (\varphi_1 \to (\varphi_1 \to \psi)) = \mathbf{1}. \tag{17}$$

The only case to show is:

$\cdot\, \psi = \mathbf{g}$

- $\cdot\, \varphi_1 = \mathbf{a}$ then $\chi \geq \mathbf{d}$ from (16) and therefore (15) does not hold.

In all other cases (17) holds. ■

Corollary 2 *Given* (4) *and* (5)*, and if* $a_1 = 0$*,* $a_2 = 1$*,* $b_1 = 0$*,* $b_2 = 2$*, then the condition* (6) *holds.*

Lemma 12 *Given* (4) *and* (5)*, and if* $a_1 = 0$*,* $a_2 = 1$*,* $b_1 = 2$*,* $b_2 = 0$*, then the condition* (6) *holds.*

Proof: We need to show that

$$\varphi_2 \to (\chi \to \psi) = \mathbf{1}, \tag{18}$$
$$\varphi_1 \to (\varphi_1 \to \chi) = \mathbf{1} \tag{19}$$

imply

$$\varphi_1 \to (\varphi_1 \to (\varphi_2 \to \psi)) = \mathbf{1}. \tag{20}$$

We prove it by cases:

- $\cdot\, \psi = \mathbf{g}$
  - $\cdot\, \varphi_2 = \mathbf{a}$
    - $\cdot\, \varphi_1 = \mathbf{a}, \mathbf{b}$ by (19) $\chi \geq \mathbf{e}$ and so (18) fails.
  - $\cdot\, \varphi_2 = \mathbf{b}$
    - $\cdot\, \varphi_1 = \mathbf{a}$ by (19) $\chi \geq \mathbf{d}$ and therefore (18) fails.
  - $\cdot\, \varphi_2 = \mathbf{c}$
    - $\cdot\, \varphi_1 = \mathbf{a}$ by (19) $\chi \geq \mathbf{d}$ and therefore (18) fails.

In all other cases (20) holds. ∎

Lemma 13 *Given* (4) *and* (5)*, and if* $a_1 = 1$*,* $a_2 = 0$*,* $b_1 = 0$*,* $b_2 = 2$*, then the condition* (6) *holds.*

Proof: We need to show that

$$\varphi_1 \to (\chi \to \psi) = \mathbf{1}, \tag{21}$$
$$\varphi_2 \to (\varphi_2 \to \chi) = \mathbf{1} \tag{22}$$

imply

$$\varphi_1 \to (\varphi_2 \to (\varphi_2 \to \psi)) = \mathbf{1}. \tag{23}$$

We prove it by cases:

- $\cdot\, \psi = \mathbf{g}$
  - $\cdot\, \varphi_2 = \mathbf{a}$
    - $\cdot\, \varphi_1 = \mathbf{a}, \mathbf{b}, \mathbf{c}$ by (22) $\chi \geq \mathbf{d}$ and so (21) fails.
  - $\cdot\, \varphi_2 = \mathbf{b}$
    - $\cdot\, \varphi_1 = \mathbf{a}$ by (22) $\chi \geq \mathbf{e}$ and therefore (21) fails.

In all other cases (23) holds. ∎

Lemma 14 *Given* (4) *and* (5)*, and if* $a_1 = 1$*,* $a_2 = 1$*,* $b_1 = 1$*,* $b_2 = 0$*, then the condition* (6) *holds.*

Proof: We need to show that

$$\varphi_1 \to (\varphi_2 \to (\chi \to \psi)) = \mathbf{1}, \tag{24}$$
$$\varphi_1 \to \chi = \mathbf{1} \tag{25}$$

imply

$$\varphi_1 \to (\varphi_1 \to (\varphi_2 \to \psi)) = \mathbf{1}. \tag{26}$$

We prove it by cases:

- $\cdot\, \psi = \mathbf{g}$
  - $\cdot\, \varphi_2 = \mathbf{a}$
    - $\cdot\, \varphi_1 = \mathbf{a}, \mathbf{b}$ by (25) $\chi \geq \mathbf{b}$ and so (24) fails.
  - $\cdot\, \varphi_2 = \mathbf{b}$
    - $\cdot\, \varphi_1 = \mathbf{a}$ by (25) $\chi \geq \mathbf{a}$ and therefore (24) fails.
  - $\cdot\, \varphi_2 = \mathbf{c}$
    - $\cdot\, \varphi_1 = \mathbf{a}$ by (25) $\chi \geq \mathbf{a}$ and therefore (24) fails.

In all other cases (26) holds. ∎

Lemma 15 *Given* (4) *and* (5)*, and if* $a_1 = 1$*,* $a_2 = 0$*,* $b_1 = 1$*,* $b_2 = 1$*, then the condition* (6) *holds.*

Proof: We need to show that

$$\varphi_1 \to (\chi \to \psi) = \mathbf{1}, \tag{27}$$
$$\varphi_1 \to (\varphi_2 \to \chi) = \mathbf{1} \tag{28}$$

imply

$$\varphi_1 \to (\varphi_1 \to (\varphi_2 \to \psi)) = \mathbf{1}. \tag{29}$$

We prove it by cases:

- $\cdot\, \psi = \mathbf{g}$
  - $\cdot\, \varphi_2 = \mathbf{a}$
    - $\cdot\, \varphi_1 = \mathbf{a}$ by Lemma 11.
    - $\cdot\, \varphi_1 = \mathbf{b}$ by (28) $\chi \geq \mathbf{d}$ and therefore (27) fails.
  - $\cdot\, \varphi_2 = \mathbf{b}$
    - $\cdot\, \varphi_1 = \mathbf{a}$ by (28) $\chi \geq \mathbf{d}$ and therefore (27) fails.
    - $\cdot\, \varphi_1 = \mathbf{b}$ by Lemma 11.
  - $\cdot\, \varphi_2 = \mathbf{c}$
    - $\cdot\, \varphi_1 = \mathbf{a}$ by (28) $\chi \geq \mathbf{e}$ and therefore (27) fails.

In all other cases (29) holds. ∎

Lemma 16 *Given* (4) *and* (5)*, and if* $a_1 = 0$*,* $a_2 = 1$*,* $b_1 = 1$*,* $b_2 = 1$*, then the condition* (6) *holds.*

Proof: We need to show that

$$\varphi_2 \to (\chi \to \psi) = \mathbf{1}, \tag{30}$$
$$\varphi_1 \to (\varphi_2 \to \chi) = \mathbf{1} \tag{31}$$

imply

$$\varphi_1 \to (\varphi_2 \to (\varphi_2 \to \psi)) = \mathbf{1}. \tag{32}$$

We prove it by cases:

- $\cdot\, \psi = \mathbf{g}$
  - $\cdot\, \varphi_2 = \mathbf{a}$
    - $\cdot\, \varphi_1 = \mathbf{a}, \mathbf{b}, \mathbf{c}$ by (31) $\chi \geq \mathbf{e}$ and so (30) fails.
  - $\cdot\, \varphi_2 = \mathbf{b}$
    - $\cdot\, \varphi_1 = \mathbf{a}$ by (31) $\chi \geq \mathbf{d}$ and therefore (30) fails.

In all other cases (32) holds. ∎

Theorem 2 *There is a logic with IDT$_2$ but without IDT$_3$.*

In Tables 2 and 3 we spell all variants needed by Corollary 1 to show that the logic $\mathbf{L}$ given by $\mathbf{M}$ has IDT$_2$, which together with Lemma 3 completes the proof of the previous theorem. Consequently, we obtained the complete picture of hierarchy of Implicational Deduction Theorems

$$\text{IDT}_0 \supsetneq \text{IDT}_1 \supsetneq \text{IDT}_2 \supsetneq \text{IDT}_3 = \text{IDT}_4 = \dots$$

a_1	a_2	b_1	b_2	Solution
0	0	0	0	Lemma 7
1	0	0	0	Lemma 7
0	1	0	0	Lemma 7
0	0	1	0	Lemma 7
0	0	0	1	Lemma 7
2	0	0	0	Lemma 7
0	2	0	0	Lemma 7
0	0	2	0	Lemma 7
0	0	0	2	Lemma 7
1	1	0	0	Lemma 7
1	0	1	0	Lemma 8
1	0	0	1	Lemma 8
0	1	1	0	Lemma 9
0	1	0	1	Lemma 8
0	0	1	1	Lemma 7

Table 2: Proof variants for $a_1 + a_2 + b_1 + b_2 < 3$.

a_1	a_2	b_1	b_2	Solution
3	0	0	0	Lemma 7
0	3	0	0	Lemma 7
0	0	3	0	Lemma 7
0	0	0	3	Lemma 7
2	1	0	0	Lemma 7
2	0	1	0	Lemma 8
2	0	0	1	Lemma 8
1	2	0	0	Lemma 7
0	2	1	0	Lemma 10
0	2	0	1	Lemma 8
1	0	2	0	Lemma 11
0	1	2	0	Lemma 12
0	0	2	1	Lemma 7
1	0	0	2	Lemma 13
0	1	0	2	Corollary 2
0	0	1	2	Lemma 7
1	1	1	0	Lemma 14
1	1	0	1	Lemma 8
1	0	1	1	Lemma 15
0	1	1	1	Lemma 16

Table 3: Proof variants for $a_1 + a_2 + b_1 + b_2 = 3$.

5. Summary

We presented a hierarchy of logics satisfying some Implicational Deduction Theorems. We know that any logic with IDT$_i$ has also IDT$_j$ for any $j \leq i$ and for any $0 \leq i \leq 1$ there is a logic with IDT$_i$ but without IDT$_{i+1}$. Our paper showed that there is also a logic with IDT$_2$ but without IDT$_3$. Moreover, any logic with IDT$_i$, for $i \geq 3$, has also IDT$_j$ for any $j \geq i$ and hence any j. This completes the picture of our hierarchy.

References

[1] K. Chvalovský and P. Cintula, "Note on Deduction Theorems in Contraction Free Logics," In preparation.

[2] J. Herbrand, "Recherches sur la théorie de la démonstration," *Travaux de la Société des Sciences et des Lettres de Varsovie, Classe III, Sciences Mathematiques et Physiques*, vol. 33, 1930.

[3] A. Tarski, "Über einige fundamentale begriffe der metamathematik," *Comptes Rendus des Séances de la Société et les Lettres de Varsovie*, vol. 23, pp. 22–29, 1930.

[4] R. Wójcicki, " Theory of Logical Calculi: basic theory of consequence operations". Springer, 1988.

Concept of Role in Multi-Agent Systems Development: Methodologies and Application

Post-Graduate Student:
MGR. ONDŘEJ KAZÍK
Faculty of Mathematics and Physics
Charles University in Prague
Malostranské náměstí 25
118 00 Prague 1, CZ
kazik.ondrej@gmail.com

Supervisor:
MGR. ROMAN NERUDA, CSC.
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
roman@cs.cas.cz

Field of Study:
Theoretical Computer Science

Abstract

In recent works concerning open Multi-Agent Systems (MAS), the emphasis has been laid on organizational aspects of the development of agent societies. Many models of collaboration and cooperation of agents have been proposed which allow reusability of design patterns in MAS and separation of concerns. One of these approaches is the role-based model, inspired by the importance of roles in human communities. In this paper we summarize the common features of role-based methodologies, compare frameworks and apply the concepts of role-based models to a concrete scenario in the field of Computational MAS.

1. Introduction

Agent is a computer system that is situated in some environment and that is capable of autonomous action in this environment in order to meet its design objectives [17]. Some of its features are adaptivity to changes in the environment or collaboration with other agents. Such interacting agents join in more complex societies, Multi-Agent Systems (MAS). These groups of agents gain several advantages, as are the applications in distributed systems, delegacy of subproblems on other agents, and flexibility of the software system engineering.

Many present-day applications require open societies whose agents can join, leave them or change their position in them, and which are able to cope with dynamic environments. The importance of an interaction and cooperation aspects of agents, therefore, rises. The effort to reuse MAS patterns brings the need of separation of the interaction logic from the inner algorithmic logic of an agent. There are several approaches providing such separation and modeling MAS from the organizational perspective, among others the Tuple-Spaces, Group Computation, Activity Theory or Roles [4]. We will examine the latter model which is inspired by a sociological concept of a role applied in several methodologies on the analysis, design and implementation level.

2. Roles

The concept of a *role* is well established in different fields of computer science. We can find the roles in the domain of software development, e.g. in the object-oriented programming, design patterns or computer-to-human interfaces [11]. The roles there include groups of objects with similar common behavior. The roles are also used for restricting of the access to various sources of a system to guarantee the security, for example in the Role Based Access Control (RBAC) [10], or in the Computer Supported Cooperative Work (CSCW) [1].

In the field of the multi-agent systems engineering, peculiarities of agents, such as the autonomy and ability to collaborate, are emphasized. Generally speaking, a role is an abstract representation of stereotypical behavior common to different classes of agents. Moreover, it serves as an interface, through which the agents obtain their knowledge of their execution environment and affect this environment. Such representation contains a set of patterns of interactions, capabilities and knowledge which the associated agent may utilize to achieve its goals. On the other side, the role defines constraints, which a requesting agent has to satisfy to obtain the role, and responsibilities, for which the agent playing this role holds accountable. It serves also as a mean of definition of protocols, common interactions between agents. In most approaches the assignment between roles and agents is a general relation, i.e. an agent may handle

more roles as well as a role can be embodied by different classes of agents. Moreover, the agents can change their roles dynamically.

The role-based solutions may be independent of the concrete situation in a system. This allows to design an overall organization of the multi-agent systems, represented by roles and their interactions, separately from the algorithmic issues of agents and to reuse the solutions from different application contexts. The coordination of agents is based on local conditions, especially the positions of the agent playing some role, and so even the large MAS can be built modularly out of simple organizational structures.

3. Survey of existing approaches

We will describe there briefly some of the role-based approaches. The comparison is difficult for many reasons. First, the models vary in the used terminologies and definitions of roles. In addition, the roles are not exploited in all phases of the development process. Some of them define the roles in the analysis phase and these are transformed to agent classes during design, whereas others implements the role-based support for applications.

The *Gaia* methodology [18] is a conceptual model for multi-agent system analysis and design. The phases preceding the implementation, i.e. the analysis and design, are there well separated and key concepts in each phase are identified. The goal of the analysis phase is an elaboration of functional features of the MAS and its organization, which includes: identification of the roles played by agents in the organization, interactions between these roles controlled by the protocols, and constraints on both roles and protocols maintaining their coherency. These three domains (i.e. roles, protocols and constraints) constitute the *role model*, *interaction model* and *organizational rules*, which are inputs of the next phase, the design phase. The actual agent system is there defined so that it will be suitable for an easy implementation. This definition results in the following three models: the *agent model*, which determines final classes playing concrete roles; *service model*, which identify the services associated with each agent to fulfill its role; and *acquaintance model*, which represents communication between agents and follows from the interaction and agent model.

The role model's domain in Gaia is a set of the key roles in the system. The roles are defined by means of the four following attributes: *protocols*, which define the specific patterns of interaction with other roles; *activities*, i.e. tasks associated with the role that an agent carries about without interaction with another agent; *permissions*, which determine access to information resources; and finally *responsibilities*, determining expected behavior and functionality of the role.

The interaction model consists of a set of protocol definitions, the definitions of fixed patterns of interactions between roles. These protocols are described by schemata with following attributes: the *initiating role*, *responding role*, *inputs*, *outputs* (i.e. information used by initiator and provided by responder) and description of protocol's *purpose*. The organizational rules control the coherency of the system and can be useful in the context of open systems. These models lead to the final agent model, assignment of agent classes to the roles, their services required to perform these roles and acquaintances directing the interactions between agents.

The Gaia methodology represents a role-based model of MAS engineering from the organizational perspective. The models designed by means of this methodology are independent on a choice of the final implementation. On the other side, the support for the role-based implementation is not considered and the role model is used only in the analysis phase and left in the design level. There are attempts to combine the Gaia methodology with the JADE framework [13] but the flexibility resulting from the role approach is not exploited in an application. The Gaia is suitable mainly for closed systems, where the components are known at the design time since the relation between roles and agents is firm.

The aim of the *Multiagent Systems Engineering* (MaSE) [7] methodology is a development of general-purpose multi-agent systems. The development is again split into analysis and design phase. The analysis phase consists of three steps: *capturing the goals*, which is done by identifying the goals and subgoals of a system in scenarios and by structuring goals into a goal hierarchy diagram; *applying the use cases*, where the scenarios are captured with desired system behaviors and event sequences; and *refining the roles*, where a role model is established.

The result of the analysis phase is thus a set of roles responsible for achieving of the system level goals. The purpose of the design phase is a definition of concepts more suitable for implementation: agents and conversations. It progresses through four steps: *construction of agent classes*, where agent classes and their conversations are identified and documented in Agent Class Diagrams; *constructing conversations*, i.e. detailed definition of conversations model by means of finite state automata; *assembling agent classes*, i.e. defining the agents' internal architecture; and *deployment design*,

where the actual configuration of the system is chosen and documented in a Deployment Diagram.

The MaSE has several weaknesses: there is no mechanism for modeling MAS interaction with the environment, designed MAS have rather fixed organization, the integration of sub-teams into a MAS is not allowed, the protocols are decomposed into small and simple pieces, and roles are again restricted to the analysis phase. Some of these disadvantages are solved in organizational extension of the methodology, O-MaSE [6].

The *ALAADIN* framework [8, 9] is an organization-centered generic meta-model of multi-agent systems. It defines a general conceptual structure which is utilized in the MAS development. The framework describes MAS from an organizational perspective, instead of using terms of agents' mental states (agent-centered). This model (also called AGR) focuses on three basic concepts: agent, group and role.

An *agent* is an active, communicating entity which plays roles and is a member of groups. The model does not describe internal structure and architecture of the agent, only its expected behavior. A *group* is a set of agents and serves as a context for these agents. A *group structure*, an abstract representation of a group, is described by a set of admissible roles which agents in the group can play, and by possible interactions between the role pairs in the group. Two agents can communicate if and only if both are in the same group. On the other hand, groups can intersect due to the agents handling two different roles in different groups. *Roles*, thus, are abstract representations of the functional position of an agent in a group and at the same time they are views of agents playing them, i.e. the way other agents recognize them. The AGR model also introduces the formal model of an organizational structure defining sets of roles, groups and their interactions and dependencies. In order to demonstrate the potentialities of the model, the MadKit agent platform have been implemented [8].

The positive features of the ALAADIN model are introducing modularity of MAS consisting of simpler groups and interoperability among different implementation platforms allowed by a role view of agents. However, the roles are still tightly bound to the notion of agents.

The *BRAIN* (Behavioral Roles for Agent INteractions) [3] framework is a multi-layer role-based approach to support MAS development process. The framework provides three components: model of interactions, XML-based notation and interaction infrastructures.

The *interaction model* in BRAIN represents a role by a set of capabilities and an expected behavior. The role's set of capabilities is a set of available *actions* which an agent playing the role can perform in order to fulfill its task. The expected behavior is a set of *events* that an agent playing the role ought to handle. Interactions between two agents are then represented in the form of couples `(action, event)`, where `action` is from the set of capabilities of the initiating agent and `event` is managed by the expected behavior of the requested agent.

This role model is described by means of the XML-based *notation*, called XRole. The tagged XRole representation of the model is a compromise of human and machine readability, interoperability and platform independence. An XRole document contains three main parts: the basic information, allowed actions and recognized events.

The BRAIN *interaction infrastructures* are implementations of the role model. [5] The interaction infrastructure allows assumption and dismission of roles by agents during their lifetime, translation of actions into events and their delivery, and control local policies of the system. Different infrastructures (RoleSystem or RoleX) can be plugged-in into the system according to the need of flexibility, security, efficiency or compactness of the system without any change of the two top layers.

The BRAIN model supports MAS development in all of its phases and introduces dynamism into the agent-role relation during the run-time. Moreover, the role model of a MAS is independent on the platform which controls sending messages. On the other hand, this is provided by rather complicated process of role registration including either communication with a central unit or Java bytecode manipulation.

There exist other approaches employing the concept of roles that are not so relevant to our work. Xu, Zhang and Patel's [19] approach proposes another formal role-based model of open MAS specified by means of *Object-Z* formalism, consisting of the role organization, role space and agent society. *RoleEP* (Role based Evolutionary Programming, [16]) is a system supporting construction of cooperative mobile agent applications, which are described by mean of four basic notions: objects, roles, agents and environments. The *TRUCE* (Tasks and Roles in a Unified Coordination Environment, [12]) framework consists of concurently interpreted scripts defining coordination between agents.

4. Case Study: Computational MAS

The application area of our concern is computational intelligence, namely hybrid models. These models including combinations of artificial intelligence methods, such as neural networks, genetic algorithms (GA) and fuzzy logic controllers, have shown to be promising and extensively studied research area [2]. They have demonstrated better performance over individual methods in many real-world tasks. Their disadvantage is generally higher complexity, and the need to manually set them up and tune various parameters. Also, there are not many software packages that provide a large collection of individual computational methods, as well as the possibility to connect them into hybrid schemes in various ways.

The hybrid models are thus complex systems with a large number of components and computational methods, and with potentially unpredictable interactions between these parts. Multi-agent systems are suitable solutions in order to manage this complexity. The computational MAS contains one or more computational agents, i.e. highly encapsulated objects realizing particular computational methods, collaborating with other autonomous agents to fulfill their goals. Several models of development of hybrid intelligent systems by means of MAS have been proposed, e.g. [20] and [14].

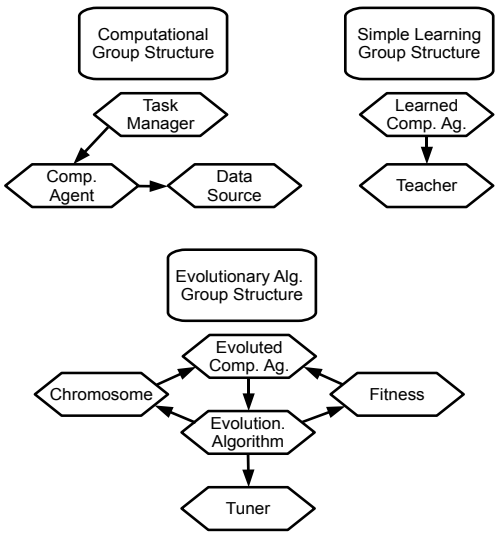


Figure 1: The organizational structure diagram of the computational MAS.

In order to verify the abilities of role-based models we will present an example of the analysis of a computational MAS scenario. We are exploiting the conceptual framework of the AGR model. Its organization-centered perspective allowing modular and variable construction of MAS is well suited especially to more complicated configurations of computational agents.

As an example we take the computational MAS from [15]. The system consists of a Task Manager agent, Data Source agent, two computational agents (RBF neural network and Evolutionary algorithm agent) and supplementary agents. In the case of RBF network, there are unsupervised (vector quantization) and supervised (gradient, matrix inverse) learning agents. The evolutionary algorithm agent needs Fitness, Chromosome and Tuner agents.

Such a computational MAS is represented by an role organizational structure showed on the Figure 1. It contains the following group structures:

- *Computational Group Structure*. It contains three roles: a Task Manager, Data Source and Computational Agent.
- *Simple Learning Group Structure* consisting of two roles: a Teacher and Learned Computational Agent. This structure is instantiated by three groups for each Teacher (Vector Quantization, Gradient and matrix inverse).
- *Evolutionary Algorithm Group Structure* which is more complicated than the previous two. It contains an Evolutionary Algorithm agent, Evolved Computational Agent, Fitness, Chromosome and Tuner.

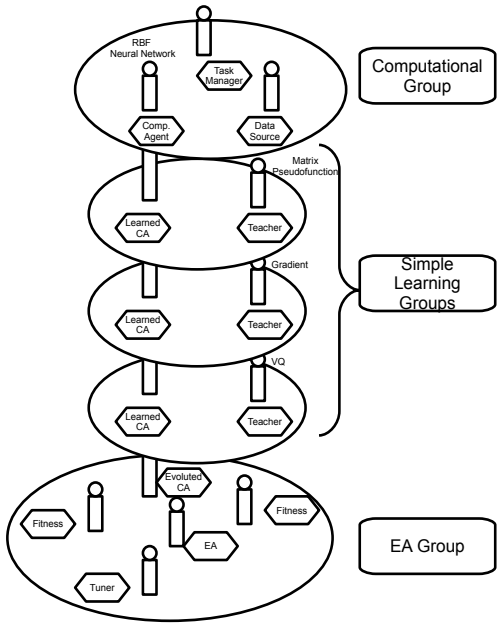


Figure 2: The organization of a concrete computational MAS scenario (cheeseboard notation).

At the beginning of the run, only the computational group exists with the RBF network in the role of a computational agent. After the request for learning the problem by the task manager, appropriate simple learning groups are created and the learning agents are constructed/reused/found. Similarly, the evolutionary algorithm group is constructed with all supplementary agents. The interactions proceed according to the definition of organizational structure. The Figure 2 shows an actual organization of such computational MAS.

We can see, that the role model allows to simplify the construction of more complicated computational multi-agent systems by its decomposition to the simple group structures and roles, to which the agents assigns. Moreover, the position of an agent in MAS in every moment of the run-time is defined by their roles without need to take into account their internal architecture or concrete method it implements. It also reduces the space of possible responding agents when interactions are established.

5. Conclusion

In large and open multi-agent systems, where agents can join and leave different groups and behave according to its changing position in the environment, the emphasis shifts from the inner structure and algorithmic logic of individual agents to their interaction and cooperation aspects. In this paper we have examined the concept of a role which allows to design multi-agent systems from this organizational perspective.

This paper has also presented a survey of the most significant role approaches for MAS. Despite their diversity in a terminology and varying support for the phases of the development, an introduction of this concept simplifies engineering of an application. In general, a role is a concept independent of agents, to which an agent can assign, and it is defined as a set of obligations and capabilities of the agent. Protocols and groups the agents can participate are also described by means of the roles. The advantages in the development result from the separation of concerns, generality and reusability of solutions, modularity and locality. These features significantly reduces the development process.

We have also proved usability of the role approach in computational MAS, where individual agents migrate and establish potentially unpredictable interactions. The analysis of a scenario from this field by means of concepts of agents, groups and roles suggests possibilities of such a model.

Future research will be put in the formal ontological description of the earlier mentioned role model of computational MAS by means of Description Logics. This formalization allows to link the role-based analysis with run-time management of the system. Maintaining consistency and matchmaking of the responding agent can be converted to a simple reasoning, integrity constraints validation or query problem in the ontological model.

References

[1] T. Ahmed, R. Kumar, and A.R. Tripathi, “Secure management of distributed collaboration systems”, Tech. rep., Dept. of Computer and Information Science, University of Minnesota, http://www.cs.umn.edu/Ajanta/papers/secure-mgmt.pdf, 2003.

[2] P. Bonissone, “Soft computing: The convergence of emerging reasoning technologies”, *Soft Computing*, *1*, 6–18, 1997.

[3] G. Cabri, L. Ferrari, and L. Leonardi, “Brain: A framework for flexible role-based interactions in multiagent systems”, in *Proceedings of the 2003 Conference on Cooperative Information Systems (CoopIS)*, 2003.

[4] G. Cabri, L. Ferrari, and L. Leonardi, “Agent role-based collaboration and coordination: A survey about existing approaches”, in *Proc. of the Man and Cybernetics Conf.*, 2004a.

[5] G. Cabri, L. Ferrari, and F. Zambonelli, “Role-based approaches for engineering interactions in large-scale multi-agent systems”, in *Proceedings of the SELMAN 2004*, edited by C. L. et al., pp. 243–263, 2004b.

[6] S.A. DeLoach, “Engineering organization-based multiagent systems”, in *SELMAS 2005*, edited by A. G. et al., LNCS 3914, pp. 109–125, 2006.

[7] S.A. DeLoach, M.F. Wood, and C.H. Sparkman, “Multiagent systems engineering”, *The International Journal of Software Engineering and Knowledge Engineering*, *11*, 231–258, 2001.

[8] J. Ferber and O. Gutknecht, “A meta-model for the analysis and design of organizations in multi-agent systems”, in *Third International Conference on Multi-Agent Systems*, pp. 128–135, IEEE Computer Press, 1998.

[9] J. Ferber, O. Gutknecht, and M. Fabien, “From agents to organizations: An organizational view of multi-agent systems”, in *AOSE 2003*, edited by P. G. et al., LNCS 2935, pp. 214–230, 2004.

[10] D. Ferraiolo and D. Kuhn, “Role-based access control”, in *15th National Computer Security Conference*, pp. 554—563, 1992.

[11] M. Fowler, "Dealing with roles", in *Proceedings of PLoP '97, Technical Report WUCS-97-34*, Washington University Dept. of Computer Science, 1997.

[12] W.C. Jamison and D. Lea, "Truce: Agent coordination through concurrent interpretation of role-based protocols", in *Proceedings of Coordination 99*, 1999.

[13] P. Moraitis and N.I. Spanoudakis, "Combining Gaia and JADE for multi-agent systems development", in *Proceedings of the 17th European Meeting on Cybernetics and Systems Research (EMCSR 2004)*, 2004.

[14] R. Neruda and G. Beuster, "Emerging hybrid computational models", in *Proc. of the ICIC 2006*, LNCS 4113, pp. 379–389, 2006.

[15] R. Neruda and G. Beuster, "Toward dynamic generation of computational agents by means of logical descriptions", *International Transactions on Systems Science and Applications*, pp. 139–144, 2008.

[16] N. Ubayashi and T. Tamai, "ROLEEP: Role based evolutionary programming for cooperative mobile agent applications", in *Proccedings of International Symposium on Principles of Software Evolution*, 2000.

[17] G. Weiss, ed., "Multiagent Systems", MIT Press, 1999.

[18] M. Wooldridge, N.R. Jennings, and D. Kinny, "The Gaia methodology for agent-oriented analysis and design", *Journal of Autonomous Agents and Multi-Agent Systems*, *3*, 285–312, 2000.

[19] H. Xu, X. Zhang, and R.J.Patel, "Developing role-based open multi-agent software systems", *International Journal of Computational Intelligence Theory and Practice*, *2*, 2007.

[20] Z. Zhang and C. Zhang, "Agent-based hybrid intelligent systems", Springer Verlag, 2004.

Static Load Balancing of Parallel Mining of Frequent Itemsets Using Reservoir Sampling

Post-Graduate Student:
ING. ROBERT KESSL

Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ [1]

kesslr@cs.cas.cz

Supervisor:
PROF. ING. PAVEL TVRDÍK, CSC.

Department of Computer Systems
CTU FIT
Kolejní 550/2
160 00 Prague 6, CZ

tvrdik@fit.cvut.cz

Field of Study:
Computer Science

Abstract

One of the important data mining tasks is the search for co-occurences among the items in the databases, the so called frequent itemset mining. Let us consider a retail store. We track the contents of baskets of customers. The content of the baskets is stored in the database as transactions. In this database, we search for sets of items (itemsets in short) that occurs in at least $min\_support$ transactions.

The automated collection of data in companies allows to store huge amount of data. This creates the need for parallel mining of frequent itemsets. In this paper, we present a new method for parallel mining of frequent itemsets based on the reservoir sampling that statically load-balance the workload.

1. Introduction

The automated data collection causes extreme growth of the database sizes. Processing of databases of such sizes is almost impossible with a single processor. Therefore, new parallel algorithms that are able to process such amount of data are needed.

Today, large shared-memory machines parallel are still quite expensive. Distributed-memory multiprocessors can easily be built from cheap computers connected with a special network. Therefore, we consider designing algorithms for distributed-memory parallel machines.

One of the important data mining tasks is the search for co-occurences among the data. Let us consider a retail store. We track the contents of baskets of customers. The content of the baskets is stored in the database as transactions. In this database, we search for sets of items (itemsets in short) that occurs in at least $min\_support$ transactions. These itemsets are so called *frequent itemsets* (or FIs in short). From FIs, we create rules of type $X \Rightarrow Y$, where X, Y are two FIs. For example $\{$butter, bread$\} \Rightarrow \{$milk$\}$. The search of these co-occurences is divided into two parts: 1) find all *frequent itemsets*; 2) create *association rules* from the FIs.

The task of mining of FIs is computationally and memory demanding. It seems that the finding of all FIs is the most time-consuming part of the whole process. With the growth of retail-store databases it is important to design parallel algorithms for mining of FIs.

In [1] and [2] we have proposed new parallel methods for mining of FIs. We denote the set of all FIs by $\mathcal{F}$. The basic idea of the algorithms is to create a sample $\widetilde{\mathcal{F}}_s$ that is used to create disjoint partitions $F_i, F_j \subseteq \mathcal{F}$ such that $|F_i|/|\mathcal{F}| \approx 1/P$, where P is the number of processors. The relative size $|F_i|/|\mathcal{F}|$ is estimated using the sample $\widetilde{\mathcal{F}}_s$. The partitions F_i are then independently processed by each processor. In [1, 2], we have proposed a method of creation of $\widetilde{\mathcal{F}}_s$, based on a *modified* coverage algorithm. The problem with these two methods is that the sample $\widetilde{\mathcal{F}}_s$ is *non-uniform* and therefore, we do not have any guarantees on the load-balance. Additionally, the two methods needs an algorithm for mining of maximal frequent itemsets (MFIs in short). The MFIs are hold in main memory and used in the modified coverage algorithm. Since the number of MFIs can be quite large, the methods are in some cases very memory consuming.

In this paper, we show how to create a *uniform* sample $\widetilde{\mathcal{F}}_s$ using a different, faster and less memory consuming, method. The new method does not need an algorithm

[1] The author is also assistant professor at CTU FIT, Department of Theoretical Informatics and doctoral student of CTU FEE, Department of Computers.

for mining of MFIs and therefore is less memory consuming. Our new method needs only the algorithm for mining of FIs.

2. Notation

Let $\mathcal{B} = \{b_i\}$ be a *base set* of items (items can be numbers, symbols, strings etc.). An arbitrary set of items $U \subseteq \mathcal{B}$ will be further called an *itemsets*. Further, we need to view the baseset $\mathcal{B}$ as an ordered set. The items are therefore ordered using an arbitrary order $<$: $b_1 < b_2 < \ldots < b_n, n = |\mathcal{B}|$. Hence, we can view an itemset $U = \{b_{u_1}, b_{u_2}, \ldots, b_{u_{|U|}}\}$, $b_{u_1} < b_{u_2} < \ldots b_{u_{|U|}}$, as an ordered set denoted by $U = (b_{u_1}, b_{u_2}, \ldots, b_{u_{|U|}})$.

Let $U \subseteq \mathcal{B}$ be an itemset and `id` a unique identifier. We call the pair (id, U) a *transaction*. The `id` is called the *transaction id*. A database $\mathcal{D}$ is a set of transactions. In our algorithms, we need to sample the database $\mathcal{D}$. A *database sample* is denoted by $\widetilde{\mathcal{D}}$. We define the support as the number of transactions containing U, but in some literature, the relative support is defined by $Supp^*(U) = Supp(U)/|\mathcal{D}|$. We call U *frequent* in database $\mathcal{D}$ if $Supp(U, \mathcal{D}) \geq min\_support$. We can also define the frequent itemset using the relative support, denoted by $min\_support^*$, $0 \leq min\_support^* \leq 1$, i.e., an itemset is frequent iff $Supp^*(U, \mathcal{D}) \geq min\_support^*$.

We denote the set of all frequent itemsets computed from $\mathcal{D}$ by $\mathcal{F}$. The set of all frequent itemsets computed from $\widetilde{\mathcal{D}}$ is denoted by $\widetilde{\mathcal{F}}$. In our algorithms, we need to sample the set $\mathcal{F}$. A sample of frequent itemsets is denoted by $\widetilde{\mathcal{F}}_s$. In our case, the set $\widetilde{\mathcal{F}}_s$ is a sample of $\widetilde{\mathcal{F}}$, i.e., $\widetilde{\mathcal{F}}_s \subseteq \widetilde{\mathcal{F}}$.

The basic property of frequent itemsets is the so called *monotonicity of support*. It is an important property for all FIs mining algorithms and is defined as follows:

Theorem 3 (Monotonicity of support) *Let $U, V \subseteq \mathcal{B}$ be two itemsets such that $U \subsetneq V$ and $\mathcal{D}$ be a database. Then for the supports of U and V we have $Supp(U, \mathcal{D}) \geq Supp(V, \mathcal{D})$.*

The multivariate hypergeometric distribution describes the following problem: let the number of colors be C and the number of balls colored with color i is M_i and the total number of balls is $N = \sum_i M_i$. Let X_i, $1 \leq i \leq C$, be a random variable representing the number of balls colored by the ith color. The sample of size n is drawn from balls and X_i balls, such that $n = \sum_{i=1}^{C} X_i$ are colored by the ith color. Then the probability mass function is:

$$P(X_1 = k_1, \ldots, X_C = k_C) = \frac{\prod_{i=1}^{C} \binom{M_i}{k_i}}{\binom{N}{n}}.$$

We denote the number of processors by P and processor i by p_i.

3. The lattice of all itemsets

It is well known that the powerset the powerset $\mathcal{P}(\mathcal{B})$ of a set $\mathcal{B}$ is a complete lattice. The *join* operation is the *set union operation* and *meet* the *set intersection operation*.

To decompose the $\mathcal{P}(\mathcal{B})$ into the prefix-based equivalence classes, we need to order the items in $\mathcal{B}$. An equivalence relation partitions the ordered set $\mathcal{P}(\mathcal{B})$ into disjoint subsets called *prefix-based equivalence classes*:

Definition 4 (prefix-based equivalence class) *Let $U \subseteq \mathcal{B}, |U| = n$ be an itemset. We impose some order on the set $\mathcal{B}$ and hence view $U = (u_1, u_2, \ldots, u_n), u_i \in \mathcal{B}$ as an ordered set. A* prefix-based equivalence class *(PBEC in short) of U, denoted by $[U]_l$, is a set of all itemsets that have the same prefix of length l, i.e., $[U]_l = \{W = (w_1, w_2, \ldots, w_m) | u_i = w_i, i \leq l, m = |W| \geq |U|, U, W \subseteq \mathcal{B}\}$*

To simplify the notation, we use $[W]$ for the PBEC $[W]_l$ iff $l = |W|$. Each $[W], W \subseteq \mathcal{B}$ is a meet sublattice of $(\mathcal{P}(\mathcal{B}), \subseteq)$. Additionally, we use the term prefix for both: (a) *ordered set*; (b) *unordered set*; if clear from context, e.g., let $\mathcal{B} = \{1, 2, 3, 4, 5\}$ with the order $1 < 2 < 3 < 4 < 5$ and $U = \{3, 1, 2\}$ then the term prefix means $U = (1, 2, 3)$. The *extensions* of the PBEC $[U], U \subseteq \mathcal{B}$ is an ordered set $\Sigma \subseteq \mathcal{B}$ such that $U \cap \Sigma = \emptyset$ and for each $W \in [U], W \backslash U \subseteq \Sigma$. We denote the PBEC together with the extensions Σ by $[U|\Sigma]$.

Let $\mathcal{B} = \{b_1, \ldots, b_n\}, b_1 < \ldots < b_n$. Let $U_i = \{b_i\}$ and $\Sigma_i = \{b_j | b_i < b_j\}$ then $[U_i|\Sigma_i]$ forms *disjoint* PBECs. Each PBEC $[U_i|\Sigma_i]$ can be recursively divided into disjoint PBECs in the following way: let $W_k = U_i \cup \{b_k\}, b_k \in \Sigma_i$ and $\Sigma'_k = \{b \in \Sigma_i | b_k < b\}$ then $[W_k|\Sigma'_k]$ forms disjoint PBECS. We omit the extensions from the notation if clear from context.

Further, we need to partition $\mathcal{F}$ into n disjoint sets, denoted $(F_1, \ldots, F_n)$, satisfying $F_i \cap F_j = \emptyset, i \neq j$, and $\bigcup_i F_i = \mathcal{F}$. This partitioning can be done using the PBECs. The PBECs can be collated into n partitions as follows: let have *disjoint* PBECs $[U_l]$, such that $\bigcup_l [U_l] = \mathcal{F}, 1 \leq l \leq m$ and sets of indexes of the PBECs $L_i \subseteq \{l | 1 \leq l \leq m\}, 1 \leq i \leq n$

such that $L_i \cap L_j = \emptyset$ and $\sum_i |L_i| = m$ then $F_i = \bigcup_{l \in L_i}([U_l] \cap \mathcal{F})$.

The basic idea of our algorithm is to create a database sample $\widetilde{\mathcal{D}}$ that allow us to estimate the support of an arbitrary itemset and using $\widetilde{\mathcal{D}}$, we create identically (but not independently distributed) sample of all FIs $\widetilde{\mathcal{F}}_s$. Let U_i be a prefix of the PBEC $[U_i|\Sigma_i]$. Using the sample, we can estimate the relative size of an arbitrary PBEC $[U_i]$ $\frac{|[U_i]\cap\mathcal{F}|}{|\mathcal{F}|} \approx \frac{|[U_i]\cap\widetilde{\mathcal{F}}_s|}{|\widetilde{\mathcal{F}}_s|}$. The knowledge of the relative size of PBECs allow us to create the partitions $F_i, 1 \leq i \leq P$ such that $|F_i|/|\mathcal{F}| \approx 1/P$.

4. Database sample

The time complexity of the decision whether an itemset U is frequent or not is in fact the complexity of computing the *relative support* $Supp^*(U, \mathcal{D})$ in the input database $\mathcal{D}$. If we know the approximate relative support of U, we can decide whether U is frequent or not with certain probability. We can estimate the relative support $Supp^*(U, \mathcal{D})$ from a database sample $\widetilde{\mathcal{D}}$, i.e., we can use $Supp^*(U, \widetilde{\mathcal{D}})$ instead of $Supp^*(U, \mathcal{D})$ – this significantly reduces the time complexity. The approach of estimating the relative support of U was described by Toivonen [3]. Further, we denote the set of all FIs computed from the database sample $\widetilde{\mathcal{D}}$ by $\widetilde{\mathcal{F}}_s$.

Toivonen uses a *database sample* $\widetilde{\mathcal{D}}$ for the *sequential* mining of frequent itemsets and for the efficient estimation of theirs supports. Toivonen's algorithm works as follows: 1) create a database sample $\widetilde{\mathcal{D}}$ of $\mathcal{D}$; 2) compute all frequent itemsets in $\widetilde{\mathcal{D}}$; 3) check that all these FIs computed using $\widetilde{\mathcal{D}}$ are also FIs in $\mathcal{D}$ and correct the output. If an itemset is frequent in $\mathcal{D}$ and not in $\widetilde{\mathcal{D}}$, correct the output using $\mathcal{D}$. Toivonen's algorithm is based on an efficient probabilistic estimate of the support of an itemset U. We reuse this idea of estimating the support of U in our method for parallel mining of FIs, i.e., we use only the first two steps.

We define the error of the estimate of $Supp^*(U, \mathcal{D})$ from a database sample $\widetilde{\mathcal{D}}$ by: $err_{supp}(U, \widetilde{\mathcal{D}}) = |Supp^*(U, \mathcal{D}) - Supp^*(U, \widetilde{\mathcal{D}})|$

The database sample $\widetilde{\mathcal{D}}$ is sampled with replacement. The estimation error can be analyzed using the Chernoff bound without making other assumptions about the database. The error analysis then holds for a database of arbitrary size and properties.

Theorem 4 *[3] Given an itemset $U \subseteq \mathcal{B}$ and a random sample $\widetilde{\mathcal{D}}$ drawn from database $\mathcal{D}$ of size*

$$|\widetilde{\mathcal{D}}| \geq \frac{1}{2\epsilon^2_{\widetilde{\mathcal{D}}}} \ln \frac{2}{\delta_{\widetilde{\mathcal{D}}}},$$

then the probability that $err_{supp}(U, \widetilde{\mathcal{D}}) > \epsilon_{\widetilde{\mathcal{D}}}$ is at most $\delta_{\widetilde{\mathcal{D}}}$.

Using a database sample $\widetilde{\mathcal{D}}$ with size given by the previous theorem, we can estimate $Supp^*(U, \mathcal{D})$ with error $\epsilon_{\widetilde{\mathcal{D}}}$ that occurs with probability at most $\delta_{\widetilde{\mathcal{D}}}$: It follows from Lemma 4 that if we compute the approximation $\widetilde{\mathcal{F}}$ of $\mathcal{F}$ from the database sample $\widetilde{\mathcal{D}}$ of size $|\widetilde{\mathcal{D}}| \geq \frac{1}{2\epsilon^2_{\widetilde{\mathcal{D}}}} \ln \frac{2}{\delta_{\widetilde{\mathcal{D}}}}$, we should get an estimate of the supports of itemsets $U \in \widetilde{\mathcal{F}}$, i.e., potentionally, we have a close approximation $\widetilde{\mathcal{F}}$ of $\mathcal{F}$.

5. The reservoir sampling algorithm

In this section, we show the *reservoir sampling algorithm* that creates an uniformly but not independently distributed sample $\widetilde{\mathcal{F}}_s$ of $\widetilde{\mathcal{F}}$ on the contrary of the previous section.

Vitter [4] formulates the problem of *reservoir sampling* as follows: given a stream of records; the task is to create a sample of size n *without replacement* from the stream of records without any prior knowledge of the length of the stream.

We can reformulate the original problem in the terms of $\widetilde{\mathcal{F}}$ and $\widetilde{\mathcal{F}}_s$: let's consider a sequential algorithm that outputs all frequent itemsets $\widetilde{\mathcal{F}}$ from a database $\widetilde{\mathcal{D}}$. We can view $\widetilde{\mathcal{F}}$ as a stream of FIs. We do not know $|\widetilde{\mathcal{F}}|$ in advance and we need to take $|\widetilde{\mathcal{F}}_s|$ samples of $\widetilde{\mathcal{F}}$, see Theorem 5. We take the samples $\widetilde{\mathcal{F}}_s$ using the reservoir sampling algorithm. This solves our problem of making a uniform sample $\widetilde{\mathcal{F}}_s \subseteq \widetilde{\mathcal{F}}$. The sampling is done using an array of FIs (a buffer, or in the terminology of [4] a reservoir) that holds $\widetilde{\mathcal{F}}_s$.

The reservoir sampling uses the following two procedures: 1) READNEXTFI(L): reads next FI from an output of an arbitrary sequential algorithm for mining of FIs and stores the itemset at the location L in memory; 2) SKIPFIS(k): skips k FIs from the output of an arbitrary algorithm for mining of FIs. And the following function: RANDOM() which returns an uniformly distributed real number from the interval $[0, 1]$

The simplest reservoir sampling algorithm is summarized in Algorithm 2. It takes as an input an array R (reservoir/buffer) of size $n = |\widetilde{\mathcal{F}}_s|$, the function READNEXTFI(L) that reads an FI from the output of

an FI mining algorithm and stores it in memory at location L, and finally the function SKIPFIS(k) that skips k FIs. The algorithm samples $|\widetilde{\mathcal{F}}_s|$ FIs and stores them in memory into the buffer R.

The RESERVOIR-ALGORITHM follows:

Algorithm 2 The RESERVOIR-SAMPLING algorithm.

RESERVOIR-SAMPLING(**In/Out:** Array R of size n,
In: Integer n,
In: Function ReadNextFI,
In: Function SkipFIs)

1: **for** $j = 0$ to $n - 1$ **do**
2: ReadNextFI($R[j]$)
3: **end for**
4: $t = n$
5: **while** not eof **do**
6: $t = t + 1$
7: $m = \lfloor t \times \text{RANDOM}() \rfloor$ {pick uniformly a number from the set $\{1, \ldots, t-1\}$}
8: **if** $m < n$ **then**
9: ReadNextFI($R[m]$)
10: **else**
11: SkipFIs(1)
12: **end if**
13: **end while**

The RESERVOIR-SAMPLING is quite slow, it is linear in the number of input records read by READNEXTFI(R), i.e., it is linear in $|\widetilde{\mathcal{F}}|$. Vitter [4] created a *faster* algorithm with the average running time $\mathcal{O}(|\widetilde{\mathcal{F}}_s|(1 + \log \frac{|\widetilde{\mathcal{F}}|}{|\widetilde{\mathcal{F}}_s|}))$, where $|\widetilde{\mathcal{F}}_s|$ is the size of the array R used by RESERVOIR-SAMPLING. The algorithm has the same parameters as the RESERVOIR-SAMPLING and we denote the Vitter's variant of the reservoir sampling algorithm by VITTER-RESERVOIR-SAMPLING.

Now, we analyse the relative size of a PBEC using the samples taken by the reservoir sampling algorithm. The reservoir sampling samples the set $\widetilde{\mathcal{F}}$ **without replacement**, resulting in $\widetilde{\mathcal{F}}_s$. The reservoir sampling process can be modeled using the *hypergeometric distribution*. In the rest of this chapter, we analyze the bounds on the relative size of a set of itemsets using the sample made by the reservoir sampling using a *hypergeometric distribution*.

Using the bounds from [5], i.e., estimation of the relative size of a PBEC but using an uniformly but not *independently* distributed sample. From these bounds follows the following theorem:

Theorem 5 (Estimation of the relative size of $F \subseteq \widetilde{\mathcal{F}}$) *Let $F \subseteq \widetilde{\mathcal{F}}$ be a set of itemsets. The relative size of F, $\frac{|F|}{|\widetilde{\mathcal{F}}|}$, is estimated with error $\epsilon_{\widetilde{\mathcal{F}}_s}$ with probability $\delta_{\widetilde{\mathcal{F}}_s}$ from a hypergeometrically distributed sample $\widetilde{\mathcal{F}}_s \subseteq \widetilde{\mathcal{F}}$ with parameters $N = |\widetilde{\mathcal{F}}|, M = |F|$ (see [5] for details) of size:*

$$|\widetilde{\mathcal{F}}_s| \geq -\frac{\log(\delta_{\widetilde{\mathcal{F}}_s}/2)}{D(\rho + \epsilon_{\widetilde{\mathcal{F}}_s}||\rho)}$$

Where $D(x||y)$ is the Kullback-Leibler divergence of two hypergeometrically distributed variables with parameters x, y and $\rho = |F|/|\widetilde{\mathcal{F}}|$.

The expected value of the size $|F|$ is $\mathrm{E}[|F \cap \widetilde{\mathcal{F}}_s|] = |\widetilde{\mathcal{F}}_s| \cdot \frac{|F|}{|\widetilde{\mathcal{F}}|}$.

Proof: Can be found in [5]. ∎

6. Summary of the previous two methods

In [1] we have proposed the PARALLEL-FIMI-SEQ method and in [2] we have proposed the PARALLEL-FIMI-PAR method. The idea of the two methods is to partition all FIs into P disjoint sets F_i, using PBECs of relative size $\frac{|F_i|}{|\mathcal{F}|} \approx \frac{1}{P}$. Each processor p_i then processes partition F_i. The whole method consists of four phases.

The four phases are designed in such a way that they statically load-balance the computation of all FIs. Phases 1–2 prepares the static load-balancing for Phase 4. In the Phase 3, we redistribute the database partitions so each processor can proceed independently with some PBECs. In the Phase 4, we execute an arbitrary algorithm for mining of FIs. To speedup Phases 1–2, we can execute each of Phase 1–2 in parallel.

The input and the parameters of the whole method are the following: 1) Minimal support $min\_support^*$; 2) The sampling parameters: real numbers $0 \leq \epsilon_{\widetilde{\mathcal{D}}}, \delta_{\widetilde{\mathcal{D}}}, \epsilon_{\widetilde{\mathcal{F}}_s}, \delta_{\widetilde{\mathcal{F}}_s} \leq 1$, see Sections 4 and 5; 3) The relative size of a PBEC: the parameter $\rho, 0 \leq \rho \leq 1$, see Sections 5; 4) Partition parameter: real number $\alpha, 0 \leq \alpha \leq 1$; 5) Database parts $D_i, 1 \leq i \leq P$. The database partitions are loaded by each processor at the beginning of the methods.

We assume that at the beginning of the computation, processor p_i loads its database partition D_i to a local memory. The database partitions D_i has the following properties: $D_i \cap D_j = \emptyset, i \neq j$, and $|D_i| \approx \frac{|\mathcal{D}|}{P}$. Additionally, without loss of generality, we expect that each

$b_i \in \mathcal{B}$ is frequent. Therefore, each processor p_i computes local support of all items $b_j \in \mathcal{B}$ in its database part D_i. The support is then broadcast and each p_i removes all b_j that are not globally frequent. The four phases are summarized below:

Phase 1 (sampling of FIs): the input of Phase 1 is the minimal support $min\_support^*$, a partitioning of the database $\mathcal{D}$ into P disjoint partitions D_i, and the real numbers $0 \leq \epsilon_{\widetilde{\mathcal{D}}}, \delta_{\widetilde{\mathcal{D}}}, \epsilon_{\widetilde{\mathcal{F}}_s}, \delta_{\widetilde{\mathcal{F}}_s} \leq 1$. Output of Phase 1 is a sample of frequent itemsets $\widetilde{\mathcal{F}}_s \subseteq \widetilde{\mathcal{F}}$. Generally, the only purpose of the first phase is to compute a sample $\widetilde{\mathcal{F}}_s$. First, each processor samples D_i (in parallel) and creates part $\mathcal{D}'_i$ and broadcasts them to other processors (all-to-all scatter[1]). Each processor p_i then creates $\widetilde{\mathcal{D}} = \bigcup_i \mathcal{D}'_i$. The the processors compute the set of all MFIs from $\widetilde{\mathcal{D}}$, denoted by $\widetilde{\mathcal{M}}$. The set $\widetilde{\mathcal{M}}$ is the upper boundary in the sense of set inclusion of the set $\widetilde{\mathcal{F}}$, i.e., for each $U \in \widetilde{\mathcal{F}}$ exists $m \in \widetilde{\mathcal{M}}$ such that $U \subseteq m$. We need to take a sample $\widetilde{\mathcal{F}}_s \subsetneq \widetilde{\mathcal{F}}$. This can be done using a *modified coverage algorithm*. The modified coverage algorithm randomly chooses $m \in \widetilde{\mathcal{M}}$ with probability $|\mathcal{P}(m)| / \sum_{m' \in \widetilde{\mathcal{M}}} |\mathcal{P}(m')|$. Then it picks uniformly a set $U \in \mathcal{P}(m)$. This approach does not generates an identically distributed sample $\widetilde{\mathcal{F}}_s$, but it gives reasonably good results and it is very quick. The difference between PARALLEL-FIMI-SEQ and PARALLEL-FIMI-PAR methods is that the second computes a set M, such that $\widetilde{\mathcal{M}} \subseteq M \subseteq \widetilde{\mathcal{F}}$, i.e., it computes a superset of $\widetilde{\mathcal{M}}$. Computation of $\widetilde{\mathcal{M}}$ in parallel makes the PARALLEL-FIMI-PAR faster then PARALLEL-FIMI-SEQ.

Phase 2 (lattice partitioning): the input of this phase is the sample $\widetilde{\mathcal{F}}_s$, the database sample $\widetilde{\mathcal{D}}$ (both computed in Phase 1) and the parameter α. In Phase 2 the algorithm creates prefixes $U_i \subseteq \mathcal{B}$ and the extensions Σ_i of each PBEC $[U_i|\Sigma_i]$, and estimates the size of $[U_i|\Sigma] \cap \mathcal{F}$ using $\widetilde{\mathcal{F}}_s$. The PBECs $[U_i|\Sigma_i]$ are then assigned to the processors and the assignment is broadcast to the processors.

Phase 3 (data distribution): the input of this phase is the assignment of the prefixes U_i and the extensions Σ_i to the processors p_i and the database partitioning $D_i, i = 1, \ldots, P$. Now, the processors exchange database partitions: processor p_i sends $S_{ij} \subseteq D_i$ to processor p_j such that S_{ij} contains transactions needed by p_j for computing support of the itemsets of its assigned PBECs.

Phase 4 (computation of FIs): as the input to each processor are the prefixes $U_i \subseteq \mathcal{B}$, the extensions Σ_i, and the database parts needed for computation of supports of itemsets $V \in [U_i] \cap \mathcal{F}$ and the original D_i. Each processor computes the FIs in $[U_i] \cap \mathcal{F}$ by executing an arbitrary sequential algorithm for mining of FIs. Additionally, each processor computes support of $W \subseteq U_i$ in D_i, i.e., $Supp(W, D_i)$. The supports are then send to p_1 and p_1 computes $Supp(W, \mathcal{D}) = \sum_{1 \leq i \leq P} Supp(W, D_i)$

7. Proposal of a new DM parallel method

Our new method is called *Parallel Frequent Itemset MIning – Reservoir* (Parallel-FIMI-Reservoir in short). This method works for any number of processors $P \ll |\mathcal{B}|$. The basic idea is the same as in PARALLEL-FIMI-SEQ and PARALLEL-FIMI-PAR methods. The main difference is the usage of the so called reservoir sampling algorithm instead of the modified coverage algorithm. This allow us to take an identically but *not independently* distributed sample $\widetilde{\mathcal{F}}_s$. We make the sample $\widetilde{\mathcal{F}}_s$ in parallel: in Phase 1, we execute an arbitrary algorithm for mining of FIs in parallel and the output of the FI mining algorithm is sampled using the reservoir sampling (in parallel). The input parameters are the same as in the PARALLEL-FIMI-SEQ and PARALLEL-FIMI-PAR methods

7.1. Detailed description of Phase 1

In this Section, we give a detailed description of the sampling process based on the *reservoir sampling* [4] that samples $\widetilde{\mathcal{F}}$ uniformly, i.e., it creates an identically distributed sample of $\widetilde{\mathcal{F}}$.

In our parallel method, we are using the VITTER-RESERVOIR-SAMPLING Algorithm, the faster reservoir sampling algorithm. To speedup the sampling phase of our parallel method, we execute the reservoir sampling in parallel. The database sample $\widetilde{\mathcal{D}}$ is distributed among the processors – each processor having a copy of the database sample $\widetilde{\mathcal{D}}$. The baseset $\mathcal{B}$ is partitioned into P parts $B_i \subseteq \mathcal{B}$ of size $|B_i| \approx |\mathcal{B}|/P$ such that $B_i \cap B_j = \emptyset, i \neq j$. Processor p_i then takes part B_i and executes an arbitrary sequential depth-first search (DFS in short) algorithm for mining of FIs, enumerating $[(b_j)] \cap \widetilde{\mathcal{F}}, b_j \in B_i$. The output, the itemsets $[(b_j)] \cap \widetilde{\mathcal{F}}$, of the sequential DFS algorithm are read by the reservoir sampling algorithm. If a processor finished its part B_i, it asks other processors for work. For terminating the parallel execution, we use the Dijkstra's token termination algorithm.

[1] All-to-all scatter is a well known communication operation: each processor p_i sends a message m_{ij} to processor p_j such that $m_{ij} \neq m_{ik}$, $i \neq k$.

The task of the process is to take $|\widetilde{\mathcal{F}}_s| = -\frac{\log(\delta_{\widetilde{\mathcal{F}}_s}/2)}{D(\rho+\epsilon_{\widetilde{\mathcal{F}}_s}||\rho)}$ samples, see Theorem 5. Because the reservoir sampling algorithm and the sequential algorithm is executed in parallel, it is not known how many FIs is computed by each processor. Denote the unknown number of FIs computed on p_i by f_i, the total number of FIs is denoted by $f = \sum_{1 \leq i \leq P} f_i$. Because, we do not know f_i in advance, each processor samples $|\widetilde{\mathcal{F}}_s|$ frequent itemsets using the reservoir sampling algorithm, producing $\widetilde{\mathcal{F}}_s$, and counts the number of FIs computed by the sequential algorithm. When the reservoir sampling finishes, processor p_i broadcasts f_i to all other processors. The processors then pick P random variables $X_i, 1 \leq i \leq P$ from multivariate hypergeometrical distribution with parameters: number of colors $C = P$, $M_i = f_i$ and choose X_i itemsets $U \in \widetilde{\mathcal{F}}_s$ at random out of the $N = |\widetilde{\mathcal{F}}_s|$ sampled frequent itemsets computed by p_i. The samples are then send to processor p_1. p_1 stores the received samples in $\widetilde{\mathcal{F}}_s$.

7.2. Detailed description of Phase 2

In Phase 2 the method partitions $\mathcal{F}$ sequentially on processor p_1. As an input of the partitioning, we use the samples $\widetilde{\mathcal{F}}_s$, the database $\widetilde{\mathcal{D}}$ (computed in Phase 1), the set $\mathcal{B}$, and a real number $\alpha, 0 < \alpha \leq 1$. For the purpose of this section, we denote the prefixes by U_k, the extensions of U_k by Σ_k, i.e., U_k and Σ_k forms a PBEC $[U_k|\Sigma_k]$. The set of the indexes of the PBECs assigned to processor p_i is denoted by L_i, and the set of all FIs assigned to processor p_i is denoted by F_i. Each F_i is the union of FIs in one or more PBECs $[U_k|\Sigma_k]$, i.e., $F_i = \bigcup_{k \in L_i}([U_k|\Sigma_k]) \cap \mathcal{F}$. Each processor p_i then in Phase 4 processes the FIs contained in F_i. *The output of Phase 2* are the index sets L_i of PBECs, computed on p_1, and the PBECs $[U_k|\Sigma_k]$.

The DFS sequential FI mining algorithm usually dynamically changes the order of items in $\mathcal{B}$ for each PBEC, i.e., the algorithm uses different order of items in the extensions. The PBECs are still disjoint and additionally the sequential algorithm is faster. Therefore, we need to prepare the PBECs in the same way as the sequential algorithm does. Let U be a prefix and $\Sigma = \{\epsilon_1, \ldots, \epsilon_n\} \subseteq \mathcal{B}$ the extensions. The sequential algorithm orders the items ϵ_i: $\epsilon_1 < \ldots < \epsilon_n$ such that $Supp(U \cup \{\epsilon_1\}) < \ldots < Supp(U \cup \{\epsilon_n\})$. We use the supports estimated using $\widetilde{\mathcal{D}}$ for making the order of the extensions.

The partitioning of $\mathcal{F}$ is a two step process:

(1) p_1 creates a list of prefixes U_k such that the estimated relative size of the PBEC $[U_k] \cap \mathcal{F}$ satisfies $\frac{|[U_k] \cap \widetilde{\mathcal{F}}_s|}{|\widetilde{\mathcal{F}}_s|} \leq \alpha \cdot \frac{1}{P}$, where $0 < \alpha < 1$ is a parameter of the computation set by the user. The PBECs are created recursively, see Section 3. The reason for making the PBECs of relative size $\leq \alpha \cdot \frac{1}{P}$ is to make the PBECs small enough so that they can be scheduled and the schedule is balanced, i.e., each processor having a fraction $\approx 1/P$ of FIs. Smaller number of large PBECs could make the scheduling unbalanced.

(2) p_1 creates set of indexes L_i such that $|F_i|/|\mathcal{F}| \approx 1/P$.

In the second step, we need to create index sets L_i, such that $F_i = \bigcup_{k \in L_i}([U_k] \cap \mathcal{F})$ and $\max_i |F_i|/|\mathcal{F}|$ is minimized. This task is known NP-complete problem with known approximation algorithms. We use the LPT-SCHEDULE algorithm (LPT stands for least processing time). The LPT-SCHEDULE algorithm (see [6] for the proofs) is a best-fit algorithm, see Algorithm 3:

Algorithm 3 The LPT-SCHEDULE algorithm

LPT-SCHEDULE(**In:** Set $S = \{(U_i, \Sigma_i, s_i)\}$, **Out:** Sets L_i)

1: Sort the set S such that $s_i < s_j, i \neq j$.
2: Assign each (U_i, Σ_i, s_i) (in decreasing order by s_i) to the least loaded processor p_i. The indexes assigned to p_i, are stored in L_i.

Lemma 5 *[6]* LPT-SCHEDULE *is* $4/3$*-approximation algorithm.*

The index sets L_i together with U_k and Σ_k are then broadcast to the remaining processors.

7.3. Detailed description of Phase 3

The input of Phase 3 for processor p_i is the set of indexes of the assigned PBECs L_i together with the prefixes U_k and its extensions Σ_k. Processor p_i needs for the computation of $F_i = \bigcup_{k \in L_i}([U_k] \cap \mathcal{F})$ a database partition D'_i. The database partition D'_i should contain all the information needed for computation of F_i. At the begginning of this phase, the processors has disjoint database partitions D_i such that $|D_i| \approx \frac{|\mathcal{D}|}{P}$. We expect that we have a distributed memory machine whose nodes are interconnected using a network such as Myrinet or Infiniband, i.e., a network that is not congested while an arbitrary permutation of two nodes communicates with each other. The problem is the congestion of the network in Phase 3.

To construct D'_i on processor p_i, every processor p_j, $i \neq j$, has to send a part of its database partition D'_j needed by the other processors to all other processors (an all-to-all scatter takes place[2]). That is: processor p_i send to processor p_j the set of transactions $\{t|t \in D_i, k \in L_j, \text{ and } U_k \subseteq t\}$, i.e., all transactions that contain at least one $U_k, k \in L_j$ as a subset: $D'_j = \bigcup_i \{t|t \in D_i, k \in L_j, \text{ and } U_k \subseteq t\} = \{t|t \in \mathcal{D}, \text{exists } k \in L_j, U_k \subseteq t\}$. The all-to-all scatter is done in $\lfloor \frac{P}{2} \rfloor$ communication rounds.

We can consider the scatter as a round-robin tournament of P players [7], which is the following procedure: if P is odd, a dummy processor can be added, whose scheduled opponent waits for the next round and the processors performs P communication rounds. For example let have 14 processors, in the first round the following processors exchange their database portions:

1	2	3	4	5	6	7
14	13	12	11	10	9	8

The processors are paired by the numbers in the columns. That is, database parts are exchanged between processors p_1 and p_{14}, p_2 and p_{13}, etc. In the second round one processor is fixed (number one in this case) and the other are rotated clockwise:

1	14	2	3	4	5	6
13	12	11	10	9	8	7

This process is iterated until the processors are almost in the initial position:

1	3	4	5	6	7	8
2	14	13	12	11	10	9

7.4. Detailed description of Phase 4

The input to this phase, for processor $p_q, 1 \leq q \leq P$, is the database partition D_q (the database partition that is the input of the whole method, the database partition), D'_q (computed in Phase 3), the set $\pi = \{(U_k, \Sigma_k)|U_k \subseteq \mathcal{B}, \Sigma_k \subseteq \mathcal{B}, U_k \cap \Sigma_k = \emptyset\}$ of prefixes U_k and the extensions Σ_k, and the sets of indexes L_q of prefixes U_k and extensions Σ_k assigned to processor p_q.

In Phase 4, we execute an arbitrary algorithm for mining of FIs. The sequential algorithm is run on processor p_q for every prefix and extensions $(U_k, \Sigma_k) \in \pi, k \in L_q$ assigned to the processor, i.e., p_q enumerates all itemsets $W \in [U_k|\Sigma_k], (U_k, \Sigma_k) \in \pi$. Therefore, the datastructures used by a sequential algorithm, must be prepared in order to execute the sequential algorithm for mining of FIs with particular prefix and extensions. To make the parallel execution of a DFS algorithm fast, we prepare the datastructures by simulation of the execution of the sequential DFS algorithm, e.g., to enumerate all FIs in a PBEC $[U_k|\Sigma_k]$ Phase 4 simulates the sequential branch of a DFS algorithm for mining of FIs up to the point the sequential algorithm can compute the FIs in $[U_k|\Sigma_k]$.

7.5. The PARALLEL-FIMI-RESERVOIR algorithm

This algorithm samples $\widetilde{\mathcal{F}}$ using the *reservoir sampling*. The reservoir sampling samples $\widetilde{\mathcal{F}}$ uniformly. To make the algorithm faster, the reservoir sampling is executed in parallel. The method is summarized in the PARALLEL-FIMI-RESERVOIR method, see Algorithm 4.

8. Experimental evaluation

We have measured the speedup of our new method, the PARALLEL-FIMI-RESERVOIR method, on a cluster of workstations using three datasets.

The cluster of workstations was interconnected with the Infiniband network. Every node in the cluster has two dual-core 2.6GHz AMD Opteron processors and 8GB of main memory.

The datasets were generated using the IBM database generator. We have used datasets with 500k transactions and supports for each dataset such that the sequential run of the Eclat algorithm is between 100 and 12000 seconds ($\approx$ 3.3 hours) and two cases with running time 33764 (9.37 hours) and 132186 (36.71 hours) seconds. The IBM generator is parametrized by the average transaction length `TL` (in thousands), the number of items `I` (in thousands), by the number of patterns `P` used for creation of the parameters, and by the average length of the patterns `PL`. To clearly differentiate the parameters of a database we are using the string `T[number in thousands]I[items count in 1000]P[number]PL[number]TL[number]`, e.g. the string `T500I0.4P150PL40TL80` labels a database with 500K transactions 400 items, 150 patterns of average length 40 and with average transaction length 80. All experiments were performed with various values of the support parameter on 2, 4, 6, and 10 processors. The databases and supports used for evaluation of our algorithm is summarized in the Table 1.

[2] All-to-all scatter is a well known communication operation: each processor p_i sends a message m_{ij} to processor p_j such that $m_{ij} \neq m_{ik}$, $i \neq k$.

In Phase 4 in our experiments, we use the ECLAT algorithm for mining of FIs. We have used the ECLAT in Phase 1 and 4 of the PARALLEL-FIMI-RESERVOIR method. The PARALLEL-FIMI-RESERVOIR method achieves speedup up to 8.6 on 10 processors.

There is an advantage of the PARALLEL-FIMI-RESERVOIR over the two previous methods [1, 2]: the need of computation of MFIs. The number of MFIs can be very large and the program implementing the PARALLEL-FIMI-SEQ method or the PARALLEL-FIMI-PAR can run out of main memory. The PARALLEL-FIMI-RESERVOIR does not suffer from this problem.

Figure 1 clearly demonstrate that for reasonably large and reasonably structured datasets, the speedup is linear with speedup ≈ 6 on 10 processors. The numeric values of the speedup are located in Table 2.

In [2], we have evaluated the PARALLEL-FIMI-PAR as faster then the PARALLEL-FIMI-SEQ method. We can compare the speedup of the PARALLEL-FIMI-RESERVOIR method with the PARALLEL-FIMI-PAR method. In the PARALLEL-FIMI-PAR method we have used the Eclat algorithm in Phase 4 and the Fpmax* [8] algorithm in Phase 1 as the algorithm for mining of MFIs. The speedup of PARALLEL-FIMI-PAR method is shown in Figure 1 the numerical average speedup values are located in Table 2. We can see that the speedup of the PARALLEL-FIMI-PAR is a bit smaller then the speedup of PARALLEL-FIMI-RESERVOIR. Additionally, in some cases, we were not able to finish the execution of the PARALLEL-FIMI-PAR due to large amount of used memory. In such cases the speedup is shown to be 0.

Algorithm 4 The PARALLEL-FIMI-RESERVOIR method.

PARALLEL-FIMI-RESERVOIR **In:** Double $min\_support^*$,
In: Doubles $\epsilon_{\widetilde{\mathcal{D}}}, \delta_{\widetilde{\mathcal{D}}}, \epsilon_{\widetilde{\mathcal{F}}_s}, \delta_{\widetilde{\mathcal{F}}_s}, \rho, \alpha$,
Out: Set $\mathcal{F}$)

1: **for** all p_i **do-in-parallel**
// Phase 1: sampling.
2: Read D_i and set $N_{\widetilde{\mathcal{D}}} \leftarrow \frac{1}{2\epsilon^2_{\widetilde{\mathcal{D}}}} \ln \frac{2}{\delta_{\widetilde{\mathcal{D}}}}$.
3: Creates a sample $D'_i \subseteq D_i$ and broadcast it to each other processor.
4: $\widetilde{\mathcal{D}} \leftarrow \bigcup_i D'_i$.
5: Execute in parallel an arbitrary algorithm for mining of FIs on database $\widetilde{\mathcal{D}}$ in parallel and create the sample $\widetilde{\mathcal{F}}_s$ using the VITTER-RESERVOIR-SAMPLING.
// Phase 2: partitioning.
6: p_1 creates PBECs $[U_k]$ such that $\frac{|[U_k] \cap \widetilde{\mathcal{F}}_s|}{|\widetilde{\mathcal{F}}_s|} < \alpha \cdot \frac{1}{P}$.
7: p_1 creates $L_j, 1 \leq j \leq P$ using the LPT-MAKESPAN algorithm.
8: *// Phase 3: data re-distribution.*
9: Redistribute the database partition D_i
10: *// Phase 4: parallel computation of FIs.*
11: compute support of $W \subseteq U_k$ in D_i and send the supports to p_1
12: p_1 outputs W
13: all p_i executes an arbitrary algorithm for mining of FIs in parallel that computes supports of $Supp(W, D'_q), W \in \bigcup_{k \in L_q} [U_k|\Sigma_k]$, $(U_k, \Sigma_k) \in \pi$.
14: **end for**

The database replication: we define the *database replication factor* as follows: let D'_i is the database partition used by the ith processor in Phase 4, i.e., the database part received in Phase 3. The database replication factor is defined as follows: $\frac{\sum_{i=1}^{P} |D'_i|}{|\mathcal{D}|}$. One would expect that the database replication factor will be small, e.g., for $P = 10$ a replication factor between $2 - 6$ would be expected. The oposite is the true. The replication factor is in all our experiments $\approx P$. The minimalization of the database replication factor is a hard task. The minimization of the database replication factor is an opened problem.

9. Acknowledgment

This paper was supported from the Czech Science Foundation, grant number GA ČR P202/10/1333.

Dataset	Supports
T500I0.1P100PL20TL50	$0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18$
T500I0.4P250PL10TL120	$0.2, 0.25, 0.26, 0.27, 0.3$
T500I1P100PL20TL50	$0.02, 0.03, 0.05, 0.07, 0.09$

Table 1: Databases used for measuring of the speedup and used supports values for each dataset.

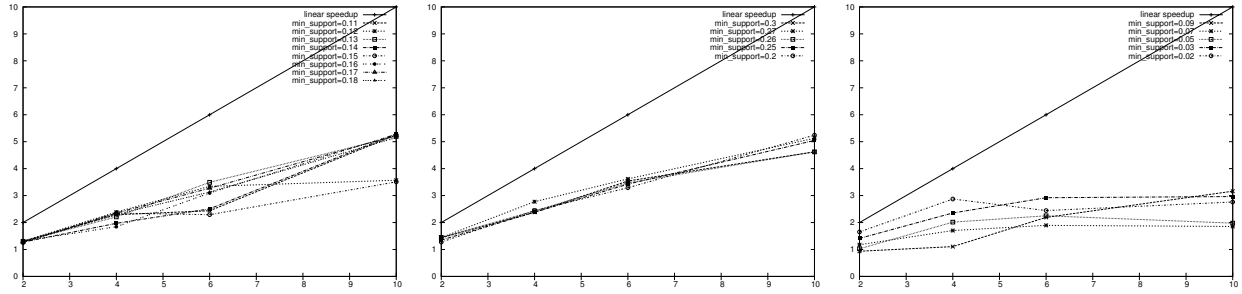


Figure 1: Speedup of the PARALLEL-FIMI-RESERVOIR method parametrized with the Eclat algorithm, measured on the `T500I0.1P100PL20TL50`, `T500I0.4P250PL10TL120`, `T500I1P100PL20TL50` (from left to right).

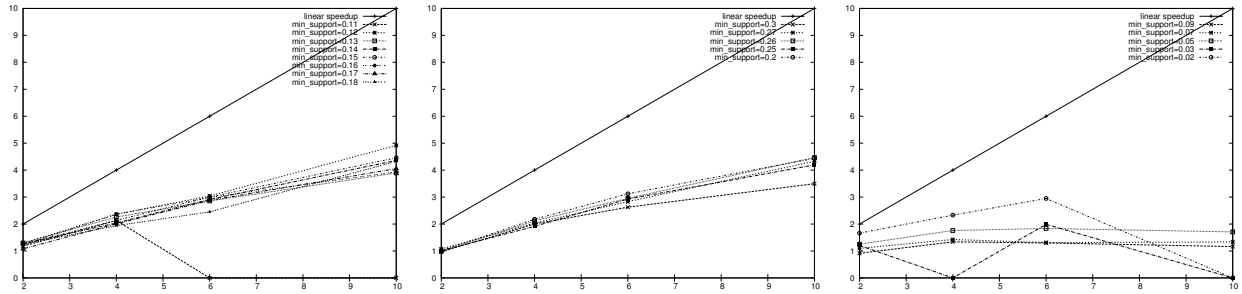


Figure 2: Speedup of the PARALLEL-FIMI-PARmethod parametrized with the Eclat algorithm, measured on the `T500I0.1P100PL20TL50`, `T500I0.4P250PL10TL120`, `T500I1P100PL20TL50` (from left to right).

datafile/PARALLEL-FIMI-RESERVOIR	2	4	6	10
T500I0.1P50PL10TL40	1.523	2.633	3.380	5.342
T500I0.4P250PL10TL120	1.389	2.481	3.470	4.932
T500I1P100PL20TL50	1.240	2.010	2.340	2.544
Total average	1.384	2.375	3.063	4.273

datafile/PARALLEL-FIMI-PAR	2	4	6	10
T500I0.1P50PL10TL40	1.596	2.668	3.438	5.135
T500I0.4P250PL10TL120	1.010	2.050	2.891	4.186
T500I1P100PL20TL50	1.227	1.714	1.876	1.401
Total average	1.277	2.144	2.735	3.574

Table 2: Numerical values of average speedup of the PARALLEL-FIMI-RESERVOIR and PARALLEL-FIMI-PAR methods for number of processors $P = 2, 4, 6, 10$.

References

[1] R. Kessl and P. Tvrdík, Probabilistic load balancing method for parallel mining of all frequent itemsets. In *PDCS '06: Proceedings of the 18th IASTED International Conference on Parallel and Distributed Computing and Systems*, pages 578–586, Anaheim, CA, USA, 2006. ACTA Press.

[2] R. Kessl and P. Tvrdík, Toward more parallel frequent itemset mining algorithms. In *PDCS '07: Proceedings of the 19th IASTED International Conference on Parallel and Distributed Computing and Systems*, pages 97–103, Anaheim, CA, USA, 2007. ACTA Press.

[3] H. Toivonen, Sampling large databases for association rules. In T. M. Vijayaraman, A. P. Buchmann, C. Mohan, and N. L. Sarda, editors, *In Proc. 1996 Int. Conf. Very Large Data Bases*, pages 134–145. Morgan Kaufman, 09 1996.

[4] J.S. Vitter, Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, March 1985.

[5] M. Skala, Hypergeometric tail inequalities: ending the insanity. http://ansuz.sooke.bc.ca/professional/hypergeometric.pdf.

[6] R.L. Graham, Bounds on multiprocessing timing anomalies. *SIAM Journal of Applied Mathematics*, 17(2):416–429, 1969.

[7] http://en.wikipedia.org/wiki/Round_robin_tournament.

[8] G. Grahne and J. Zhu, Efficiently using prefix-trees in mining frequent itemsets. In *FIMI '03, Frequent Itemset Mining Implementations, Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, 19 December 2003, Melbourne, Florida, USA*, volume 90 of *CEUR Workshop Proceedings*, 2003.

Analysis of Algebraic Preconditioning Based on Different Variants of the Gram-Schmidt Algorithm

Post-Graduate Student:
ING. JIŘÍ KOPAL

Faculty of Mechatronics and Interdisciplinary Engineering Studies
Technical University of Liberec
Hálkova 6

461 17 Liberec 1, CZ

jiri.kopal@tul.cz

Supervisor:
PROF. ING. MIROSLAV TŮMA, CSC. [1]
DOC. DR. ING. MIROSLAV ROZLOŽNÍK [2]

Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2

182 07 Prague 8, CZ

tuma@cs.cas.cz [1], miro@cs.cas.cz [2]

Field of Study:
Science Engineering

This work was supported by the international collaboration support M100300902 of AS CR and by the grant No. 102/08/H081 of the Grant Agency of the Czech Republic.

Abstract

This paper deals with preconditioning of iterative methods with a symetric positive definite matrix A which are based on A-orthogonalization (generalized Gram-Schmidt algorithm). The main goal of our work is to bound accuracy of the algorithm which provides an matrix inverse factorization, via its loss of orthogonality. In practice, this accuracy reflects stability of the preconditioner based on this factorization. The analysis presented here assumes variants of the algorithm which do not use dropping of small entries, and it allows a fully general input. The derived bounds are accompanied by results of experiments using both artificial and real-world problems.

1. Introduction

Many problems of physical and technical sciences lead to a system of linear algebraic equations which have to be solved by direct or iterative methods. There are two reasons, why an iterative solver is a method of choice. The first one is that the amount of needed memory to solve the problem is often much smaller than direct methods typically use. The second reason is that the CPU time needed to get a useful, at least approximate, solution which is also very often significantly smaller for large problems. Note that the amount of floating-point operations needed to get the solution by a direct method (Gaussian elimination) is $\approx O(n^3)$ if A is dense. But in order to get the iterative solvers reasonably robust and fast, they should be preconditioned.

There are lot of ways to compute simple preconditioners which may also take into account parallel computing architecture. A famous class of algebraic preconditioners is based on the incomplete LU factorization (see their characteristics and some notes on their history in [1] or [2]). Here we will deal with preconditioners P which directly approximate the inverse problem $\approx A^{-1}$. Such types of preconditioners are sometimes called explicit. An interesting type of such preconditioners is provided by the SPAI [3] procedure, which minimizes the functional $\|I - PA\|_F$ and enables decomposition to n-independent problems. Thus this approach is naturally parallel. Further techniques are called AINV [4] and SAINV (stabilized AINV) [5]. They are based on A-orthogonalization (generalized Gram-Schmidt algorithm). Their application can exploit a lot of available parallelism and they were proved to be efficient for solving some significant classes of problems. These techniques are the topic of this paper.

Section 2 discusses preconditioning techniques. Section 3 is dedicated to the Gram-Schmidt algorithm in general, and Section 4 presents results of our analysis of the discussed algorithms. Finally Section 5 contains some experimental results for test problems. Section 6 concludes this contribution summarizing also the future work.

2. Preconditioning

Good preconditioner of a sparse matrix should be sparse, in order to induce the product PA sparse as well. Then the matrix-vector multiplications with A and P should not be expensive. Every specific preconditioning approach has a mechanism for preserving sparsity. One such possibility is to prescribe the pattern of nonzero entries of the preconditioner in advance (e.g., using the pattern of symbolically computed powers of the matrix A).

This pattern can be also formed dynamically by dropping entries which are small in some sense. Apart from the sparsity, successful preconditioners have to satisfy also additional requirements. In particular, their *accuracy* $\|I - PA\|$ and *stability* $\|A - P^{-1}\|$ (see also [1]) should not be very large. A bound for still allowable accuracy and stability are typically very individual for each algorithm and problem to be solved. Currently, there is no general theory which connects accuracy and stability with convergence of a preconditioned iterative solver.

3. Gram-Schmidt algorithm

The generic Gram-Schmidt algorithm was historically defined in more variants which differ by the order of computational operations.

- classical - **CGS**
- modified - **MGS**

Both algorithms are numerically equivalent in exact arithmetics, but there is a strong difference in finite precision arithmetic. This can be easily demonstrated by both analytical and experimental results. The main visual computational difference between these variants of the Gram-Schmidt algorithm is an amount of their granularity. Namely, CGS can be easily parallelized on the level of vector operations, whereas MGS can be parallelized only on finer grain level. The computational difference between algorithms is clear from their schematic description as shown below:

MGS algorithm

for $k = 1 : n$
  $q_k := a_k$
  for $j = 1 : k - 1$
    $R_{j,k} := q_j^T a_k$
    $q_k := q_k - R_{j,k} q_j$
  end for
  $R_{j,j} := \|q_k\|$
  $q_k := q_k / R_{j,j}$
end for

CGS algorithm

for $k = 1 : n$
  $q_k := a_k$
  for $j = 1 : k - 1$
    $R_{j,k} := q_j^T a_k$
  end for
  for $j = 1 : k - 1$
    $q_k := q_k - R_{j,k} q_j$
  end for
  $R_{j,j} := \|q_k\|$
  $q_k := q_k / R_{j,j}$
end for

Both the above presented algorithms are in the so-called *left looking* form. In this case this means that the matrix R is obtained by columns. If we would swap inner and outer loop in the algorithms, matrix R would be constructed by rows. This would provide the so-called *right looking* form. Both right looking and left looking forms of the algorithm are identical even numerically. The main difference between these variants reveals in the parallel computing environment, where the right looking form has typically better performance then left looking [6] for large sparse problems.

Some of the main common features of the variants of the Gram-Schmidt algorithm which can be written in the matrix form as

$$A = QR$$

are

- $QQ^T = I$, R is an upper triangular matrix,
- the algorithm is backward stable,
- it can be used to compute underdetermined/overdetermined systems,
- it is more difficult to keep the algorithm sparse than the Cholesky or LU decomposition

Generalized Gram-Schmidt

As in the quoted papers, for the computation of the above introduced factorized inverse preconditioner we use a generalized Gram-Schmidt algorithm which is based on the energetic dot product $\langle z_j, z_k \rangle_A$ (A-ortogonalization) and which assumes inital selection of vector basis given by $Z^{(0)} = [z_1^{(0)} z_2^{(0)} \cdots z_n^{(0)}]$. Because of the energetic dot product there is one general limitation with respect to the non-generalized algorithm. Namely, matrix A has to be symmetric positive definite. The generalized MGS algorithm can be put down as follows:

for $k = 1 : n$
  $z_k := z_i^{(0)}$
  for $j = 1 : k - 1$
    $U_{j,k} := z_k^T A z_j$
    $z_k := z_k - U_{j,k} z_j$
  end for
  $U_{j,j} := (z_k^T A z_k)^{1/2}$
  $z_k := z_k / U_{j,j}$
end for

An incomplete version of this algorithm introduced in [5] is called SAINV. In exact case, the inverse matrix can be written using Z in the form $A^{-1} = Z^T Z$. If we choose the matrix $Z^{(0)}$ in an upper triangular form, we can write the inverse Cholesky factor of the matrix $A = LL^T$ as $Z = L^{-1}$. In particular, $A = U^T U$ for $Z^{(0)} = I$. In all the other cases, Z is a general matrix instead of being upper triangular. The loss of orthogonality can be written using the matrix Z in 3 identical forms as follows: $\|ZZ^T A - I\| = \|AZZ^T - I\| = \|Z^T AZ - I\|$. The identity of these forms we get from simple transformations of the related generalized eigenvalue problem.

The AINV algorithm is a specific combination of the generalized MGS and CGS algorithms and it can be written as follows:

for $k = 1 : n$
  $z_k := z_i^{(0)}$
  for $j = 1 : k - 1$
    $U_{j,k} := z_k^T A z_j^{(0)}$
    $z_k := z_k - U_{j,k} z_j$
  end for
  $U_{j,j} := (z_k^T A z_k)^{1/2}$
  $z_k := z_k / U_{j,j}$
  $z_k^{(0)} := z_k^{(0)} / z_k^T A z_k^{(0)}$
end for

The main practical power of the AINV algorithm results from the special choice of $Z^{(0)}$. Namely, if we choose diagonal $Z^{(0)}$ then the algorithm is matrix free and we can work in each major step of AINV only with one particular column of the matrix A. Of course, we can choose $Z^{(0)}$ as a general full-rank matrix as possible but for practical problems and with the intention to use the decomposition for preconditioning it has no sense to choose $Z^{(0)}$ other than $Z^{(0)} = diag(A)^{-1/2}$ or $Z^{(0)} = I$.

Extensions for ill-conditioned problems

For ill-conditioned problems algorithms should be modified in order to obtain good results. Let us mentione two basic possibilities of these modifications:

- iterative refinement,
- pivoting strategies,

which can be used also at the same time. Iterative refinement is based on a simple stationary postprocessing of the algorithm which corrects the already approximately orthogonalized vectors and coefficients. This can take sometimes significantly more CPU time than the generic algorithm. Pivoting strategies are connected with construction of the matrix R or U, respectively, so that the sequence of diagonal entries does not grow much. From the implementational point of view, this method is the most suitable for the right looking MGS, where there are just two sources of the additional costs caused by the pivoting. First, column permutations in matrices Q and R or in Z and U, should be applied in the standard or generalized algorithms, respectively. Second, pivoting needs additional computations of column norms. All the other cases of the Gram-Schmidt family are less suitable for the pivoting. In the general case, the sequence of the column vectors in the matrix $Z^{(0)}$ can be easily changed. The analysis of algorithms with iterative refinement and pivoting is more complicated. For the non-generalized CGS with iterative refinement see [7].

4. Analysis in finite precision arithmetics

This Section presents and summarizes the main results which we have achieved. In exact arithmetic we have $\|Z^T AZ - I\| = 0$, but for the purpose of particular preconditioning analysis it is necessary to evaluate the loss of orthogonality $\|\bar{Z}^T A\bar{Z} - I\| =?$ in the finite precision arithmetics, where $\bar{Z}$ is computed matrix Z. An important work from which our analysis stems out is [8] which deals with the (non-generalized) MGS. Further ideas for the analysis were found in [7] and [9]. No variant of generalized Gram-Schmidt algorithm have been analyzed yet. Our tools included rounding error analysis and analysis of the errors in the recurrent formulas for columns of the inverse factor Z.

CGS:

$$\|I - \bar{Z}^T A\bar{Z}\| \leq \frac{O(u)\kappa(A)\kappa^2(A^{1/2}Z^{(0)})}{1 - O(u)\kappa(A)\kappa^2(Z^{(0)})}$$

MGS (SAINV):

$$\|I - \bar{Z}^T A\bar{Z}\| \leq \frac{O(u)\kappa(A)\kappa(A^{1/2}Z^{(0)})}{1 - O(u)\kappa(A)\kappa(A^{1/2}Z^{(0)})}$$

MGS (for diagonal A):

$$\|I - \bar{Z}^T D\bar{Z}\| \leq \frac{O(u)\kappa(D^{1/2}Z^{(0)})}{1 - O(u)\kappa(D^{1/2}Z^{(0)})}$$

AINV and algorithms with iterative refinement and pivoting:

$$\|I - \bar{Z}^T A\bar{Z}\| \leq \text{still in progress...}$$

Note that the numerator of the estimates forms the main part of the expression, and it more or less determines

the bound for the loss of orthogonality. The denominator can be considered only as a restriction in the limiting case. In the other words, this denominator can influence the asymptotic relation more significantly only in very ill-conditioned cases. Note that the denominator should be positive as well.

5. Test problems and experimental results

In this section we will present our test problems as well as some experimental results for the considered class of algorithms in finite precision arithmetics. The results are compared with the ideally exact implementation computed as a sequence of SVD and QR decomposition [10].

Ideal implementation
Assume the following backward stable spectral decomposition of the matrix A:

$$A + E_0 = \bar{V}\bar{\Lambda}\bar{V}^T, \|E_0\| \leq O(u)\|A\|,$$

where $\|\bar{V}\bar{V}^T - I\| \leq O(u)$. Multiplying the matrix $Z^{(0)}$ by $\bar{\Lambda}^{1/2}\bar{V}^T$ and applying the Householder QR to the product $fl(\bar{\Lambda}^{1/2}\bar{V}^T Z^{(0)})$ we can write the following identity for the computed factors $\bar{Q}$ and $\bar{U}$

$$\bar{\Lambda}^{1/2}\bar{V}^T Z^{(0)} = \bar{Q}\bar{U}+E_1, \|E_1\| \leq O(u)\|A\|^{1/2}\|Z^{(0)}\|,$$

where $\|I - \bar{Q}^T\bar{Q}\| \leq O(u)$. The matrix $\bar{Z}$ is computed then as the product $\bar{Z} = fl(\bar{V}\bar{\Lambda}^{-1/2}\bar{Q})$ satisfying

$$\begin{aligned} \bar{Z} &= \bar{V}\bar{\Lambda}^{-1/2}\bar{Q} + E_2, \\ \|E_2\| &\leq O(u)\|\bar{\Lambda}^{-1}\|^{1/2} \leq O(u)\|\bar{Z}\|. \end{aligned}$$

5.1. Real-world problems

We have chosen as real problems a set of structural engineering matrices from MatrixMarket, for more details see [11].

The figures clearly reveal that the derived bounds for the loss of orthogonality significantly overestimate the experimental data. In order to get our estimates closer to the experiments, we will construct in the next section a specific artificially made sequence of matrices.

name	estimate of $\kappa(A)$
BCSSTK01	1.6e+06
BCSSTK02	1.3e+04
BCSSTK03	9.5e+06
BCSSTK04	5.6e+06
BCSSTK05	3.5e+04
BCSSTK06	1.2e+07
BCSSTK07	1.2e+07
BCSSTK08	4.7e+07
BCSSTK09	3.1e+04
BCSSTK10	1.3e+06
BCSSTK11	5.3e+08
BCSSTK12	5.3e+08
BCSSTK13	4.6e+10
BCSSTK14	1.3e+10
BCSSTK15	8e+09

Table 1: Selected real-world problems.

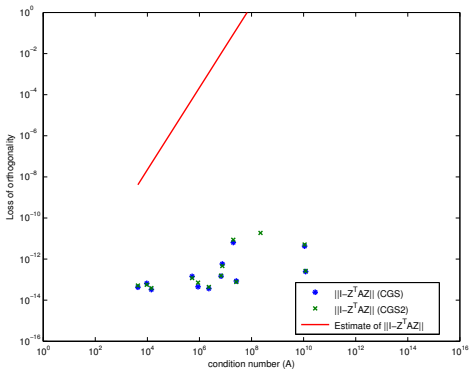


Figure 1: Loss of orthogonality for the real-world problems (CGS).

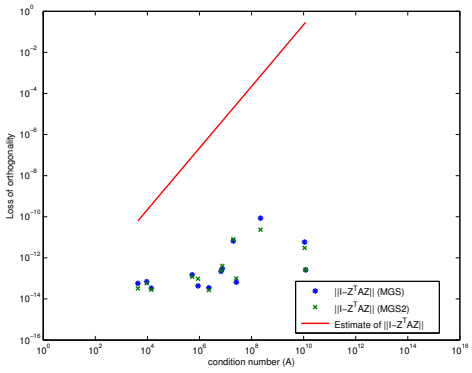


Figure 2: Loss of orthogonality for the real-world problems (MGS).

5.2. Artifical problems

Test problem 1, dimension of the problem $n = 8$, with $Z^{(0)} = A^{1/2}$, where A is the inverse Hilbert matrix $\left(\kappa(Z^{(0)}) \approx 10^5\right)$ and $A^{(i)}$ is a diagonal matrix with $\kappa(A^{(i)}) \approx 10^i, i = 0, \ldots, 15$.

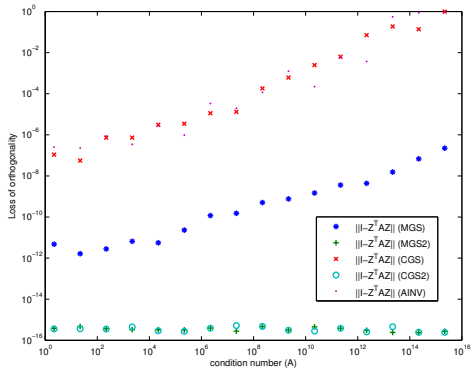


Figure 3: Loss of orthogonality for the test problem 1.

Test problem 2 was motivated by our analysis. Since the largest error in the algorithm corresponds to the singular vector with the minimal singular value. More in detail: Dimension of the problem $n = 8$, where $A^{(i)} = V\Lambda^{i/10}V^T$ is a power of the inverse Hilbert matrix $A = V\Lambda V^T$ ($\kappa(A) \approx 10^{10}$) with $\kappa(A^{(i)}) \approx 10^i, i = 0, \ldots, 15$. The matrix $Z^{(0,i)}$ is constructed as $Z^{(0,i)} = V\Lambda^{-i/20}(L^{(i)})^T$, where $L^{(i)}$ is the Cholesky factor of the matrix $A^{(i)} = L^{(i)}(L^{(i)})^T$, $\kappa(Z^{(0,i)}) \approx 10^i, i = 0, \ldots, 15$.

The experiments with this test problem clearly show that the quality of the results is strongly affected by the strategy which was used to compute the energetic dot products. The variant "a"assumes to $\bar{\alpha}_{ji} = \mathrm{fl}[\langle \bar{z}_i^{(j-1)}, \mathrm{fl}(A\bar{z}_j)\rangle]$ whereas the variant "b"corresponds to $\bar{\alpha}_{ji} = \mathrm{fl}[\langle \mathrm{fl}(A\bar{z}_i^{(j-1)}), \bar{z}_j\rangle]$.

We can see the significant difference in behaviour in particular for MGS, despite fact that the rounding errors are the same in both cases ('a' and 'b'):

$$\begin{aligned} |\mathrm{fl}[\langle \bar{z}_i^{(j-1)}, \mathrm{fl}[A\bar{z}_j]\rangle] &- \langle \bar{z}_i^{(j-1)}, \bar{z}_j\rangle_A| \\ &\leq O(u)\|A\|\|\bar{z}_j\|\|\bar{z}_i^{(j-1)}\| \\ |\mathrm{fl}[\langle \mathrm{fl}[A\bar{z}_i^{(j-1)}], \bar{z}_j\rangle] &- \langle \bar{z}_i^{(j-1)}, \bar{z}_j\rangle_A| \\ &\leq O(u)\|A\|\|\bar{z}_j\|\|\bar{z}_i^{(j-1)}\| \end{aligned}$$

Rounding error of the (non-energetic) dot product has the form:

$$|x^T y - \mathrm{fl}[x^T y]| \leq O(u)\|x\|\|y\|.$$

The estimate can be also extended to energetic dot product. More precisely, according to [10] we have:

Variant "a"
Substituting $y_a = A\bar{z}_j$, $x_a = \bar{z}_i^{(j-1)}$ we get the computed quantity $\bar{y}_a = \mathrm{fl}[A\bar{z}_j] = (A + \Delta A_1)\bar{z}_j$, where

$$\|\bar{y}_a - y_a\| \leq O(u)\|A\|\|\bar{z}_j\|.$$

$$\begin{aligned} &|(\bar{z}_i^{(j-1)})^T\bar{y}_a - \mathrm{fl}[(\bar{z}_i^{(j-1)})^T\bar{y}_a]| \leq O(u)\|\bar{z}_i^{(j-1)}\|\|\bar{y}_a\| \\ &|(\bar{z}_i^{(j-1)})^T A\bar{z}_j - \mathrm{fl}[(\bar{z}_i^{(j-1)})^T\mathrm{fl}[A\bar{z}_j]]| \\ &\leq O(u)\|\bar{z}_i^{(j-1)}\|\|A\bar{z}_j\| + O(u)\|(\bar{z}_i^{(j-1)})^T\Delta A_1\bar{z}_j\| \end{aligned}$$

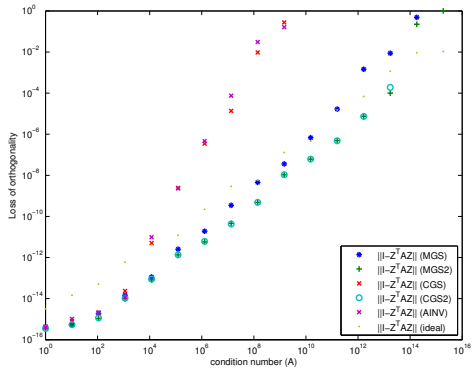


Figure 4: Loss of orthogonality for the test problem 2a.

Variant "b"
Substituting $x_b = A\bar{z}_i^{(j-1)}$, $y_b = \bar{z}_j$ we get the computed quantity $\bar{x}_b = \mathrm{fl}[A\bar{z}_i^{(j-1)}] = (A + \Delta A_2)\bar{z}_i^{(j-1)}$, where $\|\bar{x}_b - x_b\| \leq O(u)\|A\|\|\bar{z}_i^{(j-1)}\|$.

$$\begin{aligned} &|(\bar{x}_b)^T\bar{z}_j - \mathrm{fl}[(\bar{x}_b)^T\bar{z}_j]| \leq O(u)\|\bar{x}_b\|\|\bar{z}_j\| \\ &|(\bar{z}_i^{(j-1)})^T A\bar{z}_j - \mathrm{fl}[\mathrm{fl}[(A\bar{z}_i^{(j-1)})^T]\bar{z}_j]| \\ &\leq O(u)\|A\bar{z}_i^{(j-1)}\|\|\bar{z}_j\| + O(u)\|(\bar{z}_i^{(j-1)})^T\Delta A_2\bar{z}_j\| \end{aligned}$$

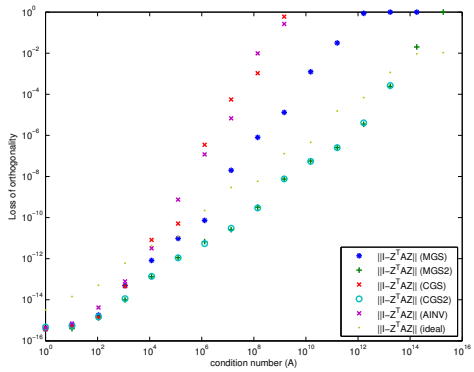


Figure 5: Loss of orthogonality for the test problem 2b.

The rounding error has two components, but in general we cannot see which is more important here.

Test problem 3 is as follows: dimension of the problem is $n = 8$, where $A^{(i)} = Z^{(0,i)} = V\Lambda^{i/10}V^T$ is a power of the inverse Hilbert matrix $A = V\Lambda V^T$ ($\kappa(A) \approx 10^{10}$) with $\kappa(A^{(i)}) \approx 10^i, i = 0, \ldots, 15$.

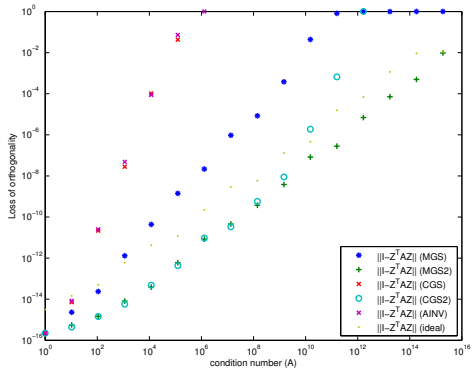


Figure 6: Loss of orthogonality for the test problem 3.

6. Conclusion and future work

In this paper we summarized the results we have achieved in analysis of the generalized Gram-Schmidt algorithms. Experimental results demonstrate the estimated theoretical differences of numerical properties of the investigated algorithms. In order to use these algorithms for preconditioning we need to extend our analysis also for incomplete generalized Gram-Schmidt algorithms. That is, for algorithms which employ dropping of small entries. Our final goal is to control the loss of orthogonality by the amount of dropping. We believe that in this case we will be able to develop robust preconditioners useful in practical applications.

References

[1] Y. Saad, "Iterative Methods for Sparse Linear Systems", *WPS*, 1996.

[2] J.A. Meijerink and H.A. van der Vorst, "An Iterative Solution Method for Linear Systems of Which the Coefficient Matrix is a Symmetric M-Matrix", *Math.Comp. 31*,148, 1977.

[3] T. Huckle, A. Kallischko, A. Roy, M. Sedlacek, and T. Weinzierl, "An efficient parallel implementation of the MSPAI preconditioner", *Parallel Computing*, 36(5-6): 273-284, 2010.

[4] M. Benzi, C.D. Meyer, and M. Tůma, "A sparse approximate inverse preconditioner for the conjugate gradient method", *SIAM J. Scientific Computing*,17, pp. 1135–1149, 1996.

[5] M. Benzi and M. Tůma, "A robust incomplete factorization preconditioner for positive definite matrices", *Numerical linear algebra factorization with applications*, vol. 10, pp. 385–400, 2003.

[6] M. Tůma, Personal comunication, 2010.

[7] L. Giraud, J. Langou, M. Rozložník, and J. van den Eshof, "Rounding error analysis of the classical Gram-Schmidt orthogonalization process", *Numerische Mathematik*, 101: 87-100, 2005.

[8] Å. Björck, "Solving linear least squares problems by Gram-Schmidt ortogonalization", *BIT*, vol. 7, pp. 1–21, 1967.

[9] N.J. Higham, "Accuracy and Stability of Numerical Algorithms - Second Edition", *SIAM*, 2002.

[10] M. Rozložník, Personal comunication, 2010.

[11] www.math.nist.gov/MatrixMarket/index.html, web page.

Harmonization of Clinical Content in EHR: HL7 Messaging vs. OpenEHR Approach

Post-Graduate Student:
MGR. MIROSLAV NAGY
Department of Medical Informatics
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
nagy@euromise.cz

Supervisor:
RNDR. ANTONÍN ŘÍHA, CSC.
Department of Medical Informatics
Instutite of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
riha@euromise.cz

Field of Study:
Biomedical Informatics

This work was partially supported by the project of the Institute of Computer Science of Academy of Sciences of the Czech Republic AV0Z10300504 and by the project of the Ministry of Education, Youth and Sports of the Czech Republic No. 1M06014.

Abstract

This paper sums up the last stage of the project called "Harmonization of Clinical Content in Electronic Health Record (EHR)", which is being solved as author's PhD project. The aim of this project was the clinical content modeling of EHR and transportation of medical data via communication standards (HL7 v3 and CEN EN 13606). An experiment was conducted where archetypes and openEHR templates were used to send patient's information to another EHR-S. The results were compared to the previous solution based on HL7 v3 messages, which was described at the PhD Conference '09. At the end of the paper the author discusses contemporary semantics description of real clinical data by means of ICD-10 code-list, which covers the qualitative point of view quite well. However, the quantitative view on semantics of clinical data is much more difficult and the inevitability of medical ontologies usage is emphasized.

1. Introduction

Contemporary initiatives in e-Health aim to carry a proposal that would enforce clinical data usage in an electronic form. Benefits of the electronic document interchange in healthcare are becoming obvious to the broader professional and laic audience. The basic prerequisite for this to carry on is a high quality source of clinical data, i.e. well structured EHR.

Defining a structure of clinical data stored in various EHRs is the most important issue, which, at the same time, causes most problems with interoperability among EHR systems (EHR-S). Therefore an indispensable amount of effort is made to unify the modeling process in order to make the semantic interoperability real.

One of the most significant initiatives in standardization of EHRs is the CEN EN 13606 norm and specifications of the openEHR Foundation. Relations between these two are described in [1] and they can be summarized as follows. Currently there is no standard available for a Shared-EHR system which supports the creation, storage, maintenance, and querying of Shared EHRs. OpenEHR is the only open specification currently available and it is a candidate for this purpose. On the other hand, CEN EN13606 is an appropriate standard for the exchange of Shared EHR Extracts, which are comprehensive collections of data designed to transfer clinical data from one EHR to another.

2. Materials

2.1. Two-level modeling

In order to use the openEHR archetypes we have to define them, as well as the basic modeling paradigm of openEHR - the two-level modeling paradigm. According to [2] an archetype (from the technical point of view) is "a computable expression of a domain-level concept in the form of structured constraint statements, based on some reference information model". Each archetype [3] is a set of constraints on the reference model, defining a subset of instances that are considered to conform to the subject of the archetype, e.g. "laboratory result". An archetype can thus be thought of as being similar to a LEGO instruction sheet.

Under the two-level approach [4], [3], a stable reference information model constitutes the first level of mode-

ling, while formal definitions of clinical content in the form of archetypes and templates constitute the second.

2.2. Ontologies

Ontology has many definitions depending on the angle of view you look at it. For the purpose of this paper we can take the definition from [5]. In the context of computer and information sciences, an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). Ontologies are typically specified in languages that allow abstraction away from data structures and implementation strategies.

In our department, a research in the field of knowledge modeling was accomplished resulting in creation of modeling concepts and tools. It will be demonstrated in the Results section.

2.3. MDMC and its mapping to SNOMED CT

The Minimal Data Model of Cardiology (MDMC) [6] is a set of approximately 150 attributes, their mutual relations, integrity restrictions, units, etc. prepared according to needs of statisticians. Prominent professionals in the field of Czech cardiology agreed on these attributes as on the basic data necessary for an examination of a patient in cardiology.

Description of encoded concept	**SNOMED CT Code**
Drug allergy (disorder)	416098002
Diabetes mellitus (disorder)	73211009
Vascular disorder of lower extremity (disorder)	373408007
Cerebrovascular accident (disorder)	230690007
Normal menopause (finding)	237123000
Body weight (observable entity)	248345008
Hip circumference (observable entity)	284472007
Respiratory rate (observable entity)	86290005
Atrial fibrillation (disorder)	49436004
ECG finding (observable entity)	271921002

Table 1: Some of the MDMC concepts mapped to SNOMED CT.

We started encoding the concepts of the MDMC using SNOMED-CT codes in order to describe the semantics in a standardized way. These coded terms were used to avoid ambiguity in the next part of our work where we integrated the HL7 v3 standard and openEHR artifacts into EHR systems based on MDMC.

Some example concepts from MDMC and their mapping to SNOMED CT can be found in the Table 1.

2.4. Clinical Knowledge Manager and Ocean Archetype Editor

The notion of archetypes has been known for several years. As a result there already exist some archetypes describing most common concepts and their sharing started to have sense. For this purpose a Clinical Knowledge Manager (CKM) is available at http://www.openehr.org/knowledge/. It is a repository designed to share (serves as a library of archetypes and templates), supports the full life cycle management of archetypes and provides governance of the knowledge artifacts.

In case that the CKM repository does not contain certain archetypes, there is a tool for designing new archetypes. The Ocean Archetype editor is a tool developed by OceanInformatics and supports creating archetypes and binding terms they contain to coding systems. After creating an archetype it can be exported in an abstract syntax, i.e. the ADL format [7].

2.5. Clinical Data Source and LIM templates

As a source of clinical data the ADAMEKj EHR-S [8] was used. ADAMEKj was developed in the Department of Medical Informatics ICS AS CR with the aim to collect outpatient genetic and clinical data in cardiology. The domain model of the application is based on the MDMC. The user interface was inspired by the former ADAMEK [9] application.

The ADAMEKj LIMFiller module was used to retrieve data from the ADAMEKj EHR-S into the form of LIM messages [10]. LIM messages were then used for further transformations - to fill the archetypes with data.

LIM templates [10] were developed in the project called "Information Technologies for the Development of Continuous Shared Healthcare"(ITDCSH). For the purpose of this paper we describe the LIM template for Patient's Physical Examination. The LIM template is depicted in the form of XML-Schema in the Figure 1.

```
<?xml version ='1.0' encoding ='UTF-8'?>
<xs:schema targetNamespace="urn:lim" xmlns:xs="XMLSchema" xmlns:limDt="urn:lim">
  <xs:import namespace="limDt" schemaLocation="limDt.xsd"/>
  <xs:element name="lim">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="header" minOccurs="1" maxOccurs="1" type="limDt:header"/>
        <!-- Šablona: Fyzikální vyšetření -->
        <!-- begin hl7class type="Act-cls" -->
        <xs:element name="Fyzikální_vyšetření" minOccurs="1" maxOccurs="1">
          <!-- LOINC: 129545-1 PHYSICAL FINDINGS FIND PT ^PATIENT NAR OBSERVED H&amp;P.PX 2 -->
          <!-- begin hl7class type="ActRelationship-cls" -->
          <xs:element name="Měření_výšky_je_součástí_fyzikálního_vyšetření" minOccurs="0" maxOccurs="unbounded">
            <!-- begin hl7class type="Observation-cls" -->
            <xs:element name="Měření_Výšky" minOccurs="1" maxOccurs="1"/>
            <!--
                LOINC: 8308-9 BODY HEIGHT^STANDING LEN PT ^PATIENT QN BDYHGT.MOLEC 2
                UMLS: CUI C0005890
            -->
          </xs:element>
          <!-- begin hl7class type="ActRelationship-cls" -->
          <xs:element name="Měření_váhy_je_součásti_fyzikálního_vyšetření" minOccurs="0" maxOccurs="unbounded">
            <!-- begin hl7class type="Observation-cls" -->
            <xs:element name="Měření_hmotnosti" minOccurs="1" maxOccurs="1"/>
            <!--
                LOINC: Weight Measured 3141-9 BODY WEIGHT MASS PT ^PATIENT QN MEASURED BDYWGT.ATOM 2
            -->
          </xs:element>
          <!-- begin hl7class type="ActRelationship-cls" -->
          <xs:element name="Měření_tlaku_je_součásti_fyzikálního_vyšetření" minOccurs="0" maxOccurs="unbounded">
            <!-- begin hl7class type="Act-cls" -->
            <xs:element name="Měření_tlaku" minOccurs="1" maxOccurs="1">
              <!-- begin hl7class type="ActRelationship-cls" -->
              <xs:element name="Je_diastolickou_součástí" minOccurs="0" maxOccurs="unbounded">
                <!-- begin hl7class type="Observation-cls" -->
                <xs:element name="Měření_Diastolického_Tlaku" minOccurs="1" maxOccurs="1"/>
                <!--
                    LOINC BP Diastolic 8462-4 INTRAVASCULAR DIASTOLIC PRES PT ARTERIAL SYSTEM QN BP.ATOM 2
                -->
              </xs:element>
              <!-- begin hl7class type="ActRelationship-cls" -->
              <xs:element name="Je_systolickou_součástí" minOccurs="0" maxOccurs="unbounded">
                <!-- begin hl7class type="Observation-cls" -->
                <xs:element name="Měření_Systolického_Tlaku" minOccurs="1" maxOccurs="1"/>
                <!--
                    LOINC: 8508-4 INTRAVASCULAR SYSTOLIC PRES PT BRACHIAL ARTERY QN BP.MOLEC 2
                -->
              </xs:element>
            </xs:element>
          </xs:element>
          <!-- begin hl7class type="ActRelationship-cls" -->
          <xs:element name="Měření_teploty_je_součástí_fyzikálního_vyšetření" minOccurs="0" maxOccurs="unbounded">
            <!-- begin hl7class type="Observation-cls" -->
            <xs:element name="Měření_Teploty" minOccurs="1" maxOccurs="1"/>
            <!--
                LOINC: 8328-7 BODY TEMPERATURE TEMP PT AXILLARY QN BDYTMP.MOLEC 2
            -->
          </xs:element>
          <!-- begin hl7class type="ActRelationship-cls" -->
          <xs:element name="Měření_pulsu_je_součástí_fyzikálního_vyšetření" minOccurs="0" maxOccurs="unbounded">
            <!-- begin hl7class type="Observation-cls" -->
            <xs:element name="Měření_Pulsu" minOccurs="1" maxOccurs="1"/>
            <!--
                LOINC: 8893-0 HEART BEAT NRAT PT PERIPHERAL ARTERY QN PALPATION HRTRATE.MOLEC 2
            -->
          </xs:element>
          <!-- begin hl7class type="ActRelationship-cls" -->
          <xs:element name="Měření_dechové_frekvence_je_součástí_fyzikálního_vyšetření" minOccurs="0" maxOccurs="unbounded">
            <!-- begin hl7class type="Observation-cls" -->
            <xs:element name="Měření_Dechové_Frekvence" minOccurs="1" maxOccurs="1"/>
            <!--
                LOINC: Respiratory Rate 9279-1 BREATHS NRAT PT RESPIRATORY SYSTEM QN RESP.ATOM 2
            -->
          </xs:element>
          <!-- begin hl7class type="ActRelationship-cls" -->
          <xs:element name="Měření_obvodu_boků_je_součásti_fyzikálního_vyšetření" minOccurs="0" maxOccurs="unbounded">
            <!-- begin hl7class type="Observation-cls" -->
            <xs:element name="Měření_Obvodu_Boků" minOccurs="1" maxOccurs="1"/>
            <!--
                UMLS: CUI: C0562350
            -->
          </xs:element>
          <!-- begin hl7class type="ActRelationship-cls" -->
          <xs:element name="Měření_obvodu_pasu_je_součásti_fyzikálního_vyšetření" minOccurs="0" maxOccurs="unbounded">
            <!-- begin hl7class type="Observation-cls" -->
            <xs:element name="Měření_Obvodu_Pasu" minOccurs="1" maxOccurs="1"/>
            <!--
                UMLS: CUI: C0455829
            -->
          </xs:element>
          <!-- begin hl7class type="Participation-cls" -->
          <xs:element name="Pacient_se_účastní_fyzikálního_vyšetření" minOccurs="1" maxOccurs="1"/>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Figure 1: LIM template for Physical Examination - XML Schema simplified for clarity.

2.6. OpenEHR templates

In the set of documentation released by openEHR explanation of the term archetype and template can be found as follows [11]:

Archetypes are a key element of the openEHR methodology. They are reusable, structured models of clinical information concepts that appear in EHRs, such as 'test result' , 'physical examination' and 'medication order', and are expressed in terms of constraints on the reference model. All data in openEHR EHRs are instances of reference model entities, configured by archetypes. Archetypes also act as mediators between data and terminology. They are language- and terminology-neutral.

Templates are (usually) locally defined models of screen forms, and ring together a selection of archetypes, terminologies, language and other details relevant to the particular local use of archetypes. For example, concepts such as 'referral' and 'prescription' are modeled as templates, which in turn use archetypes for more fine-grained concepts.

Archetypes are encoded in the Archetype Definition Language (ADL) and openEHR Templates (OETs) in the Template Definition Language (TDL). TDL forms a super-set of ADL. Each OET must have a root archetype that contains other relevant archetypes connected through so called slots. Templates may add further local constraints to the archetypes it mentions, including removing or mandating optional sections, and they may define default values.

As mentioned in [10] openEHR Templates can be used as a basis for a user interface definition or after filling with data they can be used as some sort of messages or documents suitable for data transfers. The second usage will be studied in this paper.

3. Methods

In the next section an experiment will be described, which aimed to compare the two approaches - one based on HL7 v3 messaging (described in more detail in [12]) and the other on openEHR archetypes and templates.

3.1. How to create an Archetype?

A basic procedure of Archetype creation can be found in "Help pages"supplied with the Ocean Archetype Editor. Some of advices are cited in the following two paragraphs.

The first aspect to consider is that an openEHR health record consists of just a few 'classes' which contain information about the patient or data subject. The 'container classes' are:

EHR - this is the top level class and contains all information about the data subject.

Extract - this class contains all information that is to be transferred to another EHR.

Folder - this class allows information within an EHR to be organized.

Composition (or document) - this is the class that contains information committed to the EHR by a clinician.

Section - this class allows information within a composition to be organized.

Entry - this class contains meaningful information that is to be processed by the machine and read by the clinician.

These classes contain no clinical or demographic concepts at all - and it is this feature which differentiates the openEHR approach. The classes do have attributes which ensure that it is clear how the information in the EHR was collected, by whom, and who took responsibility for it - as well as meeting many other complex requirements. The clinical or demographic requirements are met through designing archetypes for the purpose.

The usage of each one of above-mentioned classes is explained in more detail in the "Help pages", which we leave out from spatial reasons. For the purpose of this text we put only an explanation of usage of the Section class: "In summary, if you are attempting to standardize the organization of information within a document, progress note or any other Composition, then you probably need to create an SECTION archetype."

3.2. Experiment: Comparison of openEHR Template approach with HL7 v3 messaging

In the following text we describe the experiment conducted in order to compare the communication schema based on HL7 v3 messages with the data exchange using openEHR templates and archetypes. The main motivation of accomplishing such a comparison was the difficulties encountered during implementation of the communication among EHR-Ss using HL7 v3 messages. The main shortcoming was the usage of HL7 balloted storyboards. Storyboards describe the dynamic aspect of the communication and define the factual form and thus the content of messages is exchanged. And here we

come to the main problem. HL7 storyboards are ”short-message”oriented, which was not exactly matching our needs. Nature of our communication was rather document-oriented, which caused complicated transfer of LIM messages (e.g. physical examination) via several HL7 v3 messages originating in several storyboards.

Usage of openEHR templates on the other hand makes our data exchange straightforward. Another possible solution was incorporating HL7 CDA documents, which might be a future work for our team. The openEHR approach was right choice for us because the openEHR approach is infiltrating the CEN EN 13606 standard, which is gaining its popularity and is in center of interest in various research projects [13]. Therefore, we also tested the applicability of the openEHR approach while we already have had implemented data exchange via HL7 v3 messages. The data exchange via openEHR templates forms an alternative solution.

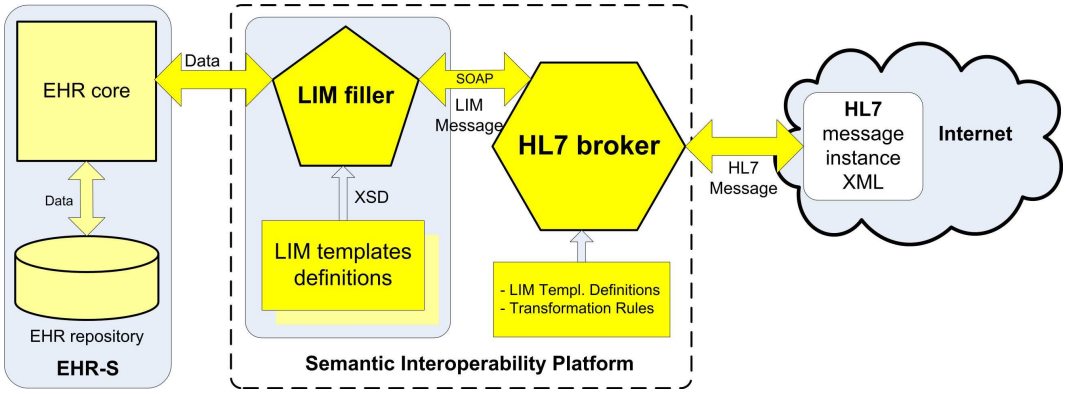


Figure 2: Former communication schema based on HL7 v3 messaging.

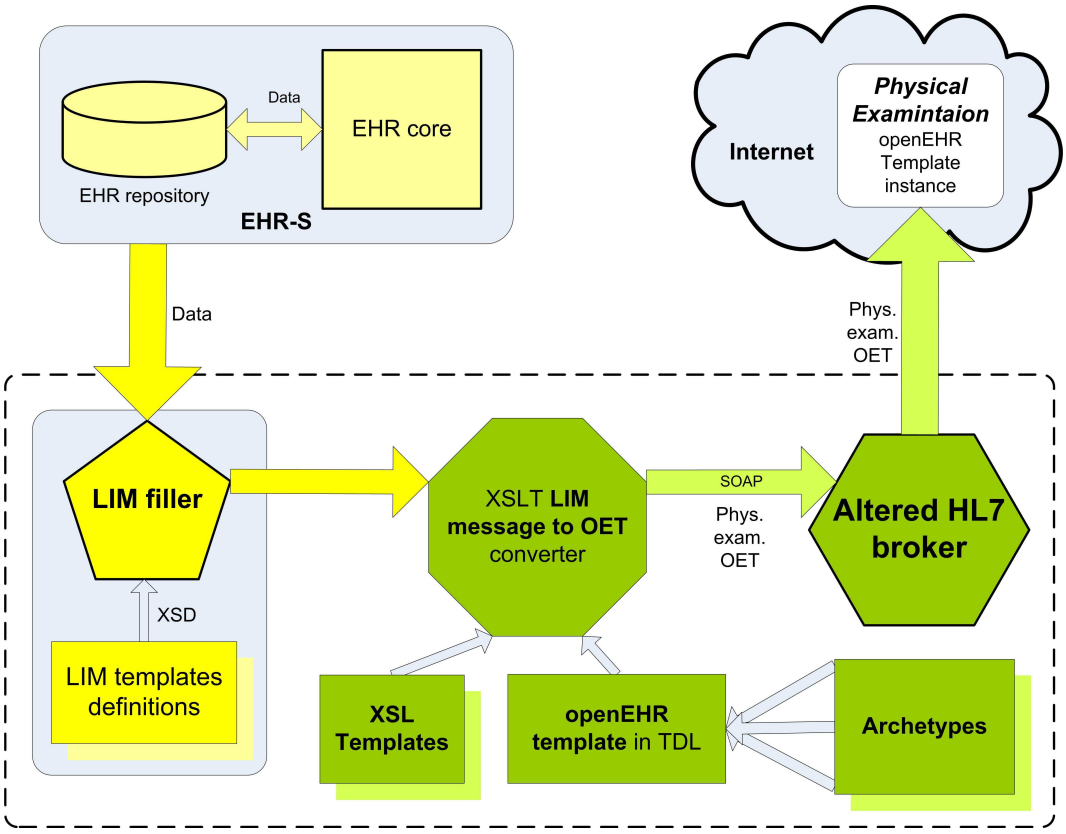


Figure 3: New communication schema based on openEHR templates and archetypes.

In the former solution (see Figure 2) we used all ”possibilities” that the HL7 v3 standards offers (RIM, storyboards, balloted messages). Since the HL7 v3 storyboards are ”action-oriented” it was quite difficult to transport most of defined LIM templates. The nature of data we were dealing with was much more suitable for HL7 CDA [14]. Our task was more or less as the CDA's - transferring ”documents”, rather than short messages.

```
definition
  SECTION[at0000] matches {    -- Physical Examination
    items cardinality matches {1; ordered} matches {
      allow_archetype OBSERVATION[at0003] occurrences matches {0..1} matches {
        -- Height
        include archetype_id/value matches
            {/openEHR-EHR-OBSERVATION\.height(-[a-zA-Z0-9_]+)*\.v1/}
      }
      allow_archetype OBSERVATION[at0002] occurrences matches {0..1} matches {
        -- Body Weight
        include archetype_id/value matches
            {/openEHR-EHR-OBSERVATION\.body_weight(-[a-zA-Z0-9_]+)*\.v1/}
      }
      allow_archetype OBSERVATION[at0004] occurrences matches {0..1} matches {
        -- Blood pressure
        include archetype_id/value matches
            {/openEHR-EHR-OBSERVATION\.blood_pressure(-[a-zA-Z0-9_]+)*\.v1/}
      }
      allow_archetype OBSERVATION[at0005] occurrences matches {0..1} matches {
        -- Body Temperature
        include archetype_id/value matches
            {/openEHR-EHR-OBSERVATION\.body_temperature(-[a-zA-Z0-9_]+)*\.v1/}
      }
      allow_archetype OBSERVATION[at0006] occurrences matches {0..1} matches {
        -- Pulse Rate
        include archetype_id/value matches
            {/openEHR-EHR-OBSERVATION\.heart_rate-pulse(-[a-zA-Z0-9_]+)*\.v1/}
      }
      allow_archetype OBSERVATION[at0007] occurrences matches {0..1} matches {
        -- Respiration
        include archetype_id/value matches
            {/openEHR-EHR-OBSERVATION\.respiration(-[a-zA-Z0-9_]+)*\.v1/}
      }
      allow_archetype OBSERVATION[at0001] occurrences matches {0..1} matches {
        -- Waist Hip
        include archetype_id/value matches
            {/openEHR-EHR-OBSERVATION\.waist_hip(-[a-zA-Z0-9_]+)*\.v1/}
      }
      allow_archetype OBSERVATION[at0008] occurrences matches {0..*} matches {
        -- Other observation
        include archetype_id/value matches {/.*/}
      }
      SECTION[at0009] occurrences matches {0..1} matches {*}
    }
  }

ontology
  terminologies_available = <"SNOMED-CT", ...>
  term_definitions = <
    ["cs"] = <
      items = <
        ["at0000"] = <
          text = <"Fyzikální vyšetření">
          description = <"unknown">
        >
...
      >
    >
  >


  term_bindings = <
    ["SNOMED-CT"] = <
      items = <
        ["at0001"] = <[SNOMED-CT::276361009-284472007]>
        ["at0002"] = <[SNOMED-CT::248345008]>
        ["at0003"] = <[SNOMED-CT::50373000]>
        ["at0004"] = <[SNOMED-CT::250765005]>
        ["at0005"] = <[SNOMED-CT::276535009]>
        ["at0006"] = <[SNOMED-CT::78564009]>
        ["at0007"] = <[SNOMED-CT::86290005]>
      >
    >
  >
```

Figure 4: Definition and Ontology sections of `openEHR-EHR-SECTION.physical_examination.v1` archetype, which is a root archetype in the Physical Examination OET.

4. Results

Which form of the communication was better? HL7 has a much better documentation, thus a straightforward development process. OpenEHR templates and the TDL are not documented well, yet, but they are much more simple than the HL7 v3 modeling methodology and suitable for smaller developer teams.

4.1. Physical Examination OET

In the Figure 4 we can see an example of the openEHR Template (OET) containing archetype `openEHR-EHR-SECTION.physical_examination.v1` which we created using the Ocean Archetype Editor tool. This archetype was developed to be a root archetype of composed OET. The archetype is derived from the SECTION class from the openEHR Reference Model. According to the Archetype Building Guide: A SECTION is an organizing class, contained within a COMPOSITION. Archetypes of Sections standardize the organization of information within a Composition. Examples of Sections are: Physical examination - organized by System, History - organized by presenting complaint, social history, review of systems etc.

The slots of the `openEHR-EHR-SECTION.physical_examination.v1` archetype contain following archetypes found in the CKM repository:

- openEHR-EHR-OBSERVATION.height.v1
- openEHR-EHR-OBSERVATION.body_weight.v1
- openEHR-EHR-OBSERVATION.blood_pressure.v1
- openEHR-EHR-OBSERVATION.body_temperature.v1
- openEHR-EHR-OBSERVATION.heart_rate-pulse.v1
- openEHR-EHR-OBSERVATION.respiration.v1
- openEHR-EHR-OBSERVATION.waist_hip.v1

Finally, concepts covered by archetypes slots were encoded in the ontology section by codes from SNOMED-CT (the lower part of the Figure 4).

4.2. Components of the Experiment

The data flow in the experiment depicted in Figure 3 is as follows. The former LIM Filler module attached to ADAMEKj EHR creates the LIM message containing data about a physical examination. The LIM message is then transformed by XSL templates into the OET instance conforming to the `openEHR-EHR-SECTION.physical_examination.v1` archetype, described in the previous section. Both the LIM message and the OET instance are encoded in XML. The OET instance is sent via SOAP to the former HL7

Broker, which was reconfigured for the purpose of this experiment. The HL7 broker then sends the OET instance to the recieving system. In this experiment the HL7 broker does not transform the OET instance in any way. It serves as an inbox/outbox for the communicating EHR-S.

4.3. Archetype modeling methodology

To propose a modeling guideline or some sort of methodology for archetypes' creation, the content of EHR they describe must be clarified. According to "EHR information model" these are commonly used types of clinical information: *Basic information* (e.g. date of birth, sex, height, weight, pregnancy), *Problem list*, *Medications list*, *Therapeutic precautions* (allergies and alerts), *Patient preferences*, *Patient consents*, *Family history*, *Social history/situation*, *Lifestyle*, *Vaccination record* and *Care plan*.

This is a high level structure of EHR which can be found in a more or less complete form even in contemporary systems. The ADAMEKj also corresponds with this structure. This fact suggests an idea that if the modeling process of EHR adheres commonly agreed guidelines or methodologies, which are independent on the implementation aspects (i.e. used approach - object oriented programming, archetypes, relational database etc.), much better interoperable EHRs can be implemented. This implies creation of ontologies of various domains in medicine.

Within the frame of research in the EuroMISE Center a so called knowledge base (KB) tree of the MUDR EHR [15] was created. This tree was based on concepts of the MDMC. Another KB tree was created for the domain of dentistry. MUDR KB trees can be considered as some kind of ontologies, because they comprise vertices interconnected with various kinds of edges. The tree structure is formed only by the edge of the *superior* type. From the modeling point of view MUDR KB trees do not depend on any kind of implementation technology. After creation of the KB tree in the "MUDR Knowledge Base Editor" tool, it can be imported to MUDR or it can serve as a basis for further modeling process. Hence, the process of enabling the creation of EHRs with harmonized clinical content formulated in [10] can be extended by the *ontology creation* step at the beginning of the process.

Although ontology creation process can end up with various ontologies describing more or less the same domain, they can be aligned and merged [16] thank to the research conducted in the field of knowledge modeling.

5. Discussion

The most important thing, in order to achieve information consistency among various openEHR archetype based EHRs, is the archetype creation methodology. Such a methodology should guarantee consistent archetype creation. This means that two authors describing the same concept via archetype should end up with approximately the same structure of the definition part of the archetype. However, such an aim is really hard to accomplish.

After we mapped all concepts from the MDMC to SNOMED-CT we fixed their meaning. This helps in the further work, however the CKM repository does not support searching archetypes using SNOMED-CT codes. Only archetype identifiers are supported.

In the experiment we use the clinical data in form of LIM messages only from technical reasons. The more straightforward solution would comprise an archetype-specific "filler"module.

The usage of openEHR approach was a reaction to the situation in the e-Health community. The considerable amount of experts and professionals deal with archetypes, therefore we realized the experiment based on openEHR approach. For example NEHTA recommends to use archetypes in Australian e-Health [17].

The advantage of HL7 v3 being an ISO standard is gradually being eliminated by the activities of the openEHR Foundation. The archetypes were added to the CEN EN 13606 norm as Part 2. The implementation of communication via OET was considerably simpler than searching for correct storyboard in HL7 v3 ballot.

Despite the clinical content of the ADAMEKj EHR (concepts of MDMC) was modeled using HL7 v3 RIM classes, we used the openEHR approach as the first choice mainly thank to the stricter policy on data structures compared to HL7 CDA. The openEHR template contains only structured data conforming to given archetypes, the CDA document consists mainly from the free-text part which is annotated by structured pieces of information.

The WHO's ICD-10 coding list is commonly used in the real world of healthcare. The advantage it brings is obvious: even when the well structured EHR is not common, the information about patient's diagnoses are encoded in ICD-10 codes and they are processable by any other system or professional. However, the ICD-10 covers only diagnoses, i.e. true/false statements (boolean data type) if patient suffers from given disease. This can be considered the *Qualitative point of view*.

The modeling of the *quantitative* attributes is more complicated. It comprises of the description of the attribute, stating its data type (real number, integer, image data, etc.), eventually its units definition. Such a complex ”meta-information”about a given attribute cannot be covered by a coding list. Much more sophisticated tools have to be incorporated e.g. ontologies.

6. Conclusion

Without ontologies there is no progress in achieving semantic interoperability. Communication standards can only *support* (!) the semantic interoperability among heterogenous EHR-S, *not guarantee* it.

The first step in creating ontologies might be the usage of the MUDR Knowledge Base editor tool, which was successfully applied in the creation of Dental KB [18].

Standardized clinical terminology would bring benefits to physicians, patients, administrators, software developers and payers. It would help also health care providers because it would give them more reliable and complete pieces of information, which belong to the healthcare process and would lead to better, less error prone and safer care of patients.

Since the evaluation of the comparison experiment described in sections above is not fully complete at the time of writing this text, we have given a brief outline of the evaluation only. We were interested in the standardization degree (data model, design and development process), the amount of implementation work, possibilities of interconnection with classification systems and tooling support (e.g. openEHR tools: Ocean Architect Editor, EHRflex [19]).

References

[1] P. Schloeffel, T. Beale, G. Hayworth, S. Heard, and H. Leslie, “The Relationship between CEN 13606, HL7, and OpenEHR”, In: J. Westbrook, J. Callen, G. Margelis, J. Warren (Eds). HIC 2006 and HINZ 2006: Proceedings. Brunswick East, Vic.: Health Informatics Society of Australia, pp 24-28, 2006.

[2] Technical report, ISO/TR 20514 – Health informatics – Electronic health record – Definition, scope, and context. ISO. 2005.

[3] T. Beale and S. Heard, “Architecture Overview”, The openEHR Foundation. 2008. (Available at: http://www.openehr.org/releases/1.0.2/architecture/overview.pdf).

[4] T. Beale, “Archetypes: Constraint-based Domain Models for Future-proof Information Systems” In: Eleventh OOPSLA Workshop on Behavioral Semantics: Serving the Customer. K. Baclawski, H. Kilov (Eds). Northeastern University, Boston, 2002, pp. 16-32.

[5] T. Gruber, “Ontology”, Encyclopedia of Database Systems, Ling Liu and M. Tamer Özsu (Eds.), Springer-Verlag, 2009. (Available at: http://tomgruber.org/writing/ontology-definition-2007.htm).

[6] M. Tomeckova, et al., “Minimal data model of cardiological patient”. (in Czech). Cor et Vasa 2002; 4: 123.

[7] T. Beale and S. Heard, “Archetype Definition Language - ADL 1.4” The openEHR Foundation. 2008. (Available at: http://www.openehr.org/releases/1.0.2/architecture/am/adl.pdf).

[8] M. Nagy, P. Hanzlicek, P. Preckova, P.Kolesa, J. Misur, M. Dioszegi, and J. Zvarova, “Building Semantically Interoperable EHR Systems Using International Nomenclatures and Enterprise Programming Techniques”, CeHR 2007: Conference Proceedings. Amsterdam: IOS press; 2008. p. 105-10.

[9] R. Mares, M. Tomeckova, J. Peleska, P. Hanzlicek, and J. Zvarova, “User interface for patients’ database systems - an example of application for data collection using minimal data model of cardiology” (in Czech), Cor et Vasa 2002; 4: 76.

[10] M. Nagy, “Clinical contents harmonization of EHRs and its relation tosemantic interoperability”, PhD Conference ’09. Matfyzpress 2009; pp. 65-74.

[11] The openEHR Foundation, “Introducing openEHR”, The openEHR Foundation 2007. (Available at: http://www.openehr.org/releases/1.0.2/openEHR/introducing_openEHR.pdf).

[12] M. Nagy, P. Hanzlicek, P. Preckova, A. Riha, M. Dioszegi, L. Seidl, and J. Zvarova, “Semantic Interoperability in Czech Healthcare Environment Supported by HL7 version 3”, Methods Inf Med, 2010; 49(1):186-195.

[13] D. Moner, J. Maldonado, D. Bosca, C. Angulo, M. Robles, D. Perez, and P. Serrano, “CEN EN 13606 Normalisation Framework Implementation Experiences”, In: B.Blobel, E. Hvannberg, V. Gunnarsdottir (Eds). Seamless Care - Safe Care, IOS Press 2010. pp: 136-142.

[14] R.H. Dolin, et al., “The HL7 Clinical Document Architecture”, J Am Med Inform Assoc. 2001;8 (6):552-69.

[15] P. Hanzlicek, J. Spidlen, and M. Nagy, Universal Electronic Health Record MUDR. In: Duplaga, M., Zielinski, K., Ingram, D. (eds.) Transformation of Healthcare with Information Technologies. pp. 190–201. IOS Press, Amsterdam (2004).

[16] A. Burger, D. Davidson, and R. Baldock, "Anatomy Ontologies for Bioinformatics: Principles and Practice", Springer London 2008.

[17] National E-Health Transition Authority, "Review of Shared Electronic Health Record Standards", Version 1.0 - 20/02/2006. (Available at: http://www.nehta.gov.au/component/docman/doc_download/68-review-of-shared-electronic-health-records-standards-v10).

[18] M. Nagy, P. Hanzlicek, J. Zvarova, T. Dostalova, M. Seydlova, R. Hippmann, L. Smidl, J. Trmal, and J. Psutka, "Voice-controlled Data Entry in Dental Electronic Health Record", Studies in Health Technology and Informatics, 2008. 136: pp. 529-534.

[19] A. Brass, D. Moner, C. Hildebrand, and M. Robles, "Standardized and Flexible Health Data Management with Archetype Driven EHR System (EHRflex)", In: B.Blobel, E. Hvannberg, V. Gunnarsdottir (Eds). Seamless Care - Safe Care, IOS Press 2010. pp: 212-218.

Portál poznámek aneb polidštění sémantického webu

doktorand:
ING. MARTIN ŘIMNÁČ
Ústav informatiky AV ČR, v. v. i.
Pod Vodárenskou věží 2
182 07 Praha 8
rimnacm@cs.cas.cz

školitel:
ING. JÚLIUS ŠTULLER, CSC.
Ústav informatiky AV ČR, v. v. i.
Pod Vodárenskou věží 2
182 07 Praha 8
stuller@cs.cas.cz

obor studia:
Databázové systémy

Práce byla částečně podpořena projektem 1M0554 Ministerstva školství, mládeže a tělovýchovy ČR "Pokročilé sanační technologie a procesy" a záměrem AV0Z10300504 "Computer Science for the Information Society: Models, Algorithms, Applications".

Abstrakt

Příspěvek[1] se s ohledem na aktuální dění v oblasti webových technologií věnuje možnosti implementace portálu, kde by popis objektů byl realizován formou krátkých, sémanticky ohraničených poznámek. Taková forma zajišťuje čitelnost jak pro koncového uživatele (člověka), tak pro softwarové agenty pracující s dokumenty sémantického webu, neboť tyto poznámky mohou být převedeny na RDF data. Další zásadní výhodou použití sémanticky ohraničených poznámek je jejich přímočarý překlad do jiných jazyků, kterého lze využít při tvorbě mulitjazyčných webových prezentací.

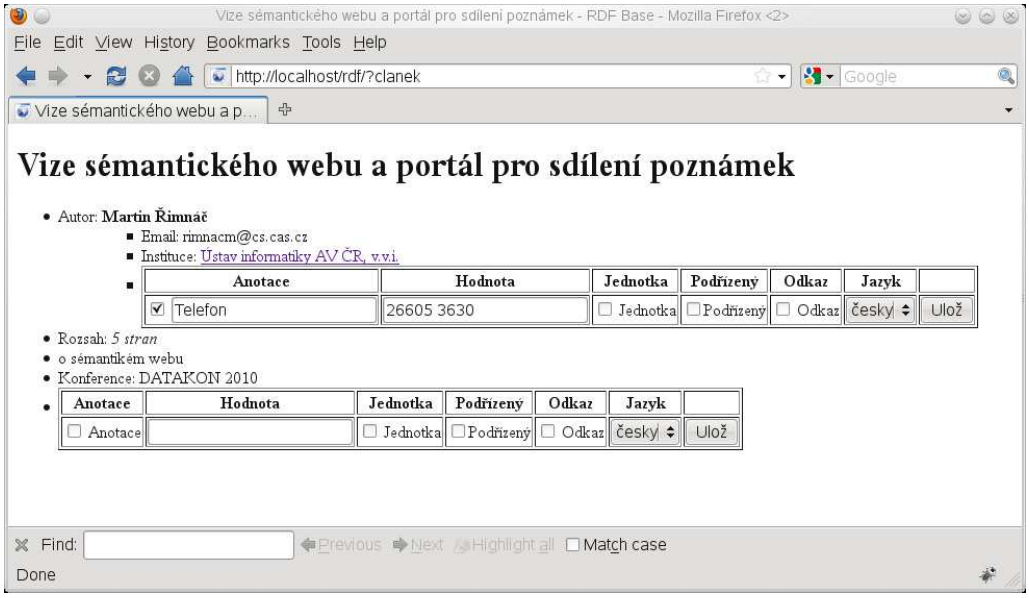
Jedním z aktuálně řešených problémů v oblasti sémantického webu je hledání způsobu, jak v této oblasti zajistit dostatečný objem reálných dat, po kterých by se poptávali také běžní uživatelé webových technologií. Taková data je možné získat buď semiautomatickou extrakcí z klasických webových stránek do formátu sémantického webu anebo manuálně tvorbou ontologií a následnou anotací existujících dat. Zatímco tvorba ontologií je doménou ontologických inženýrů (čemuž odpovídají i podpůrné aplikace), u mnohých jiných formátů (blogy, sociální sítě, sdílení dat) se o tvorbu starají přímo uživatelé bez podrobnějšího obeznámení s použitými technologiemi. Příspěvek se proto snaží poukázat na možnost zadávání RDF dat ve formě sémanticky ohraničených poznámek, kdy daný subjekt (resource) je popsán pomocí (potenciálně anotovaných) hodnot, respektive klíčových slov. K podobné formě dnes přistupují i mnohé sociální sítě; jejich omezení délky sdělení vede uživatele k návyku, aby namísto strojově složitě zpracovatelných vět v přirozeném jazyce použili pouze několik klíčových slov ve stejném významu.

Navrhovaná forma může interaktivním způsobem nabádat uživatele, aby rozšířil popis daného objektu o vlastní postřehy (stejný model tvorby obsahu je znám například z wikipedie). Každý uživatel může buď označit poznámku, kterou navrhuje smazat (se kterou nesouhlasí), může poznámku potvrdit nebo opravit, případně může vložit poznámku novou. Poznámka může být anotována názvem vlastnosti, na rozdíl od klasických ontologií je však většina vlastností definována pouze lokálně. Ukázka takového zadávání RDF dat je na obrázku 1.

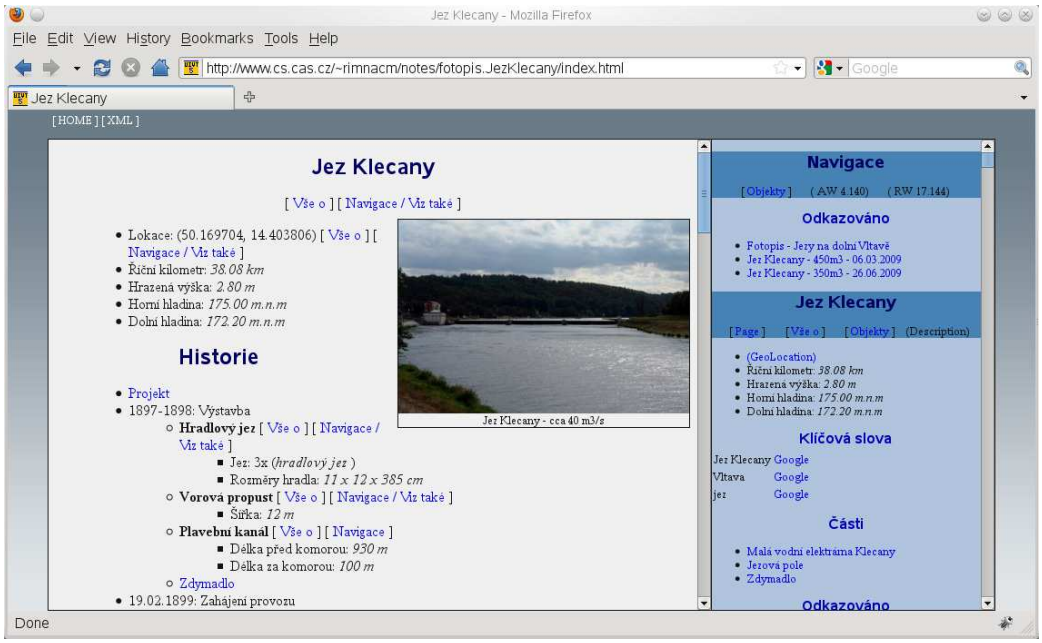
Takto koncipovaná RDF data mohou být transformována XSLT procesorem do XHTML kódu, ideálně s RDFa anotacemi. Taková transformace zajistí indexaci dokumentů publikovaných na portálu dnešními klasickými fulltextovými nástroji - tak ani člověk ani fulltextový nástroj "nepozná", že se vůbec jedná o dokument sémantického webu. Portál tímto způsobem umožňuje spolupráci mezi fulltextovými vyhledávači a nástroji pro zpracování dokumentů sémantického webu.

S ohledem na transformaci je vhodné zavést i specializované, například multimediální prvky, které obohatí strohý seznam poznámek o grafiku. Tím se portál dále přibližuje současným uživatelským požadavkům na webovou prezentaci. Ukázka takového výstupu včetně pokročilých navigačních nástrojů je ilustrována na obrázku 2.

[1] Příspěvek je rozšířeným abstraktem příspěvku přijatému na mezinárodní konferenci DATAKON 2010.



Obrázek 1: Studie interaktivního vkládaní strukturovaných poznámek



Obrázek 2: Ukázka prezentující RDF data ve formě poznámek

Vzhledem k rozličné kvalitě vkládaných poznámek je nanejvýše vhodné každou poznánku ohodnotit fuzzy mírou udávající její důvěryhodnost. Ta zajistí, že při vyhledávání budou nalezeny všechny varianty k dané poznámce, avšak každá z nich bude ohodnocena pouze určitým stupněm (varianty mohou být seřazeny podle této míry). Analogicky lze postupovat i u hodnocení uživatelů a vhodně tak iniciovat váhy uživatelem nově vložených poznámek, popřípadě tyto váhy využít stejným způsobem jako v sociálních sítích.

Z pohledu ostatních zdrojů RDF dat může být portál chápán jako jedna součást distribuovaného prostředí vzájemně spolupracujících zdrojů. Takové prostředí umožňuje značně omezit redundanci i duplicitu prezentovaných dat, kdy namísto kopie dat je použito pouze odkazu na (externí) zdroj. Data v takovém prostředí bývají označována jako tzv. linked data. Portál tuto aktuálně se rozvíjející technologii navíc rozšiřuje o aspekt důvěryhodnosti těchto dat.

Návrh a parametry portálu jsou předmětem několika posledních prací a jsou součástí závěrečné části di-sertační práce autora.

Spiking Neural Networks, Perspective

Post-Graduate Student:
ING. DANIEL SURGENT
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
surgent@cs.cas.cz

Supervisor:
ING. MARCEL JIŘINA , DRSC.
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
marcel@cs.cas.cz

Field of Study:
Biocybernetics

Abstract

Spiking neural networks are an offshoot of neural computation reasearch. In the recent years, however, we have witnessed a growing interest and shift of the emphasis in the artificial neural network community toward pulse-coupled neural networks with spike-timing as they can encode temporal information in their signals. This paper gives a brief overview of different spiking neuron models and we discuss their ability to operate in large complex networks as well as their evolvability.

1. Introduction

Artificial intelligence is a branch of computer science for which the biological inspiration seems to be crucial. Neural networks, as used in artificial intelligence, have traditionally been viewed as simplified models of neural processing in the brain, even though the first simple models of ANNs, known as the first generation neural networks [22], are considered more as mathematical or computational model for information processing based on connectionistics approach to computation and the relation between these models and the brain architecture is not in place.

From some point of view we can see the following generations of artificial neural networks as continuous acquiring of novel biological inspirations. For instance the third generation of neural networks raised the level of biological realism by employing individual pulses which allow spatial-temporal information in communication and computation, like real neurons do [5]. In this paper, we consider this third generation, i.e., we compare and contrast various models of spiking neurons with special attention focused on their ability to efficiently operate in complex networks and on "how easily" they can change their behavior through genetic mutation - their evolvability. But first of all, we briefly explain what is common for all spiking neuron models and what differs them from their non-spiking predecessors.

2. On artificial spiking neurons

Artificial spiking neurons model the relationship between the inputs and the output of a neuron in terms of single spikes (or pulses), and describe how such input leads to the generation of output spikes. Non-spiking neuron models do not employ individual pulses, but their output signals are computed in each iteration and typically lie between 0 and 1. So they do not implement the element of time in communicating. These signals can also be seen as normalized firing rates (frequencies) of the neuron within a certain period of time, and therefore non-spiking neural network is a special case of spiking neural network from some point of view.

The classical point of view that neurons transmit information exclusively via modulations of their mean firing rates [5, 21] seems to be at odds with growing empirical evidence that the patterns can be found in the firing sequences of single neuron [31] or in the relative timing of spikes of multiple neurons [13, 29] forming a functional neuronal group [4].

In spiking neuron models the transmission of a single spike from one neuron to another is mediated by synapses at the point where the two neurons interact. An input, or presynaptic spike arrives at the synapse, which in turn releases neurotransmitter which then influences the state, or a membrane potential of the target, or postsynaptic neuron. When the value of this state crosses some firing threshold (some models do not implement a fixed threshold, see next section), the target neuron generates a spike, and the state is reset by a refractory response. The size of the impact of presynaptic spike is determined by the type and efficacy (weight) of the synapse. In biology, there are known two distinct groups of neurons: excitatory neurons, which synapses release

neurotransmitter that increases the membrane potential of a target cell, and inhibitory neurons with synapses, that decrease this potential [23]. A good discussion of artificial spiking neurons can be found in [9]. Let's take a look on some of the most useful models of spiking neurons.

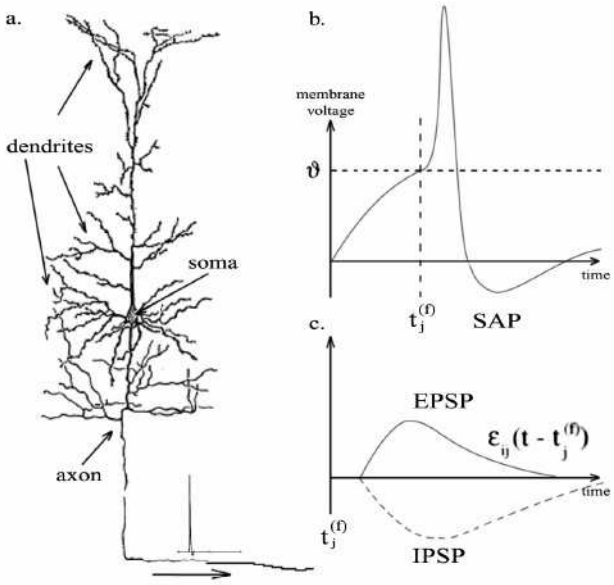


Figure 1: (a) Schematic drawing of a biological neuron. (b) Incoming postsynaptic potentials alter the membrane voltage so it crosses threshold value θ; the neuron spike goes into a refractory state. (c) Typical forms of excitatory and inhibitory postsynaptic potentials over time. [8]

3. Spiking models

There are two main approaches in creating neuron models: computational neuroscience approach, trying to understand and model biological neuron, and connectionism on the other side of the river, trying to solve artificial intelligence related problems by creating interconnected networks of simple units (i.e. artificial neurons), which can exhibit complex global behavior, determined by the connections between the processing units and unit parameters. In this paper we try to bridge this two concepts in the sake of finding a neuron model (from the pool of the most useful models of spiking neurons) best suited for the use in large complex networks for solving some interesting problems outside biological modeling.

Below we provide a brief review of some widely used spiking neuron models as a shortened version of review [16]. But instead of Izhikevich's concern in simulating cortical spiking neurons, we ask different question: which model to use as a processing unit in some large network architecture? Through this section, v denotes the membrane potential and v' denotes its derivative with respect to time. All the parameters in the models are chosen so that v has mV scale and the time has ms scale.

3.1. I&F

First group of spiking neuron models are known as integrate and fire neurons (I&F). The Leaky I&F neuron is one of the most widely used models in computational neuroscience

$$v' = I + a - bv, \text{if } v \geq v_{thresh}, \text{ then } \; v \leftarrow c,$$

where I is the input current, and a, b, c and v_{thresh} are the parameters. When the membrane potential v reaches the threshold value v_{thrash}, the neuron is said to fire a spike, and v is reset to c. From computational neuroscience point of view is the leaky I&F one of the worst models to use in cortical simulations, despite its simplicity.

The Leaky I&F has many extensions and modifications like I&F with adaptation, Integrate-and-Fire-or-Burst [34], Resonate-and-Fire and Quadratic I&F [20], but for purposes of this work we describe the Resonate-and-Fire neuron, which is two-dimensional (2-D) analogue of I&F neuron

$$z' = I + (b + i\omega)z$$
$$\text{if Im } (z) = a_{thresh}, \text{ then } \; z \leftarrow z_0(z)$$

where the real part of the complex variable z is the membrane potential. Here b, ω and a_{thresh} are parameters, and $z_0(z)$ is an arbitrary function describing activity-dependent after-spike reset.

By now, I&F family neurons are the only spiking neurons used outside the computational neuroscience community [1, 35] as they are easy to implement and their computational efficiency is far better than efficiency of Hodgkin-Huxley family spiking neuron models (see next section).

3.2. Hodgkin-Huxley

Second group of spiking neuron models are known as Hodgkin-Huxley-type (conductance-based) neurons as they are basicly variations and simplifications of the model developed by [12] based on data from the squid giant axon. It consists of four equations and tens of parameters, not provided here, describing membrane potential, activation of **Na** and **K** currents, and inactivation of **Na** current.

The Hodgin-Huxley model is one of the most important models in computational neuroscience not only because its parameters are biophysically meaningful and measurable, but also because they allow us to investigate questions related to synaptic integration, dendritic cable filtering, effects of dendritic morphology, the interplay between ionic currents, and other issues related to single cell dynamics. However, the end result can be at the small end tens of parameters which one must estimate or measure for an accurate model, and for complex systems of neurons not easily tractable by computer. So careful simplifications of the Hodkgin-Huxley model were therefore needed.

Sweeping simplifications to Hodgkin-Huxley model were introduced by FitzHugh-Nagumo model [6]. The parameters in this model

$$v' = a + bv + cv^2 + dv^3 - u$$
$$u' = \varepsilon(ev - u)$$

can be tuned so that model describes spiking dynamics of many resonator neurons. Although not clearly derivable from biology, the model allows for a simplified, immediately available dynamic, without being a trivial simplification [18].

From the other Hodgkin-Huxley family models we mention Morris-Lecar model [25] as a combination of Hodgkin-Huxley and FitzHugh-Nagumo into a voltage-gated **Ca** channel model with delayed-rectifier **K** channel, and the model of thalamic neuron - Hindmarsh-Rose [32], which is built upon the FitzHugh-Nagumo model and provide extra mathematical complexity that allows a great variety of dynamic behaviors for the membrane potential.

Although these models have much better dynamic properties than I&F family neuron models, and are better suited for computer simulations than Hodgkin-Huxley, they are yet still prohibitive in terms of large-scale simulations (see next section).

3.3. Izhikevich

A simple model of spiking neurons proposed recently by Izhikevich [15] combines the biologically plausibility of Hodgkin-Huxley-type dynamics and the computational efficiency of I&F neurons

$$v' = 0.04v^2 + 5v + 140 - u + I$$
$$u' = a(bv - u)$$

with auxiliary after-spike resetting if $v \geq +30$ mV, then $v \leftarrow c$, $u \leftarrow u + d$.

Here v represents the membrane potential of the neuron and represents a membrane recovery variable, which accounts for the activation of $\mathbf{K}^+$ ionic currents and inactivation of $\mathbf{Na}^+$ ionic currents, and it provides negative feedback to v. After the spike reaches its apex ($+30$ mV), the membrane voltage and the recovery variable are reset. Synaptic currents or injected dc-currents are delivered via the variable I. If v skips over 30, then it is first reset to 30, and then to c so that all spikes have equal magnitudes.

The part $0.04v^2 + 5v + 140$ is chosen so that v has mV scale and the time has ms scale. The resting potential in the model is between -70 and -60 mV depending on the value of b. As most real neurons, the model does not have a fixed threshold (contrary to I&F neurons); Depending on the history of the membrane potential prior to the spike, the threshold potential can be as low as -55 mV or as high as -40 mV.

By now, this model was used exclusively for large-scale simulations of cortical neurons within computational neuroscience research (see [17]). In our opinion, it seems to be suitable for solving artificial intelligence related tasks as a processing unit of a complex neural network and we try to put some arguments for this statement in the next sections.

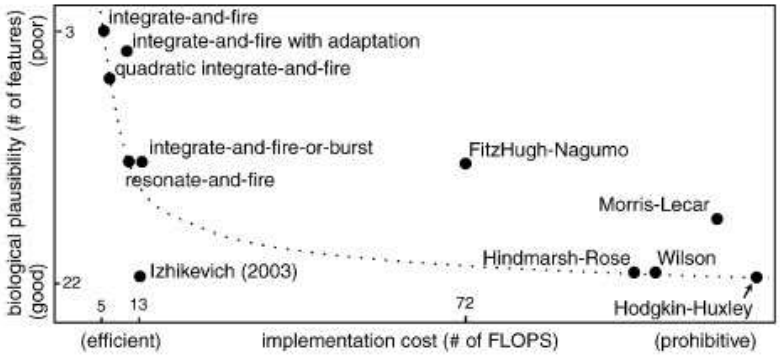
4. Computational efficiency

The notion of the efficiency of a simulation scheme is rather loosely defined in the computational neuroscience literature. [26] argue that efficiency should be defined as the simulation time required to achieve a prescribed accuracy goal. A scheme which constrains spike times to a timegrid is unsatisfactory in this respect if high accuracy is required, because the integration error drops only linearly with decreasing computation time step [11, 33].

In this section we again refer to the work by Izhikevich [16]. He compared some of the most useful models of spiking and bursting neurons from the biological plausibility and computational efficiency points of view. The summary of his comparison is in Fig. 2. To compare computational cost, he assumed that each model, written as dynamical system $x' = f(x)$, is implemented using a fixed-step first-order Euler method $x(t + \tau) = x(t) + \tau f(x(t))$ with the integration time step τ chosen to achieve reasonable numerical accuracy.

As we are interested in the idea of building a network connecting tens of thousands of neurons (maybe even more), we need to choose a neuron model that would be able to efficiently handle large numbers of neurons in

complex topologies. Considering this, it is prohibitive to use any of the Hodgkin-Huxley family neuron models (all models on the right half of the graph in Fig. 2), even the FitzHugh-Nagumo neuron model, which has "only" 72 floating-point operations is computationally too expensive. The reason why it is still not suitable, is that the efficiency of a single neuron is compared, and not the efficiency of the whole network. To estimate the computational efficancy of such network we must multiply the computational cost of its single element by the total number of its processing elements due to sequential computer processing (Fig. 3). Thus, we need to look somewhere else in a case of large-scale modeling.



Models	biophysically meaningful	tonic spiking	phasic spiking	tonic bursting	phasic bursting	mixed mode	spike frequency adaptation	class 1 excitable	class 2 excitable	spike latency	subthreshold oscillations	resonator	integrator	rebound spike	rebound burst	threshold variability	bistability	DAP	accommodation	inhibition-induced spiking	inhibition-induced bursting	chaos	# of FLOPS
integrate-and-fire	-	+	-	-	-	-	-	+	-	-	-	-	+	-	-	-	-	-	-	-	-	-	5
integrate-and-fire with adapt.	-	+	-	-	-	-	+	+	-	-	-	-	+	-	-	-	-	+	-	-	-	-	10
integrate-and-fire-or-burst	-	+	+		+	-	+	+	-	-	-	-	+	+	+	-	+	+	-	-	-		13
resonate-and-fire	-	+	+	-	-	-	-	+	+	-	+	+	+	+	-	-	+	+	+	-	-	+	10
quadratic integrate-and-fire	-	+	-	-	-	-	-	+	-	+	-	-	+	-	-	+	+	-	-	-	-	-	7
Izhikevich (2003)	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	13
FitzHugh-Nagumo	-	+	+	-		-	-	+	-	+	+	+	-	+	-	+	+	-	+	+	-	-	72
Hindmarsh-Rose	-	+	+	+			+	+	+	+	+	+	+	+	+	+	+	+	+	+		+	120
Morris-Lecar	+	+	+	-		-	-	+	+	+	+	+	+	+		+	+	-	+	+	-	-	600
Wilson	-	+	+	+			+	+	+	+	+	+	+	+	+	+		+	+				180
Hodgkin-Huxley	+	+	+	+			+	+	+	+	+	+	+	+	+	+	+	+	+	+		+	1200

Figure 2: Comparison of the neuro-computational properties of spiking and bursting models. "of FLOPS" is an approximate number of floating point operations (addition, multiplication, etc.) needed to simulate the model during a 1 ms time span. Each empty square indicates the property that the model should exhibit in principle (in theory) if the parameters are chosen appropriately, but the author failed to find the parameters within a reasonable period of time. [16]

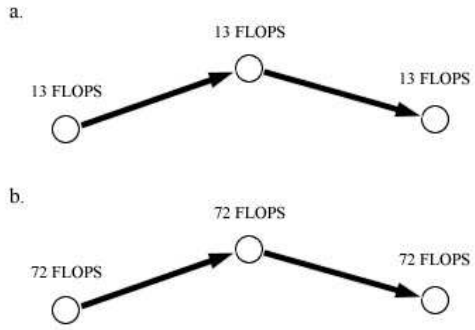


Figure 3: When the implementation is sequential (not parallel), the difference in efficiency of single neurons must be multiplied by the total number of the network's neurons to get the difference in efficiency for the networks. In this sample case the neurons efficiency difference is 59 FLOPS whereas the implementation cost difference for the networks is 177 FLOPS. (a) A sample network of Integrate-and-Fire-or-Burst neurons. (b) The same sample network with FitzHugh-Nagumo neurons.

If we want to simulate tens of thousands of spiking neurons in real time with 1 ms resolution, then there are plenty of models to choose from (all models on the left half of the graph in Fig. 2). The most efficient is the Leaky I&F neuron. It is the simplest model to implement when the integration time step τ is 1 ms. Indeed, the iteration $v(t+1) = v(t) + I + a - bv(t)$ takes only four floating-point operations (additions, multiplications, etc.) plus one comparison with the threshold v_{thrash}.

The other I & F-type models are practically as efficient as the Leaky I & F (see table in Fig. 2) and are usable for large-scale modeling. State-of-the-art solvers for networks of I&F neurons allow for routine simulations of networks of some 10^5 neurons and 10^9 connections (synapses) on moderate computer clusters [28].

5. Evolvability

First of all it should be noted that the language of this section moves between bilological and that of artificial intelligence systems, so potentially biological terms are used in less orthodox manner than that used in biological literature.

The human brain contains over 11 billion specialized nerve cells, or neurons, capable of receiving, processing, and relaying the electrochemical pulses on which all our sensations, actions, thoughts, and emotions depend [7]. But it is not the sheer number of neurons alone that is most striking about the brain, but how they are organized and interconnected.

Despite our goal is not to create an artificial brain, it is hardly imaginable to build a large-scale network with spatio-temporal structure of activity without a proper construction algorithm. In our opinion, the only way how to solve this complex problem is to use evolutionary computation techniques (e.g. genetic algorithms, artificial embryogeny). However, this approach is considerably challenging and it has many unresolved issues. Through this section we try to shed some light on the problem of system's ability to evolve, and we try to apply this information to compare different spiking neuron models.

Evolvability is a concept in evolutionary biology that tries to measure an organism's ability to evolve (see [2, 19, 27, 37]). We see this concept as an important part of the design process of a system, which we want to evolve. As estimating a system's evolvability is not an easy task, we divide the problem in two aspects, which we think are crucial, namely phenotype-genotype compression and evolutionary potential.

5.1. Phenotype-genotype compression

From the evolutionary neuroscience point of view, there is a vaxing problem with the notion that genome provides complete information for the construction of the nervous system of humans and other mammals. It is estimated that just human neocortex alone has about 10^{15} (one thousand million million) synapses [14]. Since the human genome has only about 3.5 billion (3.5 x 10^9) bits of information (nucleotide base pairs), with 30% to 70% of these appearing silent [3], some neural and molecular scientists have concluded that our genes simply do not have enough storage capacity to specify all of these connections, in addition to including information on the location and type of each neuron plus similar information for the rest of the body. Considering this, there must be some kind of phenotype-genotype compression for every biological unit with no exception for neurons.

Most probably is such representational efficiency made possible through gene reuse mechanism. Natural organisms implement gene reuse through a process of development, or embryogeny. The same genes can be used at different points in development for different purposes, and the order in which activations of genes take place determines when and where a particular gene is expressed [30].

In our opinion there is present more general phenomenon, that is, principle of reducing the number of genes by preserving phenotype functionality, either by gene reuse or by other mechanism. Let us give an exaplme for better understandig: suppose we want to compare

evolvability of two neural cells with exactly the same current behavior but different length of their genomes (the neuron with shorter genome has better phenotype-genotype compression). Let's say there is some novel function that both neurons are able to perform if their genome is changed accordingly (by some mutations). We say that the neuron with shorter genome has better chance to gain the novel function faster, simply because its genome needs fewer mutations to "search the space of its mutants". Therefore we can conclude that the neural cell with shorter genome has better ability to evolve and would be more successful than the neuron with longer genome, in case that the novel function increases organisms's chance for reproduction.

To put it in analogy with spiking neuron models, we define genome of given neuron model (i.e. some piece of genome of the network) as a set of dimensionless parameters of the model, and we define its phenotype as a full description of the model - its differential equations with given values of its parameters, i.e, the model's dynamical behavior.

If we want to compare evolvability of different spiking models considering only their phenotype-genotype compression, than the models with fewer parametes are better, with the Resonate-and-Fire neuron as the most evolvable (it has only three parameters plus an arbitrary function) and Hodgkin-Huxley with the worst phenotype-genotype compression (Fig. 4). If this two neuron models behave as integrators and we want them to respond as resonators (see [16]), we predict that genome (parameters) of the Resonate-and-Fire neuron needs much fewer mutations to change its phenotype (dynamical behavior) than genome of the Hodgkin-Huxley neuron.

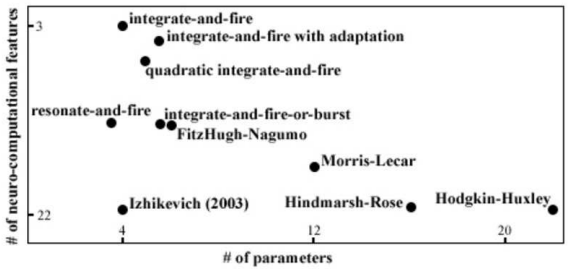


Figure 4: Comparison of the number of dimensionless parameters and the neuro-computational properties of spiking neuron models. Location of most of the dots is given approximately as the number of different parameters can vary for some of the models.

Note that we have intentionally choosed neuro-computational features that both neuron models are able to reproduce (see table in Fig. 2). What if chosen novel function would be the tonic bursting? It is clear that the Resonate-and-Fire neuron model is not able to evolve that way, thus we should not compare the neuron models only by phenotype-genotype compression, i.e, we should also compare their number of different neuro-computational properties. This conclusion leads to another important aspect of evolvability - evolutionary potential.

5.2. Evolutionary potential

The term evolutionary potential was defined from the genetics point of view as the array of successful mutants of selected gene for chosen novel function [10]. In case of neural cell we can define its evolutionary potential as the array of successful mutants of selected neuron for chosen novel behavior (i.e. response) of the neuron. Such definition does not exclude the existence of some novel behavior for which there are no mutants of given neural cell at all (its internal structure and genetic constitution simply does not allow to be changed that way). So the question is, how many different novel behaviors can a neural cell acquire through its evolution, and how many succesful mutants exist for each of these behaviors?

It seems really difficult to answer this question for biological neurons but not as difficult for artificial neurons, which we are able to investigate as dynamical systems. At least we are able to give more or less satisfying answer on the first part of the question above. If we take a look at the table in Fig. 2, we see that I&F family models can not reproduce some of the most important neuro-computational properties of real neurons, and therefore it seems that they are in general less evolvable than Hodgkin-Huxley-type neurons considering evolutionary potential. But more serious research on evolutionary potential of dynamical systems is needed and it would be interesting to investigate all of the dynamical behavior possibilities for spiking neuron models, and not "only" 20 (plus chaos) of the most prominent features of biological spiking neurons (table in Fig. 2).

6. Discussion

Neural structures are very well suited for complex information processing and it seems that the current research in computational neuroscience, evolutionary neuroscience, evolutionary computation, as well as in artificial neural networks has a promising future ahead.

Even though, almost all of the models used in computational neuroscience were created mainly to model physiologically realistic spike trains, some of these models appear also as an application in explicitly computational

contexts [36]. We see such phenomenon of overlapping of different approaches as a very important in the sake of building information systems like artificial neural networks.

In this paper some of the most useful models of spiking neurons have been proposed and compared from two perspectives - computational efficiency and evolvabilty. As our goal is to construct a large-scale complex network of spiking neurons using evolutionary computation techniques, we need a neuron model suitable for this kind of approach. From that point of view, computational efficiency and evolvability of given neuron model are its crucial properties. So which one to choose?

We argue that the neuron model proposed by Izhikevich [15] is the most suitable to fill our expectations, even though, the model was developed to understand the fine temporal structure of cortical spike trains, and to use spike-timing as an additional variable to understand how the mammalian neocortex processes information [16], i.e., it has not been used outside the biology yet. So what are the arguments for using the model as a processing element in an evolution driven complex network?

First: Izhikevich's model has the same implementation cost as the Integrate-and-Fire-or-Burst spiking neuron model (Fig. 2), and therefore it is as good as some of the I&F neurons from the computational efficiency point of view. Indeed, the model have been used to simulate a sparse network of 10.000 spiking cortical neurons with 1.000.000 synaptic connections in real time using a desktop PC and C++ programming language [15].
Second: as it has only four dimensionless parameters and it can reproduce all of the important neurocomputational features (see Fig. 4), we can conclude that the model has the best evolvabilty of all persented spiking neuron models considering both aspects - phenotype-genotype compression together with evolutionary potential.

These attributes of the model allows for the process of evolution to "experiment" with many types of model's behavior because there are plenty of them and the transition of neuron dynamics is manageable by only four parameters. In favor of this statement speaks the fact that author (in [16]) failed to find the parameters (within a reasonable period of time) for some of the neurocomputational properties for all of the Hodgkin-Huxley-type models, not surprisingly, the models with the worst phenotype-genotype compression.

In conclusion, having a "good-looking" efficient and evolvable spiking neuron model is only a beginning of the story of creating complex large-scale neural networks for information processing. There are still many problems to be solved but the main idea of this paper is that our most prominent tools for building complex systems should be the principles of emergence and self-organization.

References

[1] K.J. Cios, W. Swiercz, and W. Jackson, Networks of spiking neurons in modeling and problem solving. Neurocomputing, 61, 99-119, 2004.

[2] R. Dawkins, The evolution of evolvability. In C.G. Langton, editor, Artificial life, the proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems. Addison-Wesley, Redwood City, CA, 1989.

[3] J.C. Eccles, Evolution of the Brain: Creation of the Self, Routledge, 1989, Reprint edition, 1991.

[4] G.M. Edelman, Neural Darwinism: Selection and reentrant signaling in higher brain function. Neuron, 10, 115-125, 1993.

[5] D. Ferstner and N. Spruston, Cracking the neuronal code. Science, 270, 756-757, 1995.

[6] R. FitzHugh, Impulses and physiological states in models of nerve membrane. Biophys. J., 1, 445-466, 1961.

[7] M.S. Gazzaniga, Nature's Mind. New York, Basic Books. 1992.

[8] W. Gerstner and C.M. Bishop, (eds.), Spiking Neurons in Maass, Pulsed Neural Networks. MIT-press, 1999.

[9] W. Gerstner and W.M. Kistler, Spiking neuronmodels: Single neurons, populations,plasticity. Cambridge, Cambridge University Press, 2002.

[10] B.G. Hall and H.S. Malik, Determining the evolutionary potential of a gene. Mol. Biol. Evol., 15(8), 1055-1061, 1998.

[11] D. Hansel, G. Mato, C. Meunier, and L. Neltner, On numerical simulations of integrate-and-fire neural networks. Neural Comput., 10, 467-483, 1998.

[12] A. Hodgkin and A. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve. J. Physiol., 117, 500-544, 1952.

[13] E.Y. Chang, K.F. Morris, R. Shannon, and B.G. Lindsey, Repeated sequences of interspike intervals in baroresponsive respiratory related neuronal assemblies of the cat brain stem. J. Neurophysiol., 84, 1136-1148, 2000.

[14] J.P. Changeux, L'homme neuronal (Neuronal Man: The Biology of Mind, 1985), Paris, Hachette Litterature (1998).

[15] E.M. Izhikevich, Simple model of spiking neurons. IEEE Transactions on Neural Networks, 14, 1569-1572, 2003.

[16] E.M. Izhikevich, Which model to use for cortical spiking neurons? IEEE Trans. on Neural Networks, 15, 1063-1070, 2004.

[17] E.M. Izhikevich and G.M. Edelman, Large-scale model of mammalian thalamocortical systems. Proc. Natl. Acad. Sci. USA, 105, 3593-3598, 2008.

[18] E.M. Izhikevich and R. FitzHugh, FitzHugh-Nagumo Model. Scholarpedia, 1(9), 1349, 2006.

[19] M. Kirschner and J. Gerhart, Evolvability. PNAS, 95(15), 8420-8427, 1998.

[20] P.E. Latham, B.J. Richmond, P.G. Nelson, and S. Nirenberg, Intrinsic dynamics in neuronal networks. I. Theory. J. Neurophysiol., 83, 808-827, 2000.

[21] V. Litvak, H. Sompolinsky, I. Segev, and M. Abeles, On the transmission of rate code in long feedforward networks with excitatory-inhibitory balance. J. Neurosci., 23, 3006-3015, 2003.

[22] W. Maass, The Third Generation of Neural Network Models. Technische Universitat Graz, 1997.

[23] H. Markram, M. Toledo-Rodriguez, Y. Wang, A. Gupta, G. Silberberg, and C. Wu, Interneurons of the neocortical inhibitory system. Nat. Rev. Neurosci., 5, 793-807, 2004.

[24] M.E. Mazurek and M.N. Shadlen, Limits to the temporal fidelity of cortical spike rate signals. Nat. Neurosci., 5, 463-471, 2002.

[25] C. Morris and H. Lecar, Voltage oscillations in the barnacle giant muscle fiber. Biophys. J., 35, 193-213, 1981.

[26] A. Morrison, S. Straube, H.E. Plesser, and M. Diesmann, M., Exact subthreshold integration with continuous spike times in discrete time neural network simulations. Neural Comput., 19 , 47-79, 2007.

[27] C.L. Nehaniv, Evolvability. BioSystems, 69, 77-81, 2003.

[28] H.E. Plesser and M. Diesmann, Simplicity and efficiency of integrate-and-fire neuron models. Neural Comput., 21, 353-359, 2009.

[29] Y. Prut, E. Vaadia, H. Bergman, I. Haalman, H. Slovin, and M. Abeles, Spatiotemporal structure of cortical activity: Properties and behavioral relevance. J. Neurophysiol., 79, 2857-2874, 1998.

[30] R.A. Raff, The shape of life: Genes, development, and the evolution of animal form. Chicago, The University of Chicago Press, 1996.

[31] P. Reinagel and R.C. Reid, Precise firing events are conserved across neurons. J. Neurosci., 22, 6837-6841, 2002.

[32] R.M. Rose and J.L. Hindmarsh, The assembly of ionic currents in a thalamic neuron. I The three-dimensional model. Proc. R. Soc. Lond. B, 237, 267-288.

[33] M.J. Shelley and L. Tao, Efficient and accurate time-stepping schemes for integrate-and-fire neuronal networks. J Comput. Neurosci., 11, 111-119, 2001.

[34] G.D. Smith, C.L. Cox, S.M. Sherman, and J. Rinzel, Fourier analysis of sinusoidally driven thalamocortical relay neurons and a minimal integrate-and-fire-or-burst model. J. Neurophysiol., 83, 588-610, 2000.

[35] W. Swiercz, K. Cios, K. Staley, L. Kurgan, F. Accurso, and S. Sagel, A new synaptic plasticity rule for networks of spiking neurons. IEEE Trans. on Neural Networks, 17(1), 94-105, 2006.

[36] D. Tal and E.L. Schwartz, Computing with the Leaky Integrate-and-Fire Neuron: Logarithmic Computation and Multiplication. Neural Comput., 9(2), 305-318, 1997.

[37] A. Wagner, Robustness, evolvability and neutrality. FEBS, 579, 1772-1778, 2005.

Základní algoritmy modularizace rozsáhlých ontologií

doktorand:
ING. PETRA ŠEFLOVÁ

Fakulta mechatroniky, informatiky a interdisciplinárních studií
Technická Univerzita v Liberci
Hálkova 6

461 17 Liberec 1

seflova@cs.cas.cz

školitel:
ING. JÚLIUS ŠTULLER, CSC.

Ústav informatiky AV ČR, v. v. i.
Pod Vodárenskou věží 2

182 07 Praha 8

stuller@cs.cas.cz

obor studia:
Technická kybernetika

Tento příspěvek je realizován za finanční podpory Evropského sociálního fondu a státního rozpočtu ČR prostřednictvím projektu "Technologické a ekonomické kompetence pro Evropský výzkumný prostor TE-ERA".

Abstrakt

Výhody modulárních ontologií ve smyslu jednoduššího vytváření a údržby byly rozpoznány mnoha vědci. Nicméně, v reálném světě, jsou ontologie stále navrhovány jako monolitické, a proto jsou potřeba algoritmy a metody, které umožní rozdělit existující rozsáhlé ontologie do sady menších částí (modulů).

V tomto článku jsou podrobněji prezentovány dva hlavní algoritmy.

1. Úvod

Rostoucí povědomí o výhodách ontologií v oblasti zpracování dat, informací a znalostí, vedlo k vytváření velkého počtu ontologií, které se snaží popsat různé oblasti z reálného prostředí. Např. v oblasti lékařství - ontologie NCI [1], Gene Ontology a další, v oblasti elektronického obchodování - ontologie UNSPSC, NAICS.... Tyto ontologie jsou navrženy jako monolitické a obsahují desetitisíce konceptů.

Návrh, údržba a integrace takovýchto ontologií je velmi obtížná a vyžaduje velké týmy pracovníků. Tyto problémy proto vedly ke snaze rozdělit podobně rozsáhlé ontologie do menších, ucelených celků - tzv. modulů. Tento proces se nazývá modularizace ontologií.

Příklad: *Vývojář má navrhnout ontologii, která má popisovat různé typy výzkumných projektů podle témat jejich zaměření. Pro tuto ontologii potřebuje použít pojmy* Cystická fibróza *a* Genetická onemocnění *pro popis konkrétního lékařského výzkumného projektu. Vývojář je sice expert na výzkumné projekty, ale není obeznámen s některými tématy, která má cílová ontologie pokrývat (např. výše uvedené pojemy Cystická fibróza a Genetická onemocnění). Z tohoto důvodu se rozhodne využít již osvědčenou ontologii, která se zabývá popisem lékařských pojmů (např. NCI nebo SNOMED). Protože ontologii lze chápat jako sadu (logických) tvrzení (tzv. axiomů), jeví se jako nejjednodušší způsob použít logické spojení odpovídajících axiomů. Tohoto spojení může být dosaženo v OWL použitím* owl:imports konstrukce. *Nicméně ontologie jako jsou NCI nebo SNOMED jsou poměrně rozsáhlé a vložení celé ontologie by mohlo mít za následky jak znepřehlednění výsledné ontologie pro vývojáře, tak snížení výkonu. Z tohoto důvodu se jeví jako praktičtější, extrahovat z ontologie pouze část (modul), který se následně použije.*

Přestože je potřeba modularizace ontologií zřejmá, v současné době ještě nejsou pevně stanovena kritéria pro tvorbu *dobrých* modulů a také není přesně stanoveno, jaké mají být jejich vlastnosti. Výsledkem je, že u aktuálně používaných přístupů pro extrakci modulů z ontologie, každý obyčejně implementuje svoji vlastní *představu toho co je a co není modul* .

V současné době jsou používány v oblasti sémantického webu 2 hlavní algoritmy pro extrakci modulů z dané ontologie:

- rozdělení na základě struktury ontologie
- segmentace ontologie

Příspěvek je rozdělen následovně: V úvodu jsou popsána úskalí rozsáhlých monolitických ontologií. V další části jsou prezentována základní kritéria používána pro modularizaci ontologií. V třetí části jsou představeny 2 hlavní algoritmy používané pro vytváření modulů z rozsáhlých ontologií.

2. Kritéria modularity

V současné době je navrženo několik formálních i neformálních kritérií modularizace [2], která jsou často ovlivněna konkrétními případy použití modularizace. Navíc, na rozdíl od softwarového inženýrství, není v oblasti ontologií ještě pevně zakotven pojem modulu (ontologie) a z toho plynoucí vlastnosti modulu.

Z analýzy vlastností modulů používaných v softwarovém inženýrství vyplynuly dvě hlavní kritéria, která mohou být použita pro modularizaci ontologií: **zapouzdření a nazávislost**.

Zapouzdření

V softwarovém inženýrství zapouzdření odkazuje na rozdíl mezi rozhraním a implementací konkrétního modulu. Tento rozdíl není přímo aplikovatelný, ale může souviset s jinými pojmy jako je např. *nahraditelnost modulu*.

Nezávislost

Vhodně navržený softwarový modul by měl být nezávislý na jiném modulu použitém ve stejné aplikaci, to znamená, že je použit a využit samostatně. To samé můžeme aplikovat v ontologickém inženýrství, kde modul má být celistvá a znovupoužitelná komponenta.

2.1. Strukturální kritéria

Tato kritéria jsou odvozena na základě struktury ontologie, kterou chceme modularizovat. Souvisí neodmyslitelně s předchozími kritérii převzatými ze softwarového inženýrství a jsou navržena jako kompromis mezi *udržovatelností a výkonností*.

Velikost

Velikost modulu má zásadní vliv na jeho údržbu: přirozená složitost znalostí v jednom modulu vede k menší pružnosti při jeho využití a vývoji. Na druhou stranu, příliš malé moduly nemohou dostatečně pokrýt širokou oblast a mohou vést k problémům souvisejících s dalšími kritérii (např. spojitost - ontologii můžeme vidět jako graf, kde koncepty jsou uzly a hrany jsou spojení mezi dvěmi konkrétními koncepty. Spojitost je určena na základě počtu konceptů, které sdílí daný modul s jinými moduly).

Vzdálenost uvnitř modulu

Vzdálenost uvnitř modulu je určena jako počet vztahů v nejkratší cestě z jednoho elementu ontologie k druhému, spojených těmito vztahy. Toto je důležité v případě stanovení modulů pomocí extrakčních technik, kde omezení vzdáleností mezi vstupními pojmy usnadňuje jejich vizualizaci a pomáhá v porozumění vztahů v původní ontologii.

2.2. Kvalita modulů

Rozlišujeme dva různé druhy přístupů k určení kvality existující ontologie, které mohou být použity pro hodnocení kvality ontologických modulů:

- jeden analyzuje strukturu ontologie na základě aktuálního obsahu,
- druhý porovnává obsah ontologie s alternativní reprezentací.

Doporučuje se kombinovat oba tyto způsoby, abychom získali nejvhodnější modul pro popis určitého aspektu v dané oblasti.

3. Algoritmy modularizace

3.1. Algoritmus rozdělení ontologie na základě její struktury

Základem algoritmu je převedení ontologie do grafu na základě stanovení závislostí mezi jednotlivými elementy (koncepty, vztahy, instancemi) ontologie a nalezení závislých uzlů. Algoritmus rozdělení ontologie na základě její struktury lze rozdělit do tří základních úloh.

- *Vytvoření grafu* na základě závislostí mezi jednotlivými elementy v ontologii.
- *Nalezení sad uzlů* (modulů) s danou maximální velikostí, pro které je závislost mezi nimi uvnitř modulu větší než závislosti spojení s uzly v ně tohoto modulů.
- *Optimalizace* vytvořených modulů.

3.1.1 Závislostní graf: První úlohou je vytvoření *ohodnoceného grafu* na základě struktury závislostí ontologie. Tato část algoritmu se skládá ze dvou kroků

1. vytvoření grafu
2. výpočet ohodnocení grafu

Krok 1

V prvním kroce je na základě zdrojové ontologie vytvořen závislostní graf. Jednotlivé uzly v grafu reprezentují elementy ontologie. Spojení mezi jednotlivými

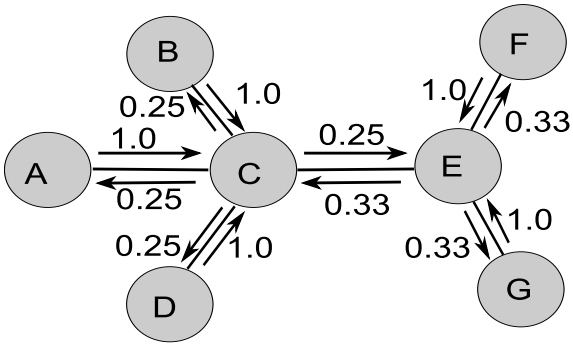
uzly je vytvořeno pouze tehdy, pokud odpovídající elementy ve zdrojové ontologii jsou též spojeny určitým vztahem. Rozlišujeme pět typů vztahů, které mohou být mezi jednotlivými elementy a to: *podtřída, vlastnost, definice, podřetězec a vzdálenost mezi elementy.*

Krok 2

Druhý krok spočívá v *určení síly závislostí spojení* mezi jednotlivými uzly závislostního grafu. Při určení síly závislosti se využívá algoritmus síťové analýzy, pomocí kterého se stanoví tzv. *proporcionální síla* grafu.

Síla závislosti spojení mezi uzly c_i a c_j je určena jako proporcionální síla spojení: tato proporcionální síla **w** je rovna podílu součtu vah spojení mezi uzlem c_i a uzlem c_j a součtem vah všech spojení uzlu c_i s dalšími uzly.

$w(c_i, c_j) = \frac{a_{ij}+a_{ji}}{\sum_k a_{ik}+a_{ki}}$, kde a_{ik} je váha stanovená pro spojení mezi uzly c_i a c_k .



Obrázek 1: Příklad grafu s proporcionální silou závislosti.

3.1.2 Určení modulu (krok 3): Proporcionální závislostní graf nám poskytuje základ pro určení sady *souvisejících konceptů*. Pro detekci těchto konceptů se používá algoritmus, který vypočítá všechna maximální spojení uzlů v grafu.

Definice řady uzlů

Mějme proporcionální závislostní graf G=(C,D,w), kde C je sada všech vrcholů v grafu, D je sada hran spojujících C, w je proporcionální síla.

Sadou vrcholů (Line Islands) nazveme množinu I jestliže platí

- $I \subseteq C$

- I je souvislý podgraf z G

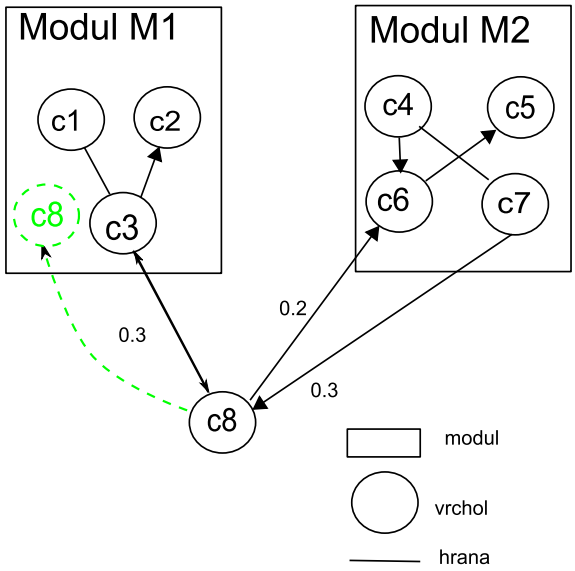
- existuje vážený graf $T = (C_T, D_T, w_T)$ takový, že

- *- T je podgraf G*
- *- T je maximální větvící se strom s ohledem na I*

3.1.3 Optimalizace modulu:

Optimalizace - převod izolovaných konceptů (krok 4): V některých případech nám mohou po rozdělení dané ontologie na moduly zůstat uzly, které nejsou přiřazeny žádnému modulu. V tomto kroku automaticky přiřadíme tyto zbylé uzly k modulu, se kterým má daný uzel největší váhu spojení.

Na obrázku 2 je zobrazen příklad grafu, který vznikl rozdělením ontologie. Graf obsahuje dva moduly (M1 a M2) a uzel c8, který nenáleží ani modulu M1 ani modulu M2. Uzel c8 je spojen jednou (obousměrnou) hranou s modulem M1 s váhou 0.3 v obouch směrech. S modulem M2 je uzel c8 spojen dvěma hranami s váhou 0.2 a 0.3. Výsledná váha spojení mezi uzlem a jednotlivými moduly se rovná součtu vah jednotlivých hran. V našem příkladu je výsledná váha spojení mezi uzlem c8 a modulem M1 0.6, mezi uzlem c8 a modulem M2 0.5. Z těchto výsledků nám vyplývá, že uzel c8 bude přiřazen modulu M1.



Obrázek 2: Přiřazení zbylého uzlu k modulu.

Optimalizace - slučování (krok 5): V některých případech algoritmus generuje moduly, které nesplňují naší intuitivní představu o povaze modulu, dochází k rozdělení souvisejících pojmů (souvislého tématu) do různých modulů (např. při dosažení maximální velikosti modulu). Po provedení kontroly závislostí v příslušné části ontologie, zjistíme, že většina modulů, které nesplňují naší intuitivní představu o rozdělení, obsahuje velmi silnou závislost spojení mezi koncepty. Pro stanovení takovýchto případů potřebujeme znát míru síly

vnitřní závislosti. Tato míra používá minimální strom T k identifikaci modulu: celková síla vnitřní závislosti se rovná síle nejslabšího spojení v daném stromě.

Všechny moduly jejichž úroveň dosáhne daného prahu jsou slučovány do odpovídajícího modulu s nižší úrovní. V mnoha případech je pouze jeden sousední modul, se kterým se může sloučit. Pokud je zde více sousedících modulů, vhodných pro sloučení, je použita síla závislosti mezi moduly pro určení kandidáta pro sloučení.

Optimalizace - kritéria minimalizace/maximalizace (krok 6): Pro optimalizaci jsou použita tato kritéria: *počet modulů, průměrná velikost modulu a odchylka velikosti*.

Algoritmus rozdělení ontologie na základě její struktury pomocí váhového grafu je v současné době implementován např. v editoru PATO (*Partitioning Tool*).

3.2. Algoritmus segmentace ontologie

Algoritmus segmentace ontologie slouží pro extrakci souvisejících částí (modulů) z rozsáhlé ontologie [4]. Moduly by neměly být pouhým zlomkem dané ontologie, ale měly by tvořit sami o sobě novou ontologií. Tato technika využívá výhod podrobného zachycení sémantiky OWL ontologií.

V současné době je tento algoritmus optimalizován pro práci s ontologiemi podobnými ontologii GALEN [17], tzn. pracuje s rozsáhlými ontologiemi majícími přes 10 000 konceptů a *celistvou spojitost* (tzn. neobsahují samostatné, nezávislé koncepty), v průměru jedno *omezení tvrzení* (tvořená např. existenciálními či všeobecnými kvantifikátory) na koncept. Dalším předpokladem je, že ontologie je normalizována (cílem normalizace je zajistit jednoznačnost použitých pojmů; normalizace velmi zjednodušuje údržbu ontologie).

Základní koncepty v normalizované ontologii nemají více než jeden základní nadřízený koncept (superclass). Definované koncepty mají přesně jedno pojmenování konceptu v jejich definici. Toto označení konceptu může být viděno jako základní koncept jednotlivých nadřízených konceptů a proto tento segmentační algoritmus může pracovat stejně dobře jak na základních tak definovaných konceptech.

Základní segmentační algoritmus začíná na jednom nebo více konceptech vybraných uživatelem a vytváří extrakt založený na těchto konceptech a souvisejících konceptech. Tyto související koncepty jsou identifikovány pomocí struktury spojení ontologie.

Tento algoritmus je založen na 4 krocích:

`1. Průchod směrem vzhůru hierarchií`

Předpokládejme, že chceme vygenerovat modul týkající se *konceptu srdce* z ontologie GALEN. Je zřejmé, že první koncept, který do modulu zahrmene, je koncepta srdce. Nadřízeným konceptem tohoto konceptu je koncept *Vnitřní orgány* atd. Takto postupujeme směrem vzhůru hierarchií ontologie, dokud není dosaženo vrcholu. Protože je ale hierarchie ontologie často obsáhlá (např.v tomto případě se jedná o 13 nadřízených konceptů), mohli bychom uvážit sloučení několika nadřízených konceptů. Nicméně toto může zničit některé sémantické detaily dané ontologie. Tato strategie se jeví jako vhodná, když konstruujeme pohled ontologie, ale není vhodná pro extrakci, která má být použita v aplikaci, protože každý nadřízený koncept může obsahovat znalosti potřebné pro danou aplikaci.

`2. Průchod směrem dolů hierarchií`

Tento algoritmus prochází ontologii směrem dolů od zvoleného konceptu, zahrnuje všechny její podřízené koncepty.

`3. Sourozenecké koncepty v hierarchii`

Sourozenecké koncepty nejsou zahrnuty v extraktu. V našem příkadě jsou sourozenci koncepty jako *plíce, žaludek a ledviny*. Je rozumné předpokládat, že nejsou dostatečně relevantní, aby byly zahrnuty standardně. Uživatel ale může vždy explicitně určit výběr konceptů pro zahrnutí do extraktu.

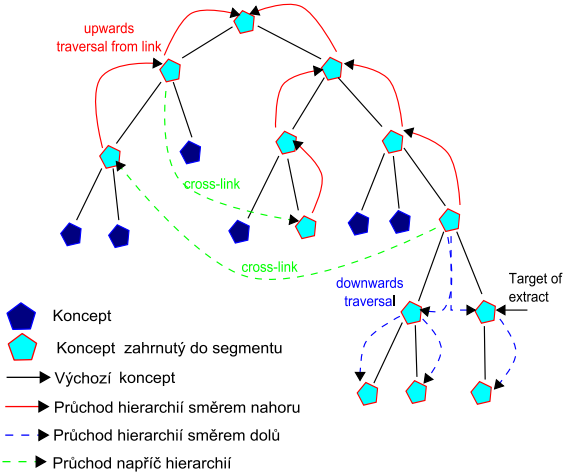
`4. Četnost průchodů vzestupně a sestupně podle spojení`

V této chvíli již máme vybrané koncepty podle cílového konceptu, jejich omezení, průnik, spojení a ekvivalentní koncepty.

Obrázek 3 nám poskytuje názorný příklad segmentačního algoritmu. Algoritmus začíná na konceptu zvoleným uživatelem a prochází hierarchií ontologie nejprve směrem vzhůru všemi možnými způsoby až ke kořenovému konceptu. V dalším kroku prochází všemi možnými způsoby podřízené koncepty daného konceptu směrem dolů. Nakonec může algoritmus, pokud je to vhodné, procházet strukturou ontologie přes sourozence.

Tento segmentační algoritmus generuje extrakt všech souvisejících konceptů s cílovým konceptem. Nicméně, v hustě vnitřně spojených ontologiích, jako je GALEN, nová ontologie (modul) má obvykle velikost jedné pětiny původní ontologie (tato hodnota byla získána na

základě provedených testů tohoto algoritmu). Z toho vyplývá potřeba dalších omezení segmentace.



Obrázek 3: Průchod ontologií směrem nahoru a dolů.

3.2.1 Omezení velikosti segmentu: Jestliže je naším cílem vyrobit segment pro použití člověkem nebo pro speciální aplikace, jeví se jako užitečné filtrovat určité vlastnosti. Např. jestliže se uživatel nezajímá o nemoci, může určit vyjmutí všech *lokálních vlastností* souvisejících s nemocemi.

3.3. Porovnání algoritmů

Vývojáři využívají algoritmus segmentace ontologie pro rychlé a snadné vytváření relevantních zákaznických ontologií, na místo toho, aby se spoléhali na dekompozici ontologie navrženou jejím autorem. Tento algoritmus využívá výhod ontologických principů: normalizace, nadontologie a bohatou hierarchii vlastností. U algoritmu segmentace není nutný převod ontologie do závislostního grafu.

Algoritmus rozdělení ontologie na základě její struktury se jeví jako univerzálnější (oproti algoritmu segmentace pracuje i s nemormalizovanými ontologiemi), ale již vyžaduje určité zkušenosti vývojáře.

4. Závěr

S rostoucí velikostí ontologií roste potřeba používat principy modularity ontologií, aby se usnadnilo vytváření, údržba a využití znalostí zachycených v těchto ontologiích. Tento příspěvek se věnoval dvěma základním algoritmům používaných pro tvorbu modulů. Plán na nejbližší období mého studia počítá s provedením testů jednotlivých algoritmů pro různé ontologie. Dále bych se ráda věnovala problematice zjišťování změn v původních ontologiích a jejich automatické implementaci do již vygenerovaných modulů.

Literatura

[1] J. Golbeck at al. (Eds.), The national cancer institute's thesaurus and ontology , *Journal of Web Semantics*, 1(1), 2003.

[2] H. Stuckenschmidt at al. (Eds.), Modular Ontologies , *LNCS 5445, 2009, Springer/Verlag Berlin Heidelberg 2009*.

[3] S. Spaccapietra and A. Tamilin, D2.1.3.1 Report on Modularization of Ontologies *KWEB/2004/D2.1.3.1/v1.1 July 30, 2005*.

[4] J. Seidenberg and A.Rector, 'Web Ontology Segmentation: Analysis, Classification and Use', *WWW2006, Edinburgh,Scotland,May 23 - 26, 2006*, page 13 - 22.

[5] P. Šeflová, Metody modularizace rozsáhlých ontologií, *Doktorandské dny '09, Matfypress, 2009, s. 108 - 112, ISBN: 978-80-7378-087-6*.

[6] V. Bagatejl, 'Analysis of large network islands', *Dagstuhl Semina 03361, University of Ljubljana, Slovenia, August 2003. Algorithmic Aspects of Large and Complex Network*.

[7] S. Brin and L.Page, 'The anatomy of a large-scale hypertextual Web search engine', *Computer Networks and ISDN Systems, 30 (1-7):1007-117,1998*.

[8] B.C. Grau, B. Parsia, E. Sirin, and A. Kalyanpur, 'Modularizing OWL Ontologies', *In K-CAP 2005 Worshop on Ontology Management, October 2005*.

[9] A. Magkanaraki, V. Tannen, V. Christophides, and D. Plexousakis, 'Viewing the Semantic Web through RVL Lenses', *Journal of Web Semantics, 1(4):29, October 2004*.

[10] D.L. McGuinness and F. Van Harmelen, 'OWL Web Ontology Language Overview', *February 2004, W3C Recommendation*.

[11] N. Noy and M.A. Musen, 'Specifying ontology views by traversal', *In S. A. McIlraith, D. Plexousakis, and F. Van Harmelen, editors, International Semantic Web Conference (ISWC), volume 3298 of Lecture Notes in Computer Science, pages 713-725. Springer, November 2004*.

[12] A. Pease, I.Niles, and J. Li, 'The suggested upper merged ontology: A large ontology for the semantic web and its applications', *In Working Notes of the AAAI-2002 worshop on Ontologies and the Semantic Web, July 28 – August 1, 2002*.

[13] A.L. Rector, 'Normalisation of ontology implementations: Towards modularity, re-use, and maintainability', *In EKAW Worshop on Ontologies for Multiagent Systems, 2002*.

[14] H.A. Simon, 'The Sciences of the Artifical', *chapter 7, pages 209-217. MIT Press, 1969*.

[15] H. Stuckemschmidt and M. Klein, 'Structure-based partitioning of Large Class Hierarchies', *In Proceedings of the 3rd ISWC,2004*.

[16] M. Bhatt, C. Wouters, A. Flahive, W. Rahayu, and D. Taniar, 'Semantic completeness in sub-ontology extraction using distributed methods', *In A. Lagana, M.L. Gavrilova, and V. Kumar, editors, Computational Science and Its Applications (ICCSA), volume 3045, pages 508 - 517. Springer-Verlag GmbH, May 2004*.

[17] A. Rector and J. Rogers, Ontological Issues in using a Description Logic to Represent Medical Concepts: Experiemnce from GALEN, *In IMIA WG6 Workshop, 1999*.

Fuzzy t-conorm Integral as an Aggregation Operator in Dynamic Classifier Systems

Post-Graduate Student:
ING. DAVID ŠTEFKA

Department of Mathematics
Faculty of Nuclear Science and Physical Engineering
Czech Technical University
Trojanova 13

120 00 Prague 2, CZ

david.stefka@gmail.com

Supervisor:
DOC. ING. RNDR. MARTIN HOLEŇA, CSC.

Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2

182 07 Prague 8, CZ

martin@cs.cas.cz

Field of Study:
Mathematical Engineering

The research reported in this paper was partially supported by the grant ME949 of the Ministry of Education, Youth and Sports of the Czech Republic (D. Štefka), and by the grant No. 201/08/0802 of the Grant Agency of the Czech Republic and by the Institutional Research Plan AV0Z10300504 (M. Holeňa).

Abstract

Fuzzy integral is a general aggregation operator, which encompasses many common aggregation operators like weighted mean, ordered weighted mean, weighted minimum and maximum, etc. In classifier combining, it can be used to aggregate the outputs of the individual classifiers in the team with respect to a fuzzy measure, based on the classifier confidences. In practice, the Choqet integral and the Sugeno integral are used most often. However, they both belong to the more general family of fuzzy t-conorm integral. In this paper, we theoretically examine which fuzzy t-conorm integrals are useful for classifier aggregation, and we experimentally compare the individual methods on 27 benchmark datasets.

1. Introduction

Classifier combining is a popular approach for improving the quality of classification. By combining outputs of several classifiers into one final decision, the combined classifier can perform better than any of the individual classifiers [1–3].

There are two main approaches to classifier combining: *classifier selection* [4–7], where a single classifier is selected from the team according to some criterion, and *classifier aggregation* [1–3, 8, 9], where outputs of all the classifiers in the team are aggregated into a single prediction. Classifier combination can be either *static*, i.e., the combining process is the same for all patterns, or *dynamic*, where the combination process is adapted to the currently classified pattern [4, 6, 10–12].

One of the popular aggregation operators is the *fuzzy integral* [1, 9, 13, 14]. Fuzzy integral aggregates the input with respect to a *fuzzy measure*, which is a generalization of the additive probabilistic measure, where the additivity is replaced by a weaker condition, monotonicity. Leaving out the additivity allows us to model interactions between different elements of the fuzzy measure space. Two main types of fuzzy integrals are used most often – the Choquet integral, which generalizes the Lebesgue integral, weighted mean, ordered weighted mean, etc., and the Sugeno integral, which generalizes the weighted minimum and weighted maximum.

Both the Choquet integral and the Sugeno integral can be further generalized using the fuzzy t-conorm integral framework [13–15]. In this approach, the spaces of the integrand, fuzzy measure and integral are separated and each space has its own t-conorm operation. The t-conorms, along with a multiplication-like operation which interconnects them, build a *t-conorm system for integration*.

By choosing particular t-norms and multiplication-like operations, we obtain different fuzzy integrals, including the Choquet and Sugeno integrals. By using the t-conorm integral framework in classifier aggregation, we obtain additional degrees of freedom in the aggregation, and thus we can fine-tune the aggregation to a particular data. However, not all t-conorm systems for integration lead to meaningful aggregation operators.

In this paper, we introduce the theory of classifier combining and the framework of fuzzy t-conorm integral. We theoretically investigate which combinations of

t-conorms and multiplication-like operations lead to meaningful aggregation operators. This will show that the family of t-conorm integrals is not as rich as it may seem on the first glance, and that it reduces to the Choquet integral, Sugeno integral, and $\vee_S$-type systems, including the quasi-Sugeno integral. In the experimental section, we compare the representatives of different fuzzy t-conorm integrals, i.e., the Choquet integral, the Sugeno integral, and the quasi-Sugeno integral.

The paper is structured as follows. Section 2 presents the formalism of classifier combining, classification confidence, and dynamic classifier systems. In Section 3, we deal with the fuzzy integral and its use in classifier aggregation. Section 4 introduces the fuzzy t-conorm integral and examines possible t-conorm systems for integration. Section 5 contains the experiments, and Section 6 then concludes the paper.

2. Classifier Combining

In this section, we present the formalism of dynamic classifier combining, as introduced in [10]. Throughout the rest of the paper, we use the following notation. Let $[0,1]$ denote the unit interval, let $\mathcal{X} \subseteq \mathbf{R}^n$ be a n-dimensional *feature space*, let $C_1, \ldots, C_N \subseteq \mathcal{X}$, $N \geq 2$ be disjoint sets called *classes*. A *pattern* is a tuple $(\vec{x}, c_{\vec{x}})$, where $\vec{x} \in \mathcal{X}$ are *features* of the pattern, and $c_{\vec{x}} \in \{1, ..., N\}$ is the index of the class the pattern belongs to. The goal of classification is to predict $c_{\vec{x}}$ for unclassified patterns. We assume that for every $\vec{x} \in \mathcal{X}$, there is a unique classification $c_{\vec{x}}$, but when we are classifying a pattern, we do not know it – due to this fact, we will denote unclassified patterns only as $\vec{x} \in \mathcal{X}$.

Definition 1 *We call a* classifier *any mapping* $\phi : \mathcal{X} \to [0,1]^N$*, where for* $\vec{x} \in \mathcal{X}$*,* $\phi(\vec{x}) = (\gamma_1(\vec{x}), \ldots, \gamma_N(\vec{x}))$ *are* degrees of classification (d.o.c.s) *to each class.*

The d.o.c. to class C_j expresses the predicted extent to which the pattern belongs to class C_j (if $\gamma_i(\vec{x}) > \gamma_j(\vec{x})$, it means that the pattern $\vec{x}$ belongs to class C_i rather than to C_j). Depending on the classifier type, it can be modelled by probability, fuzzy membership, etc. The prediction of $c_{\vec{x}}$ for an unknown pattern $\vec{x}$ is done by converting the continuous d.o.c.s of the classifier into a *crisp output* $\phi^{(cr)}(\vec{x}) = \arg\max_{i=1,\ldots,N} \gamma_i(\vec{x})$ if there are no ties, or arbitrarily as $\phi^{(cr)}(\vec{x}) \in \arg\max_{i=1,\ldots,N} \gamma_i(\vec{x})$ in the case of ties.

2.1. Classification Confidence

In addition to the classifier output (the d.o.c.s), which predicts to which class a pattern belongs to, we will work with the *confidence* of the prediction $\kappa_\phi(\vec{x})$, i.e., the extent to which we can "trust" the output of the classifier. The higher the trust in the classification, the closer $\kappa_\phi(\vec{x})$ is to 1. In the rest of the paper, we will use the indicator operator I, defined as $I(\text{true}) = 1$, $I(\text{false}) = 0$.

Definition 2 *We call a* confidence measure *of classifier* ϕ *every mapping* $\kappa_\phi : \mathcal{X} \to [0,1]$*. Let* $\vec{x} \in \mathcal{X}$*.* $\kappa_\phi(\vec{x})$ *is called* classification confidence *of* ϕ *on* $\vec{x}$*.*

A confidence measure can be either *static*, i.e., it is a constant of the classifier, or *dynamic*, i.e., it adjusts itself to the currently classified pattern. Static confidence measures include accuracy, precision, sensitivity, resemblance, etc. [16, 17], and they are usually computed on a validation set after the classifier is trained. An example of a static confidence measure is the Global Accuracy measure:

$$\kappa_\phi^{(GA)} = \frac{\sum_{(\vec{y}, c_{\vec{y}}) \in \mathcal{V}} I(\phi^{(cr)}(\vec{y}) \stackrel{?}{=} c_{\vec{y}})}{|\mathcal{V}|}, \qquad (1)$$

where $\mathcal{V} \subseteq \mathcal{X} \times \{1, \ldots, N\}$ is the validation set.

Dynamic confidence measures [4, 6, 10–12, 18] predict the local quality of the classification for the particular pattern $(\vec{x}, c_{\vec{x}})$. An example of a dynamic confidence measure is the Euclidean Local Accuracy (ELA):

$$\kappa_\phi^{(ELA)}(\vec{x}) = \frac{\sum_{(\vec{y}, c_{\vec{y}}) \in \mathcal{V}(\vec{x})} I(\phi^{(cr)}(\vec{y}) \stackrel{?}{=} c_{\vec{y}})}{|\mathcal{V}(\vec{x})|}, \qquad (2)$$

where $\mathcal{V}(\vec{x}) \subseteq \mathcal{V}$ is the set of validation patterns neighboring with $\vec{x}$ (for example k nearest neighbors under Euclidean metric).

2.2. Classifier Systems

In classifier combining, instead of using just one classifier, a team of classifiers is created (sometimes called an *ensemble of classifiers*), and the team is then aggregated into one final classifier. If we want to utilize classification confidence in the aggregation process, each classifier must have its own confidence measure defined.

Definition 3 *Let* $r \in \mathbf{N}$*,* $r \geq 2$*.* Classifier team *is a tuple* $(\Gamma, \mathcal{K})$*, where* $\Gamma = (\phi_1, \ldots, \phi_r)$ *is a set of classifiers, and* $\mathcal{K} = (\kappa_{\phi_1}, \ldots, \kappa_{\phi_r})$ *is a set of corresponding confidence measures.*

If a pattern $\vec{x}$ is submitted for classification, the team of classifiers gives us information of two kinds – outputs of

the individual classifiers (a *decision profile*), and classification confidences of the classifiers on $\vec{x}$ (a *confidence vector*).

Definition 4 *Let $(\Gamma, \mathcal{K})$ be a classifier team, and let $\vec{x} \in \mathcal{X}$. Then we define* decision profile $\Gamma(\vec{x}) \in [0,1]^{r \times N}$

$$\Gamma(\vec{x}) = \begin{pmatrix} \phi_1(\vec{x}) \\ \phi_2(\vec{x}) \\ \vdots \\ \phi_r(\vec{x}) \end{pmatrix} = \begin{pmatrix} \gamma_{1,1}(\vec{x}) & \gamma_{1,2}(\vec{x}) & \dots & \gamma_{1,N}(\vec{x}) \\ \gamma_{2,1}(\vec{x}) & \gamma_{2,2}(\vec{x}) & \dots & \gamma_{2,N}(\vec{x}) \\ & & \ddots & \\ \gamma_{r,1}(\vec{x}) & \gamma_{r,2}(\vec{x}) & \dots & \gamma_{r,N}(\vec{x}) \end{pmatrix}, \tag{3}$$

and confidence vector $\mathcal{K}(\vec{x}) \in [0,1]^r$

$$\mathcal{K}(\vec{x}) = \begin{pmatrix} \kappa_{\phi_1}(\vec{x}) \\ \kappa_{\phi_2}(\vec{x}) \\ \vdots \\ \kappa_{\phi_r}(\vec{x}) \end{pmatrix} \tag{4}$$

After the pattern $\vec{x}$ has been classified by all the classifiers in the team, and the confidences have been computed, these outputs have to be aggregated using a *team aggregator*, which takes the decision profile as its first argument, the confidence vector as its second argument, and returns the aggregated degrees of classification to all the classes. A classifier team with an aggregator will be called a *classifier system*, which can be also viewed as a single classifier.

Definition 5 *A* team aggregator *of dimension (r, N) is any mapping $\mathcal{A} : [0,1]^{r \times N} \times [0,1]^r \to [0,1]^N$.*

Definition 6 *Let $(\Gamma, \mathcal{K})$ be a classifier team, and let $\mathcal{A}$ be a team aggregator of dimension (r, N), where r is the number of classifiers in the team, and N is the number of classes. The triple $\mathcal{S} = (\Gamma, \mathcal{K}, \mathcal{A})$ is called a* classifier system. *We define an* induced classifier of $\mathcal{S}$ *as a classifier Φ:*

$$\Phi(\vec{x}) = \mathcal{A}(\Gamma(\vec{x}), \mathcal{K}(\vec{x})).$$

An example of an aggregation operator is the mean value, which defines the aggregated d.o.c. to class j as the arithmetic mean of the d.o.c.s to class j given by the classifiers in the team:

$$\gamma_j(\vec{x}) = \frac{\sum_{i=1,\dots,r} \gamma_{i,j}(\vec{x})}{r}, \tag{5}$$

where $\Phi(\vec{x}) = \mathcal{A}(\Gamma(\vec{x}), \mathcal{K}(\vec{x})) = (\gamma_1(\vec{x}), \dots, \gamma_N(\vec{x}))$.

We can distinguish three types of classifier systems: confidence-free (which do not utilize the classification confidence at all), static (which use only static classification confidence), and dynamic (which utilize classification confidence in a dynamic way, i.e. the aggregation is adapted to the particular pattern).

In classifier aggregation, many aggregation operators have been studied in the literature [1–3, 8, 9]: simple arithmetic operations (voting, sum, maximum, minimum, mean, weighted mean, weighted voting, product, etc.), probability-based approaches (e.g., product rule, bayesian aggregation, Dempster-Shafer fusion), fuzzy logic methods (fuzzy integrals, decision templates), and others.

In classifier selection [4–7], a single classifier from the team is selected according to some criterion (which corresponds to the confidence measure in our framework), and the result of the classifier is taken as the final result. Using the notation from this section, $\Phi(\vec{x}) = \mathcal{A}(\Gamma(\vec{x}), \mathcal{K}(\vec{x})) = \phi_k(\vec{x})$, such that $k \in \arg\max\{\kappa_{\phi_i}(\vec{x}) | i = 1, \dots, r\}$.

However, our key interest in this paper lies in studying dynamic classifier aggregation using the fuzzy integral, which is described in the following section.

3. Fuzzy Integral

Fuzzy integral [13–15] is an aggregation operator, based on a *fuzzy measure* (sometimes called *capacity*), which is a generalization of the additive measure, such that the additivity is replaced by a weaker condition – monotonicity. Several definitions of a fuzzy integral exists in the literature – among them, the Choquet integral and the Sugeno integral are used most often. In this section, we briefly summarize the needed definitions, and we show how the fuzzy integral can be used in classifier aggregation. For simplicity reasons, we restrict ourselves to the discrete case, i.e. the universe $\mathcal{U}$ is finite, and to functions in $[0,1]$ with discrete domain (so called simple functions). As the universe $\mathcal{U}$ will correspond to the set of classifiers in the team, we use r to denote the universe size.

3.1. Fuzzy Measure

Definition 7 *A fuzzy measure μ on a set $\mathcal{U} = \{u_1, \dots, u_r\}$ is a set function $\mu : \mathcal{P}(\mathcal{U}) \to [0,1]$, such that:*

1. $\mu(\emptyset) = 0, \mu(\mathcal{U}) = 1$ *(boundary conditions)*
2. $A \subseteq B \Rightarrow \mu(A) \leq \mu(B)$ *(monotonicity)*

The fuzzy measure is determined by all its 2^r values. In contrast, if the fuzzy measure is additive, i.e., $\mu(A \cup B) = \mu(A) + \mu(B)$ for disjoint A, B, then we need to know only the measure values for the singletons, $\mu(u_i),\ i = 1, \ldots, r$ (usually called *fuzzy densities*), and all the remaining values can be computed using the additivity condition.

Since it is hard to define all the 2^r values, and since it is quite easy to define the fuzzy densities, methods for computing the fuzzy measure values from the fuzzy densities have been developed [13,14]. $\perp$-decomposable fuzzy measures ($\perp$ being a t-conorm), defined as

$$\mu(A \cup B) = \mu(A) \perp \mu(B), \tag{6}$$

for disjoint $A, B \in \mathcal{U}$, are examples of such approaches. Particular cases of $\perp$-decomposable fuzzy measures are additive measures ($\perp$ being the bounded sum), and the Sugeno λ-measure [13, 14], defined as

$$\mu(A \cup B) = \mu(A) + \mu(B) + \lambda\mu(A)\mu(B), \tag{7}$$

for disjoint $A, B \in \mathcal{U}$, and some fixed $\lambda > -1$. The value of λ is computed as the unique non-zero root greater than -1 of the equation

$$\lambda + 1 = \prod_{i=1,\ldots,r} (1 + \lambda\mu(u_i)), \tag{8}$$

if the densities do not sum to 1. If they do sum to 1, $\lambda = 1$ and the fuzzy measure is additive.

3.2. Fuzzy Integrals

After the fuzzy measure is defined, we can define the fuzzy integral. The most common fuzzy integrals are Choquet and Sugeno integrals. Throughout the rest of the paper, we will use the following notation. Let $\mathcal{U} = \{u_1, \ldots, u_r\}$ be a universe, let $f : \mathcal{U} \to [0, 1]$, $f(u_i) = f_i,\ i = 1, \ldots, r$ be a simple function on $\mathcal{U}$, and let μ be a fuzzy measure on $\mathcal{U}$. Then we will use the notation $f_{<i>}$, $A_{<i>}$, where $f_{<i>}$ indicates that the indices have been permuted, such that $0 \leq f_{<1>} \leq \cdots \leq f_{<r>} \leq 1$, and $f_{<0>} = 0$, and $A_{<i>} = \{u_{<i>}, \ldots, u_{<r>}\}$. Using this notation, $\mu(A_{<i>})$ represents the value of the fuzzy measure for the set of elements of $\mathcal{U}$ corresponding to the $r - i$ highest values of f.

Definition 8 *Let μ be a fuzzy measure on $\mathcal{U}$. Then the* Choquet integral *of a function $f : \mathcal{U} \to [0, 1]$, $f(u_i) = f_i,\ i = 1, \ldots, r$, with respect to μ is defined as:*

$$\int_C f d\mu = \sum_{i=1}^{r} (f_{<i>} - f_{<i-1>})\mu(A_{<i>}). \tag{9}$$

Definition 9 *Let μ be a fuzzy measure on $\mathcal{U}$. Then the* Sugeno integral *of a function $f : \mathcal{U} \to [0, 1]$, $f(u_i) = f_i,\ i = 1, \ldots, r$, with respect to μ is defined as:*

$$\int_S f d\mu = \max_{i=1}^{r} \min(f_{<i>}, \mu(A_{<i>})), \tag{10}$$

Both the Choquet and Sugeno integrals belong to the more general class of fuzzy t-conorm integrals, which will be described in the following section. Prior to that, we demonstrate the use of fuzzy integrals as aggregation operators in classifier aggregation.

3.3. Fuzzy Integral in Classifier Aggregation

In classifier aggregation, the universe $\mathcal{U}$ corresponds to the set of classifiers Γ in the team, i.e., $\mathcal{U} = \Gamma$. For $\vec{x} \in \mathcal{X}$, the individual columns of the decision profile $\Gamma(\vec{x})$ (d.o.c.s to a particular class given by all the classifiers in the team) are integrated using the fuzzy integral, i.e., the aggregated d.o.c. to class j is defined as

$$\gamma_j(\vec{x}) = \int \Gamma_{*,j} d\mu, \tag{11}$$

where $\int$ is a fuzzy integral, $\Gamma_{*,j}$ is the j-th column of Γ (d.o.c.s to class C_j), and μ is a fuzzy measure based on the confidences $\mathcal{K}(\vec{x})$. The fuzzy measure μ represents the importance of a particular set of classifiers used in the integration ($\mu(A_{<i>})$ represents the importance of the classifiers corresponding to the $r - i$ highest d.o.c.s). Usually, Sugeno λ-measure is used as the fuzzy measure, with the fuzzy densities defined as the confidences, $\mu(\{\phi_i\}) = \kappa_{\phi_i}(\vec{x})$.

4. Fuzzy t-conorm Integral

Through the rest of the paper, we use the following notation for common t-norms and t-conorms:

- Standard: $x \wedge_S y = \min(x, y), x \vee_S y = \max(x, y)$
- Łukasiewicz: $x \wedge_L y = \max(x+y-1, 0), x \vee_L y = \min(x + y, 1)$
- Product: $x \wedge_P y = xy, x \vee_P y = x + y - xy$
- Drastic: $x \wedge_D y = x$ if $y = 1$, y if $x = 1$, and 0 otherwise; $x \vee_D y = x$ if $y = 0$, y if $x = 0$, and 1 otherwise.

Although only Sugeno and Choquet fuzzy integrals are used routinely in many applications, they belong to the class of the so-called *fuzzy t-conorm integrals*, which were introduced in [15]. In this section, we present the

framework of t-conorm fuzzy integral for simple functions, following [13, 14].

The individual types of fuzzy t-conorm integrals differ in the way how they bind together the spaces of integrand, measure, and integral. To formalize this, the spaces are linked together by the following definition.

Definition 10 *Let $\triangle, \perp, \underline{\perp}$ be continuous t-conorms, each of which is either Archimedean, or $\vee_S$. Let $([0,1], \triangle), ([0,1], \perp)$, and $([0,1], \underline{\perp})$ denote the spaces of values of integrand, measure, and integral, respectively. Let $\odot : ([0,1], \triangle) \times ([0,1], \perp) \to ([0,1], \underline{\perp})$ be a binary operation, non-decreasing in both variables, and satisfying the following:*

1. *$\odot$ is continuous on $(0,1]^2$,*
2. *$a \odot x = 0$ if and only if $a = 0$ or $x = 0$,*
3. *If $x \perp y < 1$, then $\forall a \in [0,1] : a \odot (x \perp y) = (a \odot x) \underline{\perp} (a \odot y)$,*
4. *If $a \triangle b < 1$, then $\forall x \in [0,1] : (a \triangle b) \odot x = (a \odot x) \underline{\perp} (b \odot x)$*

Then $\mathcal{F} = (\triangle, \perp, \underline{\perp}, \odot)$ is called a t-conorm system for integration*. If all the three t-conorms $\triangle, \perp, \underline{\perp}$ are Archimedean, $\mathcal{F}$ is then called* Archimedean*.*

Prior to defining the t-conorm integral, we have to define the *pseudo-difference*.

Definition 11 *Let $\triangle$ be a t-conorm. An operation $-_\triangle : [0,1]^2 \to [0,1]$, defined as*

$$a -_\triangle b = \inf\{c | b \triangle c \geq a\}, \tag{12}$$

is called pseudo-difference *of a and b with respect to $\triangle$.*

The pseudo-difference operation is dual to the residue $\Rightarrow_\wedge$, defined as $a \Rightarrow_\wedge b = \sup\{c | a \wedge c \leq b\}$, $\wedge$ being a t-norm.

Example 1 *If $\triangle = \vee_S$, then*

$$a -_{\vee_S} b = \begin{cases} a & \text{if } a \geq b \\ 0 & \text{if } a < b \end{cases} \tag{13}$$

Example 2 *If $\triangle = \vee_L$, then*

$$a -_{\vee_L} b = \max(a - b, 0) \tag{14}$$

Now we are ready to define the fuzzy t-conorm integral.

Definition 12 *Let μ be a fuzzy measure on $\mathcal{U}$, and let $\mathcal{F} = (\triangle, \perp, \underline{\perp}, \odot)$ be a t-conorm system for integration. Then the* fuzzy t-conorm integral *of a function $f : \mathcal{U} \to [0,1]$, $f(u_i) = f_i,\ i = 1, \ldots, r$, with respect to μ is defined as:*

$$\int_{\mathcal{F}} f \odot d\mu = \underset{i=1}{\overset{r}{\underline{\perp}}} [(f_{<i>} -_\triangle f_{<i-1>}) \odot \mu(A_{<i>})]. \tag{15}$$

4.1. Classification of t-conorm Systems

The framework of fuzzy t-conorm integral provides many different types of fuzzy integrals, depending on the t-conorm system for integration used. For example, if we choose $\mathcal{F} = (\vee_S, \vee_S, \vee_S, \wedge_S)$, we get the Sugeno integral, while for $\mathcal{F} = (\vee_L, \vee_L, \vee_L, \cdot)$, where $\cdot$ is the ordinary multiplication, we get the Choquet integral.

However, not all combinations of t-conorms give t-conorm systems for integration, and moreover, not all t-conorm systems for integration provide useful approach to classifier aggregation. In this section, we show several classes of t-conorm systems, and we investigate each class in detail.

If the fuzzy integral of a function with respect to a fuzzy measure is understood as some kind of expected value of the integrand, then the spaces of the integrand $([0,1], \triangle)$ and the integral $([0,1], \underline{\perp})$ should be the same [13–15]. Since this is the case in classifier aggregation, we restrict ourselves to the t-conorm systems for integration with $\triangle = \underline{\perp}$.

One important class of t-conorm systems are Archimedean systems, i.e. $\mathcal{F} = (\triangle, \perp, \underline{\perp}, \odot)$, where all the t-conorms are Archimedean. As shown in [14], if $\mathcal{F}$ is Archimedean, the corresponding fuzzy integral can be expressed as a Choquet integral with transformed domains, and thus it has nearly the same properties as the Choquet integral. More precisely, let $h_\triangle, h_\perp, h_{\underline{\perp}}$ denote the generators of $\triangle, \perp, \underline{\perp}$, respectively, then the following holds:

$$\int_{\mathcal{F}} f \odot d\mu = h_{\underline{\perp}}^{-1} \left[\min\left(h_{\underline{\perp}}(1), \int_C h_\triangle \circ f \mathrm{d}(h_\perp \circ \mu) \right) \right], \tag{16}$$

where $\circ$ denotes function composition. Therefore, Archimenean t-conorm systems bring nothing new to classifier aggregation, since they reduce to the Choquet integral.

The class of non-Archimedean t-conorm systems (i.e. at least one of the t-conorms is $\vee_S$, and the rest is Archi-

medean) is not as large as it may seem, because of the following proposition.

Proposition 1 *Let $\mathcal{F} = (\triangle, \perp, \underline{\perp}, \odot)$ be a non-Archimedean t-conorm system for integration, such that $\triangle = \underline{\perp}$, and $\odot$ is not constant in the second argument on (0,1). Then $\triangle = \perp = \underline{\perp} = \vee_S$.*

Proof: Since $\mathcal{F}$ is not Archimedean, i.e. at least one of $\triangle, \perp, \underline{\perp}$ is $\vee_S$ and the rest is Archimedean, and $\triangle = \underline{\perp}$, there are only two situations possible:

- $\mathcal{F} = (\triangle, \vee_S, \triangle, \odot)$, where $\triangle$ is $\vee_S$, or continuous Archimedean t-conorm. Suppose that $\triangle \neq \vee_S$, i.e. it is continuous and Archimedean. According to Req. 3 from Def. 10, when $x \vee_S y < 1$, then $\forall a \in [0,1]$

$$a \odot (x \vee_S y) = (a \odot x) \triangle (a \odot y). \tag{17}$$

  Let $x = y < 1$. Then (17) reduces to

$$a \odot x = (a \odot x) \triangle (a \odot x). \tag{18}$$

  This conflicts with the fact that $\triangle$ is Archimedean, i.e. $u \triangle u > u \; \forall u \in (0,1)$.

- $\mathcal{F} = (\vee_S, \perp, \vee_S, \odot)$, where $\perp$ is $\vee_S$, or continuous Archimedean t-conorm. Suppose that $\perp \neq \vee_S$, i.e. it is continuous and Archimedean. According to Req. 3 from Def. 10, when $x \perp y < 1$, then $\forall a \in [0,1]$

$$a \odot (x \perp y) = (a \odot x) \vee_S (a \odot y). \tag{19}$$

  Let $x, y > 0$, $y = x$, such that $x \perp x < 1$. Then (19) reduces to

$$a \odot (x \perp x) = a \odot x, \tag{20}$$

  and since $\odot$ is non-decreasing in both arguments and $\perp$ is Archimedean this implies that $\odot$ is constant in the second argument on $[x, x \perp x]$.

  Now let $d_\perp : [0,1] \to [0,1]$, defined as $d_\perp(m) = m \perp m$ be the diagonal of the t-conorm $\perp$. Because $\perp$ is continuous, $d_\perp$ is also continuous, and moreover $d_\perp(0) = 0, d_\perp(1) = 1$. We define:

$$\mathcal{M} = d_\perp^{-1}(0,1) = \{m > 0 | m \perp m < 1\}.$$

  $\mathcal{M}$ is an open interval, and $\inf \mathcal{M} = 0, \sup \mathcal{M} = s$ such that $s \perp s = 1$.

  According to (20), $\forall m \in \mathcal{M}$, $\odot$ is constant in the second argument on $I_m = [m, m \perp m]$. Let $I = \bigcup_{m \in \mathcal{M}} I_m = (0,1)$. Summarizing the former, the following holds:

$$\forall \xi \in (0,1) \; \exists \eta \in (\xi, 1), \text{ such that } a \odot \xi = a \odot \eta.$$

  The following lemma shows that this is in conflict with the fact that $\odot$ is not constant on $(0,1)$, which means $\perp = \vee_S$.

In both cases $\triangle = \perp = \underline{\perp} = \vee_S$, which proves the proposition. ■

Lemma 1 *Let $f : [0,1] \to [0,1]$ be continuous and non-decreasing on $(0,1)$. Let*

$$\forall \xi \in (0,1) \; \exists \eta \in (\xi, 1) \text{ such that } f(\xi) = f(\eta).$$

Then f is constant on $(0,1)$.

Proof: We will prove this theorem by contradiction. Assume to the contrary that f is not constant on $(0,1)$, i.e. there exist x_1, x_2, $0 < x_1 < x_2 < 1$ such that $f(x_1) < f(x_2)$ (because f is non-decreasing). Now we define:

$$A = \{x \in (0,1) | f(x) = f(x_1)\},$$

and let $\overline{x} = \sup A$. There are two situations possible:

- $\overline{x} < 1$. Therefore there exists $\eta \in (\overline{x}, 1)$ such that $f(\eta) = f(\overline{x})$. Because f is continuous on $(0,1)$, $f(\overline{x}) = f(x_1)$, which means $\eta \in A$, which contradicts $\overline{x} = \sup A$.
- $\overline{x} = 1$. Therefore there exists $x_3 \in [x_2, 1)$ such that $x_3 \in A$, i.e. $f(x_3) = f(x_1)$. Because $f(x_3) \geq f(x_2)$, i.e. $f(x_1) \geq f(x_2)$, which contradicts $f(x_1) < f(x_2)$.

■

We could theoretically still construct t-conorm systems for integration of the form $\mathcal{F} = (\vee_S, \perp, \vee_S, \odot)$, with $\odot$ constant in the second argument (because then Eq. (20) holds), i.e., in fact, integrals with no respect to the measure. However, this is not very useful in practice.

Fuzzy t-conorm systems of the form $\mathcal{F} = (\vee_S, \vee_S, \vee_S, \odot)$ are called $\vee_S$*-type* systems. The following proposition implies that for $\vee_S$-type systems, Req. 3 and 4 from Def. 10 are satisfied automatically.

Proposition 2 *Let $\odot : [0,1] \times [0,1] \to [0,1]$ be a non-decreasing operation satisfying requirements 1 and 2 from Def. 10. Then $\mathcal{F} = (\vee_S, \vee_S, \vee_S, \odot)$ is a t-conorm system for integration.*

Proof: We have to prove the Req. 3 and 4 from Def. 10. Since the proof of the latter is analogous to the proof of the former, we will prove only Req. 3, i.e.: when $x \vee_S y < 1$, then $\forall a \in [0,1] : a \odot (x \vee_S y) = (a \odot x) \vee_S (a \odot y)$. Without loss of generality, we can assume that $x \leq y$. This implies $x \vee_S y = y < 1$, and $a \odot (x \vee_S y) = a \odot y$. Since $\odot$ is non-decreasing, $(a \odot x) \leq (a \odot y)$, thus $(a \odot x) \vee_S (a \odot y) = a \odot y$, which proves the proposition. ∎

Among $\vee_S$-type systems, *quasi-Sugeno* systems, for which $\odot$ is a t-norm (this t-norm will be further denoted by $\wedge$), play an important role. However, due to the following proposition, not all t-norms can be used:

Proposition 3 *$\mathcal{F} = (\vee_S, \vee_S, \vee_S, \wedge)$, where $\wedge$ is a t-norm, is a t-conorm system for integration if and only if $\wedge$ is continuous on $(0,1]^2$ and without zero divisors.*

Proof: Recall that an element $a \in (0,1)$ is called a zero divisor of a t-norm $\wedge$ if there exists some $x \in (0,1)$, such that $a \wedge x = 0$. The implication $\Rightarrow$ is trivial. The other implication can be proved using Prop. 2 (with taking in mind that $a \wedge 0 = 0$ for any t-norm $\wedge$). ∎

For example the Łukasiewicz or the drastic t-norms have zero divisors and hence cannot be used in quasi-Sugeno systems.

We can summarize the results to create the following classification of t-conorm systems for integration:

1. Archimedean systems – can be expressed using Choquet integral, with the Choquet integral as a special case.
2. Non-Archimedean systems
   (a) Systems with $\triangle = \underline{\perp}$, which lead to $\vee_S$-type systems, with quasi-Sugeno systems as a special case (and the Sugeno integral in particular).
   (b) Systems with $\triangle \neq \underline{\perp}$, which do not express the expected value of the integrand.

5. Experiments

To find out how the individual fuzzy t-conorm integrals perform in practice, we created a Random Forest classifier system [19], and we aggregated the system vith different fuzzy t-conorm integrals. As the theoretical results from the previous section show, the only meaningful t-conorm systems are the Choquet integral, Sugeno integral, and the $\vee$-type systems, from which we used the quasi-Sugeno integrals. To obtain different fuzzy integrals from the quasi-Sugeno family, we used the t-conorm system $\mathcal{F} = (\vee_S, \vee_S, \vee_S, \wedge_{AA})$, where $\wedge_{AA}$ is the Aczél-Alsina t-norm [20], defined as:

$$x \wedge_\alpha^{AA} y = \begin{cases} x \wedge_D y & \text{if } \alpha = 0 \\ x \wedge_S y & \text{if } \alpha = \infty \\ e^{-[(-\log x)^\alpha + (-\log y)^\alpha]^{1/\alpha}} & \text{if } \alpha \in (0, \infty) \end{cases} \tag{21}$$

With increasing α, the Aczél-Alsina t-norm is continuously increasing, varying from drastic t-norm ($\alpha = 0$), through product t-norm ($\alpha = 1$), to standard t-norm ($\alpha = \infty$).

In the experiment, we compared the performance of the following classifier systems:

- single best [SB] classifier – for reference purposes, representing the "worst case" scenario; result of the classifier in the team with the lowest validation error rate
- Choquet integral [CI]
- Sugeno integral [SI]
- quasi-Sugeno integral with Aczél-Alsina t-norm [qSI AA]
- oracle [OR] – representing the theoretical "best-case" scenario, which, for a given pattern, gives correct class prediction if and only if any of the classifiers in the team gives correct class prediction

We measured the mean value and the standard deviation of the error rate of the classifier systems from 10-fold crossvalidation on 7 artificial and 20 real-world datasets with varying size, dimensionality, and class count. The methods were implemented in the Java programming language. The size of the ensemble (number of trees) was set to r = 20, the number of features to explore in each node varied between 2 and 5 (depending on the dimensionality of the particular dataset), the maximal size of a leaf was set to 10 (see [19] for description of the parameters). As the confidence measure, we used the ELA method (2). The number of neighbors was set to $k = 5$, $k = 10$, or $k = 20$, depending on the size of the dataset. The values of the parameters were set based on some preliminary testing, no optimization or fine-tuning was done. The optimal value of α for a given dataset was chosen from the values $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.2, 1.5, 2, 3\}$ on an independent validation set.

Table 1: Mean error rate ± standard deviation of the methods from 10-fold crossvalidation. SB – single best, CI – Choquet integral, SI – Sugeno integral, qSI AA – quasi-Sugeno integral with Aczél-Alsina t-norm, OR – Oracle. The best method (excluding OR) for each dataset is displayed in boldface.

Dataset	ref	size	classes	dim	SB (non-combined)	CI	SI	qSI AA α_{opt}	qSI AA	OR
Artificial										
clouds	[21]	5000	2	2	14.9 ± 1.8	$\mathbf{12.4 \pm 1.9}$	12.6 ± 1.6	0.3	13.1 ± 1.5	1.6 ± 1.0
concentric	[21]	2500	2	2	5.25 ± 1.5	3.9 ± 1.4	4.0 ± 1.4	1.2	$\mathbf{3.5 \pm 0.9}$	0.0 ± 0.0
gauss 3D	[21]	5000	2	3	30.1 ± 4.0	24.6 ± 2.3	24.3 ± 2.6	0.3	$\mathbf{24.2 \pm 2.8}$	1.5 ± 0.4
gauss 8D	[21]	5000	2	8	25.6 ± 1.8	$\mathbf{14.7 \pm 1.5}$	15.2 ± 2.1	0.6	$\mathbf{14.7 \pm 1.8}$	0.1 ± 0.1
ringnorm	[22]	3000	2	20	$\mathbf{9.4 \pm 1.7}$	13.4 ± 1.3	13.6 ± 2.2	3.0	12.3 ± 2.0	0.1 ± 0.2
twonorm	[22]	3000	2	20	18.5 ± 1.7	$\mathbf{7.9 \pm 2.0}$	8.3 ± 2.2	0.3	$\mathbf{7.9 \pm 2.0}$	0.0 ± 0.1
waveform	[22]	5000	3	21	28.25 ± 1.5	$\mathbf{18.0 \pm 2.5}$	19.3 ± 1.7	2.0	18.2 ± 1.9	0.2 ± 0.2
Real-world										
balance	[22]	625	3	4	20.3 ± 5.1	13.9 ± 4.0	13.7 ± 3.7	1.5	$\mathbf{13.6 \pm 3.5}$	2.7 ± 1.8
breast	[22]	699	2	9	5.6 ± 2.7	3.0 ± 2.0	$\mathbf{2.9 \pm 2.0}$	0.7	3.8 ± 2.0	0.7 ± 1.4
glass	[22]	214	7	9	22.8 ± 12.5	$\mathbf{20.7 \pm 8.7}$	24.9 ± 10.8	0.7	22.7 ± 8.3	0.0 ± 0.0
iris	[22]	150	3	4	9.3 ± 7.8	$\mathbf{8.0 \pm 5.2}$	9.3 ± 4.6	0.5	9.3 ± 8.4	0.7 ± 2.1
letter-recg.	[22]	20000	26	16	22.4 ± 1.3	$\mathbf{8.2 \pm 0.7}$	8.6 ± 0.5	1.0	8.6 ± 0.6	0.3 ± 0.2
pendigits	[22]	10992	10	16	8.2 ± 0.9	3.6 ± 0.6	3.5 ± 0.8	0.5	$\mathbf{3.3 \pm 0.5}$	0.1 ± 0.1
phoneme	[21]	5427	2	5	18.3 ± 1.0	$\mathbf{13.1 \pm 1.5}$	13.2 ± 1.8	0.5	13.2 ± 1.7	0.5 ± 0.3
pima	[22]	768	2	8	30.3 ± 5.0	$\mathbf{25.7 \pm 3.6}$	25.9 ± 3.8	0.5	26.0 ± 6.1	0.5 ± 1.2
poker	[22]	4828	3	10	49.9 ± 1.6	46.9 ± 2.3	$\mathbf{46.6 \pm 3.0}$	0.5	47.9 ± 2.1	3.8 ± 1.4
satimage	[21]	6435	6	4	19.0 ± 1.8	$\mathbf{15.7 \pm 1.6}$	15.9 ± 2.0	0.9	16.2 ± 2.1	2.9 ± 0.6
segmentation	[22]	2310	7	16	13.8 ± 2.4	$\mathbf{8.6 \pm 1.8}$	9.5 ± 1.9	1.5	9.2 ± 2.2	0.5 ± 0.4
sonar	[22]	208	2	60	32.3 ± 11.8	23.5 ± 11.0	$\mathbf{22.3 \pm 9.6}$	1.5	22.6 ± 9.4	0.0 ± 0.0
texture	[21]	5500	11	10	14.1 ± 2.6	$\mathbf{5.2 \pm 1.0}$	5.4 ± 0.5	3.0	$\mathbf{5.2 \pm 0.8}$	0.0 ± 0.0
transfusion	[22]	748	2	4	23.4 ± 3.6	$\mathbf{23.1 \pm 5.4}$	25.0 ± 5.7	0.5	23.5 ± 3.2	6.1 ± 2.8
vehicle	[22]	946	4	18	35.6 ± 5.3	28.1 ± 3.7	$\mathbf{27.1 \pm 3.9}$	0.4	27.7 ± 4.9	0.7 ± 0.8
vowel	[22]	990	11	10	42.7 ± 4.0	$\mathbf{14.8 \pm 2.3}$	18.7 ± 4.7	0.9	18.9 ± 6.3	0.0 ± 0.0
wine	[22]	178	3	13	9.6 ± 7.0	8.4 ± 7.9	6.7 ± 6.8	2.0	$\mathbf{5.6 \pm 5.3}$	0.0 ± 0.0
wineq-red	[22]	1600	3	11	40.8 ± 4.5	$\mathbf{31.3 \pm 3.8}$	31.4 ± 2.0	1.2	31.7 ± 5.6	0.5 ± 0.7
wineq-white	[22]	4898	3	11	44.7 ± 2.0	$\mathbf{32.6 \pm 2.8}$	33.6 ± 2.2	1.2	33.7 ± 2.7	0.7 ± 0.4
yeast	[22]	1484	4	8	46.3 ± 5.8	39.5 ± 4.7	38.2 ± 4.7	0.9	$\mathbf{37.8 \pm 2.8}$	3.1 ± 1.7

The results of the testing are shown in Tables 1 and 2. Table 1 contains the raw error rates, while Table 2 compares the different aggregation methods, showing the number of datasets, for which a particular aggregation method was at least as good as another aggregation method.

Table 2: Comparison of the individual aggregation operators. The (i,j)-th cell of the table shows the number of datasets (out of 27), on which aggregator i obtained better (or the same) results than aggregator j. The last column shows the number of datasets on which the aggregator was the best.

↓ superior to →	SB	CI	SI	qSI AA	all
SB	-	1	4	3	1
CI	26	-	18	18	16
SI	24	9	-	13	4
qSI AA	25	12	17	-	9

The results of the testing show that there is a big improvement in the classification error rate between the SB classifier, and all the fuzzy integrals. On the other hand, the performance of the particular fuzzy integrals is similar. The Choquet integral is usually better than the (quasi) Sugeno integral. The quasi-Sugeno integral is slightly better than the Sugeno integral (which is not surprising, because the parameter α was fine-tuned for the particular data). Still, there is a huge theoretical gap between any of the individual methods and the Oracle classifier system.

6. Conclusion

In this paper, we have shown the framework of fuzzy t-conorm integral, which includes both Choquet and Sugeno fuzzy integrals, and we have shown the way how it can be used in dynamic classifier combining. We have theoretically shown that the family of fuzzy t-conorm integrals is not as rich as it may seem, and that the only reasonable fuzzy t-conorm integrals are the Sugeno integral, the Choquet integral, and fuzzy t-conorm integrals using the $\vee_S$-type systems (with quasi-Sugeno integral as their representative).

In the experimental section, we have compared the performance of the Choquet integral, the Sugeno integral and the quasi-Sugeno integral on 27 datasets. The results indicate that even if the differences of the methods are small, the Choquet integral usually obtains the best results.

References

[1] L.I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[2] L. Rokach, "Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography," *Comput. Stat. Data Anal.*, vol. 53, no. 12, pp. 4046–4072, 2009.

[3] D. Ruta and B. Gabrys, "An overview of classifier fusion methods," *Computing and Information Systems*, vol. 7, pp. 1–10, 2000.

[4] G. Giacinto and F. Roli, "Dynamic classifier selection based on multiple classifier behaviour," *Pattern Recognition*, vol. 34, no. 9, pp. 1879–1881, 2001.

[5] M. Aksela, "Comparison of classifier selection methods for improving committee performance.," in *Multiple Classifier Systems*, pp. 84–93, 2003.

[6] K. Woods, J.W. Philip Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 405–410, 1997.

[7] X. Zhu, X. Wu, and Y. Yang, "Dynamic classifier selection for effective mining from noisy data streams," in *ICDM '04: Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)*, (Washington, DC, USA), pp. 305–312, IEEE Computer Society, 2004.

[8] L.I. Kuncheva, J.C. Bezdek, and R.P.W. Duin, "Decision templates for multiple classifier fusion: an experimental comparison.," *Pattern Recognition*, vol. 34, no. 2, pp. 299–314, 2001.

[9] L.I. Kuncheva, "Fuzzy versus nonfuzzy in combining classifiers designed by boosting," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 6, pp. 729–741, 2003.

[10] D. Štefka and M. Holeňa, "Dynamic classifier systems and their applications to random forest ensembles," in *Proceedings of the ICANNGA 2009 Ninth International Conference on Adaptive and Natural Computing Algorithms, Kuopio, Finland*, vol. 5495 of *Lecture Notes in Computer Science*, p. 458–468, Springer, 2009.

[11] A.H.R. Ko, R. Sabourin, and A.S. Britto, Jr., "From dynamic classifier selection to dynamic ensemble selection," *Pattern Recogn.*, vol. 41, no. 5, pp. 1718–1731, 2008.

[12] A. Tsymbal, M. Pechenizkiy, and P. Cunningham, "Dynamic integration with random forests.," in *ECML* (J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, eds.), vol. 4212 of *Lecture Notes in Computer Science*, pp. 801–808, Springer, 2006.

[13] V. Torra and Y. Narukawa, *Modeling Decisions: Information Fusion and Aggregation Operators*. Springer, 2007.

[14] M. Grabisch and H.T. Nguyen, *Fundamentals of Uncertainty Calculi with Applications to Fuzzy Inference*. Norwell, MA, USA: Kluwer Academic Publishers, 1994.

[15] T. Murofushi and M. Sugeno, "Fuzzy t-conorm integral with respect to fuzzy measures: Generalization of Sugeno integral and Choquet integral," *Fuzzy Sets and Systems*, vol. 42, no. 1, pp. 57–71, 1991.

[16] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[17] D.J. Hand, *Construction and Assessment of Classification Rules*. Wiley, 1997.

[18] S.J. Delany, P. Cunningham, D. Doyle, and A. Zamolotskikh, "Generating estimates of classification confidence for a case-based spam filter," in *Case-Based Reasoning, Research and Development, 6th Int. Conf., ICCBR 2005, Chicago, USA* (H. Muñoz-Avila and F. Ricci, eds.), vol. 3620 of *LNCS*, pp. 177–190, Springer, 2005.

[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[20] E.P. Klement, R. Mesiar, and E. Pap, *Triangular Norms*. Kluwer Academic Publishers, 2000.

[21] UCL MLG, "Elena database," 1995. http://www.dice.ucl.ac.be/mlg/?page=Elena.

[22] C.B. D.J. Newman, S. Hettich and C. Merz, "UCI repository of machine learning databases," 1998. http://www.ics.uci.edu/ mlearn/MLRepository.html.

Ontology Matching in the Context of Web Services Composition

Post-Graduate Student:
ING. PAVEL TYL

Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ

Faculty of Mechatronics, Informatics and Interdisciplinary Studies
Technical University of Liberec
Hálkova 6
461 17 Liberec 1, CZ

pavel.tyl@tul.cz

Supervisor:
ING. JÚLIUS ŠTULLER, CSC.

Institute of Computer Science of the ASCR, v.v.i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ

stuller@cs.cas.cz

Field of Study:
Technical Cybernetics

This project is partly realized under the state subsidy of the Czech Republic within the research and development project "Advanced Remediation Technologies and Processes Center" 1M0554 – Programme of Research Centers PP2-DP01 supported by the Ministry of Education, under the financial support of the ESF and the state budget of the Czech Republic within the research project CZ.1.07/2.2.00/07.0008 – ESF OP EC "Intelligent Multimedia E-Learning Portal" and cofinanced from the student grant SGS 2010/7821 "Interactive Mechatronics Systems Using the Cybernetics Principles".

Abstract

Web services became one of the best means for web application interoperability. There is a need to have a scalable and extensible model to deliver distributed information and functionality integrated as independently provided, interoperable services in a distributed environment. Several distributed services can be dynamically composed (chained) as a new service to accomplish specific tasks. Such a model of service composition (chaining) is one of the most important research topics of next generation web services.

This paper discusses possibilities of using ontology matching techniques for web services interoperability and composition, describes such processes, explain their difficulties and propose a model for web service composition based on suitable ontology matching techniques.

1. Motivation

Let's suppose this motivation scenario:
We want to deliver some electronic product from a web shop to some address by a shipping service. Online electronic shop service provides its output description in some ontology. Shipping service uses a second ontology for its input description. Then the matching of these ontologies could be used for:

- checking that what is delivered by the first service, e. g., a `DVD_Player`, matches what is expected by the second one, e. g., some `Object` (shipping service does not accept life animals),
- verifying preconditions of the second service, e. g., `Size` in centimeters, etc.

We can see only two parts of a chain in this short example, but there could be many more. For example there are web services able to compare products (e. g., `DVD_Players`) from different data sources (catalogues), some web services do it even more sophisticated using user preferences, etc.

2. Introduction

2.1. Ontology matching

Ontology matching is the process of finding "correspondences" (also called relationships [3]) between elements within different ontologies which have to be (semantically) compared and, eventually, joined. The output of the matching process is a set of such correspondences between two (or, in general, more ontologies) called an *ontology alignment*. The "oriented" version of an ontology alignment is an *ontology mapping*.

Given two source ontologies o and o', an input ("preliminary") alignment A, a set of *parameters* (e. g.,

threshold) and *resources* (e. g., provenance metadata), the **matching process** (see Fig. 1) can be described by function *f* returning a *new alignment* A' between ontologies o and o':

$$A' = f(o, o', A, p, r).$$

Ontology matching is in most cases performed *manually* or *semiautomatically*, often with support of some *graphical user interface*. Manual specification of ontology parts for matching is *time consuming* and moreover *error prone process*. It results in a strong need for development of faster and/or less laborious methods, which can process ontologies at least semiautomatically.

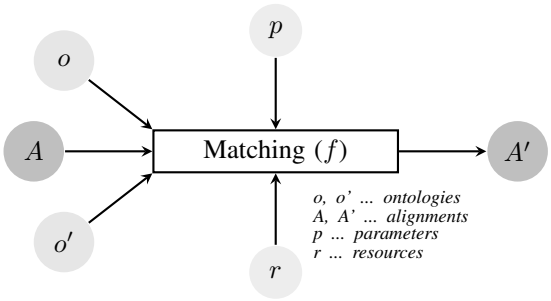


Figure 1: Schema of a matching process [3].

2.2. Web services

A web service is a network accessible interface to web application functionality. It is described in machine-readable format, most often in standardized web service description language, WSDL [15]. Way of communication between other computers and web service is specified in the web service's description with the help of Simple Object Access Protocol, SOAP [12]. SOAP messages are transfered by well-established protocols[1]. SOAP and WSDL protocols have easy machine-readable XML [17] syntax. Both SOAP and WSDL were designed to be independent on selected version of XML language, but obligated to be XML compatible.

The W3C [13] defines a web service as "a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically Web Services Description Language WSDL). Other systems interact with the web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other web-related standards" [14].

Web services expose their interfaces to the web so that users (agents) can invoke them. **Semantic web services** provide a richer and more precise way to describe services through the use of knowledge representation languages and ontologies [4], e. g., OWL-S [11] or WSDL-S [16].

2.3. Web service composition

Web service discovery and integration is the process of finding web service able to deliver a particular service and composing several services in order to achieve a particular goal [8].

Web services are often designed to be independent and replaceable and, therefore, web service processors are able to incorporate new services in their workflows and customers can dynamically choose new and more promising services. For that purpose, they must be able to compare the descriptions of these services (in order to know if they are really relevant) and to route the knowledge they process (in order to compose different services) by routing the output of some service to the input of another service.

Both for finding the appropriate service and for interfacing services, some data "mediator" is important as a bridge between different vocabularies [9]. Based on the correspondences between the terms of the descriptions, mediators must be able to translate the output of one service into a suitable input for another service (see Fig. 2).

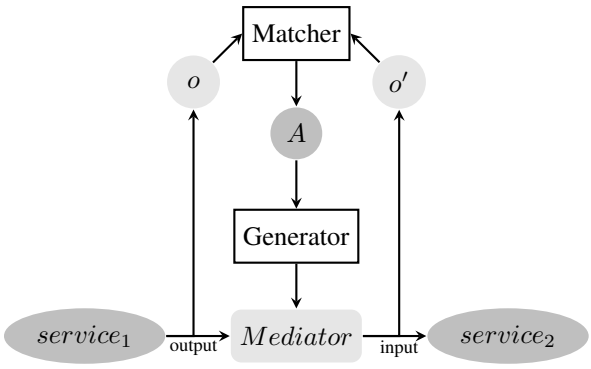


Figure 2: Web service composition.
In this case it is useful to (1) match relevant parts of ontologies o and o', thus resulting in alignment A and (2) $generate$ a $mediator$ between $service_1$ and $service_2$ in order to enable transformation of actual data [3].

The headstone of a mediator definition is an alignment between two ontologies. And this can be provided through matching the corresponding ontologies either offline when someone is designing a preliminary service

[1] Reason for SOAP messages "encapsulation" is an absence of trust from existing systems.

composition, or online (dynamically) [5], when new services are searched for completing a request.

3. Web service composition by ontology matching

There are two possibilities one could use ontology matching techniques for web service composition:

1. Entire web service is described by the service ontology (WSMO[2]).
2. Web service is described by the traditional means and only its inputs and outputs are described by the ontologies.

Ontologies classifying and describing services are called service ontologies. According to our opinion it is not necessary to describe web services by ontologies (i. e., using WSML[3] [5]), because inner behavior of a web service *need not* (sometimes rather *must not*) be always transparent or accessible. But at least web services' outputs and inputs have to be described using ontologies for successful application of ontology matching techniques.

Every web service has its input(s) and output(s), in our case described as input ontologies $o(in\_s_1)$ and $o(in\_s_2)$ and output ontologies $o(out\_s_1)$ and $o(out\_s_2)$ (see Fig. 3). They can be part of the *web service I/O interface* or can be stored outside the web service itself.

Web service in our model is devided into two main parts – its *internal structure* and a *repository*. The *internal structure* is responsible for functional achievment of the exposed service, finding direct or intermediate answers. If the web service is able to provide a direct answer (reply to the *primary request*), the input ontology $o(in\_s_1)$ is processed in the *internal structure* and results are transferred to the output ontology $o(out\_s_1)$. In the case of intermediate answer, if the *web service 1* is compliant to be a part of a chain, $o(out\_s_1)$ is produced and devolved upon a *web service chain repository* with a goal of searching for the best available web service for the chain, so that its appropriate alignment (e. g., A_{12}) is in the *alignments repository*.

When a preliminary alignment A_{12} exists (provided manually or by (semi)automatic means), it should be stored in the *alignments repository* for simple identification of reuse opportunities (see Sec. 3.1). Input/output ontologies $o(out\_s_1)$ and $o(in\_s_2)$, alignments $A_{12}, A_{13}, \ldots, A_{1n}$ and saved readymade *web service chains* are in the web service repository identified by their URIs[4]. It allows interaction with other services in order to negotiate operations the current service just cannot provide (e. g., when the current service is not available). Therefore at least the alignments (or the whole repository) should be always exposed in the same way as the inputs and the outputs of the web service.

If the suitable service (e. g., *web service 2*) is found and preliminary alignment exists, $o(out\_s_1)$ and $o(in\_s_2)$ are checked for their compatibility in a *compatibility checker* and if they pass, $o(out\_s_1)$ is easily converted in an *I/O converter* by using stored alignment A_{12} into $o(in\_s_2)$ and the request is passed on. Successful conversion and checks should be stored and cached.

If there is no alignment related to *web service 2* in our *alignments repository*, traditional matching methods (*Matcher* in Fig. 3) or manual matching have to be used. If related alignment exists, we can successfully apply an alignment reuse methods (see Sec. 3.1). According to [7] there are four matchmaking functions based on which web services can be chained: *Exact*, *PlugIn*, *Subsume* and *Intersection*. Otherwise (*Disjoint*), services are incompatible:

- **Exact** – if the output parameter out_s_y of s_y and the input parameter in_s_x of s_x are equivalent concepts (e. g., `DVD_Player` from our motivation example could be certainly delivered, because it is an `Object` and its `Size` is less than maximal allowed),
- **PlugIn** – if out_s_y is a subconcept of in_s_x (e. g., `DVD_Player` will be be delivered, if a shipping service is able to deliver whatever we want, any `owl:Thing`),
- **Subsume** – if out_s_y is a superconcept of in_s_x,
- **Intersection** – if the intersection of out_s_y and in_s_x is satisfiable,
- **Disjoint** – if the out_s_y and in_s_x are incompatible.

With *Exact* and *PlugIn* functions we are always able to match required web services, the matcher can fail in case of *Subsume* and *Intersection*.

[2] Web Service Modelling Ontology – http://www.wsmo.org.
[3] Web Service Modelling Language.
[4] Uniform Resource Identifier.

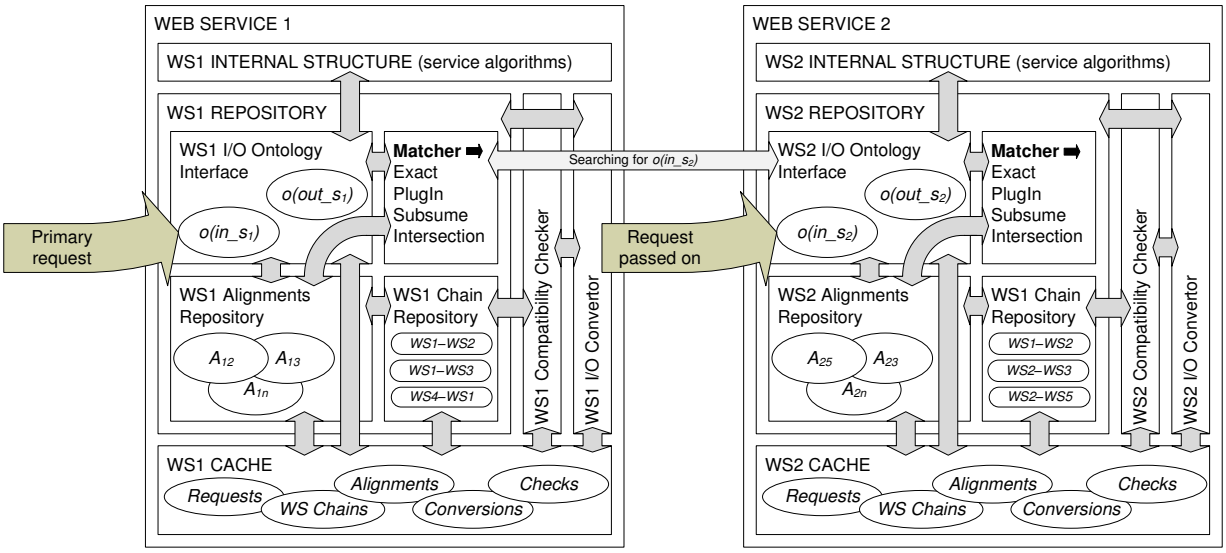


Figure 3: Proposed model for the web service composition with using of an ontology matching.

3.1. Candidate matching techniques for web service composition

It would be nice if we could always automatically create input and output ontology alignments at runtime. But it is not an easy task in the case of heterogenous web service compositions. In addition such algorithms should be fast enough, there is no time for tuning parameters, manual corrections, etc. Therefore in our model we suppose at least preliminary ontology alignments of $o(out\_s_y)$ and $o(in\_s_x)$ at design time. Consequently we can always take an advantage of them. And at this moment an alignment reuse can come on scene.

Alignment reuse is motivated by an idea that many ontologies that should be matched are similar to already matched ones, especially if they describe the same domain(s). Ontologies from the same application domain usually contain many similar elements, typical for this domain. Therefore their mappings can provide good reusable candidates.

At first, matching problems are decomposed, then a set of ontology fragments is generated and finally previous match results can be applied at the level of ontology fragments rather than at the level of the whole ontologies [3]. According to [2] alignments of ontologies (e. g., o_y and o_x) can be saved in a repository in three possible forms:

- **Direct mappings** ($o_y \leftrightarrow o_x$) – ideal for reuse, one or multiple mappings are already available for the given match problem. Such mappings represent the shortest possible mapping paths, which do not involve any intermediary ontologies.
- **Complete mapping paths** ($o_y \leftrightarrow o_i$, $o_i \leftrightarrow o_x$ or $o_j \leftrightarrow o_y$, $o_x \leftrightarrow o_j$) – such mapping paths consist only of existing mappings.
- **Incomplete mapping paths** (same as complete, but i. e., $o_i \leftrightarrow o_x$ and $o_j \leftrightarrow o_y$ are to be matched) – the default match operation is first applied and missing alignments can be computed with less effort than directly matching the input ontologies.

All match results are compared (e. g., average similarity in the path, expected computational effort expressed by the path length, etc.) and ontologies composed.

Although alignment reuse seems to be the most important technique in the proposed model (more than technique we could call it matching strategy), there are some other basic techniques that cannot be omitted in the web service composition. In the following we list some of them together with the reason for their use:

- **Internal structure (constraint) based techniques** – before creating an ontology alignment, but much more for later use, we can do a verification of criteria as the set of the entity properties (e. g., their multiplicity), the range and the domain of the properties, cardinality, datatypes, etc. These techniques are easy to implement and if the ontologies ($o(out\_s_y)$ versus $o(in\_s_x)$) pass them, it

will provide a basis on which other parts of an application can rely.

- **External ontologies based techniques** – external reference ontology (o_{ext}) can provide a common ground on which an alignment can be based. It can help in the case of disambiguity of multiple possible meanings of terms in given domain of interest. For example an alignment between $o(out\_s_y)$ and $o(in\_s_x)$ can be derived from two other alignments with external ontology ($A(o(out\_s_y), o_{ext}$ and $A(o(in\_s_x), o_{ext})$[5]. External ontology is in the most cases a general reputable upper-level ontology (e. g., FMA[6] in medicine).
- Further we can use **relational structure techniques** (e. g., taxonomy relations), **propositional** and **description logic techniques** (these techniques cannot find an alignment alone, but when alignment is generated, we can ensure its completeness and consistency), etc.

4. Potential problems of web services composition

Issues worth mentioning to deal with when composing web services are:

Third-party sources – Two things remain unchanged with the web services in general:

- they use third-party sources and
- they have questionable reliability.

This is not to say that web services are unreliable, but it simply means that we have not a primary control of the source for our web application. When our sources are offline, our web service or web application is also offline. One way to avoid this problem is to keep an actual cache of all queries issued to our data sources in case of a service failure.
Caching is a good idea in general because it will definitely speedup repeated requests.

Rate limiting – Many public service interfaces may have to limit the number of requests an application or user can make within certain period of time. (This can be done by tracking the number of requests made by a single IP address or the system may require authentication.) This is another issue that could be partially or fully solved by request caching. Fig. 4 shows the position of web services among other web applications. As can be seen, they are supposed to be dynamic, but in contrast to P2P systems they should stay always correct.

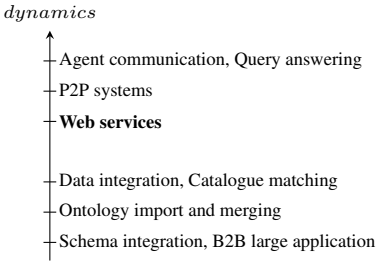


Figure 4: Example applications ordered by their *dynamics*. Space under semantic web services shows that three top applications are considered to be dynamic.

Reliability – Keeping the current cache of recent requests can help keeping our service online until our sources are back online. If more than one public service interface is available to provide the information, our composite web service requires then a fallback mechanism to be implemented. It allows our web service to switch to another source until our primary source has been reestablished or to find another reliable (data) source forever.

Vendor locking – This could be a huge problem in the future as more and more web service compositions will be created. What to do if a public application interface that serves alignments to thousands of web services and web applications suddenly goes offline for one day or forever or starts charging for their service? It is therefore necessary to share accessible sources or prepare mechanisms for rapid finding of other appropriate services.

Licensing restrictions – Some public web services restrict for what we can use them and sometimes which web services can we use together with them. We have to thoroughly read restrictions which may apply before adopting them to our web service chain [7]. This problem is similar to the above one and has the same possible solutions.

5. Conclusion and future work

In this paper the possibility of using ontology matching techniques in web service composition is presented and a complex model for such composition is proposed. There are many different applications which require or could take an advantage of ontology matching. But in

[5] Here we can omit the input alignment, the resource and the parameters.
[6] Foundational Model of Anatomy – http://sig.biostr.washington.edu/projects/fm.
[7] One more disadvantage is that these restrictions or rules can be time invariant and practically stochastic.

the comparison with traditional applications such as information or schema integration, web service composition has its specific requirements – after preliminary steps (creating and processing alignments) it should be automatic and dynamic enough. Therefore we have to store these alignments and find the way how to replace them if necessary.

The next step I would like to work on is a design and implementation of an application according to the proposed model that will be able to compose e-learning systems for advanced testing (could be seen as web services) with the help of ontology matching (or, in general, ontology integration) techniques.

References

[1] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, "Web Services. Concepts, Architectures and Applications". Springer, Berlin, 2004. ISBN 3-540-44008-9.

[2] H. Do, "Schema matching and mapping-based data integration". PhD thesis, University of Leipzig, Leipzig, DE, 2005.

[3] J. Euzenat and P. Shvaiko, "Ontology Matching". Springer-Verlag, Berlin/Heidelberg, 2007. ISBN 978-3-540-49611-3.

[4] D. Fensel, H. Lausen, A. Polleres, J. de Bruijn, M. Stollberg, D. Roman, and J. Domungue, "Enabling semantic web services: the web service modelling ontology". Springer, Heidelberg, 2004. ISBN 978-3-540-34519-0.

[5] F. Guinchiglia, F. McNeil, and M. Yatskevich, "Web service composition via semantic matching of interaction specifications". Technical Report DIT-06-080, University of Trento, 2006.

[6] D. McCandless, L. Obrst, and S. Hawthorne, "Dynamic Web Service Assembly Using OWL and a Theorem Prover". In Proc. 3^{rd} IEEE International Conference on Semantic Computing, Berkeley, USA, 2009.

[7] F. Lécué, A. Delteil, and A. Léger, "Applying Abduction in Semantic Web Service Composition". In Proc. 2007 IEEE International Conference on Web Services, p. 94–101, IEEE CS, 2007.

[8] M. Paolucci, T. Kawamura, T. Payne, and K. Sycara, "Semantic matching of web services capabilities". In Proc. 1^{st} International Semantic Web Conference (ISWC), volume 2342 of LNCS. p. 333–347, Chia Laguna, IT, 2002.

[9] D. Roman, H. Lausen, and U. Keller, "Web service modeling ontology standard (WSMO standard)". Working Draft D2v0.2, WSMO, 2004.

[10] OWL – Web Ontology Language / W3C Semantic Web Activity [online]: http://www.w3.org/2004/OWL.

[11] OWL-S – Semantic Markup for Web Services [online]: http://www.w3.org/Submission/OWL-S.

[12] SOAP – Simple Object Access Protocol [online]: http://www.w3.org/TR/soap.

[13] W3C – World Wide Web Consortium [online]: http://www.w3.org

[14] W3C – Web Services Glossary [online]: http://www.w3.org/TR/ws-gloss

[15] WSDL – Web Services Description Language [online]: http://www.w3.org/TR/wsdl.

[16] WSDL-S – Web Service Semantics [online]: http://www.w3.org/Submission/WSDL-S.

[17] XML – Extensible Markup Language / W3C XML Activity [online]: http://www.w3.org/XML.

Použití klasifikačních a nomenklaturních systémů v Katalogu klinických doporučených postupů

doktorand:
MUDR. MIROSLAV ZVOLSKÝ

Oddělení medicínské informatiky
Ústav informatiky AV ČR, v. v. i.
Pod Vodárenskou věží 2

182 07 Praha 8

zvolsky@euromise.cz

školitel:
DOC. ING. ARNOŠT VESELÝ, CSC.

Oddělení medicínské informatiky
Ústav informatiky AV ČR, v. v. i.
Pod Vodárenskou věží 2

182 07 Praha 8

vesely@pef.czu.cz

obor studia:
Biomedicínská informatika

Článek vznikl s podporou projektu 1M06014 MŠMT ČR.

Abstrakt

Katalog klinických doporučených postupů shromažďuje podrobné informace o v České republice platných dokumentech klinických doporučených postupů. Pro třídění a vyhledávání relevantních dokumentů a také pro budoucí snadné propojení s elektronickým zdravotním záznamem v navazujících systémech podpory rozhodování využívá mimo jiné klasifikační systémy MKN-10 a MeSH. V zahraničních i domácích publikacích se však vyskytují také termíny kódované dalšími klasifikačními a nomenklaturními systémy, možností je také pokrytí všech použitých klinických termínů jedním komplexním systémem.

1. Úvod

Klinické doporučené postupy (KDP) jsou sytematicky vyvíjené dokumenty napomáhající osobám zainteresovaným v léčebném procesu v rozhodnutí o adekvátním postupu péče. KDP popisují určenou část procesu zdravotnické péče, ať se jedná o diagnostiku, terapii, či prevenci jednotlivého onemocnění (nebo jejich skupin). Někdy bývají tyto dokumenty pojednány jako shrnutí komplexní problematiky skupin onemocnění, popis diagnostického nebo terapeutického výkonu, ale také například normami pro technologické zázemí či nutné administrativní procesy vyskytující se v procesu léčebné péče [1, 2].

Katalog klinických doporučených postupů (KKDP) je projekt shromažďující informace o publikovaných dokumentech klinických doporučených postupů v České republice v elektronické formě a dostupných prostřednictvím Internetu. [3] Jako databáze a aplikační rozhraní je vyvíjen od roku 2007 a je průběžně přizpůsobován zjištěnému aktuálnímu stavu publikační aktivity v dané oblasti. U každého evidovaného dokumentu je možné uložit přes třicet parametrů definujících vlastnosti dokumentu nejen z hlediska snadného zatřídění a vyhledávání v katalogu, ale také kvality obsahu (viz Tabulka 1). Dále vyvíjená verze 2.0 KKDP je volně přístupná na adrese **http://neo.euromise.cz/kkdp** .

KKDP má dvě různá webová rozhraní. Jedním je veřejně přístupný portál shromažďující seznam veřejně přístupných informací o dokumentech KDP publikovaných v prostředí Internetu a funkce vyhledávání. Druhým rozhraním je speciální aplikace pro editory systému s omezeným a jasně definovaným a zabezpečeným přístupem, která umožňuje procházení i veřejně nepřístupných záznamů, úpravu všech záznamů, správu databáze a další administrativní funkce. Samostatně použitelným prvkem je vlastní databáze, která může být použita pro další aplikace či projekty. V budoucnu se počítá s využitím nashromážděných informací jako repozitáře a zdroje dat pro klinické systémy podpory rozhodování či vytvořením volně přístupného internetového aplikačního rozhraní.

Protože KKDP může sloužit v budoucnu také jako zdroj informací pro vyvíjené systémy pro podporu rozhodování, je třeba počítat s návazností na klinické informační systémy a elektronický zdravotní zaznam.

Nezbytnou součástí informace o každém dokumentu v KKDP je určení relevantních klinických termínů, kterými se dokument zabývá. Kromě možnosti výčtu souvisejících termínů ve slovní poznámce k dokumentu je v KKDP k dispozici také definice klíčových slov ke každému dokumentu. Tato klíčová slova jsou ovšem použita pouze při fulltextovém vyhledávání v rámci webového rozhraní KKDP - pokud některé důležité

klíčové slovo nebo jeho synonymum není uvedeno přímo v názvu dokumentu, lze ho vyplnit do tohoto parametru. Jednotlivá klíčová slova jsou oddělena čárkami. Vyplnění tohoto parametru není povinné.

Interní kód
Název dokumentu
Plný název tištěné verze/publikace
Odborná společnost/autorita
Autoři
Kontaktní informace
Poznámka k autorům
Aktuálnost dokumentu
Datum vydání/zahájení platnosti dokumentu
Datum ukončení platnosti dokumentu
Jazyk dokumentu
Kategorie dokumentu
Související léčebná péče
Síla použitých důkazů
MKN-10
MeSH
DRG
Cílová odbornost dokumentu
Související odbornosti
Cílová populace
Abstrakt
Poznámka
Klíčová slova
Geografické určení
Název odkazu/externího umístění
Velikost odkazovaného dokumentu
Formát odkazovaného dokumentu
Formalizovaná verze dokumentu
Poznámka k formalizaci
Autoři formalizace
Datum formalizace
Související dokumenty
Identifikace a datum založení záznamu o dokumentu
Identifikace a datum poslední změny záznamu o dok.

Tabulka 1: Tabulka parametrů každého katalogizovaného dokumentu v KKDP ve verzi 2.0

2. Metodika

2.1. Klasifikační a nomenklaturní systémy a tezaury

Pro sjednocení a standardizaci klinických termínů vznikly ve zdravotnictví mnohé klasifikační a nomenklaturní systémy. Jejich pomocí je možné jednoznačně definovat pojmenování jednotlivých klinických termínů, hierarchicky je zařadit a definovat jejich vztahy [4].

V českém prostředí je nejrozšířenější používání Mezinárodní klasifikace nemocí v desáté revizi (MKN-10), což je klasifikační a kódovací systém zaměřený především na diagnostiku onemocnění a jejich původce. Anglická resp. mezinárodní verze International Classification of Diseases (ICD-10) je garantována Světovou zdravotnickou organizací. Český překlad je rozšířen v klinické praxi pro kódování diagnóz a spravován Ústavem zdravotnických informací a statistiky ČR [5,6].

Především pro kódování klinických výkonů je používán v českých zdravotnických zařízeních klasifikační a kódovací systém Diagnosis-related Group (DRG) s nadřazenými kategoriemi označovanými jako Major Diagnostic Categories (MDC). V českém prostředí je používána lokalizovaná verze systému, název klasifikace se nepřekládá. Použití systému má usnadnit rozhodovací procesy především v ekonomické části zdravotnictví a poskytování zdravotní péče a mimo jiné usnadnit porovnávání zdravotnických zařízení či efektivity jednotlivých druhů zdravotnických výkonů navzájem [7].

V oblasti odborných lékařských publikací a především periodik je široce mezinárodně rozšířen medicínský oborový slovník a kódovací systém MeSH (Medical Subject Headings) vyvíjený a udržovaný Národní lékařskou knihovnou USA - v anglicky psaných dokumentech jeho originální anglická verze, v některých českých publikacích a periodicích pak český překlad [8].

V jednotlivých oborech jsou dále používány oborově úzce zaměřené klasifikační systémy jako například TNM nebo ICPC. V zahraničí jsou rozšířené např. systémy LOINC nebo SNOMED, které ovšem nemají prozatím českou lokalizaci, naopak český systém Národní číselník laboratorních položek (NČLP jako součást DASTA - Datový standard Ministerstva zdravotnictví) není přímo navázán na mezinárodní systémy.

2.2. Použití v zahraničních katalozích

Zahraniční portály a katalogy věnující se problematice klinických doporučených postupů využívají nejčastěji systém MeSH nebo vlastní třídění do skupin nejčastěji podle druhu v textu popisovaného onemocnění (viz Tabulka 2). Velkou výhodou systému MeSH je jeho mnohaleté používání v zahraniční literatuře a periodikách (kde jsou především anglicky psané KDP publikovány) a tudíž také snadná návaznost klinických doporučení na další odbornou literaturu vymezenou konkrétními MeSH termíny.

Jednodušší katalogy KDP využívají vlastní členění podle kategorií odborností nebo jiné systematiky onemocnění, ve vyhledávání požadovaných dokumentů se pak spíše spoléhají na fulltextové vyhledávání zadaných slov či frází v názvu, případně abstraktu KDP.

Guidelines.gov	MeSH
Leitlinien.de	MeSH
National Library of Guidelines	klíčová slova, odbornost
Gidelines International Network (G-I-N)	MeSH
Scottish Intercollegiate Guidelines Network (SIGN)	”topics”
New Zealand Guidelines Group	”therapeutic category”
Clinical Practice Guidelines Portal	hierarchicky uspořádané skupiny onemocnění

Tabulka 2: Zahraniční katalogy klinických doporučených postupů a použité hlavní třídění dokumentů

3. Výsledky

Pro vyhledávání ve webovém rozhraní KKDP jsou použity kódy MKN-10 a kódy MeSH (nikoliv pouze deskriptory, které mají obecnější platnost a nezařazují termíny jednoznačně do hierarchie systému).

Kódy mezinárodní klasifikace nemocí v desáté revizi jsou jednak v českém zdravotnictví běžně používané a známé, jednak jsou v mnoha dokumentech KDP přímo zmíněny. Navíc je lze ve valné většině případů snadno dohledat ze zaměření textu - většina dokumentů KDP se týká jednotlivého onemocnění, výčtu nebo skupiny onemocnění.

Výhodou kódů MeSH je jednak jejich přímé použití v některých dokumentech (často například v překladech KDP s nadnárodní působností či publikovaných v angličtině), jednak snadné napojení na jinou tematicky zaměřenou odbornou literaturu.

V databázi a administračním rozhraní pro editory je prostor pro ukládání kódů dalších systémů a sice DRG/MDC a SNOMED CT. V případě DRG/MDC proto, že je v českém zdravotnictví již částečně používán a především se s ním počítá v Národní sadě standardů zdravotních služeb. Protože se uvažuje o vytvoření české lokalizace systému SNOMED CT a především pro jeho rozsah a komplexnost je KKDP připraven na zadávání kódů i tohoto systému, ačkoliv vyhledávání a zadávání klinických termínů by muselo probíhat prostřednictvím termínů v anglickém jazyce.

U každého kódu klasifikačního nebo nomenklaturního systému navázaného v KKDP jako parametr ke konkrétnímu dokumentu KDP je možné přiřadit relativní váhu a vymezit tak, jak silně se klinický termín dané problematiky týká (může být hlavním tématem, či jen okrajově zmíněn).

4. Diskuse

Kromě oborově medicínsky zaměřených klasifikačních a nomenklaturních systémů používá KKDP vlastní číselníky pro zdravotnické odbornosti (vychází z číselníku VZP a oborů specializačního vzdělávání ve zdravotnických profesích podle IPVZ), geografické a jazykové určení, formáty odkazovaných souborů a další. V ideálním případě by bylo možné některé z nich definovat na základě některé obecné a s medicínou provázané nomenklatury.

Přesné zadávání a případně aktualizace kódů klinických termínů užitých v KDP je zdlouhavý a náročný proces především v případech, kdy tyto termíny (nebo dokonce kódy) jsou zmíněny ve vlastním textu (a nikoliv s výhodou například v hlavičce dokumentu, jak tomu bývá v případě klíčových slov tezauru MeSH) nebo dokonce pouze v synonymech nebo opisech. Přesnost ručního zadávání editory je limitována a pravděpodobně neumožní plnohodnotné využití získaných údajů jako podklad pro systémy podpory rozhodování. Vhodnější cestou se jeví tvorba přesných datových modelů a ontologií pro jednotlivé KDP (nejlépe přímo ve spolupráci s autory dokumentů) a jejich případná budoucí integrace do KKDP.

Použití celkem čtyř různých klasifikačních a nomenklaturních systémů pro popis klinických termínů (2 pro vyhledávání a 2 prozatím pro sběr dat a jako možná alternativa do budoucna) zvyšuje složitost KKDP a možnost chyb - některé termíny se v různých systémech opakují. Jednotlivé systémy mají i své výhody. Alternativou je použití v budoucnu jen jednoho komplexního systému nebo metatezauru.

5. Závěr

V aktuálně vyvíjené verzi 2.0 Katalog klinických doporučených postupů využívá možnosti připojit ke každému dokumentu klinického doporučeného postupu klinické termíny v až čtyřech mezinárodně uznávaných klasifikacích. Jejich další používání bude záviset na rozšíření novějších z nich v klinické praxi. Zaběhnutým standardem zůstávají Mezinárodní klasifikace nemocí a Medical Subject Headings. Další méně systematickou možností pro upřesnění vyhledávání je vyjmenování klíčových slov v plnotextově prohledávaných parametrech.

Literatura

[1] M.J. Field and K.N. Lohr, (Eds.), "Guidelines for clinical practice: from development to use", *Institute of Medicine, Washington, D.C*, National Academy Press, 1992.

[2] R. Líčeník, "Klinické doporučené postupy", *Obecné zásady, principy tvorby a adaptace*, UP Olomouc, 2009.

[3] M. Zvolsky, "Katalog lékařských doporučených postupů v ČR", *Doktorandské dny '08*, MATFYZPRESS 2008.

[4] J. Zvárová a P. Přečková, "Medicínské klasifikační systémy a nomenklatury", http://www.cittadella.cz/euromise/sites/ File/ Zvarova/MP302/zvaraj02.pps.

[5] World Health Organization, "International Classification of Diseases ", http://www.who.int/classifications/icd/en/.

[6] Ústav zdravotnických informací a statistiky ČR, "Mezinárodní klasifikace nemocí", http://www.uzis.cz/cz/mkn/index.html.

[7] Národní referenční centrum, "DRG", http://www.nrc.cz/cs/drg.

[8] A. Šímová a L. Maixnerová, "Aktualizace českého překladu MeSH", *IKAROS*, Roč. 12, č. 5, 2008.

Ústav informatiky AV ČR, v. v. i.
DOKTORANDSKÉ DNY '10

Vydal
MATFYZPRESS
vydavatelství
Matematicko-fyzikální fakulty
Univerzity Karlovy
Sokolovská 83, 186 75 Praha 8
jako svou *– not yet –* . publikaci

Obálku navrhl František Hakl

Z předloh připravených v systému LaTeX
vytisklo Reprostředisko MFF UK
Sokolovská 83, 186 75 Praha 8

Vydání první

Praha 2010

ISBN *– not yet –*