ITRIA 2012 | INFORMATION TECHNOLOGIES: RESEARCH AND THEIR INTERDISCIPLINARY APPLICATIONS

ITRIA 2012

INFORMATION TECHNOLGIES: RESEARCH AND THEIR INTERDISCIPLINARY APPLICATIONS 2012

CONFERENCE PROCEEDINGS

INSTITUTE OF COMPUTER SCIENCE
POLISH ACADEMY OF SCIENCES

CONFERENCE PROCEEDINGS

INFORMATION TECHNOLOGIES: RESEARCH AND THEIR INTERDISCIPLINARY APPLICATIONS 2012

Warsaw, July 30, 2012.
ITRIA 2012

Institute of Computer Science Polish Academy of Sciences, Warsaw, 2013

Publication issued as a part of a project: 'Information technologies: research and their interdisciplinary applications', activity 4.1 of Human Capital Operational Programme.

Agreement number UDA-POKL.04.01.01-00-051/10-00

Publication is co-financed by European Union from resources of European Social Fund

Project leader: Institute of Computer Science, Polish Academy of Sciences

Project partners: System Research Institute, Polish Academy of Sciences
Nałęcz Institute of Biocybernetics and Biomedical Engineering, Polish Academy of Sciences

Reviewers: Krzysztof Ciesielski
Konrad Furmańczyk
Marek Gągolewski
Olgierd Hryniewicz
Jacek Koronacki
Juliusz Kulikowski
Małgorzata Marciniak
Jan Mielniczuk
Wojciech Penczek
Maciej Romaniuk
Maciej Szreter
Jacek Waniewski

Layout: Piotr Borkowski
Cover design: Waldemar Słonina

ISBN 978-83-63159-04-7
e-ISBN 978-83-63159-05-4

Preface

This volume contains selected papers from the first conference 'Information Technologies and their Interdisciplinary Applications', which took place in Warsaw on July, 28, 2012 and has been organized within the framework European Social Fund, supported by POKL.04.01.01-00-051/10-00. The speakers were students of the PhD programme bearing the same name as the conference. The programme is co-organized by three institutes of the Polish Academy of Sciences: Institute of Computer Science, System Research Institute and Institute of Biocybernetics and Biomedical Engineering. Selected speakers at the conference were invited to submit their extended contributions to the conference volume. The submissions were then reviewed by the editors and external reviewers. The volume comprises papers which have been finally accepted. The contributions deal with subjects which pertain to information processing and analysis, in particular data mining, mathematical and statistical modeling, natural language processing, and model checking.

"Modeling Consumer Decision Making in the Framework of Fuzzy Sets Theory" by Agnieszka Jastrzębska presents a descriptive model of consumer's decision making processes. The proposed model, based on the operators used in the theory of fuzzy sets, allows to model consumers who have different emotional and rational cognitive abilities.

"Sequential Monte Carlo and Bayesian methodology in the stochastic event reconstruction problems" by P. Kopka, A. Wawrzyńczak, and M. Borysiewicz addresses an important problem of the atmospheric contaminant source localization. The localization of the dangerous substance release is based only on the results of measurements of concentration of the released substance at the points where sensors are placed. The applied methodology combines the Bayesian methodology of estimation with Monte Carlo computer simulation methods.

"Prediction of the structure of the social network" by Robert Kłopotek presents a system for visualisation, exploration, and prediction of the characteristics of medium-size social networks. Beside a detailed description of the software, the experimental section includes some results of the predictive analysis of the three selected social networks, evolving in time (Enron, Autonomous, Hep-Ph).

The paper "Generalized autosort FFT framework" prepared by Michał Lenarczyk describes a concept of improving the fast Fourier Transform (FFT) algorithm and presents some new ideas in this direction developed by the author. This is not a new problem in signal processing and computer science, nevertheless, its practical importance follows from a large area of FFT applications in sciences and engineering.

"Simultaneous deleting and merging regressors for linear model selection" by Aleksandra Maj, Agnieszka Prochenka, and Piotr Pokarowski deals with an important problem of regression model choice when predictors may be quantitative or qualitative. For the latter case deletion or merging factorial levels may be needed. The authors present a new method and algorithm for doing this.

Piotr Przybyła's contribution entitled "Issues of Polish question answering" considers perspectives for building a Polish question answering system. To assess the applicability of existing solutions for English, their language dependence is discussed, preceded by an outline of distinctive features of Slavonic languages, especially Polish, and their impact on text processing. Some already existing question answering solutions for Polish are also discussed.

"Selection and prediction for linear models using Random Subspace Methods" by Paweł Teisseyre and Jan Mielniczuk transfers some methods applied in classification to regression modeling, namely those which use randomly sampled predictors to built small regression models and use specially designed overall performance measure of predictors for model choice and prediction. Special care is devoted to lessen computational cost of the method.

Agnieszka Zbrzezny's paper "Comparing SAT-based bounded model checking RTECTL and ECTL properties" compares two SAT-based bounded model checking algorithms for properties expressed in the existential fragment of the soft real time computation tree logic and in the existential fragment of the computation tree logic. To this aim two standard benchmarks are used: the faulty train controller system and the generic pipeline paradigm.

We wholeheartedly thank the authors of the papers included in this volume for their efforts to extend and revise their contributions. We are also grateful to the reviewers who devoted their time to go through submissions and to comment on them.

Editors:

Olgierd Hryniewicz
Jan Mielniczuk
Wojciech Penczek
and *Jacek Waniewski*

Table of contents

Modeling Decision Making with Respect to Consumer's Psychographical Portrait

Agnieszka Jastrzebska

[1] Institute of Computer Science, Polish Academy of Sciences
ul. J.K. Ordona 21 01-237 Warsaw, Poland
[2] Faculty of Mathematics and Information Science, Warsaw University of Technology
ul. Koszykowa 75, 00-662 Warsaw, Poland

Abstract

Decision making processes are driven by various needs. Needs determine our actions, directing us towards reducing motivational tension. This article presents developed consumer representation model. The focus is set on how, using chosen mathematical operators, we can model consumers decision making with different psychographical description. Developed approach to decision making modeling is versatile for consumers with both extremely emotional and extremely rational cognitive abilities. In the case study introduced is exemplar consumers segmentation and showed is that developed two-step procedure of obtaining the decision works.

Keywords: consumers choice theory, decision making, psychographical segmentation, triangular norms

1 Introduction

Research on mechanisms of motivational stimuli is in the scope of interest of many diverse sciences, including biology, artificial intelligence, economics and psychology. Understanding how human cognitive abilities are related to the decision making is crucial, as having appropriate knowledge would allow to control these processes. Analyzing preferences and human reactions to so called marketing communication are of interest in marketing and economics since many decades.

The objective of this paper is to present an interdisciplinary approach to modeling human behavior. Mathematical framework, which is applied is standard and well known. The originality of the idea lies in the fact that it is applied to a consumer representation model developed on the basis of two prominent theories: Maslow's need hierarchy and Lewin's field theory. Consumer representation model plus chosen operators used to describe human behavior allow to describe the whole spectrum of human decision making, including behavioral biases. Developed approach can be applied also to marketing communications research. In this article discussed is how developed model can be used to describe consumers with different psychographical portraits (a concept commonly applied in marketing). Presented are several hand-picked operators, which combined with our consumer representation model provide us with satisfactory results. Please note, that the

objective is not to describe applied, and at the same time commonly known operators, but to describe how consistent and coherent are psychological theories that lay at the background of our consumer representation model with recent marketing communications research, and how our approach can be applied to model human behavior. The originality of ideas presented in this paper lie in the fact that known theories were applied and joined together to form a consumer decision making model capable to reflect decision making process from the point of view of social sciences. Moreover, presented approach is compliant with decision making perspective present in marketing research.

The ground rule of successful marketing communication is to gather and process relevant data. As it turns out, drawing meaningful conclusions from gathered information might be aided by including psychophysical aspects of human behavior and analyzing, how it may participate in the decision making process. Hence, important questions arise: how to model consumer responses to prepared and directed (promotional) messages. Being able to understand how humans react to received information would allow to plan and execute successful promotional campaigns. In the article discussed is how consumers react at the most basic levels of motivational stimuli recognition and described is how a single decision might be taken. The explanation is supported with a model of multi-criteria consumer's decision making process based on our own consumer representation model. The article's perspective focuses on the decision making processes from the point of view of marketing communications. Described is psychographical consumer's segmentation technique. Presented is methodology of how positive preferences can be aggregated into the decision. The article is structured as follows: section 2 introduces the reader into the topic of consumer's decision making. In section 3 described is developed approach to consumer decision making modeling. In section 4 introduced is a case study, where the model is applied.

2 Link between marketing communications and decision making

Psychologists indicated that susceptibility to marketing communications depends on subject's personality. One of the most often used approach to the theoretical research on motivation and behavior concerns susceptibility to various forms of marketing. Recognizing mechanisms of the decision making and skilful targeting of particular groups of consumers brings revenue.

Most scientists from the area of motivational stimuli and personality research discuss dualistic divide between emotion and reason [1], [p. 2]. Naturally, there is unequal division between the impact of emotional selves and reasonable selves on one's behavior. Our personality determines, which side of nature influences one's actions. Compartmentalizing our character into these two parts led into the understanding of behavior as a reaction to certain set of motivational stimuli. Psychologists evaluate our actions according to one's rationality. According to this scale, on the one end there is extremely rational behavior, and on the other end extremely emotional behavior. Susceptibility to marketing communications can be evaluated in exactly the same manner.

Decision making process, according to marketing specialists, is divided into four steps:

1. Need's recognition.
2. Search for information.
3. Options comparison.
4. Decision, [9] [pp. 214 - 215].

My aim was to model a general decision making process, applicable for the microeconomic theory of consumer's choice, but also, which would be valid from the point of view of marketing communications. Each of these four steps is present in developed model. This article focuses on presenting compatibility of marketer's perspective on decision making with our model. Applied are mathematical operators, with different properties, which allow to compute correct decisions, without prior knowledge, which psychographical type of consumer we are dealing with. In further parts of this paper discussed is how, with the use of developed model involved can be causality. Described is a case study of 8 consumers with different psychographical segmentation. The reason, why it is important to investigate, how one's psychological portrait influences decision making is very important. It has been shown, that knowing target's rationality type helps in choosing right marketing communication's channels and messages. Specialists developed models of psychographical segmentation of consumers, that allow them to successfully plan promotional campaigns [9] [p. 213]. Traditional demographical and geographical segmentation is now enriched with these new aspects.

3 Developed model

In this section discussed is developed approach. Most importantly, showed is, how this model is able to represent decision making process accordant with 4-step definition given in section 2.

Decision making is a cognitive process, which is initiated by needs recognition. Consumer is able to name motivational stimuli influencing some decision in general. Number of factors influencing a decision may be infinite, but in practice we never have to analyze infinite amount of arguments. The so called cognitive simplification process usually takes place and consumer makes the decision based on few arguments, [12] [pp. 111 - 128]. Simplification of cognitive processes is a theme thoroughly and successfully researched by H. A. Simon, [11].

Consumer representation model was built on two prominent psychological theories: Maslow's needs theory and Lewin's field theory. For detailed description of these two please refer to: [6], [7] and [8]. Most important elements of these two theories, which were inspiration for this model are following:

- humans can be represented mathematically (K. Lewin),
- this representation has to describe individual's psychophysical field: all forces influencing one's behavior (K. Lewin),
- these forces (which are in up-to-date psychology called cognitive processes and their sources) determine our decisions (K. Lewin),

- on the most prime level forces influencing our decisions can be translated into needs (A. Maslow).

Combining these information to form a model that will represent a consumer in the context of a single decision problem we obtain a vector-based model in a following form:

$$V_A = [\ m_1,\ m_2,\ ...,\ m_n\ ]$$

where V_A is a vector of needs that concerns particular decision problem. Factors $m_1, ..., m_n$ represent all motivational stimuli that influence discussed decision.

As was mentioned before, human beings tend to simplify cognitive processes. Therefore, though we can name plenty needs, real-life decision making process is based on a few. In order to include causality in the decision we will take into account typically 2 vectors gathering motivational stimuli. First one will contain initial evaluations of factors influencing a particular decision. It will correspond to the situation when a person considers certain purchase for the first time. These initial evaluations might be corrected, when a consumer faces a particular product. In such case constructed is second vector that corrects evaluations gathered in the first vector. If the person is undecided and considers the same product several times, more vectors can be created, but typically we will analyze two vectors. We name these two vectors as premises (initial vector) and priorities (second vector). Detailed description on chosen approach is in [3] and [4].

Numbers contained in these vectors describes consumers attitude towards certain factors influencing the decision. Information gathered in premises and priorities vectors have uncertain character. As at this point my research on consumer behavior is at the initial state, in this article analyzed are only fuzzy sets, as a framework applied to describe consumer's attitude towards certain stimuli. We say that an element x belongs to the fuzzy set A with a degree $\mu(x)$ and denote it as the pair $(x, \mu(x))$. A fuzzy set A is defined as $A = \{(x, \mu(x)) : x \in X\}$. A fuzzy set A in the universe X is represented by the membership function $\mu_{A,X} : X \to [0, 1]$. Each element of premises and priorities vector describe how strongly given factor is influencing the consumer. Please note, that in future author plans to analyze also other models of uncertain information and their suitability for developed approach.

To sum up, each factor can be evaluated on the scale $[0, 1]$, where the greater is the value, the stronger influence has the particular factor on the decision. Such flexible scale of arguments evaluation allows to model real-life processes more accurately. To premises and priorities vectors applied is an operation of moderation. Functions, which may be applied are t-norms. Of course this is not an exclusive list. In this paper main focus is put on compatibility of presented model with decision making approach based on psychographical segmentation, not on description of functions. With the use of moderation procedure obtained is final vector, based on which performed is final decision aggregation. Again, there is plenty of aggregation operators discussed in literature, but for the purpose of model demonstration and due to space limitations in this article discussed are only t-conorms.

In order to compare how two different products satisfy one's needs it will be convenient to produce a single numeric value. This number should inform us how good is given product. In order to do so, at first included are premises, which gather general attitudes towards the decision. Then, premises are moderated with

priorities. In the frame of fuzzy logic employed can be, for example, lattice operations max and min for aggregation of information. Utilized can be also a generalized concept of the fuzzy sets connectives *union* and *intersection*. These are the mentioned before triangular norms, for short t-norms and t-conorms respectively, [5]. t-norm t and a t-conorm s are mappings $t, s : [0,1] \times [0,1] \to [0,1]$, which satisfy conditions:

- associative, i.e. $t(a, t(b,c)) = t(t(a,b),c)$, $s(a, s(b,c)) = s(s(a,b),c)$ for all $a, b, c \in [0,1]$,
- commutative, i.e. $t(a,b) = t(b,a)$, $s(a,b) = s(b,a)$ for all $a, b \in [0,1]$,
- monotonic, i.e. $t(a,b) \leqslant t(c,d)$, $s(a,b) \leqslant s(c,d)$ for all $a, b, c, d \in [0,1]$ such that $a \leqslant c$ and $b \leqslant d$,
- 1 is the neutral element of t-norm and 0 is the neutral element of t-conorm, i.e. $t(a,1) = a$, $s(a,0) = a$ for all $a \in [0,1]$, [5].

If t-norm t and t-conorm s satisfy the generalized De Morgan law $s(a,b) = 1 - t(1-a, 1-b)$, they are called dual triangular norms.

Apart from some popular triangular norms, applied will be a special type of triangular norms called strict (continuous and strictly monotone) generated by additive generators. The justification of suitability of strict norms has been already discussed in [3]. Strict norms are discussed to greater extent in [2].

In order to moderate premises with priorities used are t-norms. Applying a t-norm to correct general attitudes with specific preferences towards a particular product gives a vector of moderated values. Next, in order to be able to compare the extent to which different products satisfy the need, applied are t-conorms for aggregation. The decision ranges from 0 to 1. The stronger is the result, the better this particular product fulfills consumer's needs. This article is focused only on mappings from $[0,1]$ to $[0,1]$ interval. Bipolar information processing is out of scope of this article.

In the case study applied are following t-conorms:

- max,
- probabilistic sum,
- bounded sum,
- Nilpotent maximum,
- Einstein sum,
- t-conorm generated with arc sinus function,
- t-conorm generated with tangent (argument is multiplied by $\frac{\pi}{2}$),
- t-conorm generated with arc tangent hyperbolic.

Moderation is performed with t-norms dual to t-conorms listed above. Please note, that some of chosen t-conorms have the property of saturation. Different t-conorms have different tempo of saturation. Thanks to this, cognitive simplification process can be modeled.

Chosen were several popular operators. There are also other functions, which can be applied in this model. The objective of this paper is to show compatibility of presented model with marketing communications research and highlight the benefit of a 2-step procedure in this context.

In next section presented is a case study, where applied is described procedure.

4 Case Study

In this case study illustrated is how different consumer behaviors (analyzed in accordance with psychographical distinction known in marketing studies) might be reflected with aggregation techniques described in section 3. Case study revolves around 8 consumers facing a decision regarding a purchase of a TV. Case study is based on three different types of consumer profiles. Customers are intentionally grouped and categorized, according to common set of features, in this case according to psychographical description. Distinguished are following consumer profiles:

1. Emotional Type (EE) - profile of a person, who behaves in unexpected way. He might change his mind quickly and inconsistently with previous opinions. EE's priorities substantially differ from premises.
2. Rational Type (RR) - profile of a person, who is partially susceptible to communications. RR is not changing his mind substantially and immediately.
3. Insusceptible Type (SS) - profile of a person, who is not susceptible to stimuli arising during the phase of priorities evaluation. Insusceptible person is not changing his opinions. His premises are exactly the same as priorities.

Suggested three profiles correspond to the standard categories, describing one's susceptibility to marketing communications. It is often repeated, that susceptibility, meaning openness, sensitivity or impressionability is a feature, for which marketers fight for, [13] [p. 78]. In future author plans also to research different methodological approaches to consumers segmentation.

To be able to compare models described in section 3, used are the same arguments as premises and as priorities for all 8 consumers. Evaluations present in vectors are individual, depending on person's characteristics. Discussed are three consumers with rational attitude towards decision making, three insusceptible (or in other words stubborn) and two consumers with emotional type of personality.

For the purpose of this case study, discussed are following 5 premises (and priorities) influencing the purchase of a TV.

1. Consumer is entertained while watching TV.
2. He needs the TV for educational reasons.
3. One thinks that the TV is a nice furniture-like gadget and fits to his interiors.
4. The TV has a Wi-Fi module.
5. The TV comes with extra equipment, like stand-alone speakers.

These are common criteria, which are analyzed in the context of the decision. The character of these arguments is varied. Part of them should be more carefully

analyzed, when discussing a general decision regarding the purchase of a TV. Some are more reasonable to consider, when (after) consumer faces a particular product. In the next subsection discussed are individual premises and priorities vectors for rational, stubborn and emotional types of personalities.

4.1 Consumers' vectors description

The main interest of this case study is to investigate various values of premises and priorities and the influence of this variety on the decision. Consumer's profiles are accordant with mentioned segmentation and we will see that developed approach provides results coherent with what marketing studies show. The case study is based on positive premises only. Evaluation of premises and priorities is performed separately for each consumer profile. Vectors of premises are denoted as P_{XX}, where XX is the name of consumer (case), to which this vector belongs. Vectors of priorities are named R_{XX}.

First described are vectors of premises and priorities characterizing insusceptible kind of consumers. Distinguished are three cases. First case (SS1) describes a person, who is highly convinced that he needs the TV. All evaluations of premises and priorities are equal to 0.8. Second (SS2) describes a person, who is weakly convinced, that he needs the TV. In this case all evaluations are equal to 0.2. Third example (SS3) concerns a situation, when one motivational stimuli is evaluated as very high (equal to 0.9) and all other arguments are weak. For an insusceptible person, the evaluations of all priorities are exactly the same as evaluations of premises. Below present are six vectors describing consumers SS1, SS2 and SS3.

$$P_{SS1} = [\ 0.8,\ 0.8,\ 0.8,\ 0.8,\ 0.8\ ]$$

$$R_{SS1} = [\ 0.8,\ 0.8,\ 0.8,\ 0.8,\ 0.8\ ]$$

$$P_{SS2} = [\ 0.2,\ 0.2,\ 0.2,\ 0.2,\ 0.2\ ]$$

$$R_{SS2} = [\ 0.2,\ 0.2,\ 0.2,\ 0.2,\ 0.2\ ]$$

$$P_{SS3} = [\ 0.9,\ 0.2,\ 0.2,\ 0.2,\ 0.2\ ]$$

$$R_{SS3} = [\ 0.9,\ 0.2,\ 0.2,\ 0.2,\ 0.2\ ]$$

Next, presented are vectors of premises and priorities for rational consumers. Distinguished are 3 cases: RR1, RR2 and RR3.

$$P_{RR1} = [\ 0.8,\ 0.8,\ 0.8,\ 0.8,\ 0.8\ ]$$

$$R_{RR1} = [\ 0.9,\ 0.9,\ 0.9,\ 0.9,\ 0.9\ ]$$

$$P_{RR2} = [\ 0.2,\ 0.2,\ 0.2,\ 0.2,\ 0.2\ ]$$

$$R_{RR2} = [\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1\ ]$$

$$P_{RR3} = [\ 0.9,\ 0.2,\ 0.2,\ 0.2,\ 0.2\ ]$$

$$R_{RR3} = [\ 1.0,\ 0.1,\ 0.1,\ 0.1,\ 0.1\ ]$$

In the case of rational consumers, priorities do not significantly differ from premises. They might be slightly strengthened or weakened. In the case of RR1, consumer has all strong premises and slightly strengthened priorities. Consumer RR2 has all premises weak (equal to 0.2) and even weaker priorities (equal to 0.1). RR3's vector of premises contains first one strong argument and all other weak. His priorities vector contains one argument evaluated as 1.0 (he is certain that the product satisfies his needs in this criterion). All other priorities are weak, equal to 0.1.

Next, described are cases of emotional consumers (EE1 and EE2). Their behavior is very difficult to predict. Priorities vectors contain evaluations significantly different from premises evaluations. Such person is most likely to change his mind rapidly. Below visible are vectors P_{EE1}, R_{EE1}, P_{EE2} and R_{EE2}.

$$P_{EE1} = [\ 0.8,\ 0.8,\ 0.2,\ 0.2,\ 0.2\ ]$$

$$R_{EE1} = [\ 0.2,\ 0.2,\ 0.8,\ 0.8,\ 0.8\ ]$$

$$P_{EE2} = [\ 1.0,\ 0.2,\ 0.2,\ 0.2,\ 0.2\ ]$$

$$R_{EE2} = [\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1\ ]$$

Next section discusses differences between results of the decision making procedure.

4.2 Results interpretation

Decision making process can be simplified to the evaluation of a single set of arguments - premises influencing given decision. In this case, we do not perform operation of moderation. Including second set of arguments evaluations - priorities vectors allows to compute results, which are more compliant with intuition. Table 1 gathers results of decision aggregation based only on premises. Table's headers contain information, which t-conorm was used to calculate particular decision.

Omitting second set of input arguments provided us with outputs based on general attitudes gathered in premises vectors. Decisions calculated for consumers with all high evaluations of premises are strong positive. It is not a surprise. This is the case of consumers SS1 and RR1. Also, when a consumer expresses at least one very strong positive attitude, saturation happens fast and computed decisions are strong positive. These are cases of RR3, SS3, EE1 and EE2. When premises are moderated with priorities and then aggregated results should be different. Change should be especially visible for EE1 and EE2, who have very volatile preferences. In cases of SS2 and RR2 (both these consumers had all premises evaluated as weak: 0.2) chosen operators provided varied decisions. Results vary from 0.2 (maximum and Nilpotent maximum) to 0.88 (arc sinus). We see that weak arguments get strengthened. This phenomena is explainable with behavioral bias, described in prospect theory.

In prospect theory explained is that preferences - or attitudes can be perceived "[...]as a function of decision weights, and it assumes that these weights do not always correspond to probabilities" [10][p. 98]. Kahneman and Tversky - authors

TABLE 1: Decisions obtained for consumers RR1, RR2, RR3, SS1, SS2, SS3, EE1 and EE2 based on premises only

consumer	maximum	probabilistic sum	bounded sum	Nilpotent maximum	Einstein sum	arc sinus	tangent	arc tangent hyperbolic
SS1	0.8	1.0	1.0	1.0	1.0	1.0	0.96	1.0
SS2	0.2	0.67	1.0	0.2	0.77	0.88	0.65	0.77
SS3	0.9	0.96	1.0	1.0	0.98	0.98	0.92	0.98
RR1	0.8	1.0	1.0	1.0	1.0	1.00	0.96	1.0
RR2	0.2	0.67	1.0	0.2	0.77	0.88	0.65	0.77
RR3	0.9	0.96	1.0	1.0	0.98	0.98	0.92	0.98
EE1	0.8	0.98	1.0	1.0	0.99	0.99	0.91	0.99
EE2	1.0	1.0	1.0	1.0	1.0	0.8	1.0	0.67

of the prospect theory, postulate that people tend to overweight small probabilities and underweight moderate and high probabilities [14][p. 179]. Translating prospect theory into developed model, means that weak evaluations of premises and priorities in the process of arguments aggregation should get strengthened. This happens for most of t-conorms, which are applied, with the exception of maximum and Nilpotent maximum. The question, which operator is most suited to model decision making, is a difficult one. Nevertheless, choosing the right function would require taking a closer look into how it aggregates small arguments. The optimal choice shall include described bias.

Next, presented are decisions obtained with a 2-step procedure, when first premises are moderated with priorities, and then the decision is aggregated. Table 2 contains results for consumers SS1, SS2, SS3, RR1, RR2, RR3, EE1 and EE2. Table's headers inform about t-conorm applied for aggregation. For moderation used were dual t-norms.

Decisions obtained for consumers with insusceptible type of personality are consistent with intuition. For cases SS1 and SS3 computed were strong positive decisions. These cases are consistent with previous results, were priorities were neglected. In a case, when a person was convinced that the purchase of the TV was practically a necessity, according to at least one argument (case SS3), the output is also strong positive. The same solid positive output concerns a case, when all arguments are fairly strong (case SS1). In case of SS2, where all premises and priorities are equal to 0.2, decisions obtained with various dual triangular norms are fairly weak. Some pairs of operators strengthen the decision, but the strength-

Table 2: Decisions obtained for consumers RR1, RR2, RR3, SS1, SS2, SS3, EE1 and EE2 based on premises and priorities

consumer	maximum	probabilistic sum	bounded sum	Nilpotent maximum	Einstein sum	arc sinus	tangent	arc tangent hyperbolic
SS1	0.8	0.99	1	1	1	1	0.92	1
SS2	0.2	0.18	0	0	0.51	0.24	0.43	0.12
SS3	0.9	0.84	0.8	0.9	0.92	0.81	0.84	0.83
RR1	0.8	1	1	1	1	1	0.94	1
RR2	0.1	0.1	0	0	0.34	0.16	0.31	0.06
RR3	0.9	0.91	0.9	0.9	0.94	0.85	0.91	0.91
EE1	0.2	0.58	0	0	0.75	0.74	0.7	0.6
EE2	0.1	0.17	0.1	0.1	0.37	0.31	0.59	0.15

ening effect is significantly lower. For SS2 the effect of output strengthening, when premises are moderated with priorities, is much lower than in the case, when the decision is obtained with the use of premises vector only. This is the first solid proof that involving causality is extremely important and modeling decision making has to include time flow and possibility of change of preferences. It is accordant with intuition - facing a concrete product changes consumer's perspective, not only about particular features associated with given product. The change concerns also evaluations of needs (factors). In table 2 we see relatively higher strengthening effect for t-norm/t-conorm associated with tangent generating function and for Hamacher product/Einstein sum. These pairs allow to involve behavioral bias described in the prospect theory. Some operators behave differently. For example, Lukasiewicz t-norm/bounded sum and Nilpotent minimum/Nilpotent maximum pairs of dual triangular norms lower the decision. The output in these two cases is equal to 0.

Decisions calculated for consumers, who express rational attitude towards preferences evaluation are visible in 4th, 5th and 6th rows of table 2. Rational consumer may change his mind and priorities evaluations might be different from premises. They can be either lowered - like in case RR2, or strengthened - like for RR1. Also they can be partially lowered and partially strengthened - the case of RR3. Naturally, rational consumers are also susceptible to behavioral biases. Decisions obtained for RR1, who has all premises and priorities relatively high are also very high. These results are consistent with intuition. Decisions for RR2, whose all premises are weak and priorities are even weaker are all weak too (see row 5, table 2). In the case of RR3, when one strong factor was evaluated in R_{RR3} as a certain positive (equal to 1.0) all decisions obtained using chosen dual triangular

norms are strong positive. In the cases of RR1 and RR3 results obtained based only on premises are consistent with results obtained, when applied were both premises and priorities vectors. For RR2 results obtained without moderation are too high.

Observing results for EE1 and EE2 obtained with premises moderated with priorities differ from results obtained before (without moderation). Especially for EE2 we see the lowering effect of moderation. Though P_{EE2} contains one premise evaluated as certain positive (1.0), significantly weaker priorities lower the decision. This is highly desired property. Without taking into account priorities, we neglect the disappointment of EE2 expressed with low numbers in R_{EE2}. For the case of EE1, we see mostly moderate positive decisions. This output is also compliant with intuition - all arguments in P_{EE1} and R_{EE1} are positive.

To sum up, in the article presented was a procedure of how decision making can be modeled based on developed consumer representation scheme. In the case study introduced was consumers segmentation and it was shown that described approach produces results coherent with results assumed in marketing communication studies.

5 Conclusions

The article presented developed approach to decision making modeling. Shown were benefits of involving causality into decision making process. Investigated were several pairs of operators, with which it was possible to compute appropriate outputs. Case study examples show, that applied operators allow to return correct decisions for various consumers profiles. Technique of premises and priorities moderation and decision aggregation allows to build a model applicable for consumers, who match different psychographical description without prior knowledge about it. Choosing certain operators allows also to include various spectrum of human behavior, including behavioral biases. Presented model of consumer's representation is very versatile, as it was built with psychological models on the background. Interdisciplinarity: joining psychology with behavioral economics into one model of consumer representation is the main advantage of presented approach.

Acknowledgment

The research is supported by National Science Center, grant No 2011/01/B/ST6/06478, and by research fellowship within "Information technologies: research and their interdisciplinary applications", agreement No POKL.04.01.01-00-051/10-00.

References

[1] Frijda N. H., Manstead A.S.R., Bem S., *The influence of emotions on beliefs*, in: Emotions and Beliefs. How Feelings Influence Thoughts, ed. N. H. Frijda, A.S.R. Manstead, S. Bem, Cambridge University Press, 2000.

[2] Homenda W., *Balanced Fuzzy Sets*, Information Sciences 176 (2006) 2467 - 2506, 2006.

[3] Homenda W., Jastrzebska A., *Modeling Consumer's Choice Theory: Using Fuzzy Sets and their Generalizations*, Proc. of the 2012 IEEE World Congress on Computational Intelligence (IEEE WCCI 2012), Brisbane, 2012, pp. 1633-1640.

[4] Jastrzebska A., Homenda W., *Modeling Consumer Decision Making Process with Triangular Norms*, in: Lecture Notes in Computer Science 7564, 2012, pp. 382-394.

[5] Klement E. P., Mesiar R., Pap E., *Triangular norms*, Dordrecht: Kluwer Academic Publishers, 2000.

[6] Lewin K., *Field theory in social science; selected theoretical papers.* D. Cartwright (Ed.)., USA, New York: Harper & Row, 1951.

[7] Maslow A., *A Theory of Human Motivation*, in: Psychological Review, 50(4), 1943, p. 97.

[8] Maslow A., *Towards a Psychology of Being, 2nd ed.*, D. Van Nostrand Co., New York, 1968, p. 204.

[9] Nickels W. G., McHugh J.M., McHugh S.M., *Understanding Business 8th edition*, McGraw-Hill, 2008.

[10] Plous S., *The Psychology of Judgment and Decision Making*, New York: McGraw-Hill, 1993.

[11] Simon H. A., Egidi M., Viale R., Marris R. L., *Economics, Bounded Rationality and the Cognitive Revolution*, Edward Elgar Publishing, 2008.

[12] Schwenk Ch. R., *Cognitive simplification processes in strategic decision-making*, in: Strategic Management Journal, Volume 5, Issue 2, April/June 1984

[13] Varey, R. J., *Marketing Communication. Principles and Practice*, Routledge, London, 2002.

[14] Wakker P. P., *Prospect Theory: For Risk and Ambiguity*, New York: Cambridge University Press, 2010.

Prediction of characteristics of dynamic social networks

Robert A. Kłopotek

Institute of Computer Science, Polish Academy of Sciences,
ul. Jana Kazimierza 5, 01-237 Warsaw, Poland
Email: robert@klopotek.com.pl

Abstract

The paper describes a new framework for visualization and inspection of a social network graph evolving over time. One of the new features is prediction of future structural properties of this graph. For this purpose we use time series analysis, based on a machine learning framework of Weka. We evaluate this approach for four learning algorithms for time series prediction: SVM (support vector machine), linear model, multilayer perceptron and Gaussian process. A series of experiments was designed to assess the quality of prediction of the dynamics of the social network with selected methods.

1 Introduction

Social network analysis (SNA) is a rapidly growing area of research with many current and potential practical applications in the economy, government, science, the organization of information society, ensuring the security and defense. Application examples are early warning systems of impending epidemics of infectious diseases [1, 2], seeking expert in certain areas [3], for designing efficient share of information and education [4, 5], search people with leadership qualities [6], to identify the instability of social relations [7], the detection of activities of terrorist groups (eg [8, 9, 10, 11, 12]), identifying network vulnerability [13] etc.

Social network often means social structure between actors, which are generally individuals or individual organizations. It shows the relationship of various types, ranging from random acquaintance to the close relationship or to object flows (e.g. information, goods, money, signals, intermediate in the production cycle, etc.) between the members of the community [14].

Social network analysis is focused on mapping and measuring relationships and information flows between people, their groups, organizations or other entities in transforming the information and / or knowledge. The SNA attempts to make prediction on the basis of the characteristics of the network as a whole entity, the properties of individual nodes based on network structure, etc. The subject of research can be a complete social network or parts of it related to a specific node.

As part of a research project[1], we developed a set of methods and algorithms, and created a system called BEATCA-SNA for supporting important tasks in SNA:

- enriching the description of the analyzed social network with features derived from texts, structural information with regard to different objects in the network, and from the characteristics of aggregates based on the cluster analysis of objects,
- enriching the description of the dynamics of the network analyzed by time-dependent prediction based on original and derived features (predictors) of objects (nodes) and aggregators of the network,
- searching, visualization and exploration of social networks based on the characteristics of nodes, their aggregates and structural relationships, and supporting these processes by recommendation techniques.

In this paper we will focus on the aspect of visualization and prediction of the structure development of the network graph. Section 2 discusses related work on dynamic social networks. In section 3 we will outline the design of the implemented system, concentrating on dynamic social network graph exploration capabilities. In section 4 we will describe the set of experiments we performed to evaluate its predictive capabilities. The experimental results are summarized in section 5.

Some of the implementations are based on selected ideas described in [15] and [16] and [17]. A completely new approach to graph visualization is presented in [18].

2 Related work

Recently finding characteristics of social networks and prediction of this characteristics grew to a field of intense research activity. The results of Barabási's research into the behavior of networks presented in book [13] show how even fairly robust systems like the Internet could be crippled by taking out a few super-connected nodes, or hubs. In [19] Albert and Barábasi present a statistical approach to modeling the networks dynamics, based on empirical data. The authors discuss the main models and analytical tools, covering random graphs, small-world and scale-free networks, the emerging theory of evolving networks, and the interplay between topology and the network's robustness against failures and attacks. They use the following network characteristics as measurement metrics: number of nodes, average degree, average path length, local clustering coefficient and spectral density of a graph.

Lin et al. [20] use first-order Markov model to analyze communities[2] and their evolutions in dynamic networks. They analyze impact of network history to predict communities in current state. Unlike traditional two-stage techniques that

[1]Research project grant from the MNiSW *N N516 443038 - 4430/B/T02/2010/38 Analysis and visualization of structure and dynamics of social networks by methods inspired by nature* realized from 9.4.2010 to 8.4.2012.

[2]A community is often thought of as a set of nodes that has more connections between its members than to the remainder of the network.

separate the task of community extraction and the task of evolution extraction, the FacetNet framework presented in their paper combines these two tasks in a unified process. Authors use node degree, modularity (as defined by Newman et al. [21]) and propose new measurement soft modularity.

Statistical properties of communities in a large dynamic networks were investigated by Laskovec at al. [22]. In their parer, they characterize as a function of size the statistical and structural properties of communities. They measure community goodness by conductance, also known as the normalized cut metric (see [23, 24, 25]). The practical problem with this metric is that it is hard to compute and we must make some approximations. They also introduce the network community profile plot, which measures the quality of the best possible community in a large network, as a function of the size of the presumed community. In network dataset they examined, the conductance score of the best possible set of nodes gets gradually worse and worse as those sets increase in size. This suggests that steadily increasing clusters are "blended in" more and more with the rest of the network.

Laskovec et al. [26, 27] present two unusual phenomena of network growth in time. First, most of real graphs densify over time, with the number of edges growing superlinearly in the number of nodes. Second, the average distance between nodes often shrinks over time, in contrast to the conventional wisdom that such distance parameters should increase slowly as a function of the number of nodes. They propose two models, which can explain this phenomenas: "Community Guided Attachment" (CGA) and more complicated "Forest Fire Model" (FFM), the latter being based on having new nodes attach to the network by "burning" through existing edges in epidemic fashion.

Berger-Wolf and Saia [28] propose a new mathematical and computational framework that explicitly makes use of information about the time that social interactions occur. They analyze structure of social network by creating various groups on graph representing this network. Interactions between individuals are recorded at every timestep and for them the membership into groups and meta-groups. They measure similarities of group defined by a social event by specially built similarity measure.

Laskovec et. al [29] present usefulness of social network analysis in viral marketing. Dynamic network is constructed from person-to-person recommendation network. Authors established how the recommendation network grows over time and how effective it is from the viewpoint of the sender and receiver of the recommendations. They used traditional epidemic (see [30]) and innovation diffusion models (proposed by Bass [31]). They concluded that in network-based epidemic models, extremely highly-connected individuals play a very important role.

In this paper we present a slightly different approach to analysis of the dynamics of social networks. Instead of providing the user with ready made reports on network properties we allow him to investigate himself network properties both at global and local level. Our work differs also in the approach to time intervals and measurement handling. For given interval (time slice) we measured metrics value considering only those graph nodes and edges that were "active" or existent only in this interval, instead of a cumulated graph till given time point.

3 A system for local contextual properties of social graphs

Prof. Tukey, who inspired a new trend in exploratory data analysis, believed that an important aspect of the analysis is data visualization and exploration. Therefore, a very important component of data mining system is an opportunity to present and search network properties seen at the local level.

The social network graph node in our system is seen in three different contexts:

- in the context of its position in the graph
- in the context of the passing time
- in the context of a group to which it was assigned during clustering process (with clustering of nodes being based on their textual content)

The first and second aspect implies that in fact we are not dealing with a single social graph, but with many of them, associated with individual quanta of time, where an individual node becomes the next incarnation.

On the other hand, search engine treats all these incarnations as a single entity e.g. during clustering, search, retrieval, presentation etc.

The system BEATCA [32] reflects this duality as a dual representation of the node. On the one hand we have the node as an entity, which is assigned to all the related texts, on which it can be searched, as well as time invariant attributes. On the other hand, there is its next incarnation, which is related to the attributes change over time. From a technical point of view the search of time varying and time invariant node features runs somewhat in separate spaces, which are combined on the fly when it comes to answer the user's query.

At the same time, node is characterized not only by its graph properties, but also by the text. For example, a node in time quanta can be associated with e-mails sent by it (e.g. Enron base).

Note that with appropriate design of experiments there is a possibility that for example, some data processing may deliver some new aggregation attributes, on which you can perform searches.

A new quality in developed tool is the opportunity to observe local and global graph metrics in time (after division of the timeline to the corresponding quanta) and the ability to search nodes with specific value (or values in the range) of this metrics in a specific time interval or at any time. A set of metrics (treated as node attributes) is by definition extensible.

In subsection 3.1 we will explain the kind of graph properties (metrics) that can be inspected by the user and in subsequent subsections various available inspection methods will be presented.

3.1 Local and global properties of social graphs in BEATCA-SNA

In this project we borrowed from a rich set of traditional coefficients known from literature, which represents local and global properties of social networks. Many such measures are given by Vázquez in [33]. However we expanded this set by an important one, called bipartite clustering coefficient (BLCC), proposed and investigated widely in [34].

We measure local properties of the graph nodes by following metrics:

1. degree - the number of edges incident to the vertex
2. indegree - the number of edges going to the vertex
3. outdegree - the number of edges going out of the vertex
4. Local Clustering Coefficient (LCC) - the ratio of the number of edges connecting the neighbours to the maximum possible number for simple graphs (see [35]). When applied to a single node, is a measure of how complete the neighbourhood of a node is.
5. betweenness centrality - number of the shortest paths between nodes (from any to any) passing through the node
6. closeness centrality - average distance from a given node to all other nodes in the network
7. eccentricity - measure captures the distance between a node and the node that is furthest from it, a high eccentricity means that the furthest away node in the network is a long way away, and a low eccentricity means that the furthest away node is actually quite close
8. modularity class - identifier of community[3] to which node was assigned
9. eigenvector centrality - the component of the principal eigenvector of the network adjacency matrix corresponding to a given node
10. pagerank - the component of the principal eigenvector of the network modified adjacency matrix corresponding to a given node; the modification consists in the following: If a node does not have an outgoing edge it is connected to all the other nodes. Then each one in the matrix is divided by the number of outgoing edges of a given node. Then all the entries are multiplied by one minus decay factor, then each entry is added decay factor divided by the umber of nodes in the network
11. Strongly Connected Component Number - identifier of a strongly coherent component the node belongs to (strongly coherent component is maximum subgraph of directed graph in which each node is reachable from each on a directed path)
12. Weakly Connected Component Number - identifier of a weakly coherent component the node belongs to (weakly coherent component is the maximum subgraph of directed graph, in which after changing it to the undirected graph, each node is reachable from each other one. Numbers range from 0 to (Weakly) Connected Components Count -1),
13. `source_second_neighbours_number` - number of second neighbours of vertex source

[3]In our system communities are obtained based on Newman's modularity concept. The algorithm runs as follows: initially each node constitutes its own community, then nodes are moved between neighbouring communities until a stopping criterion is reached. The obtained communities receive distinct identifiers called modularity class. A node is moved to the community of one of its neighbours if this would increase the modularity of the entire network. At each step the node giving the maximum modularity gain is selected. Modularity is the difference between the quotient of the number of edges inside of communities and of the total number of edges minus the sum of squares of the shares of edges that have at least one end in the community. The process is terminated if no gain of modularity can be achieved.

14. `sum_of_neighbours_degree` – the sum of the out-degrees of the neighbours of the source vertex
15. `BLCC` – BLCC coefficient (bipartite graphs clustering coefficient) for the source node, equals 1 minus the number of second neighbours of given node divided by the number of the degrees of the first neighbours after removal of the node from the network; it measures how much the local environment of the node differs from a tree

We measure local properties of the graph edges by following metrics:

1. `x-z-y-transitivity` – if we have edge $x \to y$ this is the number of such z, that $x \to z \to y$
2. `z-xy-transitivity` – if we have edge $x \to y$ this is the number of such z, that $z \to x$ i $z \to y$
3. `xy-z-transitivity` – if we have edge $x \to y$ this is the number of such z, that $x \to z$ i $y \to z$

Global properties of graphs are measured by metrics:

1. Total Nodes - the total number of vertices
2. Total Edges - the total number of edges
3. Average degree - average degree of vertices in graph
4. Diameter - diameter of the graph (the longest simple undirected path between vertices)
5. Average Path Length - the average length of a simple path
6. Average Local Clustering Coefficient - the average over the entire network of the local clustering coefficient defined above
7. (Weakly) Connected Components Count - number of coherent components
8. Density - measures how close the network is to complete graph. A complete graph has all possible edges and density equal to 1. It is the ratio of the number of edges to the number of edges to the corresponding complete graph.
9. Modularity - modularity as defined by Newman (see footnote in description of modularity class above), measures how well a network decomposes into modular communities, high modularity score indicates sophisticated internal structure

3.2 Methods for predicting the dynamics of local changes and visualization of predicted layers in time

The user has the possibility to analyze the time behavior of selected indicators (from the list given previously) both for the whole social network and individual nodes.

The source of this dynamics are appearing and / or disappearing edges in the graph of social network in subsequent time intervals. The system has two main perspectives of these processes:

- edge exists in the graph in a given time interval, if it was in the source data in the graph at any point in time in the past

- edge exists in the graph in a given time interval, if it was in the source data in the graph at any point in time in the time interval

Naturally in the first variant the edges do not disappear.

To enable the user to predict future behavior of indicators in the time points, we incorporated the standard time series analysis software proposed in [36] (with the possibility to choose a learning algorithm as a basis for prediction).

Learning algorithms are available from the implementation of Weka 3.7.5 described in [36].

- SVM (support vector machine)
- Multilayer Perceptron
- Gaussian Process
- Linear Regression

Prediction needs training data with at least two different values. The prediction horizon is restricted by the number of observations in a learning sample.

3.3 Methods for visualization of dynamics in social networks

The developed tool allows to model the variability in the time of social networks. With the methods of time series analysis there can be visualized the "stability" and anomalies in the behavior of individuals and / or groups as well - for anomalies - looking for these predictors.

In subsequent moments of time for each vertex in social graph we create a time series for a particular indicator. On this basis, we predict what will happen in time $(t+1)$, for example, one can ask if there is increase in the number of edges, if the number of connections between neighbours gets concentrated, if there is increase of centrality of a node, etc.

On the basis of the analysis mentioned earlier and taking into account the expected values of certain predictors, we can construct a graph prediction at the time $(t+1)$. This graph can be used e.g. for recommendations or search (for more details see [34]).

In BEATCA system the node of the graph is viewed also as a textual document. One of the features of BEATCA is that text documents can be clustered into co called document map using WebSOM like methods [32]. Therefore the user can view the local changes around the node in graph in time and watch changes of the map of documents in time.

3.4 Methods of visualization of the answer to the question in the social network

The system can search information on social network in the mapping search. There were used known algorithms for visualization (WebSOM) of static and dynamic graphs (with the possibility of visualization hierarchically grouped social networks, statically and dynamically) based on earlier experience of our team with implementation of BEATCA system.

The possibility to use queries are extended with elements related to the structure of the relationship, e.g. the question of the nodes satisfying the conditions imposed on their local graph properties such as centrality. We can specify the point in time in which the value is obtained. So we can issue a query where we want to find nodes with degree 2 within time interval (1999-01-01,1999-07-01)

```
~degree=2 ~time=[1999-01-01 00:00:00, 1999-07-01 00:00:00]
```

We can combine graph features with textual one, e.g. looking for nodes with PageRank over 0.001 at any point in time containing the word "Susan"

```
Susan ~pageranks>0.001
```

We can refer to average values of node features with textual one, e.g. looking for nodes with Average Path Length over 3.4 containing the word "Mark" in third quarter of 1997

```
Mark ~"Average Path Length">3.4 ~time-label=1997Q3
```

Visualization response includes: maps of documents, HTML presentation of the description of the node, a graphical representation of a node in the graph surrounding network and a graphical representation of the time series of coefficients associated with the node.

Designed methods were implemented in the interface, as described in the next section.

3.5 Design and implementation of the user interface

The graphical interface allows the user to choose social network, to which queries will be directed and to formulate queries and view the results in the form of:

- list of nodes
- map of nodes

The user can navigate both the list and the map of nodes. He can also preview the global social network graph properties (average and over time, and predictions).

When user selected node in the list, he has the possibility to:

- see a text description of the node
- view the local properties of graph node
- see the changes of these properties in time
- see predicted changes in the future
- view its graph surroundings and changes of the environment in time
- navigate to the nodes surroundings in graph

Formulation of queries can include:

- text features of node (textual information)
- average local properties of node features
- local properties at selected time intervals

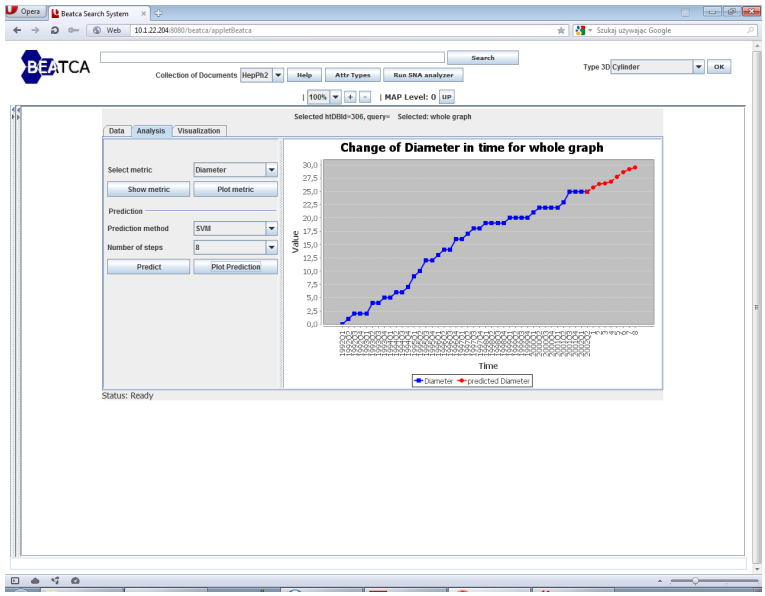


FIGURE 1: Example of usage of the "Analysis" tab to display textual dynamics of graph representing the diameter of the analyzed network.

The system allows for visualization and analysis of data, in particular the dynamic aspects of data as a graph of the social networks.

The interface of this subsystem consists of three tabs for the different aspects of the functionality of the system:

- "Data" - allows the user to select a data set to be analyzed
- "Analysis" - (see fig. 1) groups the user interface elements that allows to:
  - detailed control over analysis of the data
  - choose from a large range of attributes available for the analysis of the data
  - control of the analysis process of the network dynamics or the selected aspect
- "Visualization" - allows the user to interactively visualize respectively selected section of the data set as a graph.

The order of mentioning these tabs corresponds to one possible scenario of the sequence of operations by the user during inspection of the data, but of course the system can handle any other order selected by the user. In particular, it is possible to repeatedly switch between tabs to modify the entered parameters.

It is required to choose a data set for analysis before any other operations. While working with the system, user can select a different data set for analysis, etc.

4 Experiments

A number of experiments for prediction of future properties of social networks have been performed to verify the usefulness of the implemented predictive algorithms.

We performed experiments on prediction of the global graph properties described in section 3.1. It turned out, however, that only for a part of them reasonable results were obtained, that is for average clustering coefficient, connected component count, average path length, graph density and diameter. Only for these ones we will report the results.

We applied Weka's time series framework for the purpose of prediction of these properties. This framework differs from classical statistical techniques such as ARMA and ARIMA, because it uses standard machine learning algorithms trained on data from which the time factor was removed. The trained models are then applied in such way that the past values are used as predictors and the future value is the predicted one. The advantage of this approach lies in the possibility to use any learning algorithm and features not only belonging to the time series of the predicted index. It is claimed (see [36]) that such an approach may be even more effective than classical statistical methods.

In this study, as a learning algorithm for time series analysis we used the following algorithms: SVM (support vector machine), linear model, multilayer perceptron and Gaussian process.

A bad property that we need to live with is that the used prediction algorithms require at least two different points in the learning sample. Whenever this requirement is violated, they do not return any value, which is represented by the value -1, because value -1 is not possible to obtain from the absolute or relative error.

The usefulness of an algorithm is reciprocal to the error it makes when predicting future values.

We considered relative and absolute prediction errors.

The absolute error of prediction was calculated using the formula:

$$error_A = \sqrt{\sum_{i=1}^{k}(realValue_i - predictedValue_i)^2} \tag{1}$$

where k means the number of steps predicted ahead.

Calculation of the relative error was done as follows:

$$error_R = \sqrt{\frac{\sum_{i=1}^{k}(realValue_i - predictedValue_i)^2}{\sum_{i=1}^{k}(realValue_i)^2}} \tag{2}$$

where k is the number of steps predicted ahead.

4.1 Preparation of experimental data

The experimental work used three datasets :

- set "CitePh" - citations in physics papers
- set "Autonomous" - network of routers
- set "Enron" - email correspondence in Enron company

4.1.1 CitePh dataset

The dataset was downloaded from the web page: `http://snap.stanford.edu/data/cit-HepPh.html`. It is a graph describing the citations between scientific publications, for example, if paper A quotes B then there is an edge from A to B.

The original dataset was processed according to the following rules:

- Remove vertices (together with incident edges) whose date of creation can not be determined
- Remove the edges, where A is published earlier than B and A quotes B

In this way, we obtain a set in which papers (vertices) that appeared later cite papers that have been published previously.

Graph contains 37,622 nodes and 345,375 edges. This is a directed graph with no multiple edges. Neither vertices nor edges contain text data.

The network is divided into 42 time slices (each lasting one quarter), from Q1 1992 to Q2 2002 inclusive.

4.1.2 Autonomous dataset

The dataset was downloaded from the web page: `http://snap.stanford.edu/data/as.html`. This is a communication network between routers on the Internet.

Graph contains 7716 vertices and 11,965,534 edges. This is the undirected graph, there may be multiple edges. Both vertices and edges do not contain text data.

The network was divided into 785 one-day intervals, from 8 November 1997 to 2 January 2000.

4.1.3 Enron dataset

Database is available on the web page: `http://www.isi.edu/~adibi/Enron/Enron.htm` as well as `http://www-2.cs.cmu.edu/~enron/`.

The data were collected as a part of the so-called project CALO (Cognitive Assistant that Learns and Organizes). It contains information about users of Enron who sent about 500,000 messages.

Database created from this set includes 75,542 nodes connected by 431,573 edges.

The database is divided into 16 quarters. Each quarter has the same number of vertices, thus prediction of number of vertices is not possible. Moreover, in the last quarter (16) there are very few edges.

4.1.4 Database statistics

Basic statistics of databases are presented in table 1.

4.2 Experimental results

Let us present in this section a couple of results out of a large set we obtained that we think are of particular interest. Detailed results are presented in Appendix: for

Table 1: Basic statistics of databases

database	Enron	CitePh	Autonomous
number of time slices	16 quarters	42 quarters	785 days
number of all nodes	75542	37621	7716
number of all edges	431573	4708242	11939066

database Enron in tables 6, 4, 3, 5, 2, for database CitePh in tables 9, 7, 10, 11, 8 and for database Autonomous in tables 14, 12, 15, 16, 13. Results encompass: absolute errors calculated from equation (1) and relative ones form equation (2). Prediction model needs at least two different points, so prediction error in 1 step ahead starts from time slice 3: we have 2 observed values to build model (we will call it *learning sample*) and predict value for next (third) time slice and so on. Cases of 2 and steps ahead predictions are reported accordingly.

Subsequently we will use the following abbreviations for names of learning algorithms in prediction methods: SVM (Suport Vector Machine), MLP (Multilayer Perceptron), GP (Gaussian process) and LR (Linear Regression).

It turned out that not all global properties can be predicted by the methodologies used. Therefore only results for the following metrics are presented: *Average Clustering Coefficient* (see tables 6, 9, 14), *Average Path Length* (see tables 4, 7,12), *Connected Component Count* (see tables 3, 10, 15), *Density* (see tables 5, 11, 16) and *Diameter* (see tables 2, 8, 13).

Enron database was divided into 16 quarters. Each quarter has the same number of vertices.Moreover, in the last quarters (14,15,16) number of edges is getting smaller very fast, causing enormous errors in the prediction for those quarters. In the plots of absolute errors for *Density* (figure 2, 3, 4) where 1, 2, 3 step ahead prediction was applied resp., we see that the absolute error increases with increasing learning sample, while the relative error (figure 5) 1 step ahead is small except for the last unusual quarters. Unusuality of the graph in the last quarters reflects negatively on the quality of prediction when we forecast more than one step ahead. The best learning algorithm for time series turned out to be a *Gaussian process*, which was the most resistant to last unusual quarters. For numerical values of relative prediction errors and the value of attribute Density see table 5. In table 3 we can see a similar pattern for prediction of *Connected Components Count*. Here the best prediction was for *Multilayer Perceptron*. Time series method for prediction was much better for predicting *Diameter* (table 2), *Connected Component Count* (table 3) and *Average Path Length* (table 4). Here even for the unusual quarters error was not so huge but for *Diameter* it is over 100 %.

CitePh database contains 42 time slices in which both the number of edges and vertices are changing. Figure 8 shows a comparison of the relative error of prediction for the attribute *Average Path Length* predicted with learning algorithm SVM (see table 7). Prediction error here is big in first part and gets smaller later (after 25 time slice). From 25 elements in the learning sample error is stabilizing and very small about 3%. Errors of prediction 1, 2 and 3 steps ahead are similar. Figure 7 shows a comparison of the relative error of prediction for the attribute *Average Clustering Coefficient* (see table 9). Early prediction error is very high

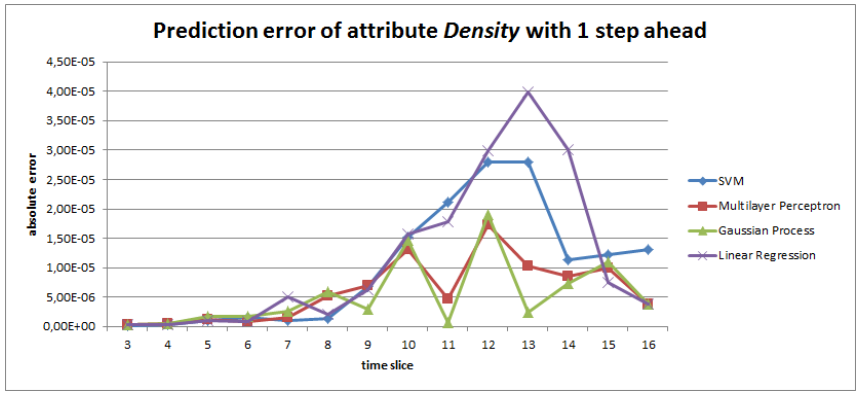


FIGURE 2: Absolute prediction error for 1 step ahead for the attribute Density in Enron database

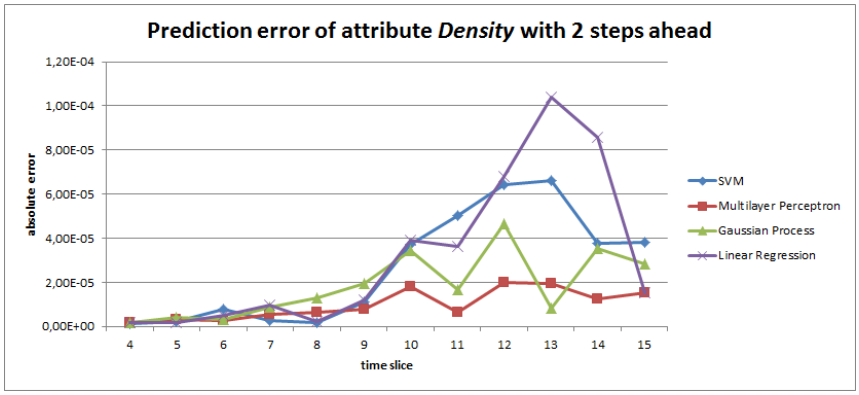


FIGURE 3: Absolute prediction error for 2 steps ahead for the attribute Density in Enron database

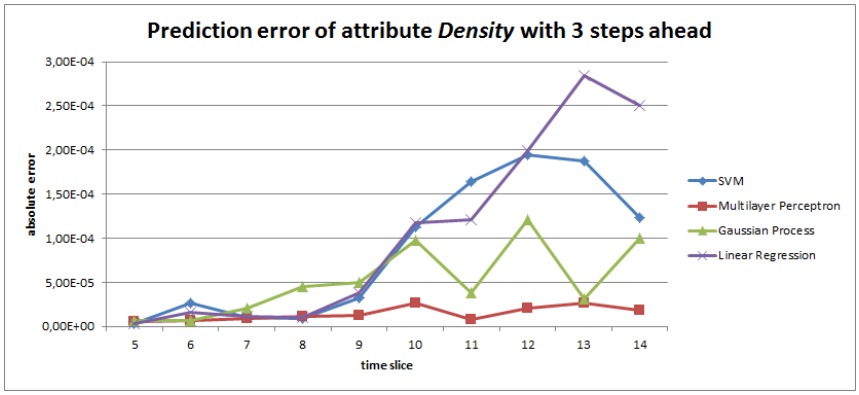


FIGURE 4: Absolute prediction error for 3 steps ahead for the attribute Density in Enron database

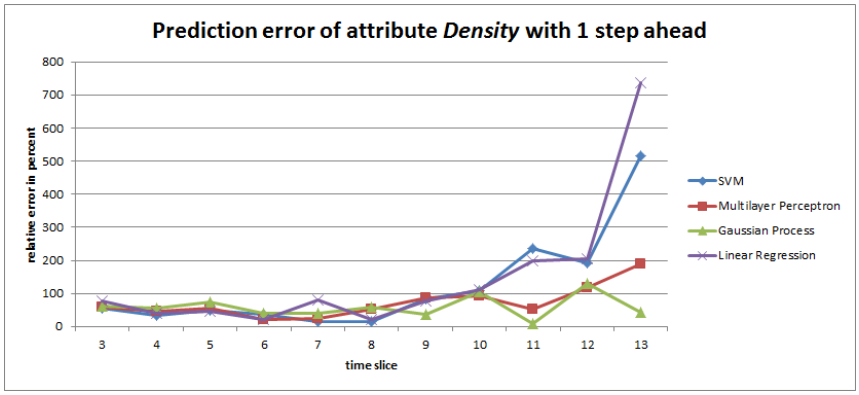


FIGURE 5: Relative prediction error in percent for 1 step ahead for the attribute Density in Enron database

above 100% for 2,3 steps ahead and at 20 elements in the learning sample is small up to 4%. Here we can see clearly that the number of steps in increasing the forward prediction becomes worse, in contrast to prediction of attribute *Average Path Length*. Moreover smaller increase in prediction for 1 step ahead causes much worse prediction error for 2 and 3 steps ahead. In those 2 cases SVM and MLP prediction are the best, when we predict more steps ahead GP is getting much worse than SVM and MLP. LR prediction is 30-50% from 20th time slice. We got similar results for other metrics *Connected Component Count* (table 10), *Density* (table 11) and *Diameter* (table 8), but for the first one SVM, MLP and GP does not differ so much.

Autonomous database contains 785 time slices, but for analysis several initial ones were selected. For the rest of the time series the results are comparable except for the time slices where no activity was registered and hence the graphs contain no edges. The number of vertices in time does not change. In figure 6 we see prediction results of attribute *Density* when SVM is used as learning algorithm for the time series. Relative prediction error is small, about 5%. With the increase in the number of observations in the training set error gradually decreases and stabilizes. Surprisingly all methods were working quite well, even the least advanced LR. Similar results we get for other metrics *Average Clustering Coefficient* (table 14), *Average Path Length* (table 12), *Connected Component Count* (table 15), and *Diameter* (table 13). Note that unusual table 13 for metric *Diameter* starts from time slice 22, because learning algorithm needs at least two different values and up to slice 19 time real value of metric was 9 and for time slice it was 10, so prediction form time slice 22 was possible.

5 Summary

This paper presented a framework for inspection of the dynamics of social networks. Its particular feature is the possibility to predict the development of some structural features of the network over time.

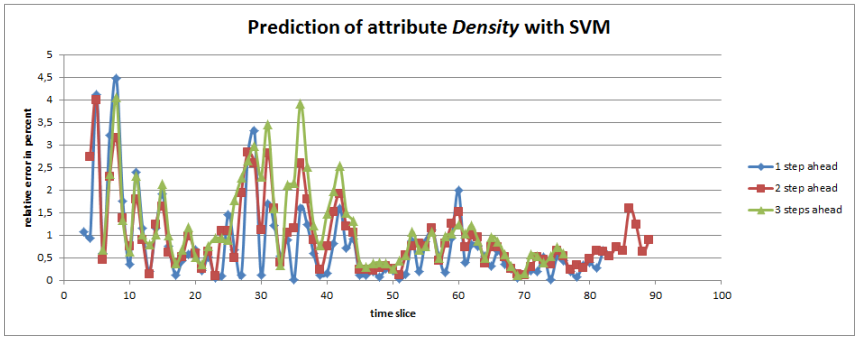


FIGURE 6: Comparison of the relative error in percent of prediction for the attribute **Density** of database Autonomous

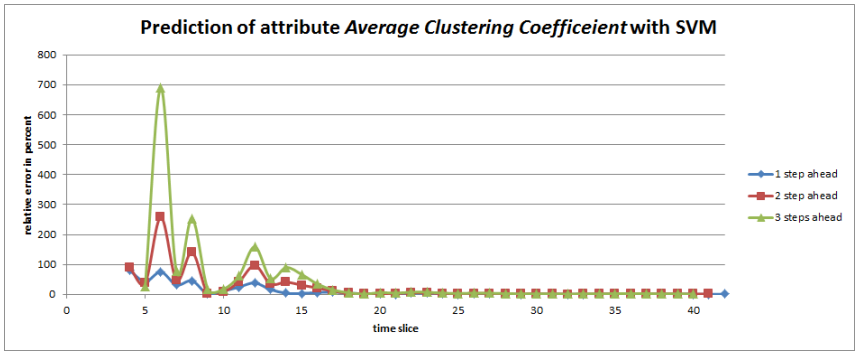


FIGURE 7: Comparison of the relative error in percent of prediction for the attribute **Average Clustering Coefficient** of database CitePh

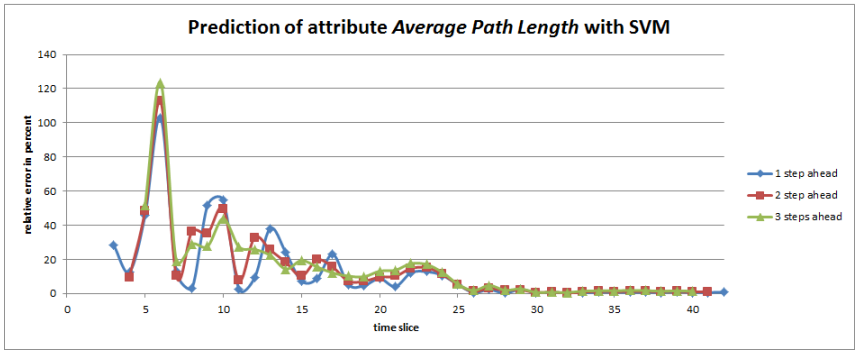


FIGURE 8: Comparison of the relative error in percent of prediction for the attribute **Average Path Length** of database CitePh

In this paper we evaluated the capability to predict the properties of social networks described in section 3.1 for three well-known dynamic social network databases, listed in section 4.1. The prediction is based on the Weka time series analysis framework, capable of exploiting machine learning algorithms for prediction of future attribute values. We compared 4 learning algorithms: support vector machine (SVM), linear model (LR), multilayer perceptron (MLP) and Gaussian process (GP).

It turned out that the best prediction (smallest error) was obtained for attributes *Average Clustering Coefficient*, *Average Path Length*, *Connected Components Count*, *Diameter* and *Density* for databases CitePh and Autonomous..

The best performing learning algorithm for time series prediction (for all the investigated databases) turned out to be SVM and MLP. Surprisingly prediction was quite with linear regression algorithm for database Autonomous. Learning using Gaussian process was characterized by the biggest error in the initial phase and rarely a little better than the MLP or SVM algorithm.

For databases Autonomous and CitePh, which have more time slices than Enron, prediction errors decreased up to overfitting the model. Unfortunately, sudden disturbances (no edge in some intervals) resulted in a dramatic increase in the prediction error.

Our approach to take measurements only from graphs existing in given time interval and not cumulated to given time point gave us interesting results. It was expected that with the increasing number of predicted steps prediction error will decrease, but as it has turned out, in the presented example of database Enron is not always true.

From experimental results of prediction 3 steps forward it can be seen that, with sufficient number of observations in the learning sample, the method of prediction using time series is doing quite well.

Due to the limitations of linear regression learning algorithm, as it has turned out in experiments, it is not recommended to use this one. General remark of our approach is that it could be expected that finding a constant trend should be easy. However, the disadvantage of the prediction algorithm is that at least two point values in the sample have to be different. For all learning algorithms this limitation from the Weka package seems strange.

Acknowledgments

The author would like to acknowledge the contributions of BEATCA SNA team: Marcin Sydow, Grzegorz Sobczak, Dariusz Czerski, Krzysztof Ciesielski, Piotr Borkowski and Michał Dramiński. This research project was supported by funds from the science budget of the Republic of Poland and is cofounded by the European Union from resources of the European Social Fund. Project PO KL "Information technologies: Research and their interdisciplinary applications".

References

[1] Pastor-Satorras, R., Vespignani, A.: Epidemics and immunization in scale-free networks. In Bornholdt, S., Schuster, H.G., eds.: Handbook of Graphs and Networks. Wiley-VCH (2003) 37–41

[2] Huang, C., Sun, C., Cheng, C., Tsai, Y.: Resource limitations, transmission costs and critical thresholds in scale-free networks. In: Proceedings of the 7th international Joint Conference on Autonomous Agents and Multiagent Systems (Estoril, Portugal, May 12 - 16, 2008). Volume 2., International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2008) 1121–1128

[3] Chirita, P.A., Damian, A., Nejdl, W., Siberski, W.: Search strategies for scientific collaboration networks. In: The 2005 ACM Workshop on information Retrieval in Peer-To-Peer Networks (Bremen, Germany, November 04 - 04, 2005), ACM, New York, NY (2005) 33–40

[4] Penuel, W.R., Sussex, W., Korbak, C., Hoadley, C.: Investigating the Potential of Using Social Network Analysis in Educational Evaluation. American Journal of Evaluation **27**(4) (December 2006) 437–451

[5] Cho, H., Stefanone, M., Gay, G.: Social information sharing in a cscl community. In: Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community. CSCL '02, International Society of the Learning Sciences (2002) 43–50

[6] Kim, D.K.: Identifying Opinion Leaders by Using Social Network Analysis: A Synthesis of Opinion Leadership Data Collection Methods and Instruments. Ohio University (2007)

[7] van Groenou, M.B., Deeg, D.: Changes in social participation. In de Boer, A., ed.: Report on the Elderly 2006, Transaction Publishers, New Brunswick (USA) (2006) 216–239

[8] Shaikh, M.A., Jiaxin, W.: Network structure mining: locating and isolating core members in covert terrorist networks. WSEAS Trans. Info. Sci. and App. **5**(6) (June 2008) 1011–1020

[9] Yang, C.C., Sageman, M.: Analysis of terrorist social networks with fractal views. Journal of Information Science **35**(3) (June 2009) 299–320

[10] Xu, J., Chen, H.: Criminal network analysis and visualization. Commun. ACM **48**(6) (June 2005) 100–107

[11] Carley, K.: Destabilization of covert networks. Computational and Mathematical Organization Theory **12** (2006) 51–66

[12] Krebs, V.: Connecting the Dots Tracking Two Identified Terrorists. `http://www.orgnet.com/tnet.html` (2008)

[13] Barabasi, A.L.: Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life. Plume Books (April 2003)

[14] Sharma, A.: Social networks. `http://www.ischool.utexas.edu/~i385q/archive/sharma_social_networks.ppt` (2009)

[15] Borzymek, P., Sydow, M.: Trust and distrust prediction in social network with combined graphical and review-based attributes. In Jedrzejowicz, P., Nguyen, N.T., Howlett, R.J., Jain, L.C., eds.: KES-AMSTA (1). Volume 6070 of Lecture Notes in Computer Science., Springer (2010) 122–131

[16] Borzymek, P., Sydow, M., Wierzbicki, A.: Enriching trust prediction model in social network with user rating similarity. In Abraham, A., Snasel, V., Wegrzyn-Wolska, K., eds.: Proceedings of the 1st International Conference on Computational Aspects of Social Networks (CASoN 2009), Los Alamitos, NY, USA, IEEE Computer Society (2009) 40–47

[17] Kłopotek, R., Sydow, M., Sobczak, G.: Ocena predykcji struktury sieci społecznej. In Kłopotek, M., Tchórzewski, J., Wierzchoń, S., eds.: Proceedings of the 26th International Conference on Artificial Intelligence AI-26'2012, Siedlce, Poland, University of Natural and Human Sciences (2012) 46–60

[18] Sobczak, G., Pikuła, M., Sydow, M.: Agnes: A novel algorithm for visualising diversified graphical entity summarisations on knowledge graphs. In Chen, L., Felfernig, A., Liu, J., Raś, Z., eds.: Foundations of Intelligent Systems. Volume 7661 of Lecture Notes in Computer Science., Springer Berlin Heidelberg (2012) 182–191

[19] Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Rev. Mod. Phys. **74** (Jan 2002) 47–97

[20] Lin, Y.R., Chi, Y., Zhu, S., Sundaram, H., Tseng, B.L.: Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In: Proceedings of the 17th international conference on World Wide Web. WWW '08, New York, NY, USA, ACM (2008) 685–694

[21] Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69** (Feb 2004) 026113

[22] Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Statistical properties of community structure in large social and information networks. In: Proceedings of the 17th international conference on World Wide Web. WWW '08, New York, NY, USA, ACM (2008) 695–704

[23] Chung, F.R.K.: Spectral Graph Theory. American Mathematical Society (1997)

[24] Kannan, R., Vempala, S., Veta, A.: On clusterings-good, bad and spectral. In: Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on. (2000) 367 –377

[25] Shi, J., Malik, J.: Normalized cuts and image segmentation. In: Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on. (jun 1997) 731 –737

[26] Leskovec, J., Kleinberg, J., Faloutsos, C.: Graphs over time: densification laws, shrinking diameters and possible explanations. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. KDD '05, New York, NY, USA, ACM (2005) 177–187

[27] Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: Densification and shrinking diameters. ACM Trans. Knowl. Discov. Data **1**(1) (March 2007)

[28] Berger-Wolf, T.Y., Saia, J.: A framework for analysis of dynamic social networks. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. KDD '06, New York, NY, USA, ACM (2006) 523–528

[29] Leskovec, J., Adamic, L.a., Huberman, B.a.: The dynamics of viral marketing. ACM Transactions on the Web **1**(1) (May 2007) 5–es

[30] Anderson, R.M., May, R.M.: Infectious Diseases of Humans Dynamics and Control. Oxford University Press (1992)

[31] Bass, F.M.: A new product growth for model consumer durables. Manage. Sci. **50**(12 Supplement) (December 2004) 1825–1832

[32] Kłopotek, M., Wierzchoń, S., Ciesielski, K., Dramiński, M., Czerski, D.: Conceptual Maps of Document Collections in Internet and Intranet. Coping with the Technological Challenge. IPI PAN Publishing House, Warszawa, Poland (2007)

[33] Vázquez, A.: Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. Phys. Rev. E **67** (May 2003) 056104

[34] Chojnacki, S.: Analiza technicznych własności systemów rekomendujących za pomocą algorytmów losowych. PhD thesis, Institute of Computer Science, Polish Academy of Sciences, Warsaw (2012) English title: Analysis of Technical Properties of Recommender Systems with Random Graphs.

[35] Watts, D.J., Strogatz, S.H.: Collective dynamics of "small-world" networks. Nature **393**(6684) (1998) 440–2

[36] Pentaho Community: Time Series Analysis and Forecasting with Weka. `http://wiki.pentaho.com/display/DATAMINING/Time+Series+Analysis+and+Forecasting+with+Weka`.

Appendix

Table 2: Metric real value vs relative prediction error (in percent) for 1, 2 and 3 steps ahead of metric **Diameter** of database Enron for consecutive time slices. Notation: number in title is number of steps ahead. Shortcuts: TS (time slice), RV (real value), SVM (Suport Vector Machine), MLP (Multilayer Perceptron), GP (Gaussian process) and LR (Linear Regression)

TS	RV	SVM 1	MLP 1	GP 1	LR 1	SVM 2	MLP 2	GP 2	LR 2	SVM 3	MLP 3	GP 3	LR 3
3	9	7.43	4.8	4.87	38.89	-	-	-	-	-	-	-	-
4	13	24.82	30.94	32.18	33.87	20.39	26.88	30.47	33.73	-	-	-	-
5	11	7.27	4.87	27.82	15.63	35.40	31.94	19.08	29.40	53.17	36.18	20.16	41.60
6	12	58.85	57.58	9.66	58.83	78.99	58.55	17.14	75.35	119.84	67.85	43.08	115.20
7	12	4.18	1.61	2.91	11.71	5.31	10.87	16.27	20.91	9.66	8.87	13.10	18.96
8	11	14.17	4.1	19.54	9.32	12.85	29.87	17.03	18.98	21.43	27.34	16.34	18.90
9	12	33.44	26.57	34.04	23.5	24.01	24.24	26.48	16.86	27.41	22.15	24.22	27.65
10	12	47.61	23.15	15.27	47.09	87.97	37.73	43.26	89.88	100.08	34.25	51.56	102.11
11	10	35.68	15.86	35.33	37.1	24.27	25.12	29.73	24.96	22.19	42.25	24.36	26.63
12	11	38.43	0.44	1.31	55.88	60.05	21.75	18.38	87.91	52.03	44.73	18.80	79.58
13	11	20.01	14.03	16.34	19.47	51.45	47.91	28.49	59.71	89.57	93.55	64.42	104.12
14	9	103.69	50.45	59.28	108.9	172.13	99.04	125.09	183.10	248.59	155.79	199.03	260.64
15	4	35.56	106.99	152.52	86.66	92.46	155.27	271.78	106.10	-	-	-	-
16	1	251.53	337.24	535.3	145.88	-	-	-	-	-	-	-	-

Table 3: Relative prediction error (in percent) for 1, 2 and 3 steps ahead of metric **Connected Components Count** of database Enron. Notation: number in title is number of steps ahead. Shortcuts: TS (time slice), RV (real value), SVM (Suport Vector Machine), MLP (Multilayer Perceptron), GP (Gaussian process) and LR (Linear Regression)

TS	RV	SVM 1	MLP 1	GP 1	LR 1	SVM 2	MLP 2	GP 2	LR 2	SVM 3	MLP 3	GP 3	LR 3
3	74357	0.76	1.15	0.85	1.13	-	-	-	-	-	-	-	-
4	72603	2.06	2.67	2.22	2.07	4.53	5.35	4.89	4.85	-	-	-	-
5	69524	3.92	5.95	4.23	5.59	3.65	7.47	4.08	6.43	4.24	10.12	4.98	8.06
6	67711	5.85	4.74	2.72	5.33	11.09	7.81	7.25	10.43	17.44	14.34	12.18	16.78
7	64368	4.06	11.42	1.53	4.07	13.57	19.11	4.80	13.63	15.53	18.26	5.60	15.54
8	57026	7.83	14.81	7.65	24.99	7.37	11.09	8.03	20.55	7.28	16.45	7.73	26.07
9	61567	22.09	6.34	15.06	25.15	30.85	17.20	18.90	32.42	53.36	15.57	37.99	50.07
10	52151	26.52	23.96	16.70	28.06	32.12	17.74	12.08	34.30	61.88	20.56	18.16	66.14
11	58537	27.58	1.89	22.95	30.08	38.78	7.97	32.58	46.43	73.06	9.73	57.52	88.40
12	51324	21.37	27.27	11.88	26.32	22.38	16.89	7.71	27.50	33.48	14.83	5.77	38.17
13	65108	15.29	7.73	18.24	18.19	30.74	6.45	33.39	39.43	44.19	6.66	44.50	56.38
14	74029	12.54	32.04	12.34	15.57	10.43	32.68	8.64	12.20	15.95	32.92	8.31	12.98
15	75420	28.17	15.76	13.24	27.79	47.93	17.63	22.14	47.74	-	-	-	-
16	75540	10.50	2.41	2.56	28.31	-	-	-	-	-	-	-	-

Table 4: Relative prediction error (in percent) for 1, 2 and 3 steps ahead of metric **Average Path Length** of database Enron. Notation: number in title is number of steps ahead. Shortcuts: TS (time slice), RV (real value), SVM (Suport Vector Machine), MLP (Multilayer Perceptron), GP (Gaussian process) and LR (Linear Regression)

TS	RV	SVM 1	MLP 1	GP 1	LR 1	SVM 2	MLP 2	GP 2	LR 2	SVM 3	MLP 3	GP 3	LR 3
3	3.905	2.86	6.83	10.9	33.13	-	-	-	-	-	-	-	-
4	4.474	7.14	0.1	15.35	2.54	7.02	35.55	19.45	3.11	-	-	-	-
5	4.559	19.96	23.25	29.85	13.1	20.67	23	21.24	23.13	34.26	24.51	20.71	40.61
6	4.572	54.6	43.56	9.98	36.96	81.97	46.49	24.19	60.7	106.24	50.85	34.99	86.74
7	4.379	3.28	16.78	5.06	9.89	3.13	13.69	6.58	24.32	8.96	11.39	5.45	47.66
8	4.531	7.17	14.85	1.07	12.74	16.26	35.69	10.02	29.44	16.8	36.08	9.12	24.49
9	4.437	12	7.78	15.12	10.86	21.39	6.55	11.08	10.71	30.59	6.07	11.96	11.18
10	4.422	42.77	18.18	15.15	39.5	52.25	26.24	20.66	50.57	57.91	30.21	27.58	57.44
11	4.496	4.2	7.13	3.44	10.81	6.22	9.31	7.03	18.71	14.76	25.84	6.62	32.25
12	4.422	0.36	0.08	7.55	13.68	10.62	21.77	7.77	37.35	20.22	20.91	18.22	32.53
13	4.531	15.34	15.67	14.03	12.2	23.21	14.56	19.51	30.91	56.19	36.78	49.74	65.46
14	3.569	59.34	29.29	45.24	69.79	117.34	68.57	98.22	127.43	171.84	111.38	151.31	180.52
15	1.977	57.84	35.93	97.98	52.12	93.24	32.68	154.04	71.28	-	-	-	-
16	1	37.28	48.23	137.51	153.58	-	-	-	-	-	-	-	-

TABLE 5: Relative prediction error (in percent) for 1, 2 and 3 steps ahead of metric **Density** of database Enron. Notation: number in title is number of steps ahead. Shortcuts: TS (time slice), RV (real value), SVM (Suport Vector Machine), MLP (Multilayer Perceptron), GP (Gaussian process) and LR (Linear Regression)

TS	RV	SVM 1	MLP 1	GP 1	LR 1	SVM 2	MLP 2	GP 2	LR 2	SVM 3	MLP 3	GP 3	LR 3
3	4.00E-07	56.70	58.39	61.98	76.56	-	-	-	-	-	-	-	-
4	9.02E-07	34.91	45.24	54.56	40.34	61.33	73.20	79.52	67.68	-	-	-	-
5	2.28E-06	47.75	54.12	75.06	46.06	48.44	68.30	83.68	41.35	45.39	76.76	84.71	35.83
6	4.28E-06	36.21	19.74	39.61	20.80	100.96	37.50	41.41	63.67	208.65	52.25	54.82	126.85
7	6.39E-06	15.30	25.26	41.14	79.02	21.96	48.29	73.15	84.25	70.84	62.36	144.93	83.99
8	9.83E-06	14.79	53.34	59.52	20.72	14.00	49.75	103.47	17.52	46.96	59.82	237.95	55.75
9	8.09E-06	83.89	86.27	36.23	77.77	70.37	47.98	119.87	76.20	178.70	66.74	270.10	205.96
10	1.39E-05	108.57	93.78	104.87	112.25	225.90	110.76	209.11	235.88	513.35	119.00	445.96	534.85
11	8.89E-06	236.75	53.05	7.71	199.61	295.92	39.23	98.74	214.06	920.39	44.30	214.62	681.46
12	1.45E-05	193.12	118.63	130.71	206.06	414.49	130.16	301.34	439.77	1259.74	134.81	784.87	1290.00
13	5.40E-06	517.13	190.54	43.59	738.03	1222.23	360.03	152.20	1915.67	3450.26	489.80	576.12	5253.48
14	5.05E-07	2232.95	1699.92	1444.70	5957.09	7435.59	2484.79	6988.38	16959.04	24264.05	3621.83	19836.88	49426.22
15	4.36E-08	28033.22	22856.47	25111.54	17168.75	87887.27	35200.03	64978.39	35355.88	-	-	-	-
16	3.50E-10	3753948.73	1083707.12	1077747.29	1095736.97	-	-	-	-	-	-	-	-

TABLE 6: Relative prediction error (in percent) for 1, 2 and 3 steps ahead of metric **Average Clustering Coefficient** of database Enron. Notation: number in title is number of steps ahead. Shortcuts: TS (time slice), RV (real value), SVM (Suport Vector Machine), MLP (Multilayer Perceptron), GP (Gaussian process) and LR (Linear Regression). Last row (for real value 0) presents predicted values.

TS	RV	SVM 1	MLP 1	GP 1	LR 1	SVM 2	MLP 2	GP 2	LR 2	SVM 3	MLP 3	GP 3	LR 3
3	5.09E-03	82.31	82.41	82.62	83.46	-	-	-	-	-	-	-	-
4	5.34E-03	806.57	45.76	330.08	287.43	4553.60	28.22	881.01	662.52	-	-	-	-
5	9.76E-03	232.85	112.13	322.28	137.07	222.16	108.90	298.97	136.89	217.11	83.10	494.22	89.81
6	1.40E-02	509.76	22.82	140.76	68.81	497.30	21.83	121.46	77.11	664.60	37.10	152.34	82.57
7	2.21E-02	179.07	107.12	510.23	7.48	187.56	106.02	523.35	8.47	295.03	106.31	752.75	8.25
8	3.15E-02	32.96	69.99	492.84	17.56	29.80	72.63	581.97	13.12	39.82	76.66	869.18	13.94
9	3.45E-02	158.70	49.43	587.49	27.68	169.77	35.59	631.33	38.70	360.78	56.54	1141.30	111.77
10	4.38E-02	289.61	64.70	718.27	613.70	370.63	60.96	1007.10	853.16	614.17	64.44	1647.10	1357.21
11	2.57E-02	381.24	35.14	1202.81	93.09	348.02	53.93	1256.25	64.85	613.90	62.15	2340.49	94.44
12	3.57E-02	1910.05	156.09	1229.44	2112.61	2877.93	169.89	1870.78	3160.02	5344.26	179.39	3594.65	5832.14
13	1.60E-02	5951.63	339.53	162.87	5883.05	9683.50	476.45	236.36	10021.93	17431.70	606.68	351.56	17601.65
14	2.80E-03	29439.30	302.31	8827.23	64692.01	45515.57	509.68	15094.99	94628.77	83185.29	514.22	29062.00	154686.18
15	1.86E-04	284358.56	6634.57	16094.07	828268.46	383162.75	10034.51	22755.04	1204807.77	-	-	-	-
16	0	1.95E-02	5.43E-03	7.66E-03	1.46E-02	-	-	-	-	-	-	-	-

Table 7: Relative prediction error (in percent) for 1, 2 and 3 steps ahead of metric **Average Path Length** of database CitePh. Notation: number in title is number of steps ahead. Shortcuts: TS (time slice), RV (real value), SVM (Suport Vector Machine), MLP (Multilayer Perceptron), GP (Gaussian process) and LR (Linear Regression)

TS	RV	SVM 1	MLP 1	GP 1	LR 1	SVM 2	MLP 2	GP 2	LR 2	SVM 3	MLP 3	GP 3	LR 3
3	1.039	28.33	19.66	7.03	51.89	-	-	-	-	-	-	-	-
4	1.062	12.75	7.70	1.30	20.71	9.56	6.15	13.54	62.09	-	-	-	-
5	1.083	45.64	55.89	45.62	91.78	48.45	57.22	34.81	62.94	51.38	51.27	31.86	49.74
6	1.150	103.07	68.40	36.66	189.16	112.69	51.30	40.92	250.51	123.60	47.18	44.52	322.05
7	1.231	12.86	13.60	6.09	65.28	10.56	9.17	5.58	131.54	19.07	22.58	12.54	193.23
8	1.352	3.10	10.11	4.26	0.42	36.24	34.01	24.45	41.17	29.02	28.52	19.11	33.18
9	1.461	51.80	23.02	37.15	52.94	35.38	18.77	25.48	40.88	27.70	19.79	20.41	43.27
10	1.593	54.67	22.60	8.34	115.53	49.69	23.36	8.42	154.01	43.91	19.92	7.48	186.00
11	1.724	2.81	3.37	6.76	9.91	7.59	7.56	8.74	16.65	27.44	24.28	20.55	19.50
12	1.918	9.15	15.44	12.19	1.24	32.82	41.88	26.49	41.34	25.93	32.83	24.55	32.46
13	2.180	38.14	44.30	33.42	54.66	25.56	29.32	27.48	37.32	22.39	29.51	26.63	28.69
14	2.515	24.11	3.54	11.09	28.87	18.37	10.02	13.03	22.39	14.40	12.17	14.34	20.60
15	2.841	7.37	3.75	17.85	18.87	10.24	5.66	19.40	14.85	19.39	14.34	23.73	21.68
16	3.163	8.76	10.88	15.87	18.14	19.77	24.24	21.69	26.54	15.62	21.83	19.74	21.58
17	3.435	23.00	16.43	21.81	27.89	15.54	15.63	17.34	18.99	12.13	16.82	15.63	15.43
18	3.747	5.26	2.73	5.02	4.34	7.25	5.01	4.92	4.15	10.28	6.92	4.21	8.00
19	4.090	4.68	3.30	5.58	0.43	7.30	2.28	4.91	1.81	9.88	1.82	6.12	4.30
20	4.346	8.84	0.30	4.48	4.87	9.67	0.94	7.14	8.04	13.19	0.77	5.74	12.90
21	4.535	4.11	3.77	9.73	4.30	10.21	6.00	6.94	3.29	13.73	5.71	5.56	4.16
22	4.745	12.17	6.68	0.03	2.47	14.74	5.38	1.90	4.68	17.59	4.92	3.27	4.43
23	4.911	12.90	2.74	0.19	0.37	15.19	3.58	1.09	2.87	17.07	5.15	1.95	5.78
24	5.037	10.72	0.91	0.73	7.08	11.39	1.53	4.45	8.37	12.56	1.31	3.87	6.89
25	5.129	5.33	4.30	7.77	4.35	5.29	5.04	6.28	4.22	5.54	4.19	5.80	9.95
26	5.256	0.70	0.82	1.62	15.39	1.27	0.63	1.51	28.80	1.79	0.93	1.26	30.32
27	5.372	2.35	1.17	1.69	49.49	3.07	1.73	1.36	49.81	4.82	2.17	3.08	50.13
28	5.440	0.26	3.43	1.03	48.31	1.89	2.53	3.84	48.64	2.00	2.64	3.12	48.92
29	5.508	2.52	2.31	5.83	47.24	2.25	1.63	4.15	47.52	2.79	4.40	3.51	47.77
30	5.565	0.46	2.15	1.11	46.18	0.61	2.62	2.91	46.42	0.75	2.80	4.95	46.62
31	5.615	0.75	3.38	2.59	11.09	0.95	2.42	4.48	14.12	0.86	2.47	4.68	18.14
32	5.653	0.34	3.45	3.76	20.15	0.26	4.13	3.17	25.39	0.66	4.46	6.31	30.86
33	5.683	0.39	0.93	0.25	23.80	0.89	1.09	4.78	28.72	1.48	0.98	8.07	34.39
34	5.714	0.87	2.85	6.04	26.37	1.34	2.55	9.03	31.66	1.70	2.37	12.13	37.67
35	5.748	0.91	2.95	7.73	41.11	1.18	2.80	10.45	41.28	1.69	2.81	11.66	41.37
36	5.781	0.76	0.06	8.40	40.28	1.31	0.18	8.59	40.33	1.82	0.24	12.19	40.41
37	5.791	0.86	2.08	4.19	39.26	1.28	1.47	9.06	39.36	1.78	1.23	13.17	39.42
38	5.809	0.70	0.90	9.52	0.90	1.18	1.04	13.10	0.81	1.42	0.90	16.62	0.75
39	5.817	1.22	1.67	11.01	1.08	1.39	1.50	14.07	1.58	1.62	2.07	15.58	2.34
40	5.842	0.69	0.67	11.11	3.85	1.02	0.70	11.57	4.39	1.43	0.58	15.38	5.10
41	5.847	0.65	0.76	6.39	14.25	1.19	0.54	11.45	22.26	-	-	-	-
42	5.847	0.83	0.59	11.45	1.96	-	-	-	-	-	-	-	-

TABLE 8: Relative prediction error (in percent) for 1, 2 and 3 steps ahead of metric **Diameter** of database CitePh. Notation: number in title is number of steps ahead. Shortcuts: TS (time slice), RV (real value), SVM (Suport Vector Machine), MLP (Multilayer Perceptron), GP (Gaussian process) and LR (Linear Regression)

TS	RV	SVM 1	MLP 1	GP 1	LR 1	SVM 2	MLP 2	GP 2	LR 2	SVM 3	MLP 3	GP 3	LR 3
3	2	33.32	37.82	44.39	75.00	-	-	-	-	-	-	-	-
4	2	18.19	14.86	7.67	50.00	27.53	55.39	13.88	79.06	-	-	-	-
5	2	40.19	50.01	55.52	82.59	37.24	36.86	55.21	67.35	27.78	29.59	44.39	55.96
6	4	1.51	19.10	41.64	5.11	30.36	17.91	31.61	22.93	32.29	25.44	28.64	21.80
7	4	50.39	28.34	3.97	47.15	49.49	19.37	14.71	42.95	63.98	18.33	14.31	52.72
8	5	37.24	34.44	42.27	44.90	38.37	35.90	46.87	49.42	36.74	39.93	45.14	47.51
9	5	24.49	34.35	30.27	26.61	23.00	27.08	29.80	24.89	29.75	28.72	27.90	31.42
10	6	11.14	1.08	25.79	17.61	8.07	4.66	24.55	12.72	10.62	4.83	26.02	9.91
11	6	18.20	3.55	25.02	28.37	23.92	3.59	30.09	37.91	18.98	29.67	48.92	29.39
12	7	8.55	5.65	30.66	3.21	36.76	26.87	53.06	33.96	36.41	27.24	58.27	31.53
13	9	30.69	40.58	51.31	36.74	27.08	39.43	53.10	31.35	30.68	43.60	58.71	35.49
14	10	31.97	17.68	26.31	30.40	29.34	11.33	32.74	24.27	41.73	9.45	31.52	33.58
15	12	23.96	27.35	42.52	25.20	16.97	20.78	40.88	17.97	13.57	17.99	44.27	14.30
16	12	32.14	19.53	8.44	30.20	30.27	14.79	16.88	28.58	29.86	11.71	22.20	28.34
17	13	9.83	4.85	20.92	16.25	17.58	9.62	27.26	37.48	14.66	7.88	27.70	32.27
18	14	10.17	5.02	22.56	22.78	11.43	7.28	19.64	25.88	10.43	11.71	25.11	30.61
19	14	18.54	2.58	2.51	52.05	12.31	10.82	10.92	37.54	12.89	19.74	12.45	50.10
20	16	6.16	8.61	16.43	57.89	5.24	8.33	17.64	57.89	5.94	8.38	17.33	58.80
21	16	10.68	1.69	10.90	55.00	10.76	4.08	9.72	56.42	11.39	17.03	9.37	57.78
22	17	4.91	4.18	4.98	55.18	6.68	11.73	4.97	56.51	12.23	9.54	4.26	56.92
23	18	3.32	6.73	4.72	55.30	10.77	5.07	3.91	55.30	8.80	9.92	9.01	56.16
24	18	10.76	10.49	1.91	52.90	7.41	7.35	10.25	54.22	7.02	5.94	10.00	54.62
25	19	2.63	19.12	14.37	53.29	3.97	22.33	12.32	53.29	3.32	21.55	11.43	53.29
26	19	8.01	0.14	2.12	51.16	6.75	2.19	1.54	51.16	9.14	2.17	3.84	51.16
27	19	1.77	11.92	0.37	49.19	4.51	9.28	4.04	49.19	3.91	8.80	3.24	50.11
28	19	1.46	15.84	4.09	47.37	1.35	11.05	3.12	48.77	2.14	10.40	4.26	49.20
29	20	3.54	7.03	5.66	48.39	2.51	8.26	4.09	48.39	2.77	13.92	5.23	48.39
30	20	4.66	1.34	3.12	46.72	5.98	1.29	6.54	46.72	5.54	1.39	9.20	46.72
31	20	2.56	0.33	5.94	45.17	1.88	0.48	7.96	45.17	2.18	0.55	7.48	46.11
32	20	1.20	6.21	5.48	43.71	2.42	35.16	3.98	45.14	4.11	36.40	3.93	46.52
33	21	2.70	3.03	1.91	45.09	4.39	5.05	1.32	46.41	4.12	6.42	3.34	46.82
34	22	5.03	4.37	0.01	46.28	4.50	3.68	3.68	46.28	4.50	3.83	7.31	46.28
35	22	0.56	5.04	4.42	44.92	1.92	6.34	7.98	44.92	1.58	5.37	10.05	44.92
36	22	2.04	1.88	7.35	43.64	1.57	4.99	8.56	43.64	1.28	5.53	10.74	44.52
37	22	1.87	7.06	5.12	42.42	1.44	5.01	7.50	43.75	3.20	6.94	7.28	45.96
38	23	0.03	6.54	6.03	43.83	4.02	4.97	4.93	46.32	4.33	5.96	6.97	47.03
39	25	5.46	1.62	0.43	47.26	5.10	1.15	4.30	47.26	4.68	3.83	6.12	47.26
40	25	3.62	3.80	5.18	46.05	3.76	7.40	6.41	46.05	3.07	6.59	10.65	46.05
41	25	2.58	5.22	4.23	2.10	1.84	4.70	9.27	2.17	-	-	-	-
42	25	3.49	1.18	9.89	2.07	-	-	-	-	-	-	-	-

TABLE 9: Relative prediction error (in percent) for 1, 2 and 3 steps ahead of metric **Average Clustering Coefficient** of database CitePh. Notation: number in title is number of steps ahead. Shortcuts: TS (time slice), RV (real value), SVM (Suport Vector Machine), MLP (Multilayer Perceptron), GP (Gaussian process) and LR (Linear Regression)

TS	RV	SVM 1	MLP 1	GP 1	LR 1	SVM 2	MLP 2	GP 2	LR 2	SVM 3	MLP 3	GP 3	LR 3
4	0.0031	80.73	80.42	83.99	83.85	91.28	91.02	94.19	96.96	-	-	-	-
5	0.0105	43.15	58.11	64.39	69.85	39.47	72.91	71.33	77.29	26.51	78.32	70.09	76.75
6	0.0188	75.11	33.62	12.09	71.44	258.10	44.79	98.43	242.33	689.33	53.53	222.78	633.12
7	0.0239	31.98	21.24	4.71	261.00	48.09	41.12	14.59	228.45	76.80	57.43	30.44	1405.80
8	0.0304	45.29	51.85	69.73	73.25	140.79	60.24	181.41	76.08	254.35	66.18	306.45	79.45
9	0.0368	1.29	1.79	87.11	3.48	2.04	10.51	190.95	10.48	17.49	8.91	313.02	9.96
10	0.0471	11.65	28.41	66.91	1.07	8.57	28.52	139.47	37.88	17.87	31.89	225.93	79.77
11	0.0499	25.01	46.74	59.28	20.94	43.01	51.43	147.00	32.10	62.44	54.67	250.05	45.13
12	0.0561	38.44	43.24	87.86	41.30	96.83	47.56	214.32	110.31	158.97	51.06	354.16	185.13
13	0.0622	16.55	24.06	87.93	22.64	35.57	34.69	188.65	40.52	53.59	41.70	306.95	53.02
14	0.0696	4.36	80.34	61.25	13.07	41.30	91.81	132.74	82.40	89.84	95.27	214.51	168.76
15	0.0742	3.61	12.49	71.23	1.47	30.93	14.75	167.69	74.53	66.64	17.12	284.06	170.79
16	0.0787	6.50	18.84	76.97	12.56	21.88	38.47	187.56	42.70	34.74	46.16	315.01	71.87
17	0.0829	8.14	80.19	52.06	5.89	11.01	133.21	76.00	7.09	15.09	154.49	99.89	9.96
18	0.0862	4.67	3.51	17.77	6.62	4.09	6.54	22.92	7.63	5.79	8.54	26.92	11.70
19	0.0909	2.28	5.77	15.57	8.07	1.59	12.94	17.85	9.82	1.28	18.85	21.85	11.83
20	0.0936	3.08	2.29	11.23	5.77	3.55	2.59	15.59	8.56	6.50	2.82	16.95	7.86
21	0.0966	0.58	6.22	13.95	52.61	2.51	9.83	13.99	53.22	4.45	15.33	13.76	53.71
22	0.0991	4.20	5.64	6.62	51.34	6.22	5.86	5.96	51.79	7.92	5.42	5.33	52.39
23	0.1009	4.92	3.58	3.18	49.93	5.78	3.24	2.75	50.61	7.59	3.45	2.69	50.94
24	0.1036	1.90	1.42	2.56	49.13	2.50	3.09	3.35	49.31	4.63	2.61	2.83	49.56
25	0.1043	0.62	14.54	4.58	47.45	2.75	27.81	3.22	47.75	3.33	23.08	3.33	48.18
26	0.1055	3.38	3.20	1.14	46.17	3.62	2.37	3.08	46.67	4.65	2.36	6.49	46.90
27	0.1075	2.34	0.14	1.89	45.41	3.65	4.28	5.22	45.51	4.50	4.75	6.68	45.68
28	0.1079	1.26	1.30	4.48	43.95	1.89	1.39	4.83	44.14	2.94	2.37	8.56	44.34
29	0.1087	0.65	8.22	0.95	42.77	1.55	6.59	5.68	42.98	2.18	6.27	9.32	43.24
30	0.1094	1.71	0.52	6.09	41.72	2.23	1.35	9.17	42.01	2.53	1.37	13.00	42.26
31	0.1105	1.47	3.95	6.75	40.92	1.64	3.62	10.23	41.14	2.09	2.98	12.33	41.32
32	0.1114	0.48	1.26	7.86	40.06	0.85	0.95	8.84	40.21	1.65	1.30	13.06	40.31
33	0.1119	0.39	2.87	4.10	39.11	1.10	2.04	9.20	39.19	1.97	1.70	13.84	39.30
34	0.1122	1.15	1.61	9.03	38.09	1.83	1.36	13.08	38.21	2.27	1.45	17.59	38.29
35	0.1127	1.76	1.34	10.15	37.22	2.21	1.62	14.23	37.28	2.77	1.60	16.85	37.35
36	0.1129	1.47	2.58	10.86	36.28	2.04	2.28	12.34	36.36	2.69	2.46	16.69	36.42
37	0.1132	1.11	1.69	6.67	35.44	1.75	1.47	11.81	35.48	2.79	1.39	16.75	35.47
38	0.1133	1.01	0.07	10.79	34.56	1.93	0.19	15.23	34.53	2.15	0.20	19.62	34.67
39	0.1132	1.76	1.95	11.88	33.58	1.77	1.88	15.75	33.81	2.01	1.90	18.32	33.89
40	0.1140	0.66	0.16	11.57	33.18	1.15	0.89	13.14	33.18	1.72	1.66	17.34	33.18
41	0.1140	0.70	0.01	7.37	0.55	1.06	0.24	12.37	0.86	-	-	-	-
42	0.1140	1.04	0.14	11.12	31.55	-	-	-	-	-	-	-	-

TABLE 10: Relative prediction error (in percent) for 1, 2 and 3 steps ahead of metric **Connected Components Count** of database CitePh. Notation: number in title is number of steps ahead. Shortcuts: TS (time slice), RV (real value), SVM (Suport Vector Machine), MLP (Multilayer Perceptron), GP (Gaussian process) and LR (Linear Regression)

TS	RV	SVM 1	MLP 1	GP 1	LR 1	SVM 2	MLP 2	GP 2	LR 2	SVM 3	MLP 3	GP 3	LR 3
3	419	41.43	85.34	50.05	73.87	-	-	-	-	-	-	-	-
4	718	27.99	62.09	45.13	2.29	30.81	68.40	55.17	12.33	-	-	-	-
5	908	24.45	37.86	57.96	51.58	15.41	44.92	56.43	42.65	20.76	50.15	48.97	33.49
6	1119	48.31	4.50	26.24	23.13	86.86	13.52	24.51	36.33	135.78	18.56	20.01	52.96
7	1292	12.62	21.89	36.64	4.85	27.18	25.50	51.45	8.76	46.80	30.11	71.38	17.29
8	1406	2.38	43.39	36.72	10.11	8.80	46.80	66.85	23.50	7.12	49.25	84.01	25.19
9	1544	9.95	77.32	57.53	10.16	8.12	78.06	72.93	8.52	15.11	78.93	85.39	15.98
10	1644	31.04	13.81	29.37	18.19	50.59	10.13	36.18	30.92	66.01	8.24	40.93	40.62
11	1778	13.30	38.75	38.58	20.75	29.94	41.66	58.83	44.02	44.17	43.56	79.02	65.36
12	1943	7.37	70.42	38.54	5.02	9.09	71.12	61.79	7.42	7.25	71.87	76.91	6.05
13	2038	2.71	0.91	44.46	3.14	16.30	2.99	57.12	11.64	27.33	5.19	67.13	21.85
14	2143	16.58	21.40	28.56	13.83	25.39	22.75	36.73	21.21	32.11	25.59	43.28	27.44
15	2227	8.49	10.64	32.78	13.33	18.65	24.38	49.81	26.17	29.07	34.96	68.51	40.79
16	2335	8.17	14.59	22.08	10.21	15.01	28.23	34.03	20.56	16.22	32.11	40.01	24.24
17	2451	8.09	20.66	24.72	10.28	7.05	18.69	26.63	9.21	5.97	18.58	27.63	7.96
18	2575	3.12	1.33	12.38	4.12	4.17	1.08	12.41	7.40	5.21	0.91	11.84	10.03
19	2722	1.17	1.79	7.57	1.58	1.64	2.06	7.22	2.37	2.31	2.01	8.76	2.21
20	2874	0.52	7.54	5.28	0.24	1.24	8.99	8.51	0.50	1.77	9.75	8.36	0.61
21	2984	1.18	5.69	10.52	1.62	1.34	4.00	9.67	1.36	1.57	3.39	8.53	1.47
22	3138	1.15	1.50	5.25	2.40	1.49	1.11	4.26	2.94	1.75	0.88	3.43	3.20
23	3286	0.36	0.59	2.44	46.78	0.59	1.03	1.81	48.11	0.92	5.11	3.69	49.66
24	3449	0.44	2.53	1.57	47.35	1.18	8.14	5.27	48.97	1.32	8.89	5.01	50.35
25	3658	1.48	6.85	8.19	48.50	1.42	6.52	7.21	49.75	1.71	6.79	6.28	51.11
26	3834	1.29	2.26	3.71	49.01	1.89	5.57	3.05	50.39	2.75	6.38	2.59	51.90
27	4039	0.68	1.18	2.03	49.81	1.58	4.28	1.88	51.34	2.01	3.86	3.10	52.63
28	4282	1.32	4.77	2.16	50.92	1.58	4.29	4.23	52.08	1.81	4.72	4.12	53.28
29	4482	1.61	13.25	5.89	51.37	1.87	12.49	5.29	52.58	2.00	13.77	4.61	53.74
30	4702	1.32	0.98	3.23	51.96	1.46	2.43	2.65	53.09	1.42	2.02	2.15	54.24
31	4921	0.66	1.16	1.67	52.44	0.52	1.07	1.27	53.59	0.42	1.30	1.88	54.62
32	5157	0.27	0.44	1.05	53.00	0.24	1.08	2.55	53.96	0.30	0.92	2.38	55.02
33	5366	0.07	1.76	4.02	53.24	0.12	2.61	3.50	54.33	0.23	3.17	2.94	55.38
34	5613	0.19	0.68	2.17	53.76	0.27	1.95	1.68	54.78	0.29	2.48	1.36	55.89
35	5860	0.40	3.85	1.02	54.19	0.35	5.41	0.82	55.32	0.46	5.77	1.40	56.25
36	6147	0.20	4.19	0.98	0.14	0.34	4.45	2.13	0.90	0.59	4.50	1.89	0.95
37	6371	0.52	2.92	3.27	54.97	0.75	2.66	2.70	55.90	1.10	3.25	2.17	56.75
38	6632	0.58	3.07	1.48	55.32	1.13	3.15	1.03	56.12	2.18	3.60	1.80	56.78
39	6870	1.31	3.20	0.08	0.92	2.43	2.84	1.69	2.17	3.45	2.82	2.02	3.27
40	7048	2.84	0.24	1.69	3.07	3.87	0.47	1.64	4.07	6.15	2.87	4.30	6.43
41	7202	4.22	0.58	0.15	3.91	6.92	1.97	3.80	6.46	-	-	-	-
42	7202	8.11	6.54	4.65	6.94	-	-	-	-	-	-	-	-

TABLE 11: Relative prediction error (in percent) for 1, 2 and 3 steps ahead of metric **Density** of database CitePh. Notation: number in title is number of steps ahead. Shortcuts: TS (time slice), RV (real value), SVM (Suport Vector Machine), MLP (Multilayer Perceptron), GP (Gaussian process) and LR (Linear Regression)

TS	RV	SVM 1	MLP 1	GP 1	LR 1	SVM 2	MLP 2	GP 2	LR 2	SVM 3	MLP 3	GP 3	LR 3
3	2.261E-04	8.35	13.23	23.57	65.64	-	-	-	-	-	-	-	-
4	2.092E-04	21.59	10.62	1.39	23.57	13.48	23.67	23.60	27.42	-	-	-	-
5	2.631E-04	54.67	65.54	58.96	100.00	36.25	43.34	41.55	75.27	33.11	33.37	31.94	61.86
6	2.985E-04	74.60	38.78	5.99	16.37	94.17	31.04	17.33	22.25	112.28	29.14	22.40	41.34
7	3.312E-04	8.41	21.24	8.05	24.61	8.90	20.22	6.94	51.17	7.49	17.05	8.88	72.67
8	3.544E-04	6.16	7.15	15.63	19.52	20.20	12.63	21.23	28.00	17.61	11.96	17.15	22.75
9	3.535E-04	29.66	2.41	30.01	24.79	21.69	8.06	21.35	22.24	27.17	17.04	18.98	34.10
10	3.686E-04	48.11	35.19	2.01	54.78	65.86	53.56	17.02	64.13	76.90	57.25	24.17	81.33
11	3.585E-04	14.98	19.34	4.66	12.78	17.26	20.33	8.11	27.13	14.04	16.58	7.01	25.40
12	3.499E-04	4.97	14.66	1.73	2.47	19.57	33.23	14.03	26.23	16.06	27.31	11.48	21.32
13	3.585E-04	24.86	30.61	22.42	44.88	17.86	22.12	16.34	32.62	19.63	28.49	13.69	26.86
14	3.589E-04	32.61	4.72	10.04	78.40	43.38	3.58	17.67	104.02	47.80	5.01	22.22	132.47
15	3.602E-04	8.72	16.06	5.15	0.43	9.40	15.49	6.66	2.96	8.19	13.69	6.15	22.32
16	3.613E-04	0.14	2.82	3.09	19.80	12.06	26.68	8.65	20.46	10.95	22.32	8.24	20.22
17	3.671E-04	19.31	16.01	16.45	39.64	14.10	11.75	11.77	28.29	16.44	10.49	12.69	25.26
18	3.609E-04	14.95	4.34	10.30	7.96	20.33	5.05	17.84	8.60	22.50	6.22	21.14	9.85
19	3.528E-04	10.26	9.41	15.56	3.48	11.28	8.78	16.97	3.13	10.56	9.22	15.17	3.78
20	3.513E-04	5.22	0.50	8.71	13.28	4.14	1.78	6.18	13.19	6.40	1.57	10.72	12.78
21	3.505E-04	1.93	1.47	5.59	12.43	2.07	1.14	8.33	11.85	3.68	1.47	14.11	11.33
22	3.458E-04	4.05	2.26	11.72	10.63	6.31	2.73	17.53	10.11	7.49	3.73	21.06	9.38
23	3.417E-04	1.16	0.05	15.40	1.38	2.42	1.28	18.16	3.72	3.92	2.10	17.20	6.75
24	3.345E-04	2.94	0.16	12.88	7.45	4.68	2.48	10.52	11.62	6.28	2.42	15.85	14.86
25	3.278E-04	3.79	1.23	0.57	4.52	5.40	2.18	12.14	3.49	7.17	2.90	19.21	2.99
26	3.190E-04	3.96	2.40	16.11	1.69	6.35	4.51	22.42	1.61	7.56	4.71	25.78	2.55
27	3.089E-04	5.07	1.75	19.83	1.58	6.33	1.97	22.53	2.98	7.44	2.10	21.32	4.08
28	3.019E-04	3.24	2.15	16.17	3.88	4.75	1.66	13.80	4.86	4.90	2.19	17.79	5.66
29	2.967E-04	2.84	2.59	3.39	5.56	2.59	4.45	12.90	6.26	2.54	5.54	19.04	7.27
30	2.930E-04	0.19	3.87	15.86	6.70	0.14	6.30	21.28	7.82	0.16	8.11	24.13	8.85
31	2.872E-04	0.26	5.41	18.56	8.61	0.27	7.20	21.10	9.57	1.10	8.56	19.85	10.42
32	2.824E-04	0.02	5.75	15.55	4.74	1.09	7.26	13.34	6.62	1.15	7.35	17.09	8.41
33	2.785E-04	1.54	6.19	3.96	4.94	1.39	4.86	12.89	7.29	1.19	4.15	18.41	9.45
34	2.736E-04	0.38	2.42	15.81	5.76	0.28	2.06	20.61	8.48	0.24	1.71	23.47	11.07
35	2.678E-04	0.42	0.70	17.99	6.67	0.33	1.66	20.89	9.93	0.31	1.39	19.59	12.19
36	2.607E-04	0.32	1.63	16.33	7.87	0.59	1.78	14.01	10.56	0.50	1.85	17.04	12.91
37	2.571E-04	0.64	0.79	4.66	7.32	0.48	0.68	12.70	10.49	1.17	0.56	17.30	13.01
38	2.529E-04	0.71	2.07	15.29	7.85	1.49	2.02	19.20	11.00	2.17	2.60	20.96	13.58
39	2.493E-04	1.61	2.60	16.48	7.94	2.15	2.69	18.32	11.25	2.32	2.21	16.52	13.64
40	2.458E-04	1.43	1.44	13.82	8.17	1.39	1.04	11.25	11.13	1.94	2.22	12.80	13.16
41	2.440E-04	0.42	4.65	2.70	7.63	1.70	3.76	8.81	10.22	-	-	-	-
42	2.440E-04	1.81	5.10	10.94	6.79	-	-	-	-	-	-	-	-

TABLE 12: Relative prediction error (in percent) for 1, 2 and 3 steps ahead of metric **Average Path Length** of database Autonomous. Notation: number in title is number of steps ahead. Shortcuts: TS (time slice), RV (real value), SVM (Suport Vector Machine), MLP (Multilayer Perceptron), GP (Gaussian process) and LR (Linear Regression)

TS	RV	SVM 1	MLP 1	GP 1	LR 1	SVM 2	MLP 2	GP 2	LR 2	SVM 3	MLP 3	GP 3	LR 3
3	3.7624	0.39	0.16	0.18	0.02	-	-	-	-	-	-	-	-
4	3.7618	0.07	0.09	0.03	0.03	0.15	0.15	0.11	0.11	-	-	-	-
5	3.7574	0.10	0.13	0.11	0.14	0.56	0.60	0.57	0.61	0.48	0.50	0.49	0.51
6	3.7310	0.66	0.71	0.66	0.82	0.57	0.56	0.58	0.60	0.64	0.55	0.68	0.55
7	3.7697	1.86	1.22	1.98	0.35	3.04	1.26	3.53	0.47	4.35	1.28	5.97	0.39
8	3.7778	0.03	0.17	0.20	0.26	1.08	0.19	0.47	1.49	1.77	0.77	0.98	2.45
9	3.7621	1.73	1.24	1.04	0.48	3.04	1.49	1.89	3.64	3.51	1.54	1.91	3.05
10	3.7614	0.98	0.67	0.31	1.55	4.24	1.03	1.60	3.60	3.49	0.84	1.32	2.96
11	3.7562	0.92	0.41	0.37	4.25	0.65	0.54	0.39	3.25	1.35	0.90	0.74	3.80
12	3.7676	0.70	1.12	0.59	0.73	0.89	0.84	0.52	1.57	0.73	0.68	0.44	2.44
13	3.7695	0.45	1.01	0.17	2.23	0.33	1.13	0.12	1.59	0.27	0.93	0.11	2.37
14	3.7733	0.02	0.60	0.00	3.78	0.22	0.42	0.12	2.89	0.77	0.96	0.42	2.36
15	3.7763	0.11	0.14	0.02	2.41	0.63	0.91	0.30	1.77	0.52	0.75	0.25	1.82
16	3.7731	0.04	0.31	0.09	0.90	0.41	1.07	0.12	1.78	0.42	0.88	0.11	1.84
17	3.7800	0.03	0.08	0.11	1.62	0.43	0.23	0.13	1.29	0.48	0.48	0.18	1.74
18	3.7770	0.05	0.41	0.16	0.56	0.35	0.44	0.20	0.40	0.49	0.47	0.24	0.32
19	3.7714	0.36	0.36	0.20	0.15	0.47	0.47	0.28	0.14	0.40	0.39	0.28	0.48
20	3.7707	0.20	0.36	0.21	0.28	0.26	0.26	0.31	0.24	0.30	0.22	0.37	0.27
21	3.7959	0.39	0.50	0.48	0.52	0.40	0.51	0.51	0.52	0.55	0.60	0.62	0.64
22	3.8008	0.25	0.12	0.36	0.09	0.54	0.41	0.54	0.58	0.54	0.44	0.52	0.63
23	3.8103	0.58	0.38	0.56	0.93	0.57	0.38	0.51	1.35	0.66	0.52	0.59	1.11
24	3.8048	0.12	0.10	0.01	0.46	0.48	0.40	0.27	0.67	0.60	0.57	0.54	0.81
25	3.8187	0.54	0.43	0.36	0.57	0.72	0.66	0.67	0.70	0.82	0.70	0.71	0.67
26	3.7759	1.00	1.14	1.02	8.59	1.03	1.06	0.96	8.27	1.34	1.20	1.13	20.01
27	3.7826	0.64	0.24	0.54	1.23	1.21	0.64	0.86	1.58	1.58	1.39	1.24	2.82
28	3.7673	0.97	1.02	0.89	0.19	1.44	2.01	1.31	0.53	1.67	2.12	1.42	0.56
29	3.7472	1.31	1.17	1.22	0.73	1.45	0.92	1.25	0.66	1.65	1.22	1.35	0.64
30	3.7521	0.33	1.17	0.61	0.57	0.48	0.96	0.74	0.57	0.51	0.78	0.70	0.48
31	3.7518	0.12	1.07	0.43	0.56	0.10	0.77	0.33	0.41	0.39	0.72	0.42	0.44
32	3.7671	0.20	0.66	0.17	0.13	0.36	0.54	0.17	0.34	0.63	0.44	0.27	0.45
33	3.7543	0.71	0.03	0.31	0.47	1.08	0.39	0.46	0.54	1.37	0.37	0.50	0.53
34	3.7489	0.66	0.50	0.37	0.60	0.72	0.51	0.34	0.55	0.66	0.50	0.28	0.47
35	3.7530	0.10	0.36	0.05	0.47	0.47	0.33	0.14	0.38	0.56	0.28	0.12	0.36
36	3.7608	0.50	0.39	0.22	0.25	0.47	0.37	0.18	0.27	0.39	0.33	0.27	0.45
37	3.7591	0.23	0.75	0.01	0.29	0.73	0.93	0.38	0.51	0.79	0.79	0.34	0.45
38	3.7448	0.65	1.58	0.52	0.66	0.54	1.58	0.40	0.51	0.56	1.91	0.39	0.45
39	3.7586	0.16	0.36	0.12	0.20	0.12	0.27	0.09	0.16	0.10	0.22	0.09	0.14
40	3.7591	0.13	0.18	0.03	0.06	0.23	0.46	0.12	0.15	0.19	0.68	0.10	0.13
41	3.7556	0.11	0.01	0.15	0.16	0.14	0.45	0.11	0.12	0.20	0.42	0.10	0.11
42	3.7616	0.23	0.62	0.13	0.07	0.30	0.58	0.15	0.05	0.36	0.51	0.17	0.05
43	3.7617	0.18	0.02	0.10	0.04	0.21	0.04	0.11	0.07	0.17	0.27	0.16	0.28
44	3.7615	0.02	0.34	0.07	0.05	0.22	0.46	0.20	0.32	0.30	0.76	0.27	0.44
45	3.7475	0.34	0.37	0.31	0.42	0.39	0.43	0.37	0.50	0.32	0.37	0.30	0.44
46	3.7427	0.30	0.62	0.27	0.27	0.23	0.51	0.21	0.19	0.23	0.46	0.20	0.17
47	3.7553	0.54	0.21	0.25	0.18	0.68	0.26	0.28	0.13	0.76	0.37	0.28	0.11
48	3.7545	0.22	0.59	0.19	0.10	0.20	0.67	0.18	0.15	0.24	0.57	0.22	0.14
49	3.7533	0.04	0.13	0.10	0.11	0.05	0.36	0.16	0.10	0.05	0.31	0.19	0.11
50	3.7567	0.10	0.41	0.16	0.01	0.10	0.44	0.18	0.04	0.08	0.41	0.17	0.11
51	3.7565	0.01	0.17	0.12	0.06	0.04	0.13	0.11	0.14	0.07	0.21	0.09	0.22
52	3.7538	0.06	0.35	0.04	0.13	0.09	0.42	0.03	0.23	0.07	0.38	0.10	0.26
53	3.7494	0.06	0.61	0.01	0.01	0.06	0.47	0.11	0.06	0.41	0.56	0.45	0.39
54	3.7501	0.13	2.31	0.16	0.09	0.54	2.12	0.55	0.48	0.57	1.73	0.58	0.50
55	3.7702	0.60	0.61	0.61	0.61	0.53	0.48	0.54	0.56	0.52	0.40	0.55	0.59
56	3.7636	0.23	0.10	0.02	0.14	0.31	0.13	0.06	0.10	0.35	0.11	0.07	0.08
57	3.7675	0.09	0.61	0.08	0.10	0.09	0.44	0.08	0.10	0.19	0.36	0.16	0.16
58	3.7680	0.02	0.24	0.03	0.03	0.17	0.26	0.15	0.14	0.14	0.22	0.13	0.12
59	3.7737	0.21	0.46	0.20	0.17	0.15	0.38	0.14	0.14	0.31	0.55	0.25	0.20
60	3.7631	0.07	0.20	0.14	0.22	0.30	0.15	0.23	0.20	0.45	0.15	0.35	0.29
61	3.7768	0.46	0.62	0.36	0.35	0.60	0.50	0.48	0.45	0.63	0.41	0.49	0.44
62	3.7842	0.56	0.23	0.41	0.31	0.54	0.72	0.38	0.29	0.66	0.82	0.47	0.40
63	3.7804	0.24	0.58	0.17	0.00	0.43	0.52	0.32	0.19	0.47	1.10	0.36	0.21
64	3.7891	0.31	0.63	0.35	0.28	0.28	0.48	0.36	0.27	0.35	0.39	0.33	0.33
65	3.7880	0.04	0.47	0.20	0.04	0.45	0.85	0.34	0.43	0.38	0.75	0.29	0.36
66	3.7625	0.67	0.69	0.52	0.63	0.49	0.89	0.37	0.46	0.48	0.82	0.31	0.44
67	3.7802	0.19	1.13	0.29	0.42	0.15	0.85	0.22	0.33	0.13	0.84	0.25	0.33
68	3.7716	0.26	0.11	0.01	0.10	0.20	0.37	0.12	0.11	0.18	0.31	0.16	0.13
69	3.7766	0.07	0.34	0.18	0.18	0.10	0.24	0.20	0.20	0.11	0.26	0.16	0.17
70	3.7765	0.06	0.03	0.14	0.10	0.16	0.89	0.10	0.07	0.14	0.84	0.11	0.13
71	3.7700	0.26	0.50	0.08	0.07	0.22	0.49	0.08	0.12	0.26	0.42	0.06	0.10
72	3.7757	0.05	0.06	0.11	0.20	0.14	0.51	0.08	0.15	0.12	0.54	0.22	0.26
73	3.7684	0.17	0.52	0.04	0.08	0.14	0.47	0.23	0.21	0.26	0.43	0.19	0.18
74	3.7805	0.23	0.48	0.34	0.32	0.24	0.35	0.25	0.23	0.29	0.40	0.24	0.23
75	3.7641	0.36	0.28	0.19	0.22	0.46	0.47	0.29	0.33	0.45	0.47	0.25	0.28
76	3.7563	0.33	0.87	0.28	0.31	0.28	0.79	0.21	0.22	0.25	0.73	0.18	0.18
77	3.7647	0.08	1.18	0.03	0.14	0.06	0.84	0.03	0.13	0.10	0.69	0.03	0.12
78	3.7634	0.04	0.44	0.02	0.06	0.04	0.33	0.01	0.05	0.13	0.27	0.12	0.10
79	3.7625	0.09	0.11	0.01	0.01	0.20	0.08	0.15	0.14	0.17	0.07	0.13	0.11
80	3.7540	0.22	0.26	0.21	0.20	0.16	0.21	0.15	0.14	0.24	0.27	0.20	0.16

TABLE 13: Relative prediction error (in percent) for 1, 2 and 3 steps ahead of metric **Diameter** of database Autonomous. Notation: number in title is number of steps ahead. Shortcuts: TS (time slice), RV (real value), SVM (Suport Vector Machine), MLP (Multilayer Perceptron), GP (Gaussian process) and LR (Linear Regression)

TS	RV	SVM 1	MLP 1	GP 1	LR 1	SVM 2	MLP 2	GP 2	LR 2	SVM 3	MLP 3	GP 3	LR 3
22	10	9.91	9.43	7.18	6.19	9.89	12.26	6.94	5.87	9.87	14.30	6.70	5.55
23	10	0.12	0.40	2.80	0.56	0.30	0.51	2.98	1.16	0.55	0.62	2.86	1.88
24	10	0.14	0.40	0.81	8.70	0.32	0.36	1.35	8.70	6.51	6.15	7.67	7.38
25	10	0.18	0.15	0.90	8.33	7.81	7.43	8.82	6.32	9.40	8.76	10.78	5.35
26	9	11.30	11.69	12.11	2.22	11.48	11.72	12.63	2.22	11.73	11.72	13.16	2.22
27	9	8.76	3.17	1.65	2.14	12.97	2.31	2.53	2.14	16.60	1.94	3.64	2.14
28	9	0.68	3.02	1.39	2.06	0.77	2.47	2.59	2.06	0.75	2.30	4.40	2.06
29	9	0.20	1.23	1.83	1.98	0.42	1.68	3.79	1.98	4.56	3.19	7.29	1.98
30	9	0.26	2.06	3.05	1.92	5.04	3.28	6.96	1.92	4.12	5.27	5.55	5.33
31	9	6.34	6.17	6.16	1.85	4.68	5.85	4.27	6.32	6.53	5.80	7.44	5.35
32	10	11.16	9.05	7.28	8.39	8.45	6.85	6.43	6.35	7.22	5.88	6.77	5.37
33	9	16.91	6.22	13.12	2.08	19.22	8.09	14.13	2.08	21.11	7.28	15.08	2.08
34	9	0.39	5.28	3.97	2.02	2.81	3.74	4.56	2.02	2.36	4.39	5.36	2.02
35	9	3.69	0.11	2.53	1.96	2.63	0.25	3.23	1.96	5.34	2.00	2.71	1.96
36	9	1.30	7.03	2.12	1.90	2.43	6.09	1.57	1.90	2.83	5.05	3.41	1.90
37	9	3.33	4.72	2.01	1.85	2.71	3.49	3.19	1.85	4.96	3.31	6.46	1.85
38	9	3.20	5.46	5.13	1.80	7.44	7.04	8.89	1.80	8.35	6.85	9.95	1.80
39	9	7.36	0.78	8.38	1.75	7.04	0.57	8.33	1.75	6.77	0.47	8.40	1.75
40	9	2.42	1.34	3.53	1.71	2.27	0.96	3.46	1.71	2.07	0.82	3.48	1.71
41	9	1.75	0.03	1.40	0.63	2.84	0.07	1.48	0.84	3.03	0.20	2.37	1.00
42	9	0.73	0.16	0.83	0.61	0.72	0.12	2.11	0.82	0.85	0.19	1.75	0.97
43	9	0.09	0.71	2.52	0.60	0.39	0.52	1.86	0.80	0.43	0.44	1.60	0.94
44	9	0.44	0.59	1.70	0.58	0.42	0.99	1.22	0.78	0.49	1.18	1.01	0.92
45	9	0.17	0.99	0.66	0.57	0.14	1.07	0.72	0.76	0.14	1.17	0.75	0.90
46	9	0.08	0.08	0.42	0.55	0.23	0.05	0.43	0.74	0.45	0.06	0.41	0.88
47	9	0.19	1.44	0.23	0.54	0.39	4.37	0.20	0.72	0.64	6.41	0.17	0.85
48	9	0.30	0.20	0.08	0.52	0.53	0.27	0.06	0.71	0.81	0.34	0.12	0.84
49	9	0.34	0.59	0.03	0.51	0.58	0.81	0.13	0.69	0.85	1.14	0.25	0.82
50	9	0.23	0.09	0.13	0.50	0.39	0.21	0.24	0.67	0.57	0.43	0.37	0.80
51	9	0.10	3.54	0.20	0.39	0.17	9.10	0.32	0.57	0.25	8.49	0.46	0.78
52	9	0.02	0.41	0.26	0.38	0.03	1.11	0.39	0.54	0.07	6.90	0.88	0.75
53	9	0.03	1.77	0.31	0.36	0.12	11.21	0.89	2.44	0.21	12.80	1.24	3.64
54	9	0.06	2.50	1.00	0.36	0.10	2.51	1.28	0.52	0.14	2.79	1.57	0.73
55	9	0.02	0.26	0.41	0.45	0.03	0.27	0.56	0.61	0.04	0.29	0.72	0.72
56	9	0.02	0.82	0.32	0.44	0.03	0.98	0.45	0.59	0.04	1.13	0.59	0.71
57	9	0.00	0.30	0.31	0.43	0.01	0.34	0.44	0.58	0.01	0.38	0.57	0.69
58	9	0.00	0.57	0.32	0.42	0.01	0.69	0.45	0.57	0.01	0.80	0.58	0.68
59	9	0.01	1.16	0.34	0.36	0.01	1.52	0.46	0.50	0.02	2.20	0.60	0.60
60	9	0.01	0.53	0.35	0.39	0.01	0.55	0.48	0.59	0.02	0.57	0.62	0.85
61	9	0.01	0.25	0.37	0.45	0.01	0.26	0.50	0.67	0.02	0.27	0.63	0.95
62	9	0.00	1.00	0.38	0.49	0.00	0.86	0.51	0.73	0.01	0.75	0.65	1.04
63	9	0.00	0.69	0.39	0.53	0.01	0.59	0.52	0.78	0.01	0.52	0.66	1.10
64	9	0.00	0.72	0.40	0.56	0.00	1.09	0.53	0.81	0.00	1.68	0.67	1.13
65	9	0.00	1.02	0.41	0.57	0.00	1.01	0.53	0.82	0.00	1.01	0.67	1.15
66	9	0.00	0.10	0.41	0.57	0.00	0.16	0.53	0.82	0.00	0.26	0.67	1.15
67	9	0.00	1.82	0.41	0.56	0.00	1.67	0.53	0.81	6.18	7.25	6.67	7.09
68	9	0.00	0.59	0.41	0.55	7.43	6.96	7.86	8.09	6.17	5.81	6.55	6.78
69	10	9.99	8.98	10.36	10.48	7.43	6.87	7.71	7.82	6.17	6.03	6.43	6.55
70	9	10.87	26.24	5.67	8.32	10.77	31.48	5.77	10.01	10.68	30.79	5.66	10.81
71	9	0.13	15.54	3.14	6.25	0.14	12.16	2.71	5.14	0.13	14.64	2.31	5.32
72	9	0.00	13.48	0.94	0.51	0.06	10.77	0.71	0.66	0.10	9.06	1.40	0.76
73	9	0.07	14.17	0.75	0.51	0.12	12.10	1.91	0.65	0.17	11.23	2.14	0.75
74	9	0.06	1.34	2.30	0.50	0.08	3.89	2.26	0.64	0.07	4.55	1.93	0.74
75	9	0.03	12.04	1.30	0.49	0.05	15.25	1.63	0.63	0.06	13.92	1.45	0.73
76	9	0.13	8.54	2.44	0.48	0.14	6.32	2.03	0.62	0.15	5.16	1.74	0.72
77	9	0.00	1.36	0.51	0.48	0.01	1.03	0.37	0.62	0.01	0.86	0.35	0.71
78	9	0.00	1.14	0.32	0.47	0.01	0.83	0.43	0.61	0.01	0.68	0.42	0.70
79	9	0.01	2.58	0.39	0.46	0.02	1.87	0.47	0.60	0.02	1.80	1.12	0.69
80	9	0.01	1.73	0.69	0.46	0.01	1.27	1.28	0.59	0.01	1.42	1.10	0.68
81	9	0.01	3.70	1.94	0.45	0.01	2.63	1.50	0.58	0.01	2.18	1.31	0.67
82	9	0.03	0.85	0.03	0.44	0.04	0.99	0.03	0.57	0.06	1.05	0.03	0.66
83	9	0.02	0.28	0.01	0.44	0.03	0.35	0.01	0.57	6.15	5.85	6.40	5.75
84	9	0.02	0.25	0.01	0.43	7.42	6.80	7.71	6.99	8.42	7.94	8.81	7.87
85	10	10.01	9.30	10.38	9.63	10.01	9.26	10.45	9.53	10.02	9.20	10.51	9.45
86	10	9.44	6.90	0.04	4.77	14.61	6.38	1.33	6.02	25.18	11.11	7.95	11.98
87	10	0.02	1.32	1.90	2.23	7.46	8.82	9.92	9.98	8.76	10.13	11.78	11.85
88	9	11.13	11.19	13.17	12.76	11.13	8.41	13.31	13.13	11.13	6.92	12.95	12.75
89	9	7.11	5.19	4.81	5.61	8.46	8.00	4.37	5.01	9.20	7.46	4.22	4.73
90	9	0.67	5.69	0.78	0.70	1.04	4.05	1.12	2.34	-	-	-	-
91	9	0.01	4.41	0.87	2.79	-	-	-	-	-	-	-	-

TABLE 14: Relative prediction error (in percent) for 1, 2 and 3 steps ahead of metric **Average Clustering Coefficient** of database Autonomous. Notation: number in title is number of steps ahead. Shortcuts: TS (time slice), RV (real value), SVM (Suport Vector Machine), MLP (Multilayer Perceptron), GP (Gaussian process) and LR (Linear Regression)

TS	RV	SVM 1	MLP 1	GP 1	LR 1	SVM 2	MLP 2	GP 2	LR 2	SVM 3	MLP 3	GP 3	LR 3
3	0.0740	6.00	2.52	3.05	0.91	-	-	-	-	-	-	-	-
4	0.0727	1.29	1.39	1.76	1.76	1.09	0.99	1.52	1.46	-	-	-	-
5	0.0737	0.88	0.22	0.65	0.58	0.90	0.30	0.47	0.97	0.86	0.62	0.40	1.27
6	0.0724	0.00	0.11	1.01	1.37	1.23	1.15	1.77	1.63	1.25	0.95	1.52	1.41
7	0.0721	1.76	1.52	1.82	1.62	1.52	1.30	1.44	1.22	1.34	1.23	1.26	0.99
8	0.0728	2.32	2.20	1.51	0.37	3.87	3.50	2.72	0.36	4.94	4.03	3.79	0.87
9	0.0733	3.29	3.36	1.53	3.74	2.45	2.61	1.52	2.81	3.73	3.39	1.86	4.53
10	0.0741	0.15	0.40	0.83	1.41	0.62	0.28	0.64	1.33	0.65	0.24	0.53	1.28
11	0.0740	0.29	0.42	0.20	0.74	0.21	0.40	0.32	0.57	0.22	0.33	0.37	0.85
12	0.0739	0.19	0.16	0.33	0.93	0.57	0.13	0.62	1.60	0.48	0.15	0.55	1.69
13	0.0748	0.97	0.59	1.16	1.98	0.99	0.61	1.15	1.88	0.81	0.57	0.95	1.59
14	0.0746	0.36	0.37	0.41	1.62	1.30	0.90	1.17	1.21	1.18	0.89	1.01	1.62
15	0.0738	2.13	1.44	1.31	0.43	1.50	1.23	1.03	1.52	1.51	1.00	0.84	1.50
16	0.0751	1.37	1.48	1.52	2.06	1.66	1.66	1.40	1.78	1.36	1.36	1.21	1.68
17	0.0746	0.73	0.91	0.36	1.31	1.61	1.46	0.41	1.32	1.40	1.31	0.82	2.00
18	0.0746	2.90	1.90	0.67	2.97	2.04	1.34	0.90	2.08	1.67	1.11	1.14	1.70
19	0.0758	2.39	1.88	1.66	2.13	2.67	1.90	1.88	2.45	2.21	1.60	1.67	2.01
20	0.0759	0.61	0.37	1.53	2.75	1.49	0.98	1.18	2.46	1.62	1.56	1.03	2.33
21	0.0754	2.55	1.79	0.36	1.98	2.48	2.09	0.40	1.96	2.07	1.73	0.82	2.39
22	0.0754	0.49	0.16	0.50	1.83	0.35	0.75	1.05	2.47	0.42	0.76	0.88	2.37
23	0.0763	0.17	0.23	1.36	2.88	0.37	0.40	0.99	2.51	2.22	2.33	2.31	2.41
24	0.0756	0.67	0.89	1.03	1.94	2.85	3.11	3.74	2.13	2.31	2.51	3.02	2.47
25	0.0725	3.86	3.94	4.77	2.41	2.66	2.71	3.29	2.69	2.43	2.60	2.88	2.31
26	0.0765	0.00	1.47	2.05	2.43	1.62	1.36	1.47	1.75	2.44	4.03	2.31	6.81
27	0.0752	2.41	3.56	0.78	2.56	3.01	6.01	2.13	8.44	2.47	5.06	1.77	6.96
28	0.0770	3.22	7.81	3.04	3.56	2.29	5.58	2.23	3.13	2.13	4.89	2.25	3.60
29	0.0763	0.19	1.80	0.41	2.50	0.91	2.71	1.16	3.50	1.47	2.46	1.44	3.69
30	0.0777	1.25	4.24	1.63	4.16	1.85	3.41	1.76	4.07	1.64	3.44	1.76	3.94
31	0.0775	2.21	1.66	1.71	3.83	1.61	3.54	1.51	3.69	1.71	2.96	1.60	3.99
32	0.0773	0.07	3.20	1.00	3.42	0.06	2.80	1.11	3.95	0.59	2.45	1.56	4.41
33	0.0781	0.05	0.16	1.16	4.30	0.73	0.68	1.60	4.72	1.41	3.23	2.36	5.16
34	0.0787	1.06	0.93	1.85	4.96	1.72	4.67	2.50	5.40	1.86	3.81	2.58	5.77
35	0.0794	2.45	5.00	2.85	5.65	2.03	3.57	2.56	5.98	1.67	2.94	2.25	5.92
36	0.0800	2.16	1.53	1.92	6.14	1.90	1.85	1.45	5.89	1.56	2.42	1.37	5.78
37	0.0795	2.39	1.60	0.53	5.45	1.89	1.26	0.52	5.42	1.88	1.58	0.96	5.66
38	0.0795	0.92	2.15	0.51	5.23	0.77	2.16	1.03	5.61	0.64	2.98	1.39	5.91
39	0.0801	0.87	0.67	1.31	5.83	1.09	0.99	1.57	6.08	1.28	1.46	1.95	6.44
40	0.0805	0.65	0.96	1.47	1.15	0.56	1.23	1.81	1.10	0.79	1.01	1.54	2.08
41	0.0812	0.10	2.06	1.69	0.81	1.25	1.54	1.22	2.19	2.01	2.01	1.07	3.11
42	0.0801	1.85	3.02	0.24	2.85	2.54	3.17	0.87	3.61	3.29	3.76	1.51	4.56
43	0.0796	2.90	4.77	1.06	1.15	3.49	5.57	1.69	1.88	3.52	4.53	1.51	1.76
44	0.0789	2.79	1.77	1.74	2.27	2.34	1.25	1.30	1.86	2.10	1.16	1.06	1.61
45	0.0798	0.66	1.53	0.04	1.10	0.46	1.85	0.42	0.92	0.49	1.86	0.44	0.89
46	0.0804	0.63	1.49	0.63	0.37	0.45	1.61	0.57	0.28	0.38	2.13	0.55	0.25
47	0.0805	0.41	0.19	0.32	0.18	0.47	0.20	0.32	0.26	0.62	0.59	0.26	0.38
48	0.0807	0.05	2.05	0.27	0.31	0.17	1.90	0.19	0.39	0.40	1.57	0.24	0.55
49	0.0807	0.20	0.57	0.03	0.44	0.47	0.41	0.23	0.55	0.41	0.50	0.23	0.45
50	0.0807	0.41	0.27	0.24	0.61	0.29	0.74	0.30	0.45	0.53	0.61	0.25	0.38
51	0.0815	0.31	0.12	0.50	0.24	0.33	0.63	0.36	0.18	1.88	1.43	1.44	1.42
52	0.0815	0.72	2.35	0.02	0.05	2.61	3.21	1.79	1.81	3.39	6.79	3.87	2.52
53	0.0798	3.05	3.60	2.50	2.59	3.48	5.37	4.77	3.13	5.19	7.15	7.10	4.84
54	0.0792	3.14	0.30	5.39	3.40	5.13	2.11	7.68	4.96	4.57	1.74	7.10	4.35
55	0.0767	3.00	3.27	5.02	5.32	2.17	2.27	3.57	3.96	1.78	2.06	3.22	3.84
56	0.0799	1.43	7.60	1.87	3.17	1.13	9.49	1.47	3.01	0.95	11.07	1.38	3.15
57	0.0791	0.03	0.64	0.06	0.65	0.22	3.52	0.11	1.05	2.20	6.04	1.90	1.39
58	0.0794	0.29	2.98	0.19	1.12	2.71	3.60	2.30	1.70	2.63	2.99	2.44	2.81
59	0.0768	3.54	5.07	3.36	3.39	3.16	3.60	3.00	2.98	2.59	3.14	2.44	2.43
60	0.0816	3.17	3.76	3.89	3.09	2.27	2.71	2.92	2.22	2.93	4.59	2.78	2.52
61	0.0796	0.16	1.36	0.07	0.23	3.32	9.37	2.89	2.59	4.32	7.83	3.83	3.67
62	0.0772	4.74	8.59	4.10	3.69	5.25	6.74	4.69	4.52	5.88	8.01	5.18	4.90
63	0.0761	5.22	6.90	3.95	3.50	5.84	6.70	4.24	4.45	6.00	5.86	4.07	4.35
64	0.0760	5.39	2.18	3.11	4.60	5.02	3.85	2.73	4.15	4.48	6.37	2.34	3.54
65	0.0767	0.34	4.11	0.98	0.82	0.76	8.06	0.69	0.83	1.61	6.65	1.10	1.94
66	0.0776	0.70	5.79	0.47	1.51	1.61	5.70	1.53	2.93	2.52	5.82	2.10	3.96
67	0.0790	1.35	0.09	1.91	2.70	2.07	5.64	2.35	3.66	1.94	4.65	1.97	3.09
68	0.0795	1.75	5.92	1.95	1.16	1.37	4.71	1.39	1.13	1.12	4.02	1.14	1.06
69	0.0780	0.19	4.82	0.93	1.66	0.63	3.97	0.85	1.56	0.71	3.26	0.70	1.36
70	0.0782	0.76	1.00	0.35	0.11	0.69	4.02	0.30	0.54	0.73	6.90	0.63	0.57
71	0.0787	0.32	1.16	0.42	0.82	1.24	3.17	0.87	0.76	1.21	4.15	0.76	0.64
72	0.0788	1.49	6.32	1.01	0.21	1.22	5.53	0.76	0.61	1.17	4.54	0.69	0.71
73	0.0784	0.33	0.02	0.08	1.08	0.25	1.66	0.12	1.12	0.29	1.97	0.29	1.55
74	0.0790	0.47	0.90	0.20	0.81	0.33	1.44	0.32	0.77	0.29	1.71	0.48	0.76
75	0.0788	0.31	1.95	0.48	0.27	0.40	2.40	0.63	0.35	0.33	2.22	0.51	0.81
76	0.0788	0.04	0.33	0.53	0.69	0.29	0.32	0.39	0.53	0.35	0.81	0.36	0.79
77	0.0794	0.43	2.98	0.38	0.09	0.42	2.78	0.28	0.50	0.61	3.57	0.69	0.42
78	0.0792	0.64	3.04	0.21	1.08	0.66	3.01	0.74	2.27	1.03	2.69	1.12	3.03
79	0.0801	0.97	1.17	1.11	1.51	1.47	1.47	1.44	1.92	1.31	1.26	1.28	1.89
80	0.0806	1.40	1.81	1.36	0.40	1.05	1.28	1.03	0.60	1.97	2.37	2.03	1.06

TABLE 15: Relative prediction error (in percent) for 1, 2 and 3 steps ahead of metric **Connected Components Count** of database Autonomous. Notation: number in title is number of steps ahead. Shortcuts: TS (time slice), RV (real value), SVM (Suport Vector Machine), MLP (Multilayer Perceptron), GP (Gaussian process) and LR (Linear Regression)

TS	RV	SVM 1	MLP 1	GP 1	LR 1	SVM 2	MLP 2	GP 2	LR 2	SVM 3	MLP 3	GP 3	LR 3
3	4693	0.59	0.19	0.38	0.23	-	-	-	-	-	-	-	-
4	4695	0.02	0.53	0.05	0.11	0.31	0.52	0.19	0.12	-	-	-	-
5	4694	0.42	0.23	0.18	0.11	1.52	0.98	1.33	1.05	1.34	0.88	1.09	0.91
6	4769	1.67	1.57	1.64	1.49	1.21	1.16	1.19	1.12	0.99	0.97	0.98	0.95
7	4674	1.10	0.04	0.69	0.48	1.37	0.14	0.73	0.89	1.15	0.13	0.62	1.28
8	4680	1.87	1.57	1.32	0.53	2.20	1.56	1.02	0.52	3.57	1.54	2.30	0.49
9	4681	5.48	0.57	1.98	5.10	6.21	0.54	2.74	5.55	5.29	0.60	2.27	4.78
10	4684	4.84	0.18	3.09	5.89	3.47	0.13	2.22	4.26	2.97	0.20	1.89	3.85
11	4675	1.08	0.19	1.40	3.00	1.55	0.35	1.03	5.06	1.35	0.50	1.01	4.62
12	4662	1.29	0.22	0.02	2.44	1.13	0.77	0.74	2.40	0.95	0.79	1.05	1.96
13	4651	0.17	1.99	0.53	0.90	0.13	2.08	0.73	0.99	0.26	2.10	0.89	1.01
14	4643	0.07	0.30	0.67	2.25	0.22	0.54	0.73	1.98	0.36	0.55	0.65	1.64
15	4644	0.18	2.61	0.69	0.13	0.28	2.30	0.52	0.51	0.50	1.94	0.49	0.75
16	4650	0.02	0.37	0.61	0.44	0.40	0.50	0.56	0.87	0.60	0.41	0.48	0.75
17	4637	0.14	0.64	0.66	0.28	0.55	0.46	0.51	0.22	0.46	0.38	0.44	0.29
18	4630	0.82	0.32	0.26	1.05	0.58	0.30	0.21	1.01	0.48	0.26	0.18	1.10
19	4634	0.26	0.06	0.16	0.91	0.19	0.11	0.12	1.06	0.15	0.13	0.10	1.15
20	4621	0.20	0.21	0.18	1.14	0.20	0.22	0.19	1.20	0.19	0.21	0.17	1.24
21	4616	0.02	0.05	0.13	0.03	0.05	0.10	0.10	0.04	0.10	0.17	0.09	0.05
22	4613	0.09	0.13	0.03	0.02	0.14	0.20	0.03	0.07	0.23	0.30	0.09	0.08
23	4610	0.10	0.06	0.02	1.22	0.18	0.08	0.09	1.24	0.30	0.16	0.20	1.22
24	4608	0.16	0.20	0.14	0.12	0.28	0.37	0.26	0.23	0.30	0.41	0.27	0.25
25	4611	0.35	0.08	0.25	0.14	0.32	0.09	0.22	0.11	0.37	0.09	0.26	0.19
26	4603	0.16	0.32	0.07	1.22	0.25	0.31	0.14	1.22	0.21	0.47	0.12	1.35
27	4603	0.19	0.18	0.15	1.18	0.14	0.40	0.12	1.36	0.16	0.58	0.18	1.52
28	4587	0.08	0.40	0.16	1.49	0.20	0.53	0.27	1.62	0.20	0.57	0.27	1.68
29	4575	0.18	0.53	0.31	1.70	0.13	0.53	0.28	1.72	0.13	0.49	0.23	1.71
30	4573	0.04	0.14	0.18	1.68	0.13	0.28	0.13	1.65	0.11	0.27	0.13	1.73
31	4576	0.22	0.10	0.04	1.56	0.17	0.12	0.08	1.70	0.18	0.10	0.06	1.75
32	4564	0.07	0.09	0.14	1.78	0.05	0.10	0.10	1.79	0.09	0.13	0.09	1.80
33	4563	0.10	0.21	0.01	1.75	0.16	0.23	0.06	1.76	0.13	0.19	0.11	1.90
34	4562	0.10	0.15	0.06	1.71	0.11	0.30	0.15	1.91	0.15	0.24	0.13	1.93
35	4545	0.15	0.19	0.25	2.04	0.17	0.13	0.18	1.98	0.25	0.11	0.17	1.95
36	4551	0.24	0.43	0.05	1.85	0.34	0.56	0.14	1.84	0.45	0.64	0.22	1.84
37	4552	0.32	0.21	0.16	1.78	0.41	0.24	0.23	1.78	0.45	0.22	0.24	1.83
38	4552	0.32	0.17	0.21	1.73	0.33	0.12	0.19	1.81	0.35	0.11	0.20	1.86
39	4545	0.32	0.01	0.10	1.84	0.33	0.04	0.12	1.87	0.34	0.07	0.13	1.91
40	4542	0.26	0.28	0.09	1.86	0.26	0.31	0.10	1.89	0.28	0.34	0.12	1.93
41	4539	0.12	0.09	0.06	0.20	0.15	0.11	0.08	0.22	0.17	0.13	0.12	0.24
42	4536	0.04	0.03	0.06	0.10	0.05	0.05	0.10	0.11	0.09	0.08	0.15	0.13
43	4534	0.05	0.05	0.09	0.07	0.09	0.05	0.14	0.09	0.16	0.04	0.19	0.12
44	4533	0.09	0.28	0.13	0.12	0.15	0.22	0.19	0.18	0.13	0.21	0.16	0.15
45	4533	0.16	0.16	0.18	0.04	0.11	0.12	0.13	0.21	0.12	0.10	0.11	0.31
46	4522	0.05	0.14	0.03	0.08	0.10	0.19	0.05	0.16	0.08	0.16	0.08	0.13
47	4516	0.08	0.08	0.08	0.25	0.07	0.10	0.09	0.21	0.06	0.09	0.09	0.24
48	4520	0.09	0.08	0.11	0.04	0.07	0.06	0.10	0.10	0.05	0.06	0.10	0.16
49	4514	0.03	0.03	0.03	0.12	0.05	0.02	0.04	0.18	0.04	0.04	0.10	0.22
50	4510	0.03	0.06	0.03	0.12	0.05	0.04	0.10	0.14	0.04	0.04	0.10	0.17
51	4510	0.08	0.11	0.12	0.13	0.06	0.13	0.10	0.18	0.09	0.11	0.16	0.15
52	4504	0.01	0.01	0.03	0.07	0.08	0.10	0.14	0.06	0.10	0.38	0.13	0.16
53	4507	0.12	0.15	0.17	0.07	0.12	0.35	0.14	0.21	0.11	0.40	0.13	0.22
54	4508	0.11	0.08	0.04	0.27	0.09	0.09	0.04	0.25	0.09	0.08	0.04	0.24
55	4504	0.04	0.18	0.04	0.06	0.03	0.13	0.05	0.07	0.10	0.14	0.05	0.17
56	4501	0.00	0.30	0.03	0.06	0.09	0.39	0.06	0.18	0.08	0.41	0.06	0.17
57	4492	0.13	0.07	0.09	0.23	0.10	0.05	0.07	0.19	0.08	0.07	0.07	0.18
58	4494	0.03	0.05	0.06	0.08	0.03	0.07	0.09	0.28	0.22	0.15	0.14	0.27
59	4492	0.01	0.08	0.09	0.38	0.24	0.20	0.18	0.29	0.21	0.17	0.15	0.48
60	4473	0.35	0.25	0.27	0.08	0.26	0.18	0.19	0.28	0.25	0.15	0.16	0.47
61	4480	0.08	0.01	0.05	0.03	0.15	0.07	0.04	0.13	0.12	0.10	0.14	0.11
62	4472	0.16	0.04	0.05	0.21	0.12	0.22	0.16	0.15	0.12	0.24	0.20	0.13
63	4480	0.14	0.46	0.22	0.13	0.16	0.47	0.24	0.10	0.15	0.45	0.24	0.10
64	4477	0.11	0.31	0.21	0.04	0.08	0.27	0.19	0.12	0.10	0.29	0.22	0.12
65	4470	0.06	0.01	0.11	0.15	0.05	0.06	0.17	0.12	0.07	0.07	0.16	0.18
66	4471	0.08	0.09	0.18	0.02	0.06	0.18	0.14	0.14	0.06	0.18	0.17	0.13
67	4462	0.08	0.31	0.04	0.20	0.06	0.28	0.13	0.17	0.07	0.31	0.12	0.20
68	4464	0.06	0.18	0.15	0.01	0.04	0.15	0.13	0.12	0.07	0.17	0.16	0.13
69	4457	0.06	0.22	0.06	0.17	0.05	0.25	0.12	0.16	0.14	0.31	0.24	0.13
70	4457	0.10	0.04	0.14	0.03	0.21	0.08	0.26	0.07	0.22	0.07	0.28	0.06
71	4462	0.22	0.14	0.31	0.11	0.19	0.10	0.29	0.08	0.22	0.12	0.34	0.09
72	4456	0.11	0.08	0.20	0.10	0.18	0.06	0.29	0.07	0.18	0.06	0.30	0.08
73	4459	0.21	0.24	0.30	0.09	0.19	0.21	0.29	0.07	0.16	0.17	0.26	0.09
74	4453	0.11	0.07	0.19	0.08	0.08	0.15	0.16	0.14	0.12	0.27	0.13	0.26
75	4446	0.03	0.02	0.07	0.13	0.16	0.13	0.08	0.28	0.25	0.23	0.14	0.40
76	4435	0.19	0.54	0.11	0.30	0.28	0.38	0.19	0.41	0.28	0.31	0.17	0.43
77	4425	0.28	0.58	0.23	0.21	0.26	0.55	0.18	0.19	0.22	0.50	0.15	0.16
78	4426	0.18	0.27	0.09	0.03	0.13	0.22	0.09	0.06	0.14	0.22	0.08	0.05
79	4430	0.06	0.19	0.10	0.09	0.05	0.25	0.08	0.07	0.09	0.33	0.07	0.09
80	4424	0.10	0.24	0.03	0.07	0.16	0.32	0.04	0.13	0.19	0.37	0.05	0.15

TABLE 16: Relative prediction error (in percent) for 1, 2 and 3 steps ahead of metric **Density** of database Autonomous. Notation: number in title is number of steps ahead. Shortcuts: TS (time slice), RV (real value), SVM (Suport Vector Machine), MLP (Multilayer Perceptron), GP (Gaussian process) and LR (Linear Regression)

TS	RV	SVM 1	MLP 1	GP 1	LR 1	SVM 2	MLP 2	GP 2	LR 2	SVM 3	MLP 3	GP 3	LR 3
3	1.8097E-04	1.07	0.86	0.89	0.76	-	-	-	-	-	-	-	-
4	1.8035E-04	0.93	2.82	1.63	0.16	2.73	1.99	3.45	0.43	-	-	-	-
5	1.8112E-04	4.12	0.84	0.05	0.55	3.99	1.51	0.97	0.72	16.07	1.28	0.80	0.72
6	1.7859E-04	0.59	0.37	0.17	0.97	0.44	0.32	0.45	0.80	0.67	0.28	0.43	0.66
7	1.8141E-04	3.22	1.43	2.76	3.85	2.30	1.14	2.05	2.84	2.35	0.93	2.28	2.76
8	1.8013E-04	4.47	2.31	1.75	0.06	3.15	1.65	1.25	0.55	4.05	1.91	1.70	0.62
9	1.8164E-04	1.75	1.00	0.94	1.87	1.37	0.81	0.79	1.45	1.33	0.67	0.68	1.36
10	1.8158E-04	0.35	0.60	1.23	0.66	0.76	0.90	1.00	0.96	0.63	0.78	1.08	1.04
11	1.8253E-04	2.39	2.73	1.88	2.52	1.80	2.00	1.42	1.96	2.32	2.40	1.74	3.44
12	1.8255E-04	1.16	1.87	1.65	1.02	0.89	1.35	1.47	1.31	1.01	1.29	1.71	1.48
13	1.8352E-04	0.19	0.14	2.52	2.41	0.14	0.22	2.12	1.88	0.79	0.65	2.22	1.93
14	1.8393E-04	1.15	1.42	3.24	0.04	1.23	1.59	2.48	0.88	1.00	1.33	2.60	0.98
15	1.8342E-04	1.92	0.12	2.84	1.37	1.64	0.94	2.21	1.33	2.13	0.83	2.65	1.59
16	1.8279E-04	0.76	0.87	0.28	1.06	0.62	0.72	0.23	0.89	0.99	0.98	0.19	1.52
17	1.8332E-04	0.12	0.35	0.18	1.00	0.37	0.29	0.17	1.06	0.38	0.71	0.57	1.23
18	1.8354E-04	0.43	0.04	0.07	1.13	0.51	0.57	0.59	1.17	0.67	1.03	1.07	1.08
19	1.8425E-04	0.58	0.59	0.76	0.29	0.98	1.01	1.26	0.95	1.17	1.19	1.54	1.05
20	1.8589E-04	0.68	0.67	1.37	0.87	0.60	0.58	1.49	0.89	0.51	0.48	1.50	0.86
21	1.8628E-04	0.22	0.34	1.19	0.78	0.26	0.29	1.08	1.31	0.35	0.24	1.08	1.73
22	1.8606E-04	0.51	0.30	0.63	0.52	0.64	0.25	0.69	0.61	0.75	0.22	0.70	0.66
23	1.8623E-04	0.05	0.33	0.63	0.24	0.10	0.44	0.64	0.24	0.93	0.60	0.79	0.89
24	1.8648E-04	0.08	0.16	0.20	0.08	1.09	0.81	1.37	0.90	0.93	0.79	1.12	0.78
25	1.8351E-04	1.45	1.24	1.80	1.40	1.10	1.04	1.26	1.00	0.90	0.87	1.06	0.88
26	1.8735E-04	0.68	3.42	0.93	2.45	0.48	3.40	0.72	2.36	1.77	5.04	2.18	3.91
27	1.8702E-04	0.12	0.65	0.11	2.18	1.94	2.50	2.35	4.37	2.26	2.96	2.80	4.99
28	1.9399E-04	2.79	2.46	3.29	5.61	2.84	2.39	3.41	5.77	2.64	2.11	3.45	5.87
29	1.9463E-04	3.31	2.66	2.63	5.72	2.59	2.81	2.52	5.79	2.98	4.01	2.86	5.79
30	1.9491E-04	0.11	3.32	1.64	5.67	1.12	2.36	2.03	5.62	2.30	2.19	2.57	5.62
31	1.9475E-04	1.69	5.23	1.89	5.39	2.81	5.25	2.40	5.41	3.45	5.37	2.88	5.53
32	1.9481E-04	1.22	2.56	1.78	5.25	1.59	2.05	2.18	5.43	1.59	2.38	2.40	5.57
33	1.9552E-04	0.53	0.69	1.50	5.43	0.40	0.71	1.62	5.56	0.34	1.05	1.73	5.83
34	1.9606E-04	0.89	0.75	0.94	5.53	1.06	0.54	1.03	5.86	2.10	0.45	0.86	5.93
35	1.9738E-04	0.01	0.29	0.68	6.00	1.14	0.67	0.48	5.96	2.16	1.83	0.61	5.90
36	1.9720E-04	1.60	3.62	0.22	5.74	2.60	3.60	0.71	5.67	3.91	3.97	0.83	5.59
37	1.9691E-04	1.23	0.95	0.67	5.44	1.78	1.09	0.67	5.35	2.51	1.12	0.62	5.35
38	1.9653E-04	0.59	0.50	0.17	5.11	0.89	0.74	0.13	5.16	1.21	1.04	0.64	5.39
39	1.9673E-04	0.12	0.23	0.19	0.06	0.24	0.32	0.86	0.48	0.76	0.27	0.94	0.53
40	1.9800E-04	0.15	1.83	1.07	0.20	0.74	1.77	1.01	0.19	1.47	1.67	0.88	0.43
41	1.9800E-04	0.81	0.45	0.23	0.30	1.51	0.32	0.23	0.60	1.98	0.27	0.27	0.74
42	1.9762E-04	1.58	0.68	0.26	0.60	1.92	0.73	0.30	0.72	2.52	0.65	0.53	0.97
43	1.9789E-04	0.71	0.58	0.16	0.44	1.19	0.83	0.43	0.83	1.49	0.88	0.44	0.93
44	1.9747E-04	0.92	0.50	0.44	0.82	1.06	0.47	0.38	0.87	1.30	0.52	0.35	0.94
45	1.9817E-04	0.12	0.33	0.05	0.07	0.23	0.29	0.04	0.15	0.38	0.27	0.04	0.21
46	1.9861E-04	0.12	0.15	0.05	0.06	0.24	0.11	0.05	0.22	0.30	0.17	0.05	0.30
47	1.9908E-04	0.19	0.13	0.09	0.26	0.20	0.12	0.09	0.31	0.38	0.15	0.17	0.56
48	1.9957E-04	0.07	0.08	0.11	0.16	0.28	0.20	0.17	0.44	0.40	0.21	0.25	0.62
49	1.9937E-04	0.26	0.31	0.20	0.49	0.32	0.35	0.27	0.61	0.37	0.42	0.32	0.73
50	1.9957E-04	0.22	0.22	0.19	0.36	0.26	0.31	0.23	0.40	0.25	0.44	0.21	0.45
51	1.9984E-04	0.03	0.11	0.13	0.18	0.12	0.08	0.09	0.18	0.43	0.61	0.60	0.76
52	2.0051E-04	0.14	0.58	0.08	0.05	0.55	0.46	0.64	0.79	0.55	0.72	1.72	1.13
53	1.9903E-04	0.87	0.31	0.89	1.08	0.76	1.09	2.13	1.34	1.07	1.71	3.12	2.02
54	1.9879E-04	0.19	0.68	2.52	0.56	0.81	1.27	3.41	1.10	0.68	1.05	3.33	0.95
55	1.9671E-04	0.85	1.07	2.12	0.99	0.76	0.75	1.65	0.70	0.75	0.68	1.51	0.58
56	1.9950E-04	1.10	0.49	0.32	0.99	1.15	0.56	0.28	1.05	1.07	0.47	0.24	0.90
57	1.9967E-04	0.54	0.18	0.07	0.25	0.43	0.50	0.21	0.59	0.49	1.03	0.81	1.02
58	1.9937E-04	0.17	0.60	0.31	1.01	0.82	1.05	1.02	1.50	0.98	0.95	0.91	1.23
59	1.9748E-04	0.92	1.88	1.27	1.04	1.25	1.32	1.06	1.25	1.04	1.26	0.91	1.02
60	2.0182E-04	1.99	1.81	1.30	2.37	1.51	1.31	0.93	2.01	1.24	1.11	0.88	1.71
61	1.9962E-04	0.40	0.13	0.44	0.66	0.74	0.35	0.93	1.67	1.04	0.49	1.29	2.01
62	1.9834E-04	0.80	2.03	1.07	1.85	1.00	1.96	1.38	1.88	1.22	2.35	1.70	2.28
63	1.9742E-04	0.76	3.13	1.26	0.33	0.96	2.77	1.56	0.47	0.88	2.81	1.55	0.38
64	1.9673E-04	0.53	0.57	1.36	0.33	0.37	0.41	1.22	0.31	0.49	0.47	1.09	0.49
65	1.9775E-04	0.32	0.07	0.57	0.51	0.74	0.38	0.44	0.81	0.96	0.60	0.36	1.07
66	1.9861E-04	0.65	1.64	0.03	0.66	0.74	2.43	0.12	0.83	0.86	2.62	0.20	1.06
67	1.9938E-04	0.35	0.69	0.19	0.50	0.51	0.92	0.27	0.67	0.58	1.01	0.27	0.80
68	1.9982E-04	0.26	1.04	0.29	0.43	0.25	1.09	0.25	0.50	0.32	1.35	0.29	0.71
69	1.9979E-04	0.05	0.19	0.11	0.24	0.14	0.27	0.21	0.47	0.11	0.45	0.21	0.64
70	2.0012E-04	0.13	1.31	0.25	0.27	0.11	1.69	0.22	0.32	0.16	2.14	0.24	0.45
71	2.0007E-04	0.21	0.47	0.11	0.18	0.30	0.53	0.16	0.31	0.58	0.49	0.18	0.29
72	2.0034E-04	0.20	2.86	0.18	0.27	0.51	3.07	0.20	0.20	0.54	3.25	0.17	0.34
73	1.9973E-04	0.51	0.85	0.26	0.08	0.46	0.88	0.20	0.21	0.38	0.78	0.29	0.66
74	2.0027E-04	0.02	0.44	0.01	0.34	0.36	0.75	0.38	0.87	0.52	0.88	0.55	1.17
75	2.0158E-04	0.52	0.45	0.56	0.95	0.65	0.52	0.69	1.15	0.74	0.64	0.79	1.39
76	2.0230E-04	0.44	0.32	0.62	0.68	0.53	0.46	0.71	0.83	0.59	0.55	0.74	1.01
77	2.0281E-04	0.20	0.37	0.59	0.96	0.22	0.36	0.60	1.42	0.18	0.29	0.53	2.12
78	2.0293E-04	0.08	0.76	0.43	0.97	0.33	0.68	0.33	1.69	0.27	0.89	0.51	2.02
79	2.0251E-04	0.33	0.31	0.05	1.33	0.26	0.66	0.46	1.46	0.29	0.85	0.65	1.76
80	2.0382E-04	0.39	0.27	0.64	0.48	0.47	0.37	0.80	0.77	0.78	0.67	1.18	0.98

Sequential Monte Carlo and Bayesian methodology in the stochastic event reconstruction problems

Piotr Kopka[1,2], Anna Wawrzynczak[2,3], and Mieczyslaw Borysiewicz[2]

[1] Institute of Computer Science, Polish Academy of Sciences,
[2] National Centre for Nuclear Research, Świerk-Otwock, Poland
[3] Institute of Computer Science, Siedlce University, Poland

Abstract

In many areas of application it is important to estimate unknown model parameters in order to model precisely the underlying dynamics of a physical system. In this context the Bayesian approach is a powerful tool to combine observed data along with prior knowledge to gain a current (probabilistic) understanding of unknown model parameters. We have applied the methodology combining Bayesian inference with Sequential Monte Carlo (SMC) to the problem of the atmospheric contaminant source localization. The algorithm input data are the on-line arriving information about concentration of given substance registered by the downwind distributed sensor's network. We have proposed the different version of the Hybrid SMC along with Markov Chain Monte Carlo (MCMC) algorithms and examined its effectiveness to estimate the probabilistic distributions of atmospheric release parameters.

Keywords: Bayesian inference, stochastic reconstruction, MCMC methods, SMC methods

1 Introduction

Accidental atmospheric releases of hazardous material pose great risks to human health and the environment. Examples, like Chernobyl nuclear power plant accident in 1986 in Ukraine or Seveso disaster in 1978, prove that it is necessary to have properly fast response to such incidents. In the case of an atmospheric release of chemical, radioactive or biological materials, emergency responders require relatively fast tools to predict the current and future locations and concentrations of substance in the atmosphere. One of the fields of application of the Bayesian approach can be problem of the localization of the dangerous substance release based only on the measured concentration sparse data.

Knowledge of the temporal and spatial evolution of a contaminant released into the atmosphere, either accidentally or deliberately, is fundamental to adopt efficient strategies to protect the public health and to mitigate the harmful effects of the dispersed material. However, to create the model realistically reflecting the

real situation based only on a sparse point-concentration data is not trivial. This task requires specification of set of parameters, which depends on the considered model. Non-inverting problems of this type are termed inverse problems: problems that can be solved in one direction but for some physical reason cannot be solved in the opposite direction. Such problems are widely encountered in several fields [1]. For instance the group method of data handling (GMDH) [2], [3] and it modifications seem to be successful as a method of inductive modeling and forecasting of complex processes and systems. The main idea of the GMDH is to create the algorithm able to construct a model of optimal complexity based only on the data. The goal is to get mathematical model to describe the processes, which will take place at object in the future. GMDH solves it, by sorting-out procedure, i.e. consequent testing of models, chosen from set of models-candidates in accordance with the given criterion. More recent developments utilize genetic algorithms or the idea of active neurons and multileveled self-organization to build models from data e.g. [4], [5].

The key idea behind statistical inversion methods is to recast the inverse problem in the form of statistical inference by means of Bayesian statistics. In the framework of Bayesian statistics all quantities included in the mathematical model are modeled as random variables with joint probability distributions. This randomness can be interpreted as parameter variability, and is reflected in the uncertainty of the true values expressed in terms of probability distributions. The solution of the inverse problem corresponds to summarizing probability distribution when all possible knowledge of the measurements, the model and the available prior information, has been incorporated. This distribution, referred as posterior distribution, describes the degree of confidence about the estimated quantity conditioned on the measurements [6].

It is clear that given a known gas source and wind field we can calculate the expected gas concentration for any downwind location. In the case of gas dispersion, the unknown state is the source strength and its location. It is obvious that if we are able to create the model giving the same point concentration of considered substance, as we get from the sensors' network, we could say that we understand the situation we face up. However, to create the model realistically reflecting the real situation based only on a sparse point-concentration data is not trivial. This task requires specification of set of models' parameters, which depends on the applied model.

A comprehensive review of past works on solutions of the inverse problem for atmospheric contaminant releases can be found in [7]. A variety of approaches to solve the atmospheric dispersion inverse problem have been explored including non-linear optimization, back-trajectory, Greens function, adjoint, and Kalman filter methods [8]. However, these methods often fail due to the inherent complexities, high-dimensionality, and/or non-linearity of the underlying physical system [9]. In [9] was introduced dynamic Bayesian modeling and the Markov chain Monte Carlo (MCMC). In [10] and [11] were presented sampling approaches to reconstruct a contaminant source for synthetic data.

In our previous work we have presented the application of the classical MCMC methods [20]. We have applied the methodology combining Bayesian inference with MCMC algorithms to the problem of the source localization. We have shown

the advantage of the MCMC algorithms that in different ways use the source location parameters' probability distributions, obtained based on available measurements, to update the marginal probability distribution of considered parameters with use of the newly received information. In this paper we examine the application of the Sequential Monte Carlo (SMC) methods combined with the Bayesian inference to the problem of the localization of the atmospheric contamination source. We present the possibility to connect MCMC and SMC to provide additional benefit in the process of event reconstruction. Proposed algorithms are tested on the synthetic release experiment.

2 Theoretical preliminaries

2.1 Bayesian inference

A good introduction to Bayesian theory can be found in [11] and [12]. Bayes' theorem, as applied to an emergency release problem, can be stated as follows:

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)} \tag{1}$$

where M represents possible model configurations or parameters and D are observed data. For our problem, Bayes' theorem describes the conditional probability $P(M|D)$ of certain source parameters (model configuration M) given observed measurements of concentration at sensor locations (D). This conditional probability $P(M|D)$ is also known as the posterior distribution and is related to the probability of the data conforming to a given model configuration $P(D|M)$, and to the possible model configurations $P(M)$, before taking into account the sensors' measurements. The probability $P(D|M)$, for fixed D, is called the likelihood function, while $P(M)$ is the prior distribution. $P(D)$ is the marginal distribution of D and is called prior predictive distribution [10]. $P(D)$ serves as a scaling factor and is crucial for model comparison; so in our case the Bayes theorem can be written as follows:

$$P(M|D) \propto P(D|M)P(M) \tag{2}$$

To estimate the unknown source parameters M using (2), the posterior distribution $P(M|D)$ must be sampled. $P(D|M)$ quantifies the likelihood of a set of measurements D given the source parameters M.

Value of likelihood for a sample is computed by running a forward dispersion model with the given source parameters M. Then the model predicted concentrations M in the points of sensors location are compared with actual data D. The closer the predicted values are to the measured ones, the higher is the likelihood of the sampled source parameters.

We use a sampling procedure with the Metropolis-Hastings algorithm to obtain the posterior distribution $P(M|D)$ of the source term parameters given the concentration measurements at sensor locations [10], [11]. This way we completely replace the Bayesian formulation with a stochastic sampling procedure to explore the model parameters' space and to obtain a probability distribution for the source location.

2.2 The likelihood function

A measure indicating the quality of the current state of Markov chain is expressed in terms of a likelihood function. This function compares the concentrations predicted from model and observed data at the sensor locations as:

$$ln[P(D|M)] = ln[\lambda(M)] = -\frac{\sum_{i=1}^{N}[log(C_i^M) - log(C_i^E)]^2}{2\sigma_{rel}^2} \tag{3}$$

where λ is the likelihood function, C_i^M are the predicted by the forward model concentrations at the sensor locations i, C_i^E are the sensor measurements, σ_{rel}^2 is the standard deviation of the combined forward model and measurement errors, N is the number of sensors. The value of σ_{rel}^2 can vary depending on the observation errors and model formulation (the assumed here value is given in chapter 3).

After calculating value of the likelihood function for the proposed state its acceptance is performed as follows:

$$\frac{ln(\lambda_{prop})}{ln(\lambda)} \geq U(0,1) \tag{4}$$

where λ_{prop} is the likelihood value of the proposal state, λ is the previous likelihood value, and $U(0,1)$ is a random number generated from a uniform distribution in the interval $(0,1)$.

It is important to note that condition (4) is more likely to be satisfied if the likelihood of the proposal is only slightly lower than the previous likelihood value. It gives a chance to choose even a little "worse" state, because the probability of acceptance depends directly on the quality of proposed state. Different likelihood functions can also be applied [13].

2.3 Posterior distribution

The posterior probability distribution (2) is computed directly from the resulting samples defined by the algorithm described above and is estimated with

$$P(M|D) \equiv \hat{\pi}^N(M) = \frac{1}{N}\sum_{i=1}^{N}\delta(M_i - M). \tag{5}$$

$P(M|D)$ represents the probability of a particular model configuration M giving results that match the observations at sensors locations. Equation (5) is a sum over the entire samples set of length N of all the sampled values M_i. Thus $\delta(M_i - M) = 1$ when $M_i = M$ and 0 otherwise. Consequently, if a Markov chain spends several iterations at the same location value of $P(M|D)$ increases through the summation (increasing the probability for those source parameters).

2.4 Sequential Monte Carlo

Sequential Monte Carlo (SMC) is designed to sample from dynamic posterior distributions. The SMC methods are easy to parallelize - the different Monte Carlo proposals can be generated and evaluated in parallel. A good introduction to SMC is present in [14, 15, 16] .

2.5 Sequential importance resampling

Sequential importance resampling (SIR) is a sequential version of importance sampling (IS) and combines IS with resampling procedure [17] . At the center of the SMC approach in our case is the generation of a weighted sample using IS method. IS uses a proposal distribution $q(.)$, that is close to target distribution $\pi(.)$ and from which it is easy to generate samples. The basic methodology is given below.

1. Generate a sample of size N from the proposal distribution $q(M)$:

$$M_{(i)} \sim q(M), i = 1, ..., N \tag{6}$$

2. Compute the importance weights:

$$\check{w}(M_{(i)}) \propto \frac{\pi(M_{(i)})}{q(M_{(i)})}, i = 1, ..., N \tag{7}$$

and define

$$w(M_{(i)}) = \frac{\check{w}(M_{(i)})}{\sum_{j=1}^{N} \check{w}(M_{(j)})} \tag{8}$$

3. The distribution $\pi(\cdot)$ is then approximated by

$$\check{\pi}^N(M) \equiv \sum_{i=1}^{N} w(M_{(i)})\delta(M_i - M) \tag{9}$$

which places the probability mass $w(M_{(1)}), ..., w(M_{(N)})$ on the support points $M_{(1)}, ..., M_{(N)}$.

Hence, the weights would be proportional to the value of likelihood. In our case to calculate the weight we use of the following formula,which is related to the likelihood function (3):

$$\check{w}(M_{(i)}) \propto -\frac{1}{ln[\lambda(M_{(i)})]}, i = 1, ..., N \tag{10}$$

Resampling is used to avoid the situation when almost all (except only a few) of the importance weights are close to zero (problem of degeneracy of the algorithm). Basic idea of resampling methods is to eliminate samples which have small normalized importance weights and to concentrate upon samples with large weights. So,:

1. for $i = 1, ..., N$ are chosen samples with indexes $k(i)$ distributed according to the discrete distribution with N elements satisfying

$$P(k(i) = l) = w(M_{(i)}) \tag{11}$$

for $l = 1, ..., N$,

2. then for $i = 1, ..., N$ for samples $M_{k(i)}$ are assigned the weights

$$w(M_{k(i)}) = \frac{1}{N}. \tag{12}$$

2.6 MCMC prior to SMC

The SMC algorithm needs some set of samples to be initialized. An ideal way to generate this initial sample is using MCMC data from first K iterations in all time steps. The resulting equally weighted MCMC set of samples can then be passed on to SMC for processing in the subsequent iteration.

We assume that the information from the sensors arrive subsequently in intervals (time steps). We start to search for the values of the model parameters M ($M \equiv M(x, y, q, \zeta_1, \zeta_2)$ for details see chapter 3) after first sensors' measurements (based on the data in time $t = 1$, see (Fig. 2)). Thus, scanning algorithm is run just after obtaining the first measurements from the sensors. Based on this information we obtain the probability distributions of the searched parameters (9) starting from the randomly chosen set of parameters M (i.e. first we start from the "flat" priori). This assumption reflects lack of knowledge about the release. The forward calculation are performed for the actual state M and likelihood function λ is calculated. Then we apply random walk procedure "moving" our Markov chain to the new position. Precisely, we change each model M parameter by the value draw from the Gaussian distribution with the zero mean and variance σ^2_M characteristic for each parameter. Standard deviations for sampling parameters are determined by the problem's domain size and refined with a trial and error procedure to ensure that the Markov chains had access to realistic ranges with minimal occurrences of stuck problem. Problem of stuck in chains can occur when the standard deviations chosen for the next iteration lead to a large number of rejected samples, causing that the chain remains in a given position for many iterations. For the proposal state the forward calculation are performed and the likelihood function λ_{prop} is again estimated. We compare this two values λ and λ_{prop} according to (4). If comparison is more favorable than the previous chain location, the proposal is accepted (Markov chain "moves" to the new location). If the comparison is "worse", new state is not immediately rejected. Random variable from binomial distribution is used to decide whether or not to accept the new state of chain. After K iteration we pass all the samples (from all m chains)to the sequential procedure. We compute importance weights by (10) and normalize them. Next we use roulette procedure to draw N samples from the set generated by Markov Chain.

This random component is important because it prevents the chain from becoming trapped in a local minimum. The pseudo code for one time step of the algorithm is given below.

2.6.1 Pseudocode 1:MCMC Prior to SMC

```
FOR j=1:ChainNumber
    Draw M(1) from a priori distribution;
    ForwardDispersion(M(1));
```

```
    Read C^M(1);
    Compute likelihood(M(1));
    Compute W(1);
    FOR i=1:K
        ChainSample(j,i)=M(i);
        ChainWeight(j,i)=W(i);
        M'=M(i)+N(0,sigma2M);
        ForwardDispersion(M');
        Read C^M';
        Compute likelihood(M');
        Compute W';
        IF likelihood(M')/likelihood(M(i-1))
                >=RND(0,1)
        THEN
            M(i+1)=M';
            W(i+1)=W';
        ELSE
            M(i+1)=M(i);
            W(i+1)=W(i);
        END IF
    END FOR
END FOR
SAMPLES=[ChainSample(1,:) ... ChainSample(ChainNumber,:)]
WEIGHTS=[ChainWeight(1,:) ... ChainWeight(ChainNumber,:)]
WEIGHT=WEIGHT/sum(WEIGHT)
FOR i=1:N
    SumOfWeights=0;
    RuletIter=1;
    rand=RND(0,1);
    WHILE(SumOfWeights<=rand)
        SumOfWeights=SumOfWeights+WEIGHTS(RuletIter);
        RuletIter++;
    END WHILE
RESAMPLES(i)=SAMPLES(RuletIter)
END FOR
```

Statistical convergence (to the posterior distribution) is monitored by computing between-chain variance and within-chain variance [11]. If there are m Markov chains of length N, then we can compute between-chain variance B with

$$B = \frac{N}{m-1}\sum_{j=1}^{m}(\bar{M}_j - \bar{M})^2 \tag{13}$$

where $\bar{M}_j$ is the average value along each Markov and $\bar{M}$ is the average of the values from all Markov chains. The within-chain variance W is

$$W = \frac{1}{m}\sum_{i=1}^{m} s_i^2 \tag{14}$$

where

$$s_i^2 = \frac{1}{N-1}\sum_{i=1}^{N}(M_{ij} - \bar{M}_i)^2 \tag{15}$$

The convergence parameter R is then computed as

$$R = \frac{var(M)}{W} \tag{16}$$

where $var(M)$ is estimate variance of M and is computed as

$$var(M) = \frac{N-1}{N}W + \frac{1}{N}B. \tag{17}$$

In this paper, we consider the following variants of scanning algorithms:

1. **Classic MCMC**

   In this algorithm, the parameter space scan in each time step t is independent form the previous ones. So, in this case we don't use information from past calculations. Classic MCMC don't use sequential mechanism.

2. **MCMC prior to SMC**

   The SMC algorithm use the set of samples generated by K iterations of Clasic MCMC algorithm as a prior distribution, but in subsequent SMC iterations don't use information from SMC results from previous time step.

3. **MCMC prior to SMC via Maximal Weights**

   This algorithm is similar to MCMC prior to SMC, but in subsequent SMC calculations uses the results obtained by SMC in the previous time steps to run calculation with use of the new measurements. As the first location of Markov chain M_0^t it select the set of M parameters for which weight in previous time step procedure was the highest. So, for $t > 1$:

   $$M_0^t \sim \ arg\ (M \in \{M_0^{t-1}, ..., M_n^{t-1}\})\ max\{w(M_i^{t-1})\} \tag{18}$$

   With this approach, we always start with the best values of the model (previously found) and correct the result with new information from sensor.

4. **MCMC prior to SMC via Rejuvenation and Extension**

   In contrast to the MCMC prior to SMC via Maximal Weights this algorithm as the first location of Markov chain M_0^t at the time $t > 1$ chooses the set of parameters M selected randomly from previous realization of resampling procedure in $t-1$ with use of the uniform distribution:

   $$M_0^t \sim U(M_0^{t-1}, M_1^{t-1}, ..., M_n^{t-1})\ a\ uniform\ distribution\ \{1, ..., n\} \tag{19}$$

   Applying the new knowledge (new measurements) the current chain is "extended" starting from selected position with use of the new data in the likelihood function calculation.

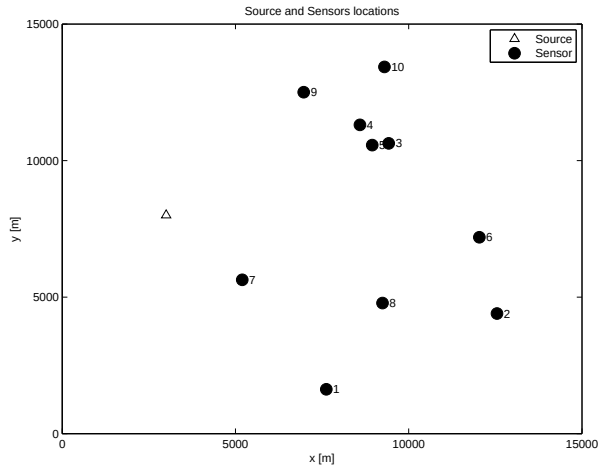


FIGURE 1: Distribution of the sensors and the release's source over the domain

3 Synthetic data

We have implemented stochastic event reconstruction algorithms grounded on the hybrid MCMC and SMC sampling to find the contamination source location based on the concentration of given substance registered by the 10 sensors distributed randomly over artificial domain 15000m x 15000m. (Fig. 1). The contamination source was located at $x = 3000m$, $y = 8000m$, $z = 30m$. The synthetic concentration measurements data (Fig. 2), used in testing the algorithm, were generated with use of the atmospheric dispersion Gaussian plume model [18], [19]. The release rate was assumed to change with time within interval $q \approx 5000g/s$ up to $q \approx 7000g/s$ which resulted in the change of the concentration measured by the sensors in subsequent time intervals (Fig. 2).The wind was directed along x axis with average speed $5m/s$.

The Markov chains are initialized by taking samples from the prior distribution.For practical reasons and to lower the computational cost we limit the prior distribution to the two dimensional coordinate space (x, y) of the source location. The vertical position of source location was fixed on $z = 30m$ at which were also located the sensors.

In our calculation we use $m = 10$ Markov chains in each time step in MCMC procedure. The traces of three independent Markov chains for the x and y parameters are presented in Fig. 3 and Fig. 4, horizontal line represents the target value. Fig. 5 presents the chain's traces in the two dimensional plane within the scanned domain, the target source location is marked by triangle and sensors by squares.The variance parameters σ_M^2 uses in random walk procedure are equal $\sigma_x^2 = 200$, $\sigma_y^2 = 200$, $\sigma_q^2 = 100$ and $\sigma_{\zeta_1}^2 = 0.02$, $\sigma_{\zeta_2}^2 = 0.02$. In this case, when we use in the reconstruction the synthetic data, the value $\sigma_{rel}^2 = 0.05$, because disorder of measurement data was low and assumed as 5%. For real measurements, if large errors in the measurements are expected, larger values of σ_{rel}^2 should be assumed.

The number of iteration for each Markov chain in Hybrid algorithm was equal $K = 10000$ (for comparision in classic MCMC $N = 20000$ to balance the number of iteration).This number was chosen based on the numerical experiments as the number of iteration needed to reach convergence for each sampled model parameters [20].One of the important aspects of stochastic procedure of calculating the posterior distribution is

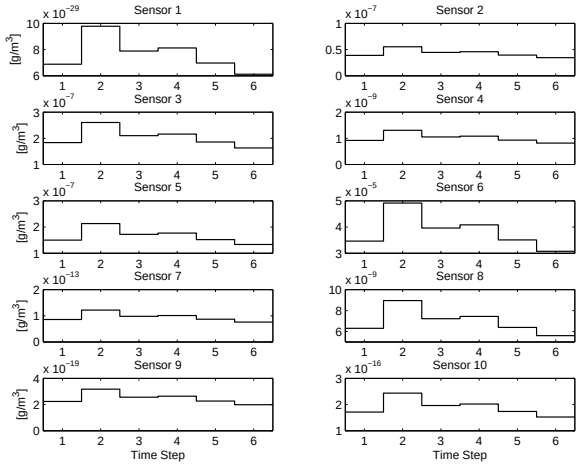


FIGURE 2: The synthetic concentration registered by the 10 sensor in 6 subsequent intervals (time steps)

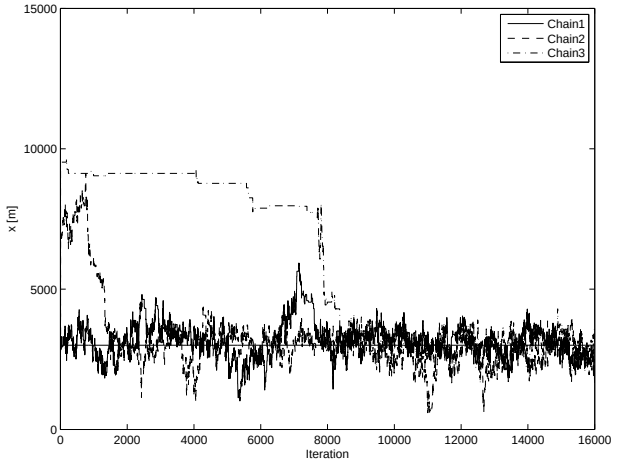


FIGURE 3: The traces of three Markov chains in the x space. The true value is marked by horizontal line. The samples came from results of Classic MCMC algorithm.

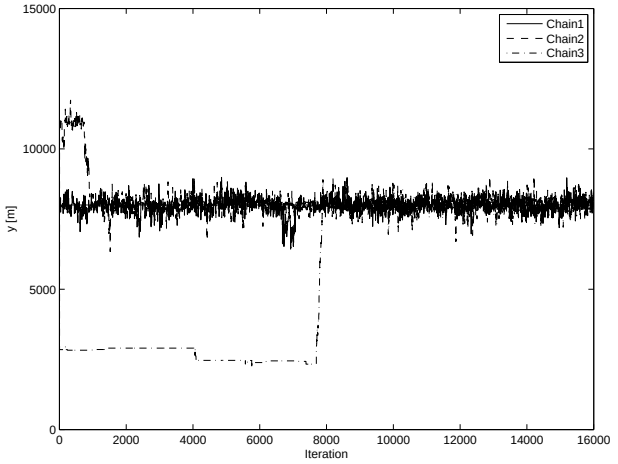


FIGURE 4: The traces of three Markov chains in the y space. The true value is marked by horizontal line. The samples came from results of Classic MCMC algorithm.

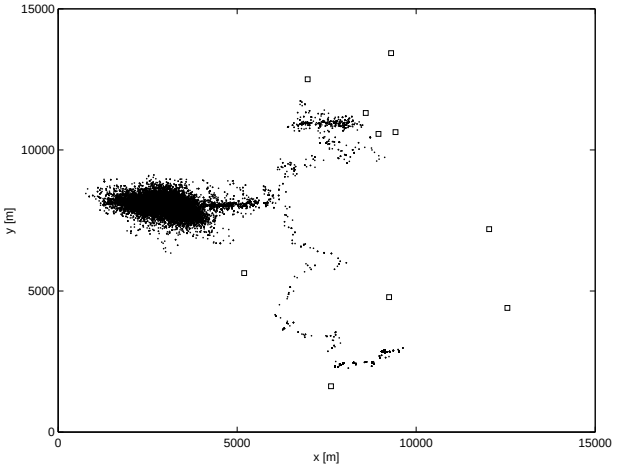


FIGURE 5: The traces of three Markov chains in the x,y space. The source location is marked by triangle and the sensors by squares. The samples came from results of Classic MCMC algorithm.

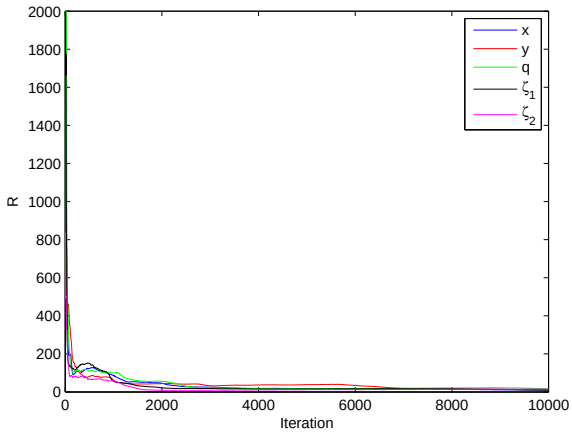


FIGURE 6: Convergence rates for position x. The samples came from results of MCMC algorithm.

choosing burn-in phase. The burn-in factor represents the number of samples needed at the beginning for the Markov chain to actually reach the search state where it is sampling from the target distribution. These initial samples are discarded and not used for inference. In our calculation the burn-in was fixed at 2000 iterations. This value was chosen based on the numerical experiments as the number of iteration needed to reach the target distribution with same approximation [20].The convergence R value vs. the number of iteration for searched parameters presents Fig. 6. One can see that the 10000 iterations satisfy the convergence condition $R \approx 1$.

3.1 Forward dispersion model

A forward dispersion model is needed to calculate the concentration C_i^M at the points i of sensors locations for the tested set of model parameters M at each Markov chain step. As a testing forward model we selected the fast-running Gaussian plume dispersion model [18],[19].

The Gaussian plume dispersion model for uniform steady wind conditions can be written as follows:

$$C(x,y,z) = \frac{q}{2\pi\sigma_y\sigma_z V} exp\left[-\frac{1}{2}\left(\frac{y}{\sigma_y}\right)\right] \times \{exp\left[-\frac{1}{2}\left(\frac{z-H}{\sigma_z}\right)^2\right] + exp\left[-\frac{1}{2}\left(\frac{z+H}{\sigma_z}\right)^2\right]\} \quad (20)$$

where $C(x,y,z)$ is the concentration at a particular location, V is the wind speed directed along x axis, q is the emission rate or the source strength and H is the height of the release; y and z are the distance along horizontal and vertical direction, respectively. In the equation (20) σ_y and σ_z are the standard deviation of concentration distribution in the crosswind and vertical direction. These two parameters were defined empirically for different stability conditions in [21] and [22]. In this case we restrict the diffusion

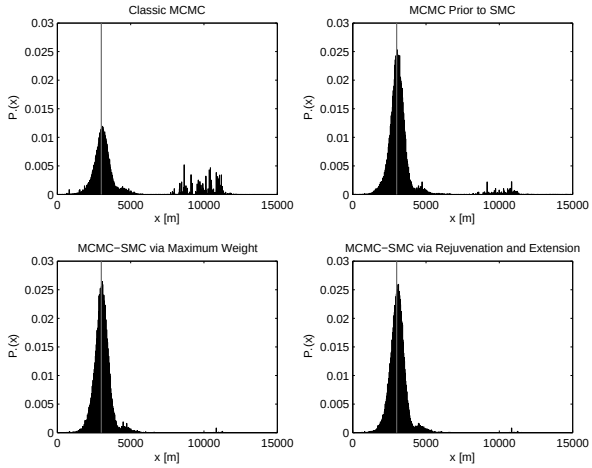


FIGURE 7: Posterior distribution as inferred by the Bayesian event reconstruction for all applied algorithms for x parameter. Posterior distributions were averaged based on the data for all time steps. Vertical lines represent the target x value.

to the stability class C (Pasquill type stability for rural area). Thus, in creation of the synthetic data we have fixed these coefficients as:

$$\sigma_y = 0.22x \cdot (1 + x \cdot 4 \cdot 10^{-5})^{-0.5}, \sigma_z = 0.2x. \tag{21}$$

However, we assume in scanning algorithm that we do not know exact behavior of the plume and consider those coefficients as not completely known. Thus, the parameters σ_y,σ_z are taken as:

$$\sigma_y = \zeta_1 \cdot x \cdot (1 + x \cdot 4 \cdot 10^{-5})^{-0.5}, \sigma_z = \zeta_2 \cdot x \tag{22}$$

where values ζ_1 and ζ_2 are sampled by scanning algorithm within interval $[0, 0.4]$. The size of the sampling interval is directly related to the choice of Pasquill Stability Class [21].
To summarize, in this paper the searched model's parameters' space is

$$M \equiv M(x, y, q, \zeta_1, \zeta_2) \tag{23}$$

where x and y are spatial location of the release, q release rate and ζ_1, ζ_2 are stochastic terms in the turbulent diffusion parameterization given in (22).

3.2 Results

All algorithms described in chapter 2.6 have been tested on the same synthetic data set. Figs. 7, 8 and 9 presents the results of calculation with use of all four above described algorithms for x, y and ζ_1 parameters. Presented distributions were calculated based on the scanning algorithms results from all time steps and all generated samples.

One can see from Figs. 7 and 8 that the classic MCMC algorithm have some the unwanted samples of x and y in ranges $x \in (1000, 1200)$ $y \in (2000, 3000)$ and marks them with a bit higher probability. MCMC prior to SMC algorithm also shows some local minima but with a lower probability value. At the same time all other algorithms

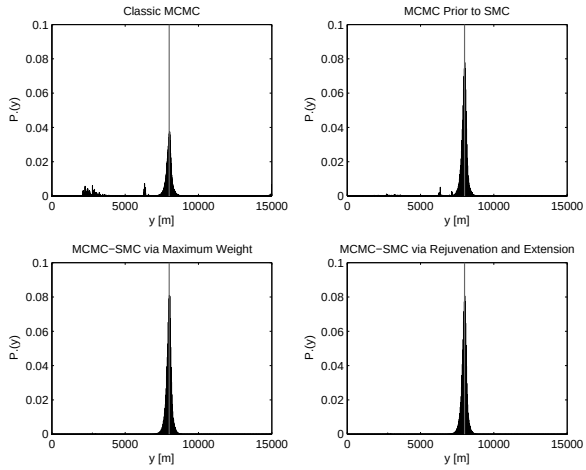


FIGURE 8: Posterior distribution as inferred by the Bayesian event reconstruction for all applied algorithms for y parameter. Posterior distributions were averaged based on the data for all time steps. Vertical lines represent the target y value.

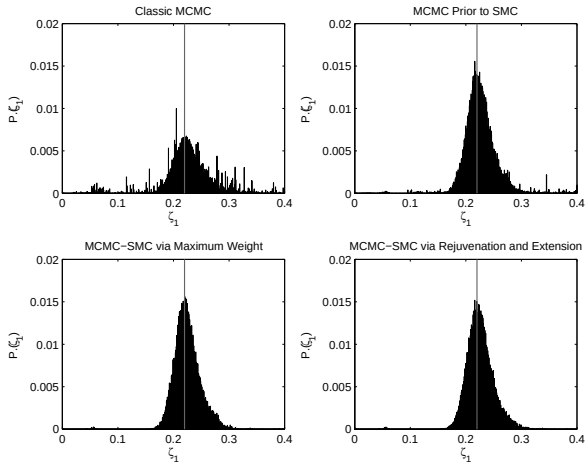


FIGURE 9: Posterior distribution as inferred by the Bayesian event reconstruction for all applied algorithms for ζ_1 parameter. Posterior distributions were averaged based on the data for all time steps. Vertical lines represent the target ζ_1 value.

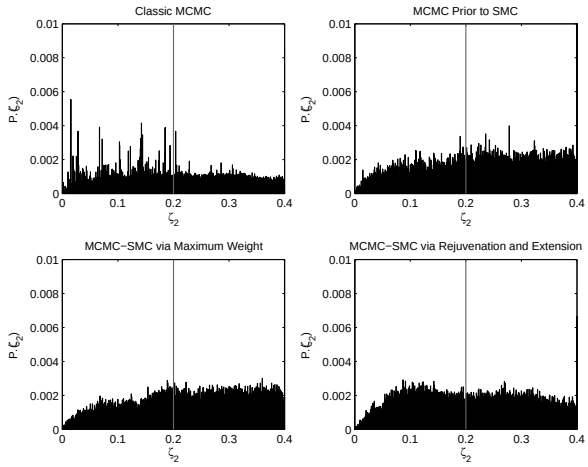


FIGURE 10: Posterior distribution as inferred by the Bayesian event reconstruction for all applied algorithms for ζ_2 parameter. Posterior distributions were averaged based on the data for all time steps. Vertical lines represent the target ζ_2 value.

reach the target value of x and y with a quite smooth and balance empirical distributions. The same in seen for the ζ_1 parameter (Fig. 9), but here the Classical MCMC algorithm do not mark the target value of ζ_1 as the most probable, while all other algorithms successfully hit its target value. The reason of the high peek in the histograms presented in Fig. 8 is that y is the crosswind direction, and applied model is quite sensitive to this parameter. In contrast, all methods do not find the target value of the ζ_2 parameter (Fig. 10) being responsible for the dispersion in vertical direction. We do not consider the probability of the release rate distribution, as far it was changing during creation of the synthetic data.

Figs. 11 and 12 presents the probability distributions of x parameter obtained in subsequent time steps by classic MCMC and SMC via Rejuvenation and Extension, respectively. One can see that in case of SMC via Rejuvenation and Extension algorithm with time the probability of the target value is reached with higher probability. Whereas in classic MCMC algorithm we observe unwanted peaks at each time step. With subsequent time steps, the algorithm MCMC prior to SMC via Rejuvenation and Extension eliminates samples with small weights, thus improves the quality of the desired distribution (e.g. local maximum is reduced to $x \in (4000, 5000)$). Confirmation of vanishing samples with small weights can be seen in Fig. 13. In first time step we can observe some outliers, while in the following steps their are discarded. A similar situation occurs for algorithms MCMC prior to SMC via Maximal Weights and MCMC prior to SMC.

4 Final conclusions

We have presented a methodology to localize a source causing an area contamination, based on a set of downwind concentration measurements. The method combines Bayesian inference with sequential Monte Carlo techniques and produces posterior probability distributions of the parameters describing the unknown source. The approach successfully provide the solution to the stated inverse problem i.e. having the downwind concentra-

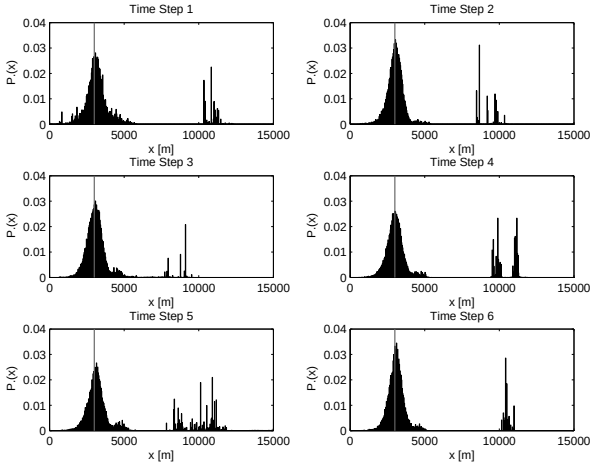


FIGURE 11: Posterior distribution of x parameter in subsequent time steps for classic MCMC algorithm. Vertical line represents the target value of x.

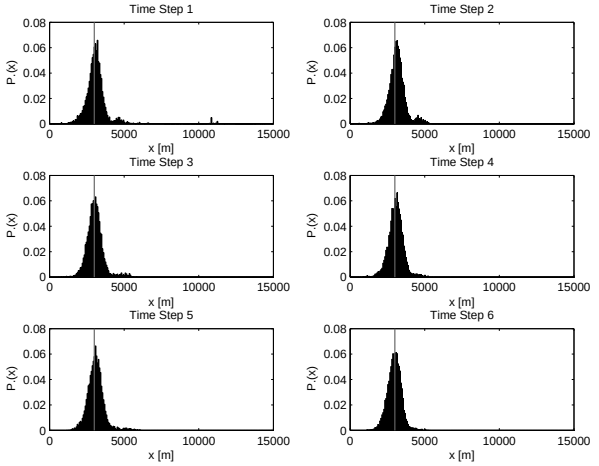


FIGURE 12: Posterior distribution of x parameter in subsequent time steps for MCMC prior to SMC via Rejuvenation and Extension algorithm. Vertical line represents the target value of x.

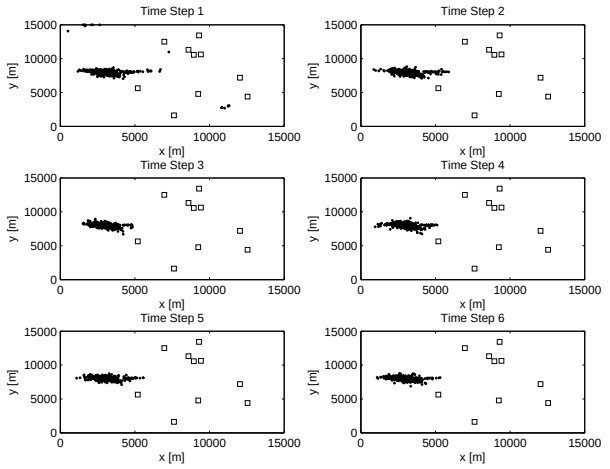


FIGURE 13: Scatter plot of all samples in subsequent time steps for MCMC prior to SMC via Rejuvenation and Extension algorithm. Squares represents the sensors.

tion measurement and knowledge of the wind field algorithm founds the most probable location of the source.

We have examined various version of the Hybrid SMC with MCMC algorithms i.e. classic MCMC, MCMC prior to SMC, MCMC prior to SMC via Rejuvenation and Extension, MCMC prior to SMC via Maximal Weights in effectiveness to estimate the probabilistic distributions of searched parameters. We have shown the advantage of the algorithms that in different ways use the source location parameters probability distributions obtained basing on available measurements to update the marginal probability distribution. As the most effective we pointed the modifications of MCMC prior to SMC.

The stochastic approach used in this paper is completely general and can be used in other fields where the parameters of the model best fitted to the observable data should be found.

Acknowledgments.

This work was supported by the Welcome Programme of the Foundation for Polish Science operated within the European Union Innovative Economy Operational Programme 2007-2013 and by the EU and MSHE grant nr POIG.02.03.00-00-013/09.

References

[1] Thomson, L. C., Hirst, B., Gibson, G., Gillespie, S., Jonathan, P., Skeldon, K. D., Padgett, M. J.: An improved algorithm for locating a gas source using inverse methods. Atmospheric Environment, 41, 1128–1134, (2007)

[2] Ivakhnenko, A.G.: Group method of data Handling- A Rival of the Method of Stochastic Approximation, Soviel Automatic Control, 13, 43–71, (1966)

[3] Madala H.R., Ivakhnenko A.G.: Inductive Learning Algorithms for Complex Systems Modeling, CRC Press, (1994)

[4] Fujimoto, K., Nakabayashi S.: Applying GMDH algorithm to extract rules from examples, Systems Analysis Modelling Simulation, 43, 10, 1311–1319, (2003)

[5] Puiga V.Witczak M.,Nejjari F.,Quevedo J.,Korbicz J.: A GMDH neural network-based approach to passive robust fault detection using a constraint satisfaction backward test, Engineering Applications of Artificial Intelligence, 20, Issue 7, 886–897, (2007)

[6] Watzenig,D.: Bayesian inference for inverse problems - statistical inversion. Elektrotechnik and Informationstechnik ,124/7/8, 240, (2007)

[7] Keats, A., Yee, E. ,Lien, F.S.: Bayesian inference for source determination with applications to a complex urban environment. Atmos. Environ., 41,465–479, (2007)

[8] Pudykiewicz, J. A.: Application of adjoint tracer transport equations for evaluating source parameters. Atmos. Environ., 32, 3039–3050, (1998)

[9] Johannesson, G. et al.: Sequential Monte-Carlo based framework for dynamic data-driven event reconstruction for atmospheric release, Proc. of the Joint Statistical Meeting, Minneapolis, MN, American Statistical Association and Cosponsors, 73–80, (2005)

[10] Gilks, W., Richardson, S., Spiegelhalter, D.: Markov Chain Monte Carlo in Practice. Chapman & Hall/CRC,(1996)

[11] Gelman, A.,Carlin, J.,Stern, H.,Rubin, D.: Bayesian Data Analysis, Chapman & Hall/CRC, (2003)

[12] Bernardo, J. M., Smith, A. F. M.:Bayesian Theory, Wiley, (1994)

[13] Senocak I.,Hengartner, N. W., Short, M. B. ,Daniel,W. B.:Stochastic Event Reconstruction of Atmospheric Contaminant Dispersion Using Bayesian Inference, *Atmos. Environ.* 42(33) 7718–7727, (2008)

[14] Doucet, A., de Freitas, J. F. G., Gordon, N. J.: Sequential Monte Carlo methods in practice. New York: Springer-Verlag, (2001)

[15] Pitt, M. K. Shephard, N.: Sequential Monte Carlo methods in practice, chapter Auxiliary variable based particle filters. New-York: Springer-Verlag, (2001)

[16] Liu, J. S.: Monte Carlo Strategies in Scientific Computing. New York: Springer, (2001)

[17] Gordon, N. J.; Salmond, D. J., Smith, A. F. M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEEE Proceedings F on Radar and Signal Processing 140 (2): 107–113, (1993)

[18] Panofsky, H. A., Dutton, J. A.: Atmospheric Turbulence, John Wiley, (1984)

[19] Turner D. B.: Workbook of Atmospheric Dispersion Estimates, *Lewis Publishers*, USA, (1994)

[20] Borysiewicz, M., Wawrzynczak A.,Kopka P.:Stochastic algorithm for estimation of the model's unknown parameters via Bayesian inference, Proceedings of the Federated Conference on Computer Science and Information Systems pp. 501–508, IEEE Press, Wroclaw, ISBN 978-83-60810-51-4, (2012)

[21] Pasquill, F.: The estimate of the dispersion of windborne material, *Meteorol Mag.*,90, 1063,: 33–49, (1961)

[22] Gifford, F. A. Jr.: Atmospheric dispersion calculation using generalized Gaussian Plum model, *Nuclear Safety*, 2(2):56–59,67–68, (1960)

Generalized autosort FFT framework

Michał Lenarczyk*

Instytut Podstaw Informatyki PAN

Abstract

In this article, the design of a mixed radix fast Fourier transform (FFT) algorithm is presented. The algorithm is general enough to be able to perform both decimation in time and decimation in frequency decompositions, and has some other advantages. It is argued it best fits signal processors, mainly because of small code footprint and efficient use of modulo addressing modes available on some DSP architectures.

1 Introduction

Among many methods for efficient computation of the discrete Fourier transform, DFT, the algorithm known as the fast Fourier transform is of greatest importance. Given data of composite size, MN, the discrete Fourier transform of the data can be reexpressed in terms DFT's of orders M and N repeated N and M times, respectively, which requires evaluation of $MN(M+N)$ terms as opposed to $(MN)^2$ when using direct formula, and this count can be further reduced by factoring M and/or N into even smaller numbers. The factorization can be done in arbitrary way, but two cases are especially useful : these are referred to as "decimation in time" and "decimation in frequency". Thus, the FFT computes the transform of composite order by a sequence of smaller transforms whose sizes are its divisors.

This approach became widely used after the famous publication of Cooley and Tukey [1] who obtained the decimation in time method for arbitrary factorizations of the transform order (mixed radix), giving special attention to the case where the order is a power of 2 (*i.e.*, radix 2). Gentleman and Sande [2] found the alternative, decimation in frequency form and also showed how reordering of data can be avoided given auxiliary storage, which is the first reported autosort framework. A particularly efficient algorithm known as "split radix" was proposed by Duhammel and Hollman [3] which is present state-of-the-art for orders containing factors 4. Of particular research interest is the adaptation of the FFT for efficient execution in vector and distributed processing architectures. More details on this subject can be found, *e.g.* in the handbook by Van Loan [9].

FFT efficiency has traditionally been measured in terms of arithmetic operations involved in its execution - the total number of multiplications, of multiplications and additions, or the greater of the two counts (see, for instance, [4]). It is

*The algorithm was developed in the time when the author was with AGH University of Science and Technology, Kraków

meaningful when the cost of these operations is dominant, which was the case in early computer architectures where especially multiplications were time consuming, and is still applicable to digital signal processors which are able to perform arithmetic calculations in parallel with other operations such as memory access, looping / execution flow, and pointer updates, which effectively come at no cost. However, in most modern general purpose architectures with slow memory and fast ALU and cache, structural overhead and memory access present a nonnegligible cost, and counting arithmetic operations may not reflect actual performance.

Recently, gains in performance have been achieved mainly by the use of larger DFT modules that can be highly optimized by hand coding. This can be viewed as roughly equivalent to loop unrolling. Decomposition into larger radices that need not be prime, has the advantage of having fewer stages, but the resulting programs tend to be large. For example, the popular software package created by Frigo and Johnson [5] uses "small" DFT modules of orders up to 64 using straightforward code with no loop structure. A similar approach led Van Buskirk [6] to improve the code of split radix and reduce the number of required multiplications, which is so far the lowest known count (more details on the method, which does not require unrolling, can be found in [7]).

While the pursuit of the lowest possible count of additions or multiplications is interesting from theoretical point of view, the approach of using large, optimized modules is neither practical nor well suited for implementation on signal processors (DSP's). Firstly, these processors usually are limited in terms of program memory ; secondly, DSP's have very low structural overhead (typically in the range 3-15% [8]), which questions the sense of loop unrolling. For example, implementation of the single 64-point DFT code with Van Buskirk's optimization requires 912 additions and 240 multiplications, the amount that exceeds the capacity of small DSP's with only a few kilowords of program memory. Even for devices with large internal memory, its amount puts a practical limit on the number of elementary DFT modules of only a few smallest orders. On the other hand, various signal processing architectures provide specific features that allow writing efficient FFT programs that are compactly structured as well.

This article introduces a general, out-of-place, mixed-radix framework for FFT that combines autosort property with the ability to incorporate linear time and frequency shifts, which can be useful e.g. for centering the spectrum or for phase rectification. The generality of the algorithm allows it to be configured both in the decimation in time and decimation in frequency manner. A link is made to practical machine implementation issues with emphasis on DSP architectures ; in particular, a novel normalization scheme is proposed for fixed-point arithmetic. The algorithm is accompanied with a proof of concept implementation in C for verification and preliminary analyses. The code has a very small footprint, as it only contains a single generic version of DFT of arbitrary order, plus an optional routine for setting up the FFT structure at run time. When implemented in native assembly, the algorithm could complement library support of some DSP families that only supply code for FFT of radix 2.

It should be pointed out that DFT with shifted indices was first proposed by Bongiovanni *et al.* [10] who termed it "generalized discrete Fourier transform" ; quite recently, generalization to nonlinear phase DFT was proposed (see Akansu

and Agirman-Tosun [11]) but is, however, not as yet permitted in this framework.

The derivation for the algorithm and analysis of its properties is presented in the next section. Section 3 considers practical implementation aspects. Finally, some concluding remarks are stated in section 4.

2 Theory and interpretation

This section focuses on how the FFT is obtained and discusses some of its properties, starting with facts well known from literature. After deriving the basic decomposition and its forms in the first paragraph, data arrangement and ordering is studied in paragraph 2.2 where an intuitive three-dimensional array interpretation is presented. Generalization of the DFT is introduced in paragraph 2.3 that allows to incorporate linear time and frequency shift into the body of the FFT, and in particular, to avoid explicit multiplication by twiddle factors. The last paragraph 2.4 considers scaling issues in the context of fixed-point arithmetic and proposes a novel approach to this question.

2.1 Derivation of the FFT

Given a signal of composite size MN, the discrete Fourier transform sum

$$\hat{x}_k = \sum_{n=0}^{MN-1} x_n \omega^{nk} \tag{1}$$

where $\omega^{nk} = e^{-i\frac{2\pi}{MN}nj}$ can, by substitution of $Nm+n$ and $Mk+j$ for original n and k, respectively, be expressed as a double summation :

$$\begin{aligned} \hat{x}_{Mk+j} &= \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} x_{Nm+n}\omega^{(Nm+n)(Mk+j)} \\ &= \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} x_{Nm+n}\omega^{Nmj}\omega^{nj}\omega^{Mnk} \end{aligned} \tag{2}$$

The innermost sum of the terms $x_{Nm+n}\omega^{Nmj}$ can be recognized as an M-point DFT over the decimated signal with argument m and offset parameter n. The outer summation is an N-point DFT over the outcome of the former transform multiplied by trigonometric phase shift coefficients ω^{nj} which are customarily referred to as *twiddle factors*, a term introduced by Gentleman and Sande [2]. Each DFT sequence spanning every possible value of the parameter which is not the index of summation will be referred to as DFT pass. The count of terms included in the double summation can be estimated as M^2N for the inner and MN^2 for the outer pass, which gives the overall formula $MN(M+N)$ for the total number of terms in both DFT passes. Considering the number of multiplications, the one-time scaling by twiddle factors requires additionally MN complex multiplications, which results in a total count of $MN(M+N+1)$ complex multiplications. The increase can be avoided if the twiddle factors are absorbed in the computation of either of the two DFT's. This idea is elaborated in paragraph 2.3 of this section.

If either of the numbers M or N is composite, further decomposition is possible and highly composite orders can be decomposed into prime factors in many ways. However, if at every step, the decomposition is made such that one selected dimension is always prime (or a number accepted as radix, *e.g.* 4), then an especially regular structure of the algorithm is obtained. If, at every step, M is prime (or non-decomposable) then the *decimation in time* results (the name reflecting the fact that decimated time index m and contiguous frequency index j are split into pairs of new indices in next pass) ; on the other hand, if the same is assured for N, then the *decimation in frequency* is obtained (decimated frequency index k and contiguous time index n are split further) - following the naming convention introduced in [12]. The resulting forms of the equation are equivalent ; the distinction concerns the association of the twiddle factors with either of the DFT passes. In case of decimation in time, multiplication by twiddle factors affects input before running the next DFT pass, whereas in decimation in frequency, the output of the DFT pass is multiplied. Apart from having some impact on implementation, the two approaches are identical.

In the following we consider the problem of composite order $N_0 = \prod_{l=1}^{L} M_l$ with radix M_l in pass l and set $N_k = \prod_{l=k+1}^{L} M_l$ so that the factorization at l-th level of decomposition is $N_{l-1} = M_l N_l$. It can be shown that in this case the overall complexity reduces to $N_0 \times \sum_l M_l = (\prod_l M_l) \times (\sum_l M_l)$ [1].

Following this convention, the first step of decomposition is $N_0 = M_1 N_1$ and for decimation in time the following index split is performed

$$\begin{aligned} n_0 &= M_1 n_1 + m_1, \quad n_1 = 0, \ldots, N_1 - 1, m_1 = 0, \ldots, M_1 - 1 \\ k_0 &= N_1 j_1 + k_1, \qquad k_1 = 0, \ldots, N_1 - 1, j_1 = 0, \ldots, M_1 - 1 \end{aligned} \tag{3}$$

or, for arbitrary level of decomposition l

$$\begin{aligned} n_{l-1} &= M_l n_l + m_l, \quad n_l = 0, \ldots, N_l - 1, m_l = 0, \ldots, M_l - 1 \\ k_{l-1} &= N_l j_l + k_l, \qquad k_l = 0, \ldots, N_l - 1, j_l = 0, \ldots, M_l - 1 \end{aligned} \tag{4}$$

which yields the formula for the first stage of FFT as follows :

$$\hat{x}_{N_1 j_1 + k_1} = \sum_{m_1=0}^{M_1-1} \sum_{n_1=0}^{N_1-1} x_{M_1 n_1 + m_1} \omega^{M_1 n_1 k_1} \omega^{m_1 k_1} \omega^{N_1 m_1 j_1} \tag{5}$$

Note the subscripts of indices m, n, j and k in (3) and (4) refer to the level of decomposition. For decimation in frequency, the following index split is applied :

$$\begin{aligned} n_0 &= N_1 m_1 + n_1, \quad n_1 = 0, \ldots, N_1 - 1, m_1 = 0, \ldots, M_1 - 1 \\ k_0 &= M_1 k_1 + j_1, \qquad k_1 = 0, \ldots, N_1 - 1, j_1 = 0, \ldots, M_1 - 1 \end{aligned} \tag{6}$$

and, generally,

$$\begin{aligned} n_{l-1} &= N_l m_l + n_l, \quad n_l = 0, \ldots, N_l - 1, m_l = 0, \ldots, M_l - 1 \\ k_{l-1} &= M_l k_l + j_l, \qquad k_l = 0, \ldots, N_l - 1, j_l = 0, \ldots, M_l - 1 \end{aligned} \tag{7}$$

leading to the following form of the first stage of FFT :

$$\hat{x}_{M_1 k_1 + j_1} = \sum_{n_1=0}^{N_1-1} \sum_{m_1=0}^{M_1-1} x_{N_1 m_1 + n_1} \omega^{N_1 m_1 j_1} \omega^{n_1 j_1} \omega^{M_1 n_1 k_1} \tag{8}$$

2.2 Arrangement and indexing of data

The index split introduced in the previous section can be interpreted in terms of arrangement of the input array in two dimensions. Let the prime symbol $'$ denote the operation of forming an array with two index variables replacing the single index of the original sequence such that, for input array x :

$$x'_{(n_1,m_1)} = x_{M_1 n_1 + m_1} \tag{9}$$

Then the double summation formula in equation 5 (DIT form) becomes

$$\sum_{m_1=0}^{M_1-1} \sum_{n_1=0}^{N_1-1} x'_{(n_1,m_1)} \omega^{M_1 n_1 k_1} \omega^{m_1 k_1} \omega^{N_1 m_1 j_1} \tag{10}$$

which is equivalent to a two-dimensional discrete Fourier transform over an $N \times M$ array, except for the intervening twiddle factors. Recall that in decimation in time, n is the decimated (by N) and m the contiguous index. From (5) it can be observed that the corresponding output indices are used in reverse order : k is contiguous and j decimated (by M). Therefore, the output can be reinterpreted as a $M \times N$ array $\hat{x}'$ with the following indexing :

$$\hat{x}'_{(j_1,k_1)} = \hat{x}_{N_1 j_1 + k_1} \tag{11}$$

Consequently, one step of DFT factorization transposes data interpreted as a two dimensional array.

Consider now the second level of decomposition according to decimation in time approach. By posing $n_1 = M_2 n_2 + m_2$ and $k_1 = N_2 j_2 + k_2$ it is possible to define x'', a three-dimensional array indexed $x''_{(n_2,m_2,m_1)} = x_{M_1 M_2 n_2 + M_1 m_2 + m_1}$ and $\hat{x}''$, indexed $\hat{x}''_{(j_1,j_2,k_2)} = \hat{x}_{N_1 j_1 + N_2 j_2 + k_2}$. It is no longer possible to recover the output by a simple transposition : the result of two level decomposition is a superposition of two transpositions applied to different array formats.

A full decomposition into L factors would yield L-dimensional input array $x^{(L)}$ indexed with an addressing sequence

$$(m_L, m_{L-1}, \ldots, m_2, m_1) \tag{12}$$

corresponding to the unidimensional data index $M_1 M_2 \cdots M_{L-1} m_L + \ldots + M_1 m_2 + m_1$, and $\hat{x}^{(L)}$, addressed

$$(j_1, j_2, \ldots, j_{L-1}, j_L) \tag{13}$$

corresponding to $N_1 j_1 + N_2 j_2 + \ldots + N_{L-1} j_{L-1} + j_L$. The addressing sequence of the input and output are reversed, which can be accomplished in two ways.

Cooley's in-place algorithm [1] did not change index order before and after each pass, and required reordering of data samples, either before or after execution of the algorithm. The special case he considered where the radix is 2 can efficiently be done on binary machines that allow reversal of bits (some processors even feature reverse-carry addressing mode which obviate bit reversal).

Sande [2] realized the possibility of absorbing the transpositions in the course of subsequent DFT passes, to undo the effect of decompositions, at the cost of

auxiliary storage. The algorithm presented in this paper follows the same *autosort* approach, since, being inherently out-of-place, it immediately satisfies the requirement for additional storage.

Consider the first level of decimation in time decomposition, and define the intermediate array y', such that :

$$\hat{x}'_{(j_1,k_1)} = \sum_{m_1=0}^{M_1-1} y'_{(k_1,m_1)} \omega^{m_1 k_1} \omega^{N_1 m_1 j_1} \tag{14}$$

and

$$y'_{(k_1,m_1)} = \sum_{n_1=0}^{N_1-1} x'_{(n_1,m_1)} \omega^{M_1 n_1 k_1} \tag{15}$$

In the above decomposition, the output y' of the inner DFT sum has the same addressing as input, whereas the outer DFT performs transposition according to

$$(k_1, m_1) \rightarrow (j_1, k_1)$$

In the second level of factorization, the intermediate array y' from (15) is replaced by a new 3-dimensional array y'' as follows :

$$y''_{(j_2,k_2,m_1)} = \sum_{m_2=0}^{M_2-1} z''_{(k_2,m_2,m_1)} \omega^{M_1 m_2 k_2} \omega^{M_1 N_2 m_2 j_2} \tag{16}$$

$$z''_{(k_2,m_2,m_1)} = \sum_{n_2=0}^{N_2-1} x''_{(n_2,m_2,m_1)} \omega^{M_1 M_2 n_2 k_2} \tag{17}$$

where a new intermediate array z'' is introduced and again the inner DFT does not change the addressing and the outer DFT transposes the output according to :

$$(k_2, m_2, m_1) \rightarrow (j_2, k_2, m_1)$$

By following the above procedure, a full decomposition defined by the input addressing (12) and output addressing (13) would be obtained. It is not necessary, however, to expand the data into L dimensions. By considering the addressing of intermediate results of consecutive DFT passes from (12) to (13),

$$\begin{array}{ccccccc}
(& m_L, & m_{L-1}, & m_{L-2}, & \ldots, & m_2, & m_1 \ ) \\
(& j_L, & m_{L-1}, & m_{L-2}, & \ldots, & m_2, & m_1 \ ) \\
(& j_{L-1}, & j_L, & m_{L-2}, & \ldots, & m_2, & m_1 \ ) \\
(& j_{L-2}, & j_{L-1}, & j_L, & \ldots, & m_2, & m_1 \ ) \\
 & \vdots & \vdots & \vdots & & \vdots & \vdots \\
(& j_1, & j_2, & j_3, & \ldots, & j_{L-1}, & j_L \ )
\end{array}$$

it can be observed that, at every pass, the address sequence consists of indices of three kinds : the already transformed indices j, the as yet untransformed indices m and the single index m_l that is subject to transformation in the present pass l.

Therefore, it suffices to use three indices for every pass. Denote by n the currently transformed index m_l, and collapse all remaining indices m into a single combined index $\vec{m}$ and all j into $\vec{j}$ (either of which can be empty, namely, $\vec{j}$ is an empty sequence in the first pass and $\vec{m}$ is empty in the last pass). Then, the following transformation of addressing sequence is applied along with DFT computation :

$$(\vec{j}, n, \vec{m}) \rightarrow (k, \vec{j}, \vec{m})$$

This leads to an elegant interpretation of the transform in terms of three-dimensional arrays with indices j, n and m. The dimensions must be defined anew in every DFT pass ; the index k from the previous pass is fused into the index $\vec{j}$ and a new n is extracted from the untransformed index sequence $\vec{m}$.

As a final remark, all the above results remain valid for decimation in frequency decomposition, except that the sequence of indices of the input and output (as well as any intermediate output from any DFT pass) is reversed. This case is depicted on the "butterfly" diagram of Fig. 1, and on Figs. 2–5, which illustrate shape transformations carried out in each DFT pass.

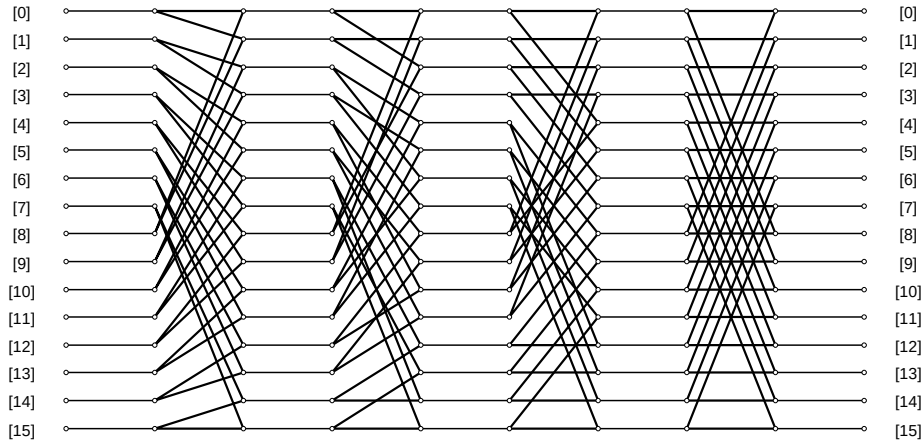


Figure 1: Autosort FFT flow diagram for $N_0 = 16$. Weight labels are omitted for clarity. Note the irregular butterfly structure of the first three passes due to transposition of output. The last pass is computed without transposition and hence its regular shape.

2.3 Generalized transform framework

Because twiddle factors only contribute insignificant amount of computation, they are usually applied as a separate stage between DFT passes, permitting deep optimization of DFT modules which are the core of the algorithm [5]. The algorithm discussed is deliberately not optimized for selected radix values but made as general as possible. As a result, the twiddle coefficients are not applied as a separate

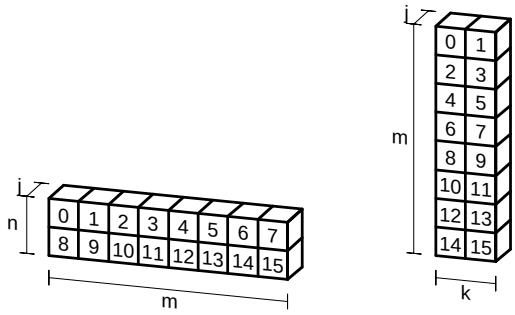


Figure 2: FFT for $N_0 = 16$, pass 1 (left : input, right : transposed output). In all cases, the transform is applied along the decimating vertical index n and results in horizontal output index k.

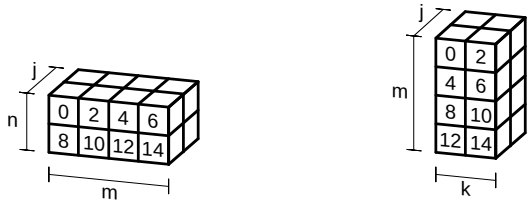


Figure 3: FFT for $N_0 = 16$, pass 2 (left : input, right : transposed output)

scaling stage but incorporated into the trigonometric coefficients of one of the neighbouring DFT passes. Again, it is possible to merge the coefficients either into the inner or the outer loop. In what follows we focus on the latter case, the former being analogous.

Let us reexpress formula (14) as follows :

$$\hat{x}'_{(j_1,k_1)} = \sum_{m_1=0}^{M_1-1} y'_{(k_1,m_1)} \omega^{m_1(N_1 j_1 + k_1)} \tag{18}$$

The frequency index k_1 which previously served as a replication parameter now becomes a variable of the DFT pass, affecting the phase of the twiddle coefficients applied in each DFT in a pass.

It is possible to absorb the twiddle factors into the body of inner DFT pass if we consider a more general form of the DFT equation (which was introduced

Figure 4: FFT for $N_0 = 16$, pass 3 (left : input, right : transposed output)

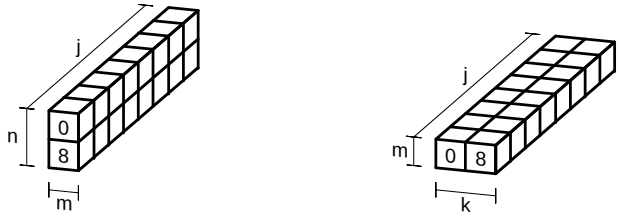


Figure 5: FFT for $N_0 = 16$, last pass (left : input, right : transposed output). Note in this case that the transposition does not affect sample ordering, which is consistent with the shape of the last pass in Fig. 1.

in [10] and termed GFT, for "generalized discrete Fourier transform") :

$$\hat{x}_k = \sum_{n=0}^{N-1} x_n \omega^{(n+P)(k+Q)} = \sum_{n=0}^{N-1} x_n \omega^{nk+Qn+Pk+PQ} \tag{19}$$

By the periodicity property of the signal and its spectrum, the above formula can easily be verified to be equivalent to

$$\hat{x}_{k-Q} = \sum_{n=0}^{N-1} x_{n-P} \omega^{nk} \tag{20}$$

which leads to interpretation of parameters P and Q as the time and the frequency shift, respectively.

Equation 20 has the disadvantage that index shifted by a nonzero value exceeds the usual range of definition $1 \ldots N-1$. Implementation of periodic signals that

are indexed in a range exceeding one period requires modulo addressing, to confine the index to the range where storage for samples is defined. Modulo adressing is assisted by hardware in some machines but most often requires software implementation, incurring some cost of execution time.

In what follows we turn to the form of equation 19 where x and $\hat{x}$ are addressed linearly. Considering the powers of $\omega = e^{-i2\pi/N_0}$, we easily note the periodicity $\omega^{N_0+n} = \omega^n$ which permits use of tabulated values of constants $\omega_n = \omega^n$ for $n = 1 \ldots N_0 - 1$ in place of calculating them during execution of the algorithm, which is the most common approach. Access to ω constants must then be performed modulo N_0.

Let us begin with the initial DFT equation involving time shift P and frequency shift Q, focusing only on the form of ω factors :

$$\omega^{(n_0+P)(k_0+Q)} = \omega^{n_0k_0+Qn_0+Pk_0+PQ} \tag{21}$$

Applying first level of decomposition (3) yields

$$\omega^{(M_1n_1+m_1+P)(N_1j_1+k_1+Q)} = \omega^{M_1n_1(k_1+Q)} \times \omega^{(m_1+P)(N_1j_1+k_1+Q)} \tag{22}$$

a formula for trigonometric constants applied : $\omega^{M_1n_1(k_1+Q)}$ for the inner DFT pass and $\omega^{(m_1+P)(N_1j_1+k_1+Q)}$ for the outer DFT pass. It is worthwile to observe that, given arbitray initial shifts P and Q, only frequency shift is propagated to the next (inner) level of decomposition, the time shift being absorbed in the outer DFT which is the last pass of the overall decomposition. Recursively applying the above decomposition, a structure composed only of DFT with time and frequency shift is obtained, and no intervening twiddle stage is required. It should be noted that the form of (22) is not a unique solution of decomposition (5) and many equivalent splits can be considerd which could be a subject of a separate study.

2.4 Scaling

The classical definition of the discrete Fourier transform pair is

$$\hat{x}_k = \sum_{n=0}^{N-1} x_n\omega^{nk} \leftrightarrow x_k = \frac{1}{N}\sum_{n=0}^{N-1} \hat{x}_n\omega^{-nk}$$

For practical implementation in fixed point arithmetic, it is necessary to include the normalizing factor N^{-1} in the forward rather than inverse formula, so that input of samples from a fixed interval, *e.g.* $(-1, +1)$, yields output bounded to the same range. Additionally, this scaling must be performed incrementally during the computation to prevent exceeding the maximum range at every stage.

Some signal processors have built-in scaling modes, allowing data to be scaled up or down during load or store by a factor of 2 through a simple bit shift. This proves particularly helpful for radix 2 fast Fourier algorithms allowing scaling to be effectuated within every pass. Good noise properties are achieved because intermediate results can make use of the full range of values allowed by the size of data words so that impact of digit truncation is minimal.

A similar feature is difficult to implement in the general mixed radix setup where each pass can be a DFT of a different order, and hence the normalizing factor can be arbitrary. Applying the scaling by the current DFT order would cost an additional real-complex multiplication of all values in every pass.

The algorithm described in this article has the property that in every DFT pass, all samples are multiplied by trigonometric factors exactly once. This leads to the observation that every output sample is a result of a process composed of L consecutive multiplications by some power of ω, where $|\omega| = 1$. To take advantage of this fact, a new vector of scaled trigonometric constants is defined :

$$\omega_S = N^{-\frac{1}{L}} \times \omega$$

The scaling can be verified to produce, after L passes, an overall factor of

$$|\omega_S|^L = \left|\left(N^{-\frac{1}{L}} \times \omega\right)\right|^L = N^{-1} \times |\omega| = \frac{1}{N}$$

Since $N = \prod_l M_l$, the constant $N^{\frac{1}{L}}$ is the geometric mean of primary orders M_l. It is easy to verify that, by chosing, in consecutive passes, DFT orders M_l from smallest to largest, the property $\left|\sum_n^{M_l} x_n \omega_S^{nk}\right| \leq 1$ for $|x_n| \leq 1$ is assured.

The choice of the the above scaling approach is a compromise between rescaling the input by N^{-1} once before execution (not acceptable due to excessive loss in signal to noise ratio) and scaling in each pass by M_l^{-1} which allows tight match of the range $(-1, +1)$ for output at the cost of additional complexity. It can be seen that when the transform order is a product of factors of very different magnitude, the factor $N^{\frac{1}{L}}$ is significantly smaller than 1 causing dynamic range compaction when small DFT's are executed in early passes and a degradation of signal to noise ratio. This fact must be taken into consideration when selecting the order of the transform. In most cases, the order can be decomposed into many small primes, in which case the algorithm should assure a good noise-performance balance.

3 Description of algorithm

This section provides details important for implementation. The algorithm has the following properties :

1. general mixed radix, out-of-place structure with one scratch array which is used together with the target output array on an alternating basis
2. employs the generalized framework described in section 2.3, allowing computation of FFT with time and frequency shift, and execution of both decimation in time and decimation in frequency structures
3. allows (but does not require) execution of autosort algorithm whereby the intermediate results are stored with transposition, to yield final result in natural order
4. has an optional scaling mode allowing the implementation in fixed range arithmetic

In the first paragraph the structure of the algorithm is described and in the second some preliminary performance metrics invoked.

3.1 Structure

The algorithm has a standard structure of four nested loops in each pass. They are discussed beginning with the innermost loop.

Loop over n **:** This loop calculates the scalar product $\sum_{n=0}^{N-1} x_{a(n)}\omega_{b(n)}$, where addressing $a(n) = K_x n + L_x$ is linear and $b(n) = K_\omega n + L_\omega$ circular (modulo N_0). Parameters $K_x, L_x, K_\omega, L_\omega$ are provided from the containing loop.

Loop over k **:** The outcome of the previous loop is a single Fourier coefficient, and execution over $k = 1 \ldots N-1$ yields the full spectrum. The result is assigned to output $\hat{x}_{c(k)}$ with linear addressing $c(k) = K_{\hat{x}} k + L_{\hat{x}}$. In every pass of the loop, the parameters passed to the inner loop K_x, L_x are supplied from the outer loop and K_ω, L_ω are found according to

$$\left\{ \begin{array}{rcl} K_\omega(k) & = & (K_{K,\omega} k + L_{K,\omega}) \mod N_0 \\ L_\omega(k) & = & (K_{L,\omega} k + L_{L,\omega}) \mod N_0 \end{array} \right.$$

Loop over j **:** Here the index j spans the already transformed dimensions of the full decomposition as discussed in paragraph 2.2. The single DFT defined by the loop over k is repeated for every value of j with parameters found according to

$$\left\{ \begin{array}{rcl} K_x(j) & = & K_{K,x} j + L_{K,x} \\ L_x(j) & = & K_{L,x} j + L_{L,x} \\ K_{\hat{x}}(j) & = & K_{K,\hat{x}} j + L_{K,\hat{x}} \\ L_{\hat{x}}(j) & = & K_{L,\hat{x}} j + L_{L,\hat{x}} \\ K_{K,\omega}(j) & = & (K_{K,K,\omega} j + L_{K,K,\omega}) \mod N_0 \\ K_{L,\omega}(j) & = & (K_{K,L,\omega} j + L_{K,L,\omega}) \mod N_0 \\ L_{K,\omega}(j) & = & (K_{L,K,\omega} j + L_{L,K,\omega}) \mod N_0 \\ L_{L,\omega}(j) & = & (K_{L,L,\omega} j + L_{L,L,\omega}) \mod N_0 \end{array} \right.$$

Loop over m **:** In this loop, m is the replication index for which the same structure (in terms of trigonometric constants) is applied to successive slices of the data interpreted as array in three dimensions. Therefore, the index only affects the addressing of inputs and outputs as follows :

$$\left\{ \begin{array}{rcl} K_{K,x}(m) & = & K_{K,K,x} m + L_{K,K,x} \\ K_{L,x}(m) & = & K_{K,L,x} m + L_{K,L,x} \\ L_{K,x}(m) & = & K_{L,K,x} m + L_{L,K,x} \\ L_{L,x}(m) & = & K_{L,L,x} m + L_{L,L,x} \\ K_{K,\hat{x}}(m) & = & K_{K,K,\hat{x}} m + L_{K,K,\hat{x}} \\ K_{L,\hat{x}}(m) & = & K_{K,L,\hat{x}} m + L_{K,L,\hat{x}} \\ L_{K,\hat{x}}(m) & = & K_{L,K,\hat{x}} m + L_{L,K,\hat{x}} \\ L_{L,\hat{x}}(m) & = & K_{L,L,\hat{x}} m + L_{L,L,\hat{x}} \end{array} \right.$$

Execution of pass l : The master loop executes the sequence of passes and uses tabulated values of the parameters defined above (there are 8 parameters defined per each of the three arrays $x, \hat{x}, \omega$). A separate routine is used to precompute the parameters according to the chosen structure. For example, the following choice of parameters is used to obtain a decimation in frequency, autosort structure for arbitrary factorization $\{M_l, l = 0, \ldots, L\}$:

$$\begin{array}{ccc}
K_{K,K,x} = 0 & K_{K,K,\hat{x}} = 0 & K_{K,K,\omega} = 0 \\
K_{K,L,x} = 0 & K_{K,L,\hat{x}} = 0 & K_{K,L,\omega} = \prod_{r=1}^{l-1} M_r \\
K_{L,K,x} = 0 & K_{L,K,\hat{x}} = 0 & K_{L,K,\omega} = 0 \\
K_{L,L,x} = 1 & K_{L,L,\hat{x}} = 1 & K_{L,L,\omega} = 0 \\
L_{K,K,x} = 0 & L_{K,K,\hat{x}} = 0 & L_{K,K,\omega} = N_0/M_l \\
L_{K,L,x} = \prod_{r=1}^{l-1} M_r & L_{K,L,\hat{x}} = N_0/M_l & L_{K,L,\omega} = 0 \\
L_{L,K,x} = N_0/M_l & L_{L,K,\hat{x}} = \prod_{r=1}^{l-1} M_r & L_{L,K,\omega} = 0 \\
L_{L,L,x} = 0 & L_{L,L,\hat{x}} = 0 & L_{L,L,\omega} = 0
\end{array}$$

for all except L-th pass, where the parameter $K_{K,L,\omega}$ is set to zero and the remaining parameters are the same.

If time and frequency offset are desired, then the parameters for ω should be changed according to the formulas from paragraph 2.3. In general, however, the parameters as found using the formulas in this paper lead to only some of the parameters that are nonzero. Since these parameters are determined uniquely, there exists a potential for further generalization, which can be an interesting subject of study. One seemingly achievable result is Good's prime factor algorithm, but for this, a modification of the implementation is required to allow modulo addressing of input and output arrays.

3.2 Complexity

The algorithm, derived and implemented in its full generality, compares infavorably with highly optimized algorithms like the popular FFTW library [5] when speed is considered. Preliminary implementation tests executed on an Intel x86 machine in a single processor core show that the FFTW is 3 to 6 times faster (see Table 1), the difference being smallest for factorizations into large primes (combinations of 17, 19 and 23 were considered) which it executes using generic routine and largest for orders being powers of 2 which are performed by highly optimized modules. This is not surprising, given the generalized structure in which all multiplications are performed directly regardless of whether they are trivial or not. On general purpose architectures (such as the Intel x86), all operations involved in the algorithm contribute to the exection time.

Table 2 shows the breakdown of the algorithm into elementary operations involved in its execution, per each of the loops from the innermost (over n) to the outermost (over m) that make up one pass.

To analyse the workload it is sufficient to consider the innermost loop, over n, which has a dominant impact. The loop iterates M_l times ; the containing loop over k also iterates M_l times and the two outer loops in combination make N_0/M_l repetitions, which makes a total of $N_0 \times M_l$ iterations of the innermost loop per

Table 1: Comparison of execution times of proposed algorithm and FFTW. Mean times and standard errors measured in microseconds. Observed measurement variability is due to processor speed stepping.

		Proposed		FFTW		
Size	Factorization	$\bar{t}$	SE t	$\bar{t}$	SE t	ratio
512	2^9	1055	5,8	177	2,9	**5,96**
1024	2^{10}	1519	731,8	284	122,8	**5,35**
2048	2^{11}	2150	3,5	385	2,7	**5,58**
323	$17 \cdot 19$	462	169,1	154	60,6	**2,99**
391	$17 \cdot 23$	1001	2,5	267	1,3	**3,75**
437	$19 \cdot 23$	1158	2,3	323	2,4	**3,59**
289	17^2	654	2,0	168	1,3	**3,9**
361	19^2	888	2,0	228	0,7	**3,89**
529	23^2	1552	148,1	412	0,7	**3,76**
7429	$17 \cdot 19 \cdot 23$	11679	24,9	3460	17,1	**3,38**
4913	17^3	6764	8,5	2155	8,1	**3,14**
6859	19^3	10471	13,9	3237	14,1	**3,23**
240	$2^4 \cdot 3 \cdot 5$	190	38,4	46	7,2	**4,11**
300	$2^2 \cdot 3 \cdot 5^2$	458	59,8	106	12,8	**4,31**
320	$2^6 \cdot 5$	585	5,1	112	0,1	**5,23**
350	$2 \cdot 5^2 \cdot 7$	554	1,4	156	0,6	**3,54**
400	$2^4 \cdot 5^2$	653	131,2	137	25,0	**4,77**
450	$2 \cdot 3^2 \cdot 5^2$	359	74,4	125	22,8	**2,88**
500	$2^2 \cdot 5^3$	741	106,0	185	23,5	**4,01**
600	$2^3 \cdot 3 \cdot 5^2$	1053	1,2	265	18,4	**3,98**
700	$2^2 \cdot 5^2 \cdot 7$	637	234,1	204	64,7	**3,12**
800	$2^5 \cdot 5^2$	860	255,0	199	50,1	**4,31**
900	$2^2 \cdot 3^2 \cdot 5^2$	1542	171,1	437	47,4	**3,53**
1000	$2^3 \cdot 5^3$	1171	514,1	320	132,8	**3,66**

Table 2: Operation counts per iteration

Operation	n-loop	k-loop	j-loop	m-loop
real multiply-accumulate	4	0	0	0
linear address update	1	1	4	8
circular address update	1	2	4	0
data word read	4	0	0	0
data word write	0	2	0	0

pass. In each iteration, a complex multiply-accumulate operation is performed, which amounts to 4 multiplies and accumulations, assuming real-imaginary representation of complex data stored in a single data word per each part. Consequently, four data reads are required to fetch the real and imaginary part of the input data and the trigonometric coefficients. In most DSP architectures, a data transfer can be performed in parallel with the multiply-accumulate operations ; sometimes, it is even possible to perform two reads or a double/quad word read. It is also common in DSP to be able to offset data pointers in parallel with arithmetic operation and data read. If a given architecture features modulo addressing, the circular address can be updated with no cost, otherwise the range check and possible index wrap will contribute to the execution time. The result is written back after the inner loop completes the calculation of the Fourier coefficient and is done in the body of the k-loop.

From the above it is clear that the execution time is determined by arithmetic instructions. The algorithm's speed could be improved by reducing their number. FFTW library uses a technique involving discrete Hartley transform to halve the number of multiplications required by computing two coefficients at a time. This partly accounts for the difference in execution speed between the library and the implementation of the presented algorithm ; an additional difference comes from the fact that FFTW avoids the problem of addressing trigonometric constants by expanding the table of all nontrivial coefficients at the cost of $(M_l - 1)^2$ memory words per pass l. In the algorithm presented, the generalized configuration in which ω factors can occur in any power prevents direct application of the mentioned technique which requires deeper study. Some known improvements, such as for real, real-odd and real-even transforms could be applied directly ; others, however, are incompatible with the framework's mixed-radix and generalized design.

The major advantage of the proposed algorithm is its small code footprint. Compiled for Intel x86, object binaries have a size of less than 16 KB, whereas FFTW library by far exceeds 500 KB. This aspect may be of key importance in many cases.

4 Concluding remarks

The algorithm presented in this article is a generalized FFT framework which also has autosort capability. In its full generality, it is able to implement many of the structures and decompositions proposed in the past. The derivations presented are implemented in the form of a test program which is available from the author. The advantages of the algorithm are its simplicity (only a single version to execute all permitted structures), which translates into a small code footprint, and the flexibility it offers. However, preliminary tests show that the implementation is inferior in terms of speed when compared with a state of the art industry implementation. Some acceleration techniques could be considered but require further study.

A true test for the algorithm would be an attempt to run it in a DSP environment, where it could take advantage of circular addressing modes it makes extensive use of, and where structural overhead is small. The incremental scaling

option which comes at no performance cost makes it directly applicable in the most typical case where fixed point arithmetic is required. It is interesting to see how this approach would perform from the perspective of noise - which was not considered in this paper because of the fact that floating point implementation is known [2] to behave in the opposite way, *i.e.*, direct formula implementation results in *higher* noise than the fast algotithm. The author plans to elaborate this aspect in the future.

Acknowledgement Study was supported by research fellowship within "Information technologies : research and its interdisciplinary applications" agreement number POKL.04.01.01-00-051/10-00.

References

[1] Cooley J. W., Tukey J. W., An algorithm for the machine computation of complex Fourier series, *Math. Computation* vol. 9, pp 297-301, 1965

[2] Gentleman W. M., Sande G., Fast Fourier transforms - for fun and profit, *Proc. AFIPS*, pp 563-578, 1966

[3] Duhamel P., Hollmann H., 'Split radix' FFT algorithm, *Electronics Letters* vol. 20 pp 14-16, 1984

[4] Stasiński R., Prime factor FFT for modern computers, *IWSSIP 2012* pp 346-349, 2012

[5] Frigo M., Johnson S. G., The design and implementation of FFTW3, *Proc. IEEE* vol. 93 no 2, pp 216-231, 2005

[6] Van Buskirk J. posted his results on comp.dsp newsgroup and the programs he wrote are available (as of 2012) at `http://home.comcast.net/~kmbtib/` and `http://www.cuttlefisharts.com/newfft/`

[7] Johnson S. G., Frigo M., A modified split-radix FFT with fewer arithmetic operations, *IEEE Trans. on Signal Processing*, vol. 55 no 1, pp 111-119, 2007

[8] Sohie G. R. L., Chen W., *Implementation of Fast Fourier Transforms on Motorola's Digital Signal Processors*, Motorola application note APR4/D, 1993

[9] Van Loan C., *Computational Frameworks for the Fast Fourier Transforms*, SIAM, 1992

[10] Bongiovanni G., Corsini P., Frosini G., One-dimensional and two-dimensional generalized discrete Fourier transform, *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 24 no 1, pp 97-99, 1976

[11] Akansu A. N., Agirman-Tosun H., Generalized discrete Fourier transform with nonlinear phase, *IEEE Trans. on Signal Processing*, vol. 58 no 9, pp 4547-4556, 2010

[12] Cochran W. T., Cooley J. W., Favin D. L., Helms H. D., Kaenel R. A., Lang W. W., Maling G. C., Nelson D. E. , Rader C. M., Welch P. D., What is the fast Fourier transform ?, *Proc. IEEE*, vol. 55 no 10, pp 1664-1674, 1967

Simultaneous Deleting or Merging Regressors for Linear Model Selection

Aleksandra Maj[1], Agnieszka Prochenka[1], and Piotr Pokarowski[2]

[1] Institute of Computer Science, Polish Academy of Sciences,
ul. Jana Kazimierza 5, 01-237 Warsaw, Poland
[2] Institute of Applied Mathematics and Mechanics, University of Warsaw,
ul. Banacha 2, 02-097 Warsaw, Poland

Abstract

In the article we consider a problem of simultaneous deleting continuous variables and merging levels of factors in linear model. We propose a backward selection procedure called DMR in two variants: the first is similar to the backward stepwise regression and the second, faster implementation combines the agglomerative clustering of levels of factors with ranking regressors by squared t-statistics. In the paper we show that our algorithm is consistent. For the formulated problem we also propose a generalization of performance measures such as sensitivity and specificity. We present a simulation study, which shows substantial advantage of DMR over other methods described in the literature.

Keywords: ANOVA; Bayes Factors; GIC; Hierarchical clustering; Merging levels; QR decomposition; Sensitivity; Similarity; Specificity; T-statistic; Variable selection.

1 Introduction

Model selection is usually understood as selection of explanatory variables. However, when a categorical predictor is considered, in order to reduce model's complexity, we can either exclude the whole factor or merge its levels.

A traditional method to examine the relationship between a continuous response and categorical variables is analysis of variance (ANOVA). However, ANOVA answers only a question of the overall importance of a factor. The next step of the analysis are pairwise comparisons of group means within important factors. Typically post-hoc analysis such as Tukey's honestly significant difference (HSD) test or multiple comparison adjustments (Bonferroni, Scheffe, Hochberg) is used. A drawback of pairwise comparisons is non-transitivity of conclusions.

As a motivating example, let us consider data `Cars93` from `R` library `MASS`. The relationship between logarithm of fuel consumption and other characteristics of 81 cars is modeled. The dependence between the response and the number of cylinders examined with the use of Tukey's HSD analysis (Figure 1) gives inconclusive answers: $\beta_4 = \beta_5$, $\beta_5 = \beta_6$, but $\beta_4 \neq \beta_6$.

In the article we introduce a novel procedure called delete or merge regressors (DMR), which enables efficient search among partitions of factor levels, hence the issue of non-transitivity does not occur. When applying DMR procedure to the `Cars93` data, the number of parameters is efficiently diminished from 31 to 11 with no considerable loss in R^2 (from 0.92 to 0.9), while a model received from the stepwise backward model selection minimizing BIC implemented in the `stepAIC` function in `R` has 14 parameters with $R^2 = 0.89$.

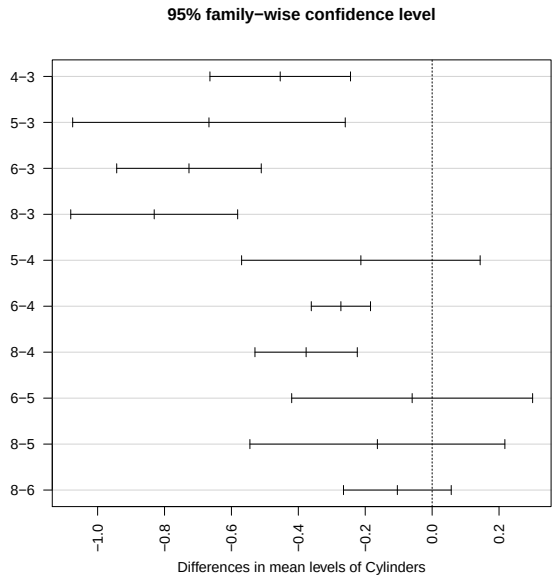


Figure 1: Results of Tukey's HSD.

The idea of partitioning a set of levels of a factor into non-overlapping subsets has already been discussed in the literature. In the article [1], 1949, Tukey proposed a stepwise backward procedure based on the Studentized range methods. Other methods performing multiple comparison procedures, based on clustering means in ANOVA, were described by Scott and Knott, 1974 [2], Calinski and Corsten, 1985 [3] and Corsten and Denis, 1990 [4]. However, these methods do not generalize directly to the problem with any number of factors.

Also the problem of simultaneous continuous variables selection and merging levels of factors is present in the literature. A method introduced by Bondell and Reich, 2009, [5] called collapsing and shrinkage ANOVA (CAS-ANOVA) solves the problem with the use of the least absolute shrinkage and selection operator (LASSO; Tibshirani, 1996, [6]), where the L_1 penalty is imposed on differences between parameters corresponding to levels of each factor. Gertheiss and Tutz, 2011, [7] proposed a modification of CAS-ANOVA, which is more computationally efficient because of using the least angle regression (LARS; Efron et al., 2004, [8]) algorithm.

We propose a backward selection procedure called delete or merge regressors

(DMR), which combines deleting the continuous variables with merging levels of factors. The method assumes greedy search among linear models with a set of constraints of two types: either a parameter for a continuous variable is set to zero or parameters corresponding to two levels of a factor are set to equal each other. DMR is a stepwise regression procedure, where in each step a new constraint is added according to ranking of the hypotheses based on squared t-statistics. As a result a nested family of linear models is obtained and the final decision is made according to minimization of generalized information criterion (GIC).

Two variants of DMR are described in the article. The first, more greedy version adapts agglomerative clustering, where squared t-statistics define the dissimilarity measure. This procedure generalizes concepts introduced by Ciampi et al., 2008, [9] and Zheng and Loh, 1995, [10]. The second version assumes recalculation of t-statistics in each step, which causes loss in computational efficiency.

In the paper we show that DMR algorithm is consistent. The time complexity of the more greedy version of DMR is $O(np^2)$, where n is the number of observations and p is the number of parameters in the full model. We describe also a simulation study and discuss a pertaining `R` package. The simulations show that DMR is several hundred times faster with significantly lower error of selection than CAS-ANOVA.

The next problem considered in the article is determining the quality of performance of model selection. Commonly used measures are for example true positive rate (TPR) or false negative rate (FNR). In the literature [5], [7] a generalization of these rates to the problem of partitioning factor levels can be found. However, these measures tend to diminish the influence of continuous predictors and factors with a small number of levels. We propose a different generalization which is based on the dimension of linear subspace of the parameter space defined by the imposed constraints.

The remainder of the article proceeds as follows. The class of feasible models considered when performing model selection is defined in Section 2. DMR procedures are introduced in Section 3, while the asymptotic properties are discussed in Section 4. Generalization of measures of performance is introduced in Section 5. Simulations and real data example are given in Section 6 to illustrate the method. All proofs are given in the Appendix.

2 Feasible models

Let us consider a full rank linear model with n observations and $p < n$ parameters:

$$y = X\beta + \varepsilon = \mathbf{1}_n\beta_{00} + X_0\beta_0 + X_1\beta_1 + \ldots + X_l\beta_l + \varepsilon, \tag{1}$$

where:

1. $\varepsilon \sim \mathcal{N}(0, \sigma^2\mathbb{I}_n)$.
2. $X = [\mathbf{1}_n, X_0, X_1, \ldots, X_l]$ is the model matrix divided as follows: X_0 is the matrix corresponding to continuous regressors and $X_1, \ldots, X_l$ are the zero-one matrices encoding corresponding factors with the first level set as reference.

3. $\beta = [\beta_{00}, \beta_0^T, \beta_1^T, \ldots, \beta_l^T]^T$ is the parameter vector divided as follows: β_{00} is the intercept, $\beta_0 = [\beta_{10}, \ldots, \beta_{p_0 0}]^T$ is a vector of coefficients for continuous variables and $\beta_k = [\beta_{2k}, \ldots, \beta_{p_k k}]^T$ is a vector of parameters corresponding to the k-th factor, $k = 1, \ldots, l$, hence the length of parameter vector is $p = 1 + p_0 + (p_1 - 1) + \ldots + (p_l - 1)$.

Definition 1. *An* elementary hypothesis *for linear model (1) is a linear hypothesis of one of two types:*

(*) $h_{jk} : \beta_{jk} = 0$ *for all* j, k *or*
(**) $h_{ijk} : \beta_{ik} = \beta_{jk}$ *for all* i, j *and* $k > 0$.

Definition 2. *A* feasible model *is defined as a sequence* $m = (C, P_1, ..., P_l)$*, where* C *denotes a subset of continuous variables and* P_k *is a partition of levels of the* k*-th factor. Such a model can be encoded by a set of elementary hypotheses. A set of all feasible models is denoted by* $\mathcal{M}$.

2.1 Change of variables

In order to replace a constrained by an unconstrained optimization problem a change of variables in model m is performed. The model can be defined by a set of following equations

$$\begin{cases} y = X\beta + \varepsilon \\ A_C \beta = 0, \end{cases}$$

where A_C is a matrix of elementary hypotheses describing constraints induced by the model.

Let us define a square matrix

$$A = \begin{bmatrix} A_1 \\ A_0 \end{bmatrix},$$

where A_0 is a matrix of elementary hypotheses in a convenient form described below, with rows spanning the same space as rows of A_C and A_1 is a complement of A_0 to a square matrix. We want A_0 and A_1 to satisfy

$$A = \begin{bmatrix} A_1 \\ A_0 \end{bmatrix} = \begin{bmatrix} \mathbb{I}_q & 0 \\ B & \mathbb{I}_{p-q} \end{bmatrix}.$$

Matrix A of such a form uniquely encodes model m and can always be obtained by appropriate permutation of the columns of the model matrix, which can be performed in the following way:

1. For each factor with partition P_k, $k = 1, \ldots, l$ with i_k clusters, where $P_k = \{U_1^k, \ldots, U_{i_k}^k\}$, rename its levels so that

$$U_j^k = \{j, i_k + \sum_{s=1}^{j-1} |U_s^k| - j + 2, \ldots, i_k + \sum_{s=1}^{j} |U_s^k| - j\},$$

for $j = 1, \ldots, i_k$.

2. Sort the columns of the model matrix in the following order:
   (a) intercept,
   (b) all continuous variables present in the model (these for which the beta coefficient is non-zero),
   (c) i_k first levels of k-th factor, $k = 1, \ldots, l$,
   (d) all remaining continuous variables,
   (e) all remaining levels of factors.

Further in the article we assume that the columns of model matrices for considered models are permuted so that the constraints matrices have a form such as A_0.

Example 1. As an illustrative example consider a model consisting of one factor with $P_1 = \{\{1,2,3\},\{4,5\},\{6,7,8\}\}$. This parametrization corresponds to the following constraint matrix:

$$A_C = \left[\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{array}\right].$$

After renaming levels in the way described above, we get partition P_1 of the form $P_1 = \{\{1,4,5\},\{2,6\},\{3,7,8\}\}$ and matrix

$$A = \left[\begin{array}{c} A_1 \\ \hline A_0 \end{array}\right] = \left[\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

in the desired form.

Thanks to such reparametrization of the model, transition from constrained to unconstrained problem is immediate.

$$X\beta = XA^{-1}A\beta = Z\xi,$$

where $Z = XA^{-1}$ and $\xi = A\beta$. From Schur complement we get

$$A^{-1} = \left[\begin{array}{cc} \mathbb{I}_q & 0 \\ -B & \mathbb{I}_{p-q} \end{array}\right] = [A^1, A^0],$$

since $A_0\beta = 0$ we have

$$Z\xi = X[A^1, A^0]\left[\begin{array}{c} \xi_1 \\ 0 \end{array}\right] = Z_1\xi_1,$$

where $Z_1 = XA^1$, $\xi_1 = A_1\beta$. The linear space of parameters changes from $\mathcal{L} = \{\beta \in \mathbb{R}^p : A_0\beta = 0\}$ to $\mathcal{L}(A^1) = \{A^1\xi_1 : \xi_1 \in \mathbb{R}^q\}$. The dimension of space $\mathcal{L}(A^1)$ is called the size of model m and denoted by $|m|$.

Note that imposing an elementary hypothesis on parameter vector β is equivalent, in terms of the new parametrization, to either eliminating one column or summing up two columns of the model matrix, which decreases its number of columns by one. These operations are explicitly visible from the form of A^1.

Definition 3. *We define* the inclusion relation between two models m_1 and m_2 *by inclusion of linear spaces spanned by columns of the corresponding matrices* $A^1_{m_1}$ *and* $A^1_{m_2}$*:*

$$m_1 \subseteq m_2 \iff \mathcal{L}(A^1_{m_1}) \subseteq \mathcal{L}(A^1_{m_2}).$$

2.2 Generalized Information Criterion

Definition 4. Generalized Information Criterion *for model m is defined as:*

$$GIC(m) = n \log(RSS_m) + r_n(|m| + 1),$$

where r_n is the penalty for model size.

The goal of our method is to find the best feasible model according to GIC, taking into account that the number of feasible models grows exponentially with p. Since for the k-th factor number of possible partitions is the Bell number $\mathcal{B}(p_k)$, the number of all feasible models is $2^{p_0}\prod_{k=1}^{l}\mathcal{B}(p_k)$. In order to significantly reduce the amount of computations, we propose the following greedy backward search.

3 Algorithms

Assuming that X is of full rank the QR decomposition of the model matrix is

$$X = Q_p R,$$

where Q_p is $n \times p$ orthogonal matrix and R is $p \times p$ upper triangular matrix. Then

$$z = Q_p^T y, \tag{2}$$

$$\widehat{\sigma}^2 = \frac{\|(I - Q_p Q_p^T)y\|^2}{n - p} \tag{3}$$

and

$$\widehat{\beta} = R^{-1}z.$$

Let us define set of indexes corresponding to continuous variables and factors

$$\mathrm{Ind}_0 = \{0, 1, \ldots, p_0\},\ \mathrm{Ind}_k = \{2, \ldots, p_k\} \text{ for } k = 1, \ldots, l.$$

Then

$$\widehat{\beta} = (\widehat{\beta}_{jk})_{j \in \mathrm{Ind}_k} = (r_{jk}^T z)_{j \in \mathrm{Ind}_k} \text{ for } k = 0, \ldots, l,$$

where

$$R^{-1} = [r_{00}^T, r_{10}^T, \ldots, r_{p_0 0}^T, r_{21}^T, \ldots, r_{p_1 1}^T, \ldots, r_{2l}^T, \ldots, r_{p_l l}^T]^T. \tag{4}$$

Elementary hypotheses of types (*) and (**) defined in Definition 1 can be now rewritten as:

(*) h_{jk}: $\beta_{jk} = 0 \iff a_{1jk}^T \beta = 0$, where $a_{1jk} = a_{st}(j,k) = \mathbb{1}(s = j, t = k)$, where $s \in \mathrm{Ind}_t$ and $t = 0, \ldots, l$,

(**) h_{ijk}: $\beta_{ik} = \beta_{jk} \iff a_{ijk}^T \beta = 0$, where $a_{ijk} = a_{st}(i,j,k) = \mathbb{1}(s = i, t = k) - \mathbb{1}(s = j, t = k)$, where $s \in \mathrm{Ind}_t$ and $t = 1, \ldots, l$.

3.1 DMR algorithm

1. Perform the QR decomposition of the full model matrix, getting matrix R^{-1}, vector z and variance estimator $\widehat{\sigma}^2$ as in equations (4), (2) and (3).
2. Calculate squared t-statistics:
   (a) for all elementary hypotheses of type (*):

$$T_{1jk}^2 = \frac{\widehat{\beta}_{jk}^2}{\widehat{Var}(\widehat{\beta}_{jk})} = \frac{(r_{jk}^T z)^2}{\widehat{\sigma}^2 \|r_{jk}\|^2} \quad \text{for} \quad k \geq 0,\ j \in \mathrm{Ind}_k \setminus \{0\},$$

   (b) for all elementary hypotheses of type (**):

$$T_{ijk}^2 = \frac{(\widehat{\beta}_{ik} - \widehat{\beta}_{jk})^2}{\widehat{Var}(\widehat{\beta}_{ik} - \widehat{\beta}_{jk})} = \frac{((r_{ik} - r_{jk})^T z)^2}{\widehat{\sigma}^2 \|r_{ik} - r_{jk}\|^2}$$

   for $k > 0$, $i, j \in \mathrm{Ind}_k$.
3. For each $k > 0$ perform agglomerative clustering using $D_k = [d_{ijk}]_{ij}$ as dissimilarity matrix, where:
   (a) $d_{1jk} = d_{i1k} = T_{1jk}^2$ for $i, j \in \mathrm{Ind}_k$,
   (b) $d_{ijk} = T_{ijk}^2$ for $i, j \in \mathrm{Ind}_k$, $i \neq j$,
   getting vectors of cutting heights g_k, $k = 1, \ldots, l$.
4. Combine all vectors g_k with g_0 defined as $g_0 = [T_{110}^2, \ldots, T_{1p_0 0}^2]$, denote the given vector as g. Sort g in increasing order: $g = [g^{(1)}, \ldots, g^{(p-1)}]^T$. Every element $g^{(i)}$ corresponds to an elementary hypothesis a_i. A sequence of nested linear constraints on model parameters $A_i \beta = 0$ is obtained, where $A_i = [a_1, \ldots, a_i]$, $i = 0, \ldots, p-1$ and $A_0 = 0$.
5. Perform QR decomposition of matrix $R^{-T} A_{p-1}^T$ getting the orthogonal matrix $W = [w_1, \ldots, w_{p-1}]$.
6. Set $\mathrm{RSS}_0 = \|y\|^2 - \|z\|^2$ and $\mathrm{GIC}_0 = n \log \mathrm{RSS}_0 + (p+1) r_n$ for model without constraints.
   For $i = 1, \ldots, p-1$

$$\mathrm{RSS}_i = \mathrm{RSS}_{i-1} + (w_i^T z)^2,$$

   where calculations are described in the Appendix A and

$$\mathrm{GIC}_i = n \log \mathrm{RSS}_i + (p - i + 1) r_n.$$

7. Selected model $\widehat{m}$ is a model with hypotheses $A_{\widehat{i}}$ accepted, where

$$\widehat{i} = \underset{0 \le i \le p-1}{\arg\min}\, GIC_i.$$

The dominating operation in the described procedure is the QR decomposition of the full model matrix. Hence, the time complexity of DMR algorithm is $O(np^2)$.

This procedure assumes transitivity of hypotheses, for example accepting hypotheses $\beta_{ik} = \beta_{jk}$ and $\beta_{jk} = \beta_{mk}$ causes acceptance of hypothesis $\beta_{ik} = \beta_{mk}$. Notice that considered hypotheses can be ambiguously encoded. In order to avoid this problem the following convention will be used while cluster merging: $U_1,\ U_2 \subseteq \{1, \ldots, p_k\}$ clusters to merge, $U_1 \cap U_2 = \emptyset$, then $i_1 = \min_{i \in U_1} i$, $i_2 = \min_{i \in U_2} i$ and the hypothesis to accept is $\beta_{i_1 k} = \beta_{i_2 k}$.

An exemplary run of DMR algorithm is shown in Figure 2. The agglomerative clustering was performed for data, which is described in Section 6.1, consisting of three categorical variables. The horizontal dotted line indicates the cutting height for the best model chosen by BIC (special case of GIC, where $r_n = \log n$).

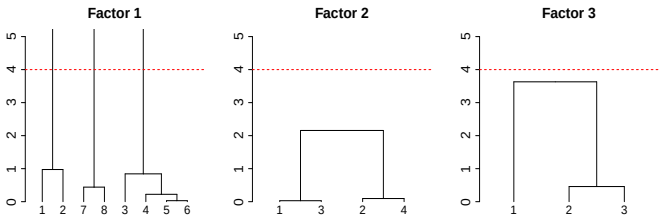


Figure 2: Dendrograms for exemplary run of DMR algorithm.

3.2 stepDMR algorithm

The stepDMR algorithm is based on the RSS calculation described in Appendix A, where the set of hypotheses under consideration is of the form $A_0\beta = 0$, hence the vector e is zero and equation (9) reduces to:

$$\|y - X\widehat{\beta}_c\|^2 = \sum_{i=1}^{p-q} (w_i^T z)^2. \tag{5}$$

By $S^{(i-1)}$ and $W^{(i-1)}$ we denote respectively the matrix corresponding to the set of possible hypotheses and the orthogonal matrix corresponding to the set of accepted hypotheses in the i-th step of the algorithm. Steps of the algorithm:

1. $S^{(0)} = R^{-T} A_{\text{all}}^T$, where A_{all} is the matrix of all possible elementary hypotheses.
2. $W^{(0)} = 0$. Calculate RSS for model without constraints getting RSS_0 and $\text{GIC}_0 = n \log \text{RSS}_0 + (p+1) r_n$.

3. Perform the orthogonal projection of possible elementary hypotheses $S^{(i-1)}$ onto the orthogonal complement of subspace spanned by already accepted hypotheses $W^{(i-1)}$:

$$V^{(i)} = (I - W^{(i-1)}(W^{(i-1)})^T)S^{(i-1)}.$$

4. Normalize columns of matrix $V^{(i)}$

$$w_j^{(i)} = \frac{v_j^{(i)}}{\|v_j^{(i)}\|}.$$

5. According to equation (9) calculate the vector of increases in residual sum of squares and choose the hypothesis corresponding to the minimal value.

$$\widehat{j} = \arg\min_j((w_j^{(i)})^T z)^2,$$

bind the chosen hypothesis with the matrix of already accepted hypotheses:

$$W^{(i)} = [W^{(i-1)}, w_{\widehat{j}}^{(i)}].$$

6. Calculate residual sum of squares

$$\mathrm{RSS}_i = \mathrm{RSS}_{i-1} + ((w_{\widehat{j}}^{(i)})^T z)^2$$

and GIC

$$\mathrm{GIC}_i = n \log \mathrm{RSS}_i + (p - i + 1)r_n.$$

7. Remove from $S^{(i-1)}$ columns linearly dependent with columns of $W^{(i)}$, getting new matrix of possible hypotheses $S^{(i)}$.
8. Go back to step 3 until the model is reduced to the intercept.
9. Selected model $\widehat{m}$ is a model with hypotheses corresponding to the columns of $W^{(\widehat{i})}$ accepted, where

$$\widehat{i} = \arg\min_{0 \le i \le p-1} GIC_i.$$

The dominating operation in each of p steps of the described procedure is the QR decomposition of the model matrix. Hence, the time complexity of stepDMR algorithm is $O(np^3)$.

4 Asymptotic properties of DMR algorithm

We first introduce some notations. In this section we use a simplifying notation $f_n \prec g_n$ which corresponds to $f_n = o(g_n)$. We allow the number of predictors p_n to grow with the number of observations n under the condition $p_n \prec n$.

We distinguish the following subsets of the set of all feasible models $\mathcal{M}$:

1. Full model f, which is a model without constraints.

2. Uniquely defined model t, which is minimal among true models in the sense of inclusion defined in Definition 3 and is fixed and does not depend on sample size. We assume that the model consists of a finite number of continuous variables and a finite number of factors with finite numbers of levels.
3. A set $\mathcal{M}_{\mathcal{V}}$ of models with one false elementary hypothesis accepted:

$$\mathcal{M}_{\mathcal{V}} = \{m \subseteq f : |m| = |f| - 1 \text{ and } t \nsubseteq m\},$$

4. A set $\mathcal{M}_{\mathcal{T}}$ of models with one true hypothesis accepted:

$$\mathcal{M}_{\mathcal{T}} = \{m \subseteq f : |m| = |f| - 1 \text{ and } t \subseteq m\}.$$

Theorem 1. *Let us denote*

$$d_n = \min_{m \in \mathcal{M}_{\mathcal{V}}} \beta_t^T X_t^T (\mathbb{I} - H_m) X_t \beta_t,$$

where X_t is the model matrix of the true model t with appropriate columns of the full model matrix X deleted or merged, $\beta_t \in \mathbb{R}^{|t|}$ is the parameter vector of t and H_m is the hat matrix corresponding to model m. Assuming that X is of full rank p_n, where $p_n \prec r_n \prec n$ and $p_n \prec d_n$ we have

$$\lim_{n \to \infty} \mathbb{P}(\widehat{m} = t) = 1,$$

where $\widehat{m}$ is the model selected by DMR procedure from Section 3.1, where the linkage criterion for hierarchical clustering is a convex combination of minimum and maximum of the pairwise distances between clusters.

Proof can be found in the Appendix B.

5 Measures of performance and quality of selection

5.1 Measures of performance

When performing simulations a researcher usually faces a problem of comparing results with the underlying truth. Furthermore, one would like to have a measure of performance which is more liberal than a binary response, whether the true model was correctly identified or not. Traditionally for model selection with only continuous predictors measures such as true positive rate (TPR) or false negative rate (FNR) are used. In the literature [7], [5] a generalization to both continuous and categorical predictors can be found.

True Positive Rate is the proportion of true differences which are correctly identified, meaning ratio of the number of true elementary hypotheses which were found by the selector to the number of all true elementary hypotheses.

False Negative Rate is the proportion of false differences which are correctly identified, meaning ratio of the number of false elementary hypotheses which were rejected by the selector to the number of all false elementary hypotheses.

However, measures defined in this way diminish the influence of the continuous variables and factors with a small number of levels. As an example, consider

a model with 5 continuous predictors and one factor with 5 levels. Then the number of parameters for the continuous predictors is 5 and the number of possible elementary hypotheses equals 5. The number of parameters for the categorical variable is also 5, whereas the number of possible elementary hypotheses is $\binom{5}{2} = 10$.

Therefore, we introduce a different generalization of traditional performance measures which treats the set of considered hypotheses as a linear subspace of the parameter space. The new measures are functions of sizes of the true and selected models. We consider two models: true model t and selected model s.

Definition 5. *Let us denote* $t \cap s = \mathcal{L}(t) \cap \mathcal{L}(s)$. Sensitivity *coefficient is defined as:*

$$Sen = \frac{TP}{FN + TP} = \frac{|t \cap s|}{|t|}.$$

Specificity *coefficient is defined as:*

$$Spe = \frac{TN}{FP + TN} = \frac{p - (|t| + |s| - |t \cap s|)}{p - |t|}.$$

However, in the article the attention is focused on values: $1 - Sen$ and $1 - Spe$, which correspond to the errors made by selector.

5.2 Measure of quality of selection

In all simulations described in the article we used BIC for model selection. One can ask how much better than the other models the selected model is. An answer to this question can be Bayes factors. Assuming uniform prior distribution on the set of models and denoting data by $\mathcal{D}$, the Bayes factor for the model m with respect to the best model $\widehat{m}$ (with minimum BIC) is expressed as

$$BF_m = \frac{\mathbb{P}(\mathcal{D}|m)}{\mathbb{P}(\mathcal{D}|\widehat{m})}.$$

Approximate Bayes factors [11]

$$\widetilde{BF}_m = \exp(-\frac{1}{2}(\mathrm{BIC}_m - \mathrm{BIC}_{\widehat{m}}))$$

are estimators of the quality of selection. Figure 3 illustrates an example of use of approximate Bayes factors for this purpose.

6 Simulation study

In order to compare DMR algorithm with other methods of model selection simulation studies were performed. All the simulations were conducted using functions implemented in `R` package called `DMR`, which is available at the `CRAN` webpage:

`http://cran.r-project.org/web/packages/DMR/index.html`

The main function of the package is `DMR`, which enables choosing a method of hierarchical clustering and a value of GIC penalty used by the algorithm. Moreover, other functions for extensions of DMR method such as `stepDMR`, which is

based on recalculation of t-statistics in each step, and function `DMR4glm` for generalized linear models can be found. Functions `roc` and `plot_bf` can be used for obtaining measures of performance and quality of selection described in Section 5.

In section 6.1 results regarding an experiment described by Bondell and Reich, 2009, [5] are presented. The data generated for the experiment consists of three factors and no continuous variables. As a continuation, simulations based on data containing one factor and eight correlated continuous predictors were carried out. The results can be found in Section 6.2. In both experiments the complete linkage method of clustering in DMR algorithm and BIC were used. Section 6.4 focuses on comparison of this clustering method with others like single linkage and Ward's minimum variance methods. The last Section 6.3 refers to a real data example where fuel consumption of cars was modeled. The data `Cars93` comes from `R` package `MASS`.

6.1 Experiment 1

The experimental model consists of three factors having eight, four and three levels, respectively. The response y is generated from the model, which can be formulated in two equivalent ways. The first one uses the notation from the model formulation (1), the second one is an illustrative version, where we can see exactly the partitions of factors.

$$\begin{aligned} y =& \beta_{00}\mathbf{1}_n + X_1\beta_1 + X_2\beta_2 + X_3\beta_3 + \varepsilon \\ =& 2 \cdot \mathbf{1}_n + X_1(0,-3,-3,-3,-3,-2,-2)^T + X_2(0,0,0)^T + X_3(0,0)^T + \varepsilon \\ =& V_1\alpha_1 + V_2\alpha_2 + V_3\alpha_3 + \varepsilon \\ =& V_1(2,2,-1,-1,-1,-1,0,0)^T + V_2(0,0,0,0)^T + V_3(0,0,0)^T + \varepsilon, \end{aligned}$$

where X_i equals V_i with first column removed for $i = 1,2,3$ and $\varepsilon \sim \mathcal{N}(0, \mathbb{I}_n)$. A balanced design was used with k observations for each combination of factor levels, which gives $n = 96 \cdot k$, $k = 1,2,4$.

The data was generated 1000 times. In the simulation study we compared four algorithms: DMR, stepDMR, CAS-ANOVA (`R`-code for CAS-ANOVA can be found at `http://www4.stat.ncsu.edu/ bondell/Software/ CasANOVA/CasANOVA.R`) and stepBIC, which is a stepwise backward procedure implemented in `stepAIC` function from `R` package `MASS`.

The results are summarized in Table 1. True model (TM) represents the percentage of time the procedure chose the entirely correct model. Correct factors (CF) represents the percentage of time the non-significant factors were eliminated and the true factor was kept. TPR represents the average percentage of true differences found, whereas FNR represents the average percentage of false differences which were correctly identified. 1-Sen and 1-Spe are defined according to Definition 5, MSEP stands for mean squared error of prediction for new data and MD is mean dimension of the selected model, both with standard deviations.

The results of Experiment 1 indicate that DMR algorithms performed at least twice better than CAS-ANOVA in terms of choosing the true model. Our procedures chose approximately smaller models with dimension closer to the dimension of the underlying true model, whose number of parameters equals three. There are

no significant differences between mean squared errors of prediction for all considered algorithms. The main conclusion, that DMR procedures choose models which are smaller and closer to the proper one, is supported by the obtained values of 1 - sensitivity and 1 - specificity, which indicates smaller errors made by our methods.

Table 1: Results of the simulation study, Experiment 1.

n	Algorithm	TM	CF	TPR	FNR	1-Sen	1-Spe	MSEP (±sd)	MD (±sd)
96	DMR	44	74	96	88	.1	.08	1.08±.17	3.5±.7
	stepDMR	44	74	96	88	.1	.08	1.08±.17	3.5±.7
	CAS-ANOVA	16	82	97	77	.06	.2	1.09±.17	4.8±1.7
	stepBIC	0	97	100	52	0	.51	1.08±.16	8.1±.4
192	DMR	67	83	99	93	.03	.04	1.03±.11	3.3±.6
	stepDMR	67	83	99	93	.03	.04	1.03±.11	3.3±.6
	CAS-ANOVA	32	92	100	86	.01	.13	1.05±.11	4.3±1.3
	stepBIC	0	99	100	53	0	.5	1.04±.11	8 ±.2
384	DMR	77	88	100	96	0	.03	1.02±.07	3.3±.5
	stepDMR	76	88	100	96	0	.03	1.02±.07	3.2±.5
	CAS-ANOVA	49	97	100	91	0	.08	1.03±.08	3.8±1
	stepBIC	0	99	100	53	0	.5	1.03±.07	8 ±.2

In Figure 3 an exemplary run of DMR algorithm is illustrated. Each row of the figure corresponds to a model on the nested path of models searched through by the algorithm. The left panel shows consecutive partitions of factors on the path of the algorithm: in the first row there is the full model, the last row shows the model containing only intercept. The true model is

$$t = (P_1 = \{\{1,2\},\{3,4,5,6\},\{7,8\}\}, P_2 = \{1,2,3,4\}, P_3 = \{1,2,3\}).$$

Bold dotted horizontal lines represent the final cut (a model with minimal BIC) of DMR algorithm. One can see that the best partition of the first factor consists of three groups, exactly the same as in the true effect vector, second and third factors are removed from the model.

The right panel of the figure shows approximate Bayes factors for models on the path. Two vertical lines represent the values of $\frac{1}{3}$ and $\frac{1}{10}$, which correspond to the Jeffreys scale [11] for interpretation of Bayes factors. We can see that there is at least substantial evidence to use the chosen model (with minimal BIC).

In Table 2 the results of computation times for several algorithms are summarized. All values are divided by the computation time of `lm.fit` function, which fits the linear model with the use of QR decomposition of the model matrix. The results for CAS-ANOVA are given only for one value of λ. By default, the searched lambda grid is of length 50. Hence, DMR is several hundred times faster than CAS-ANOVA.

6.2 Experiment 2

In the second simulation study a model containing not only categorical predictors, but also continuous variables is considered. The response y is generated from the

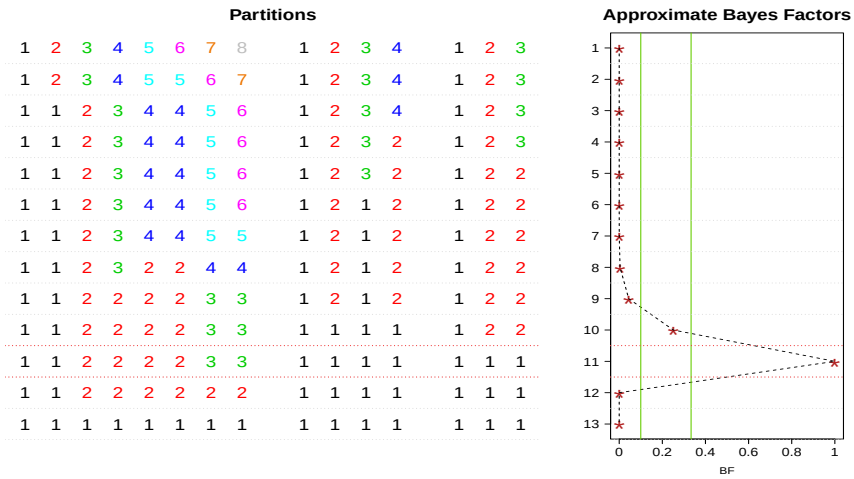


Figure 3: An example run of DMR algorithm with Bayes factors.

Table 2: Computation times divided by the computation time of `lm.fit`.

k	n	DMR	CAS-ANOVA
1	96	35	122
3	288	15	45
21	2016	4	24

model:

$$\begin{aligned} y =& V_0\alpha_0 + V_1\alpha_1 + \varepsilon \\ =& V_0(1,0,1,0,1,0,1,0)^T + V_1(0,0,-2,-2,-2,-2,4,4)^T + \varepsilon, \end{aligned}$$

where V_0 was generated from the multivariate normal distributions with autoregressive correlation structure with $\rho = 0.8$. The first $2 \cdot 16 \cdot k$ observations come from a distribution with mean vector $(1,1,0,0,0,0,0,0)^T$, then $4 \cdot 16 \cdot k$ observations with mean vector $(0,0,1,1,1,1,0,0)^T$ and the last $2 \cdot 16 \cdot k$ observations with mean vector $(0,0,0,0,0,0,1,1)^T$, according to the underlying true partition of the factor, $k = 1, 2, 4$, hence $n = 128 \cdot k$. V_1 is matrix of dummy variables decoding levels of the factor and $\varepsilon \sim \mathcal{N}(0, \mathbb{I}_n)$.

Table 3 shows the results of simulation study. The data was generated 1000 times. As in Experiment 1 we compared four algorithms: DMR, stepDMR, CAS-ANOVA and stepBIC. Despite the additional continuous correlated variables the obtained results show a considerable advantage of DMR algorithms over other methods.

Note that in both Experiment 1 and Experiment 2 DMR algorithm having lower time complexity performed comparatively to stepDMR.

Table 3: Results of the simulation study, Experiment 2.

n	Algorithm	TM	1-Sen	1-Spe	MSEP (±sd)	MD (±sd)
128	DMR	69	0	.05	1.08±.14	7.4±.7
	stepDMR	68	0	.04	1.08±.14	7.4±.6
	CAS-ANOVA	15	.09	.31	1.11±.15	9.2±1.7
	stepBIC	0	0	.58	1.11±.15	12.2±.4
256	DMR	82	0	.02	1.03±.09	7.2±.5
	stepDMR	81	0	.02	1.03±.09	7.2±.5
	CAS-ANOVA	27	.1	.25	1.05±.1	8.5±1.4
	stepBIC	0	0	.57	1.04±.1	12.1±.3
512	DMR	86	0	.02	1.02±.06	7.1±.4
	stepDMR	86	0	.02	1.02±.06	7.1±.4
	CAS-ANOVA	43	.12	.2	1.03±.06	7.9±1.1
	stepBIC	0	0	.56	1.03±.06	12 ±.3

6.3 Real data example, `Cars93`

The data `Cars93` used in this example comes from `R` library `MASS`. 81 observations of 7 categorical and 15 continuous predictors are given. The logarithm of fuel consumption is under investigation. The factors are: presence of airbags, number of cylinders, drive train, availability of manual transmission version, origin, number of passengers and type. The continuous variables are: fuel tank capacity, length, logarithm of engine size, maximum horsepower, logarithm of engine revolutions per mile, luggage capacity, price, rear seat room, revs per minute at maximum horsepower, U-turn space, weight, wheelbase and width. These give 16 and 13 parameters respectively.

Model selection was performed using four methods: DMR, stepDMR, CAS-ANOVA and stepBIC. Characteristics of the chosen models are shown in Table 4 with results for the full model added for comparison. Figure 4 illustrates partitions of factors in the model selected by DMR procedure. From the set of continuous variables weight, wheelbase and logarithm of engine size were chosen.

Table 4: Cars93 data analysis results for different selection methods. Approximate Bayes factors calculated with respect to model chosen by stepDMR.

Selection method	Number of parameters	R^2	BIC	Bayes factor
Full model	31	.92	-83.7	$2.7 \cdot 10^{-16}$
DMR	11	.9	-152.7	0.27
stepDMR	11	.9	-155.3	1
CAS-ANOVA	4	.8	-130.5	$4.1 \cdot 10^{-6}$
stepBIC	14	.89	-133.8	$2.1 \cdot 10^{-5}$

We can conclude that DMR procedures chose much better models than other

compared methods in terms of BIC. Approximate Bayes factors for the full model and models chosen by stepBIC and CAS-ANOVA indicate the decisive evidence in favor of the model chosen by stepDMR according to Jeffrey's scale for interpretation of Bayes factors [11].

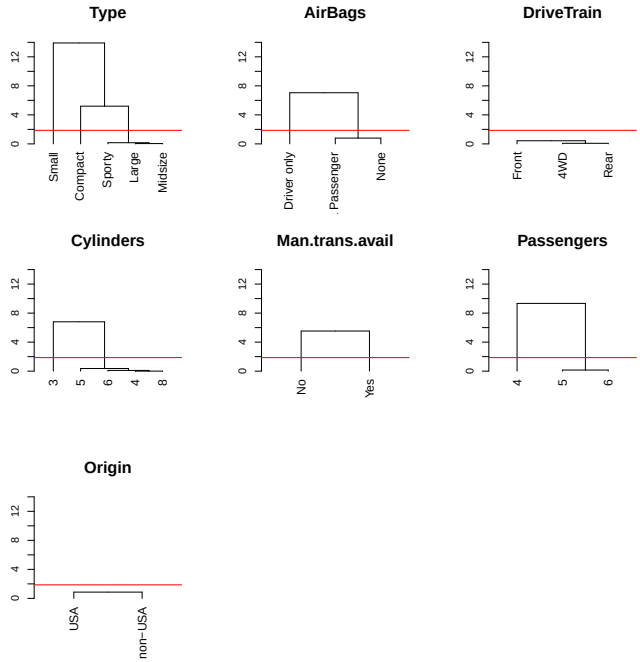


Figure 4: Partitions of factors for the model selected by DMR procedure for Cars93 data.

6.4 Clustering methods

DMR algorithm uses hierarchical clustering for generating a path of nested models. There is a wide spectrum of hierarchical clustering methods available in the statistical software. In order to compare some of them a simulation study was conducted. The results are summarized in Table 5. One can see that method complete gives the most stable results, therefore we decided to use it in simulation studies described in the article.

Table 5: Comparison of clustering methods for DMR algorithm for data as in Experiment 1.

n	Method	TM	1-Sen	1-Spe	MSEP (±sd)	MD (±sd)
96	complete	44	.11	.08	1.09±.17	3.5±.7
	single	36	.12	.09	1.10±.18	3.6±.9
	Ward's	44	.11	.08	1.09±.17	3.5±.7
192	complete	67	.03	.04	1.03±.11	3.3±.5
	single	65	.03	.04	1.03±.11	3.4±.6
	Ward's	67	.03	.04	1.03±.11	3.3±.5
384	complete	78	0	.02	1.01±.07	3.2±.5
	single	80	0	.02	1.01±.07	3.2±.5
	Ward's	78	0	.02	1.01±.07	3.2±.5

7 Conclusions

In this article novel methods of linear model selection combining deleting continuous variables with merging levels of factors were proposed. Both of them are based on ordering elementary hypotheses using squared t-statistics and choosing the best model according to GIC in the nested family of models.

We showed by simulations that DMR algorithms work well for small data sets. In comparison to other methods they gave much higher rates of choosing the true model. The time complexities of the algorithms are $O(np^2)$ and $O(np^3)$ for DMR and stepDMR respectively. In the simulations the algorithms worked several hundred times faster than for example CAS-ANOVA algorithm. For large data sets some asymptotic results were obtained. We proved that even under assumption that the number of predictors grows with the number of observations, $p_n \to \infty$, DMR algorithm is consistent.

Furthermore, a generalization of traditional measures of performance was introduced. These measures do not diminish the influence of factors with a small number of levels and continuous variables.

As a future work we plan to generalize the methods on several classes of models such as linear models with $p_n > n$, Generalized Linear Models and Cox models for survival data.

Acknowledgments

Study was supported by research fellowship within "Information technologies: research and their interdisciplinary applications" agreement number POKL.04.01.01-00-051/10-00.

Appendix

A Calculation of RSS for model with linear constraints

Let us consider a linear model:

$$\begin{cases} y = X\beta + \varepsilon \\ A_0\beta = c, \end{cases} \tag{6}$$

where A_0 is a $(p-q) \times p$ given matrix and c is a given vector of length $(p-q)$, which define a linear subspace of parameter space $\mathbb{R}^p$ described by a set of linear hypotheses concerning the vector of parameters β. The objective is to calculate residual sum of squares $\|y - X\widehat{\beta}_c\|^2$, where $\widehat{\beta}_c$ is an estimator of the parameter vector β with given constraints.

The following QR decomposition is performed

$$X = Q_p R,$$

where Q_p is $n \times p$ orthogonal matrix and R is $p \times p$ upper triangular matrix. Let us denote $S = R^{-T} A_0^T$, then

$$\begin{cases} Q_p^T y = R\beta + Q_p^T \varepsilon \\ S^T R\beta = c. \end{cases}$$

and after substitution we get

$$\begin{cases} z = \gamma + \eta \\ U^T W_{p-q}^T \gamma = c, \end{cases} \tag{7}$$

where W_{p-q} and U are respectively $p \times (p-q)$ orthogonal matrix and $(p-q) \times (p-q)$ upper triangular matrix from the QR decomposition of matrix S. If we denote

$$W_{p-q}^T \gamma = U^{-T} c = e$$

and

$$W = [W_q, W_{p-q}],$$

where W_q is an orthogonal complement of the matrix W_{p-q}, then equation (7) becomes

$$\begin{cases} W_q^T z = W_q^T \gamma + W_q^T \eta \\ e = W_{p-q}^T \gamma. \end{cases}$$

Therefore an unbiased estimator of γ with constraints satisfies the following equation

$$\begin{bmatrix} W_q^T z \\ e \end{bmatrix} = W^T \widehat{\gamma}_c, \tag{8}$$

multiplying (8) by W, we obtain

$$W_q W_q^T z + W_{p-q} e = \widehat{\gamma}_c,$$

then

$$(I_p - W_{p-q} W_{p-q}^T) z + W_{p-q} e = \widehat{\gamma}_c = R\widehat{\beta}_c.$$

The residual sum of squares for the model with linear constraints (6) can be written as

$$\begin{aligned}\|y - X\widehat{\beta}_c\|^2 =& \|Q_p^T y - R\widehat{\beta}_c\|^2 = \|W_{p-q}W_{p-q}^T z - W_{p-q}e\|^2 \\ =& \|W_{p-q}^T z - e\|^2 = \sum_{i=1}^{p-q}(w_i^T z - e_i)^2,\end{aligned} \tag{9}$$

where w_i is the i-th column of matrix W_{p-q}. Hence, for each additional hypothesis the residual sum of squares can be easily calculated from equation (9).

B Proof of Theorem 1

Lemma 1 (Klotz [12] Section 14.3)**.** *Solving an optimization problem:*

$$\widehat{\beta}_0 = \arg\min_{\beta} ||y - X\beta||^2, \text{ where } A_0\beta = 0 \text{ yields}$$

$$\widehat{\beta}_0 = \widehat{\beta} - (X^T X)^{-1}A_0^T(A_0(X^T X)^{-1}A_0^T)^{-1}A_0\widehat{\beta},$$

where $\widehat{\beta} = (X^T X)^{-1}X^T y$.

Let us denote the following matrices:

$$H = X(X^T X)^{-1}X^T,$$

$$H_0 = X(X^T X)^{-1}A_0^T(A_0(X^T X)^{-1}A_0^T)^{-1}A_0(X^T X)^{-1}X^T$$

which are matrices of orthogonal projections and matrix

$$H_1 = Z_1(Z_1^T Z_1)^{-1}Z_1^T = XA^1(A^{1^T}X^T XA^1)^{-1}A^{1^T}X^T.$$

Note that $X\widehat{\beta}_0 = (H - H_0)y$.

Lemma 2. *H_1 is a matrix of an orthogonal projection and*

$$H_1 = H - H_0.$$

Proof. Note that

$$(X^T X)^{-1} = (A^T A^{-T}X^T XA^{-1}A)^{-1} = A^{-1}(A^{-T}X^T XA^{-1})^{-1}A^{-T},$$

hence

$$\begin{aligned}(A_0(X^T X)^{-1}A_0^T)^{-1} =& (A_0A^{-1}(A^{-T}X^T XA^{-1})^{-1}A^{-T}A_0^T)^{-1} \\ =& \left[\begin{bmatrix} 0 & \mathbb{I} \end{bmatrix}(Z^T Z)^{-1}\begin{bmatrix} 0 \\ \mathbb{I} \end{bmatrix}\right]^{-1} = (G^{00})^{-1},\end{aligned}$$

where

$$G = \begin{bmatrix} G_{11} & G_{10} \\ G_{01} & G_{00} \end{bmatrix} = \begin{bmatrix} Z_1^T Z_1 & Z_1^T Z_0 \\ Z_0^T Z_1 & Z_0^T Z_0 \end{bmatrix} = Z^T Z \text{ and } G^{-1} = \begin{bmatrix} G^{11} & G^{10} \\ G^{01} & G^{00} \end{bmatrix}$$

and

$$\begin{aligned} H =& X(X^TX)^{-1}X^T \\ =& XA^{-1}(A^{-T}X^TXA^{-1})^{-1}A^{-T}X^T = Z(Z^TZ)^{-1}Z^T. \end{aligned}$$

We have also that

$$A_0(X^TX)^{-1}X^T = A_0A^{-1}(Z^TZ)^{-1}A^{-T}X^T = A_0A^{-1}(Z^TZ)^{-1}Z^T,$$

then

$$\begin{aligned} H_0 =& X(X^TX)^{-1}A_0^T(G^{00})^{-1}A_0(X^TX)^{-1}X^T \\ =& Z(Z^TZ)^{-1}\begin{bmatrix} 0 \\ \mathbb{I} \end{bmatrix}(G^{00})^{-1}\begin{bmatrix} 0 & \mathbb{I} \end{bmatrix}(Z^TZ)^{-1}Z^T \\ =& Z\begin{bmatrix} G^{10} \\ G^{00} \end{bmatrix}(G^{00})^{-1}\begin{bmatrix} G^{01} & G^{00} \end{bmatrix}Z^T = Z\begin{bmatrix} G^{10}(G^{00})^{-1}G^{01} & G^{10} \\ G^{01} & G^{00} \end{bmatrix}Z^T. \end{aligned}$$

Using above results we get

$$\begin{aligned} H - H_0 =& Z(Z^TZ)^{-1}Z^T - Z\begin{bmatrix} G^{10}(G^{00})^{-1}G^{01} & G^{10} \\ G^{01} & G^{00} \end{bmatrix}Z^T \\ =& \begin{bmatrix} Z_1 & Z_0 \end{bmatrix}\begin{bmatrix} G^{11} - G^{10}(G^{00})^{-1}G^{01} & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} Z_1^T \\ Z_0^T \end{bmatrix} \\ =& Z_1(G^{11} - G^{10}(G^{00})^{-1}G^{01})Z_1^T. \end{aligned}$$

From the definition of H_1

$$H_1 = Z_1(Z_1^TZ_1)^{-1}Z_1^T = Z_1(G_{11})^{-1}Z_1^T \tag{10}$$

and from Schur complement for matrix G^{-1} we have

$$(G_{11})^{-1} = G^{11} - G^{10}(G^{00})^{-1}G^{01}.$$

□

Hence, the predictions for constrained problem can be obtained through projecting the observations on the space spanned by columns of the model matrix for the equivalent unconstrained problem.

B.1 RSS lemmas

Lemmas concerning dependencies between residual sums of squares have similar construction to those described by Chen and Chen, 2008 [13].

It follows from Lemma 2 that for each feasible model m a hat matrix H_m can be obtained according to equation (10), so that residual sum of squares for model m is defined as

$$RSS_m = \|y - H_m y\|^2$$

and can be decomposed into three parts

$$\begin{aligned} RSS_m = \|y - H_m y\|^2 =& (X_t\beta_t + \varepsilon)^T(\mathbb{I} - H_m)(X_t\beta_t + \varepsilon) \\ =& \beta_t^T X_t^T(\mathbb{I} - H_m)X_t\beta_t + 2\beta_t^T X_t^T(\mathbb{I} - H_m)\varepsilon + \varepsilon^T H_m\varepsilon. \end{aligned}$$

When $t \subseteq m$ we have $H_m X_t = X_t$ and

$$RSS_m = \varepsilon^T H_m \varepsilon.$$

Lemma 3. *Assuming $p_n \prec n$ and r_n is a sequence of real numbers so that $p_n \prec r_n$, then*

$$\log \frac{RSS_t}{RSS_f} <_P \frac{r_n}{n},$$

which signifies $\lim_{n\to\infty} \mathbb{P}\left(\log \frac{RSS_t}{RSS_f} < \frac{r_n}{n}\right) = 1$.

Proof. Note that

$$\frac{RSS_t}{RSS_f} = 1 + \frac{RSS_t - RSS_f}{RSS_f} = 1 + \frac{p_n}{n} W_n,$$

where

$$W_n = \frac{\varepsilon^T(H_f - H_t)\varepsilon}{\varepsilon^T(\mathbb{I} - H_f)\varepsilon} \cdot \frac{n}{p_n}.$$

From Lemma 2, $H_f - H_t$ is matrix of an orthogonal projection with trace $p_n - |t|$, therefore $\psi_1 = \varepsilon^T(H_f - H_t)\varepsilon \sim \sigma^2\chi^2_{p_n-|t|}$ and $\psi_2 = \varepsilon^T(\mathbb{I} - H_f)\varepsilon \sim \sigma^2\chi^2_{n-p_n}$, we get

$$\mathbb{E}\left(\frac{\psi_1}{p_n}\right) = \frac{\sigma^2(p_n - |t|)}{p_n}, \ \operatorname{Var}\left(\frac{\psi_1}{p_n}\right) = \frac{2\sigma^4(p_n - |t|)}{p_n^2}$$

and either if $p_n \xrightarrow{n\to\infty} \infty$ then $\operatorname{Var}\left(\frac{\psi_1}{p_n}\right) \xrightarrow{n\to\infty} 0$ and from Chebyshev's inequality $\frac{\psi_1}{p_n} \xrightarrow{n\to\infty} \sigma^2$ in probability or if p_n is bounded, then $\frac{\psi_1}{p_n}$ is bounded in probability. Analogously for ψ_2 we have

$$\mathbb{E}\left(\frac{\psi_2}{n}\right) = \frac{\sigma^2(n - p_n)}{n}, \ \operatorname{Var}\left(\frac{\psi_2}{n}\right) = \frac{2\sigma^4(n - p_n)}{n^2}$$

and since $p_n \prec n$ from Chebyshev's inequality $\frac{\psi_2}{n} \xrightarrow{n\to\infty} \sigma^2$ in probability.

Therefore $W_n = O_P(1)$ and $\frac{RSS_t}{RSS_f} = 1 + O_P\left(\frac{p_n}{n}\right)$. Hence

$$\log\left(\frac{RSS_t}{RSS_f}\right) = \log\left(1 + \frac{p_n}{n}W_n\right) \le \frac{p_n}{n}W_n = O_P\left(\frac{p_n}{n}\right) <_P \frac{r_n}{n}.$$

□

Lemma 4. *Assuming that $p_n \prec d_n$ we have for all $m \in \mathcal{M}_\mathcal{V}$ and all $\delta > 1$*

$$\log\left(\frac{RSS_m}{RSS_t}\right) \ge_P \log\left(1 + \frac{d_n}{\delta\sigma^2 \cdot n}\right),$$

where $d_n = \min_{m\in\mathcal{M}_\mathcal{V}} \beta_t^T X_t^T(\mathbb{I} - H_m)X_t\beta_t$.

Proof. Using the fact that

$$\frac{1}{n}RSS_t = \frac{\varepsilon^T(\mathbb{I} - H_t)\varepsilon}{n} = \sigma^2 + o_P\big(1\big)$$

and denoting

$$RSS_m - RSS_t = d_{nm} + Z_{nm} + W_{nt} - W_{nm},$$

where $d_{nm} = \beta_t^T X_t^T(\mathbb{I} - H_m)X_t\beta_t$, $Z_{nm} = 2\beta_t^T X_t^T(\mathbb{I} - H_m)\varepsilon$, $W_{nt} = \varepsilon^T H_t\varepsilon$ and $W_{nm} = \varepsilon^T H_m\varepsilon$.

Note that

$$d_{nm} \geq d_n,\ Z_{nm} \sim \mathcal{N}(0, 4\sigma^2 d_{nm}),\ W_{nt} \sim \sigma^2\chi^2_{|t|} \text{ and } W_{nm} \sim \sigma^2\chi^2_{p_n - 1}.$$

Using assumptions, $\frac{Z_{nm}}{d_{nm}}$, $\frac{W_{nt}}{d_{nm}}$ and $\frac{W_{nm}}{d_{nm}}$ either are bounded in probability if p_n is bounded or if $p_n \xrightarrow{n\to\infty} \infty$ are $O_P\big(1\big)$ from Chebyshev's inequality.

Henceforth we have

$$RSS_m - RSS_t = d_{nm}\left(1 + \frac{Z_{nm}}{d_{nm}} + \frac{W_{nt}}{d_{nm}} - \frac{W_{nm}}{d_{nm}}\right) = d_{nm}\Big(1 + O_P\big(1\big)\Big).$$

As a result

$$\begin{aligned}\log\frac{RSS_m}{RSS_t} &= \log\left(1 + \frac{RSS_m - RSS_t}{n\frac{RSS_t}{n}}\right)\\ &= \log\left(1 + \frac{d_{nm}}{n\sigma^2}\Big(1 + O_P\big(1\big)\Big)\right) \geq_P \log\left(1 + \frac{d_n}{\delta\sigma^2 n}\right) \text{ for } \delta > 1.\end{aligned}$$

□

Lemma 5. *Assuming that $r_n \prec n$ for all $\delta > 1$ we have*

$$\max_{t\subseteq m\subseteq\mathcal{M}}\Big(\log RSS_m\Big) + \log\left(1 + \frac{d_n}{\delta\sigma^2 n}\right) \leq_P \min_{t\not\subseteq m\subseteq\mathcal{M}}\Big(\log RSS_m\Big).$$

Proof. Let us denote $a = \log\big(1 + \frac{d_n}{\delta\sigma^2 n}\big)$, then from Lemma 4 we get

$$\begin{aligned}\max_{t\subseteq m\subseteq\mathcal{M}}\Big(\log RSS_m\Big) + a &= \log RSS_t + a \leq_P \min_{m\in\mathcal{M}_\mathcal{V}} \log RSS_m - a + a\\ &\leq_P \min_{t\not\subseteq m\subseteq\mathcal{M}}\Big(\log RSS_m\Big).\end{aligned}$$

□

Corollary 1. *From Lemma 5 and properties of residual sum of squares, we have with probability tending to 1 the following order of models RSS:*

$$\begin{aligned}RSS_f &\leq \max_{|m|=|f|-1,\ t\subseteq m} RSS_m \leq \max_{|m|=|f|-2,\ t\subseteq m} RSS_m\\ &\leq \ldots \leq \max_{|m|=|t|+1,\ t\subseteq m} RSS_m \leq RSS_t \leq_P \min_{|m|=|f|-1,\ t\not\subseteq m} RSS_m\\ &\leq \min_{|m|=|f|-2,\ t\not\subseteq m} RSS_m \leq \ldots \leq \min_{|m|=2,\ t\not\subseteq m} RSS_m \leq \min_{|m|=1} RSS_m.\end{aligned} \tag{11}$$

Note that since $|t| < \infty$, there is a finite number of models bigger and having RSS not greater than the true model t.

Corollary 2. *Since models from $\mathcal{M}_{\mathcal{V}}$ and $\mathcal{M}_{\mathcal{T}}$ have the same number of parameters, we have*

$$GIC(t) \leq_P \max_{m \in \mathcal{M}_{\mathcal{T}}} GIC(m) \leq_P \min_{m \in \mathcal{M}_{\mathcal{V}}} GIC(m).$$

where the left inequality follows form Lemma 3 and right inequality from Lemma 5.

Corollary 3. *For every step of a backward stepwise elimination algorithm, if the true model t is on the path searched through, the inequality from Corollary 2 is preserved. Hence, GIC is a consistent model selection criterion.*

B.2 Ordering of squared t-statistics

Lemma 6. *Suppose $\mathcal{M}_{\mathcal{TV}} = \mathcal{M}_{\mathcal{T}} \cup \mathcal{M}_{\mathcal{V}}$ is a set of all models of size $|f| - 1$. For each $m \in \mathcal{M}_{\mathcal{TV}}$, which corresponds to one elementary hypothesis,*

$$T_m^2 = (n - |f|)\frac{RSS_m - RSS_f}{RSS_f},$$

where T_m is t-statistic for the full model with hypothesis $h: \ A_0\beta = 0$, where A_0 is $1 \times |f|$ matrix.

Proof. From Lemma 1

$$RSS_f - RSS_m = \widehat{\beta}^T A_0^T (A_0 (X^T X)^{-1} A_0^T)^{-1} A_0 \widehat{\beta},$$

hence

$$T_m^2 = \frac{(A_0\widehat{\beta})^2}{\widehat{\mathrm{Var}}(A_0\widehat{\beta})} = \frac{(A_0\widehat{\beta})^2}{A_0\widehat{\mathrm{Var}}(\widehat{\beta})A_0^T} = \frac{(A_0\widehat{\beta})^2}{\widehat{\sigma}^2 A_0 (X^T X)^{-1} A_0^T} = \frac{RSS_f - RSS_m}{\widehat{\sigma}^2},$$

where $\widehat{\sigma}^2 = \frac{RSS_f}{n-|f|}$. □

Corollary 4. *It follows from Lemma 6 that the ordering of models $m \in \mathcal{M}_{\mathcal{TV}}$ with respect to squared t-statistics is equivalent to ordering them with respect to the values of residual sum of squares for these models.*

Corollary 5. *It follows from Corollary 1 and Lemma 6, that for sufficiently large n we have*

$$\max_{m \in \mathcal{M}_{\mathcal{T}}} T_m^2 <_P c <_P \min_{m \in \mathcal{M}_{\mathcal{V}}} T_m^2,$$

where c is a positive constant.

In order to prove that hierarchical clustering implies the proper order of accepting elementary hypotheses, in the sense that the true hypotheses preface false ones, let us introduce some notations.

Let $\mathcal{T}$ stand for a set of all true elementary hypotheses and $\mathcal{V}$ for a set of all false elementary hypotheses. Suppose that $U_1^{(s)}, U_2^{(s)}, \ldots U_{k_s}^{(s)}$ are clusters of levels of a factor given by the s-th step of hierarchical clustering. $\beta_1^{(s)}, \beta_2^{(s)}, \ldots \beta_{k_s}^{(s)}$ correspond to the common value of the parameters for the factor levels for the clusters. We assume that each $\beta_i^{(s)}$ is defined as the parameter for the level with the smallest index in the cluster.

Lemma 7. *In each step of hierarchical clustering in DMR algorithm the recalculation of dissimilarity matrix using linkage criterion, which is a convex combination of minimum and maximum of the pairwise distances between clusters, preserves the inequality:*

$$\max_{\{\beta_{i_1}^{(s)}=\beta_{i_2}^{(s)}\}\in\mathcal{T}} d\left(U_{i_1}^{(s)}, U_{i_2}^{(s)}\right) <_P c <_P \min_{\{\beta_{i_1}^{(s)}=\beta_{i_2}^{(s)}\}\in\mathcal{V}} d\left(U_{i_1}^{(s)}, U_{i_2}^{(s)}\right),$$

where d is the dissimilarity measure and $c > 0$.

Proof. By induction: for $s = 1$ the inequality is preserved from Corollary 5 and the hypothesis with the smallest value of squared t-statistic is accepted. Let us assume that the inequality is also preserved for step s.

If $s + 1 > |\mathcal{T}|$ all true hypotheses are already accepted, so the inequality holds trivially for all following steps.

If $s + 1 \leq |\mathcal{T}|$ the algorithm of clustering chooses the hypothesis with minimal value of dissimilarity measure to accept. Suppose that this hypothesis has a form

$$\beta_{i_1}^{(s)} = \beta_{i_2}^{(s)}.$$

From the previous step s we know that this hypothesis is true. The dissimilarity measures between the new merged cluster and every other cluster $U_{i_3}^{(s)}$ have to be recalculated. We have two cases:

1. If $\beta_{i_3}^{(s)} = \beta_{i_1}^{(s)}$ is true then from transitivity $\beta_{i_3}^{(s)} = \beta_{i_2}^{(s)}$ is also true, hence

$$d\left(U_{i_3}^{(s)}, U_{i_1}^{(s)}\right) = d_{i_3 i_1}^{(s)} <_P c,$$

$$d\left(U_{i_3}^{(s)}, U_{i_2}^{(s)}\right) = d_{i_3 i_2}^{(s)} <_P c$$

   so the convex combination of these two values is smaller than c:

$$\begin{aligned} &d\left(\{U_{i_1}^{(s)}, U_{i_2}^{(s)}\} = U_{j_1}^{(s+1)}, U_{i_3}^{(s)} = U_{j_2}^{(s+1)}\right) \\ &\quad = \alpha \cdot \min\left(d_{i_3 i_1}^{(s)}, d_{i_3 i_2}^{(s)}\right) + (1-\alpha) \cdot \max\left(d_{i_3 i_1}^{(s)}, d_{i_3 i_2}^{(s)}\right) <_P c \end{aligned}$$

   for each $\alpha \in [0, 1]$.

2. Analogously if $\beta_{i_3}^{(s)} = \beta_{i_1}^{(s)}$ is false the convex combination of these two values is greater than c:

$$\begin{aligned} &d\left(\{U_{i_1}^{(s)}, U_{i_2}^{(s)}\} = U_{j_1}^{(s+1)}, U_{i_3}^{(s)} = U_{j_2}^{(s+1)}\right) \\ &\quad = \alpha \cdot \min\left(d_{i_3 i_1}^{(s)}, d_{i_3 i_2}^{(s)}\right) + (1-\alpha) \cdot \max\left(d_{i_3 i_1}^{(s)}, d_{i_3 i_2}^{(s)}\right) >_P c \end{aligned}$$

for each $\alpha \in [0,1]$. Note that linkage criteria: single, complete and average are a convex combination of minimum and maximum of the pairwise distances between clusters.

□

Proof of Theorem 1. It follows from Corollary 5 and Lemma 7 that for sufficiently large n on the path of models generated by DMR algorithm models with only true hypotheses accepted preface models with at least one false hypothesis accepted. Hence the true model t is on the path searched through. Therefore, from Corollary 3 we have that DMR algorithm is a consistent model selection method.

References

[1] Tukey, J.W.: Comparing individual means in the analysis of variance. Biometrics (1949) 99–114

[2] Scott, A., Knott, M.: A cluster analysis method for grouping means in the analysis of variance. Biometrics (1974) 507–512

[3] Calinski, T., Corsten, L.: Clustering means in anova by simultaneous testing. Biometrics (1985) 39–48

[4] Corsten, L., Denis, J.: Structuring interaction in two-way tables by clustering. Biometrics (1990) 207–215

[5] Bondell, H.D., Reich, B.J.: Simultaneous factor selection and collapsing levels in anova. Biometrics **65**(1) (2009) 169–177

[6] Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) (1996) 267–288

[7] Gertheiss, J., Tutz, G.: Sparse modeling of categorial explanatory variables. The Annals of Applied Statistics **4**(4) (2010) 2150–2180

[8] Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. The Annals of Statistics **32**(2) (2004) 407–499

[9] Ciampi, A., Lechevallier, Y., Limas, M.C., Marcos, A.G.: Hierarchical clustering of subpopulations with a dissimilarity based on the likelihood ratio statistic: application to clustering massive data sets. Pattern Analysis & Applications **11**(2) (2008) 199–220

[10] Zheng, X., Loh, W.Y.: Consistent variable selection in linear models. Journal of the American Statistical Association **90**(429) (1995) 151–156

[11] Kass, R.E., Raftery, A.E.: Bayes factors. Journal of the American Statistical Association **90**(430) (1995) 773–795

[12] Klotz, J.H.: A computational approach to statistics. Department of Statistics, University of Wisconsin, Madison (2006)

[13] Chen, J., Chen, Z.: Extended bayesian information criteria for model selection with large model spaces. Biometrika **95**(3) (2008) 759–771

Selection and Prediction for Linear Models using Random Subspace Methods

Jan Mielniczuk[1,2] and Paweł Teisseyre[1]

[1] Institute of Computer Science, Polish Academy of Sciences,
ul. Jana Kazimierza 5, 01-248 Warsaw, Poland
[2] Warsaw University of Technology, Faculty of Mathematics and Information Science,
ul. Koszykowa 75, 00-662 Warsaw, Poland

Abstract

In the paper two versions of random subspace method (RSM) for linear regression models are considered. In the original RSM in regression framework introduced in [1] subsets of regressors are sampled with equal probabilities from all subsets of a chosen size and then fitted in order to construct scores of all variables. Proposed modifications consist in preferential sampling of variables according to preliminary assessment of their importance and/or initial screening of features. Some properties of the proposed methods are discussed and their performance as prediction method for moderate sample sizes is studied by means of simulations. The first variant, weighted RSM, behaves promisingly when the dependence between regressors is not very strong and is also much less computationally expensive than the RSM.

1 Introduction

Model selection in high dimensional feature space plays an important role in diverse fields of sciences, engineering and humanities. Examples include microarray analysis, Quantitative Trait Loci (QTL) analysis, Genome-Wide Association Study (GWAS), drug design analysis and high-frequency financial analysis among others. In such problems it is challenging to find important variables out of thousands of predictors, with number of observations usually in tens or hundreds. In [2] the need for development of high-dimensional data analysis is discussed. Since the true relationship in data is usually unknown, very often it is worthwhile to include higher degree terms as well as interaction terms to the model. This can substantially increase the number of potential attributes. The problem recently has received much attention in the statistical and machine learning literature. An intensively studied line of research is focused on regularization (cf. e.g. [3], [4]). In many approaches a preliminary feature selection is used, e.g. in [5] a method of dimensionality reduction based on so called sure independence screening is proposed. Let us also mention procedures using information criteria modified to high-dimensional setup, see e.g. [6] or [7]. Recently a novel approach based on the adaptation of the random subspace method (RSM) in the regression context has been proposed in [1].

In the RSM a random subset m with $|m|$ features, smaller than the number of all predictors p and a number of observations n, is chosen and the model is fitted in

the reduced feature space. Selected features are assigned weights describing their relevance in the considered submodel. In order to cover a large portion of features in the dataset, the selection is repeated B times and the cumulative weights (called final scores) are computed. The results of all iterations are combined in a list of p features ordered according to final scores. The final model can be constructed based on predetermined number of the most significant variables or using selection method applied to the nested list of models given by the ordering. The simulation experiments described in [1] indicate that the proposed method behaves promisingly when its prediction errors are compared with errors of penalty-based methods such as the lasso and it has much smaller false discovery rate than the other methods considered. One drawback of this method is its computational cost. When the number of features p is large we should take large B in order to ensure that all variables are likely to be selected to random subspaces.

In this paper we propose two modifications of the original algorithm. In the first method, called weighted random subspace method (WRSM), variables are chosen to subspaces with probabilities proportional to the values of individual weights when univariate models are fitted. In the second method, called screened random subspace method (SRMS), the preliminary feature screening is performed. Both approaches reduce the computational cost of the original procedure.

This paper is organized as follows. The original RSM algorithm is recalled in Section 2.1; the choice of the weights is described in Section 2.2 and in Section 2.3 some additional properties of quantities related to the weights are discussed. The modifications of the RSM are presented in Section 2.4 and the results of numerical experiments are discussed in Section 3. The proofs are relegated to the appendix.

We define now the formal setup of the paper. Let $(\mathbf{Y}, \mathbf{X})$ be the observed data, where $\mathbf{Y} = \mathbf{Y}^{(n)}$ is an $n \times 1$ vector of n responses whose variability we would like to explain and $\mathbf{X} = \mathbf{X}^{(n)}$ is a $n \times p$ design matrix consisting of vectors of p potential regressors collected from n objects. Responses are related to regressors by means of the linear model

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{1}$$

where $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)'$ is an unobservable vector of errors, assumed to have $N(0, \sigma^2\mathbf{I})$ distribution. Vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$ is an unknown vector of parameters. We consider two scenarios: the case of deterministic and random $\mathbf{X}$. In the latter case rows of $\mathbf{X}$ constitute n independent realizations of p-dimensional random variable $\mathbf{x}$ and coordinates of vector $\mathbf{Y}$ form an i.i.d. sample distributed as $y = \mathbf{x}'\boldsymbol{\beta} + \varepsilon$. A distribution of $\mathbf{x} = (x_1, \ldots, x_p)'$ may be arbitrary, in particular the distribution of its first coordinate may be point mass at 1 corresponding to the linear model with an intercept included. The number of attributes p may be larger than n. As any submodel of (1) containing $|m|$ variables $(x_{i_1}, \ldots, x_{i_{|m|}})'$ can be described by set of indices $m = \{i_1, \ldots, i_{|m|}\}$ in order to make notation simpler it will be referred to as model m. We denote by $\mathbf{X}_m$ the matrix composed of the columns of $\mathbf{X}$ with indices in m and by $\mathbf{x}_m$ a subvector of $\mathbf{x}$ consisting of coordinates corresponding to m. Similarly, $\boldsymbol{\beta}_m \in R^{|m|}$ denotes the vector consisting of components of $\boldsymbol{\beta}$ with indices in m. For simplicity, model fitted to data $(\mathbf{Y}, \mathbf{X}_m)$ we will be denoted by $y \sim \mathbf{x}_m$. Usually some covariates are unrelated to the prediction of $\mathbf{Y}$, so that the corresponding coefficients β_i are zero. Model

containing all relevant variables, i.e. those pertaining to nonzero β_i, will be called a true model. The minimal true model $\{i : \beta_i \neq 0\}$ i.e. such that it pertains only to relevant variables will be denoted by t and $|t|$ will be the number of nonzero coefficients. It is assumed that $t \subset \{1, 2, \ldots, p\}$ is unique and t does not change with n.

2 Random Subspace Methods

2.1 Main Algorithm

We first describe the basic algorithm of Random Subspace Method.

RSM Algorithm

1. Input: observed data $(\mathbf{Y}, \mathbf{X})$, number of subset draws B, size of the subspace $|m| < \min(p, n)$. Choice of weights $w_{i,m}$ is described in Section 2.2.
2. Repeat the following procedure for $k = 1, \ldots, B = B_n$, where B_n is such that $B_n \to \infty$ when $n \to \infty$ and starting with $C_{i,0} = 0$ for any i.
   - Randomly draw a model $m^* = \{i^*_1, \ldots, i^*_{|m|}\}$ from the original feature space.
   - Fit model $y \sim \mathbf{x}_{m^*}$ and compute weight w_{i,m^*} for each $i \in m^*$. Set $w_{i,m^*} = 0$ if $i \notin m^*$.
   - Update the counter $C_{i,k} = C_{i,k-1} + I\{i \in m^*\}$.
3. For each variable i compute the final score FS^*_i defined as

$$FS^*_i = \frac{1}{C_{i,B}} \sum_{m^* : i \in m^*} w_{i,m^*}.$$

4. Sort the list of variables according to scores FS^*_i: $FS^*_{i_1} \geq FS^*_{i_2} \ldots \geq FS^*_{i_p}$.
5. Output: Ordered list of variables $\{i_1, \ldots, i_p\}$.

Two parameters need to be set in the RSM: the number of selections B and the subspace size $|m|$. The smaller the size of a chosen subspace (i.e. a subset of features chosen) the larger the chance of missing informative features or missing dependencies between variables. On the other hand for large $|m|$ many spurious variables can be included adding noisy dimensions to the subspace. Note that the subspace size is limited by $\min(n, p)$. In the following the value of parameter $|m|$ is chosen empirically. We concluded from numerical experiments that the reasonable choice is $|m| = \min(n, p)/2$. It follows from the description above that a parallel version of the algorithm is very easy to implement.

2.2 Choice of the weights $w_{i,m}$

In this section we discuss rationale for using a squared value of t-statistic as a weight in RSM procedure. Observe first that a randomly chosen model m in the second step of RSM procedure may be misspecified, in the sense that it may not

contain all significant variables. Thus it is important to investigate the performance of proposed weights in a general case when a considered model may be wrong. This is an interesting issue as it is intuitively clear that when e.g. the most important feature is mistakenly dropped from the model then a spurious feature highly correlated with it may have larger value of t-statistic than other true predictors. We discuss the problem in Theorem 1 which states the conditions under which such a situation can not occur. In particular, it follows from Corollary 2 that when variables are asymptotically uncorrelated the weighting will reflect the correct ordering of variables in the sense that all variables pertaining to the minimal true model will have larger weights than spurious ones.

Consider a submodel m of model (1) containing $|m|$ variables $i_1, \ldots, i_{|m|}$, where $|m|$ is a fixed integer such that $|m| < \min(n, p)$. Model m with i-th variable deleted will be denoted by $m \setminus \{i\}$. We assume that for the considered model m matrix $(\mathbf{X}'_m \mathbf{X}_m)^{-1}$ exists.

Let $\hat{\boldsymbol{\beta}}_m = (\hat{\beta}_{i_1,m}, \ldots, \hat{\beta}_{i_{|m|},m})'$ be the least squares estimator based on model m and

$$T_{i,m} = \hat{\beta}_{i,m}[\hat{\sigma}^2_m (\mathbf{X}'_m \mathbf{X}_m)^{-1}_{i,i}]^{-1/2}, \quad i \in \{i_1, \ldots, i_{|m|}\}$$

be t-statistic corresponding to variable i when model m is fitted to the data. In the above formula $\hat{\sigma}^2_m = (n - |m|)^{-1} RSS(m)$, where $RSS(m) = \mathbf{Y}'(\mathbf{I} - P_m)\mathbf{Y}$ is sum of the squared residuals (residual sum of squares) for model m and P_m is a projection on the column space spanned by the regressors corresponding to this model. The following equality holds

$$\frac{T^2_{i,m}}{n - |m|} = \frac{RSS(m \setminus \{i\}) - RSS(m)}{RSS(m)}. \tag{2}$$

Thus $T^2_{i,m}/(n - |m|)$ is a relative increase of RSS when variable i is dropped from the model m. It follows from (2) and generalized Cochran theorem that $T^2_{i,m}/(n - |m|)$ is a ratio of two independent chi squared distributed random variables: $\chi^2_1(\lambda_1)$ in the case of numerator and $\chi^2_{n-|m|}(\lambda_2)$ for denominator, where parameters of noncentrality are equal $\lambda_1 = ||(P_m - P_{m\setminus\{i\}})\mathbf{X}\beta||^2/(2\sigma^2)$ and $\lambda_2 = ||(I - P_m)\mathbf{X}\beta||^2/(2\sigma^2)$, respectively. It will be shown is Section 2.3 that λ_2 is equal to the Kullback–Leibler divergence between probability density function corresponding to the true model and space spanned by columns of $\mathbf{X}$ corresponding to model m. Note also that due to a variance decomposition for a linear model which includes constant regressor we have

$$\frac{T^2_{i,m}}{n - |m|} = \frac{R^2_m - R^2_{m\setminus\{i\}}}{1 - R^2_m}, \tag{3}$$

where R^2_m is a coefficient of determination for a model m. Equation (3) provides the main motivation for our choice of weights in RSM scheme, that is we consider $w_{i,m} = (n - |m|)^{-1} T^2_{i,m}$. Namely, it indicates that up to a multiplicative factor, $T^2_{i,m}$ is a decrease in R^2 due to leaving out x_i multiplied by a measure of goodness-of-fit $(1 - R^2_m)^{-1}$ of model m and thus it combines two characteristics: importance of a feature within the model m and the importance of the model itself.

In the case of random $\mathbf{X}$ the following quantities will be useful. Assume throughout for simplicity that $\mathbf{E}(x_i) = 0$ for $i \in \{1, \ldots, p\}$. Let $\mathrm{cov}(y, \mathbf{z})$ be the $1 \times |m|$ vector of covariances between y and coordinates of some $|m|$-dimensional random vector $\mathbf{z}$. Let

$$\rho^2_{y,\mathbf{x}_m} = \frac{\mathrm{cov}^2(y, P_m y)}{\mathrm{var}(y)\mathrm{var}(P_m y)} = \frac{\mathrm{var}(P_m y)}{\mathrm{var}(y)} \tag{4}$$

be the squared multiple correlation coefficient between y and its projection on a subspace spanned by coordinates of $\mathbf{x}_m$. It is easy to see that

$$\rho^2_{y,\mathbf{x}_m} = \frac{\mathrm{cov}(y, \mathbf{x}_m)\Sigma^{-1}_{\mathbf{x}_m}\mathrm{cov}(\mathbf{x}_m, y)}{\mathrm{var}(y)}, \tag{5}$$

where $\mathrm{cov}(\mathbf{x}_m, y) = \mathrm{cov}(y, \mathbf{x}_m)'$ and $\Sigma_{\mathbf{x}_m}$ is the variance-covariance matrix of variables corresponding to m. Moreover, it follows that $\rho^2_{y,\mathbf{x}_m}$ equals the maximal value of a squared correlation between y and linear combination of coordinates of $\mathbf{x}_m$, when the coefficients of the combination vary. For $m = \{i\}$ consisting of one element $\rho^2_{y,\mathbf{x}_m}$ is squared correlation coefficient $\rho^2(y, x_i)$ between variables y and x_i.

Let $\lambda_n(m) := ||\mathbf{X}\boldsymbol{\beta} - P_m\mathbf{X}\boldsymbol{\beta}||^2$. In the case of deterministic $\mathbf{X}$ let

$$\lambda(m) := \lim_{n\to\infty} n^{-1}\lambda_n(m).$$

For random $\mathbf{X}$ the limit is understood almost surely. Note that $\lambda_n(m)$ equals a squared distance of $\mathbf{X}\boldsymbol{\beta}$ from its projection $P_m\mathbf{X}\boldsymbol{\beta}$ on the columns of $\mathbf{X}$ corresponding to m and may be regarded as a measure of discrepancy between the larger and the smaller model. Since $\lambda_n(m)$ is an important object we discuss its properties in Section 2.3. Proposition 1 below gives an interpretation of $\lambda(m)$ in the terms of a limiting prediction error. The following theorem shows that ordering variables with respect to squares of their t-statistics is in the case of deterministic $\mathbf{X}$ asymptotically equivalent to ordering with respect to quantities $\lambda(m \setminus \{i\})$. It also turns out that in the case of random $\mathbf{X}$ under appropriate moment conditions $\lambda(m \setminus \{i\})$ exists almost surely and the ordering can be reexpressed in the terms of squared multiple correlation coefficients $\rho^2_{y,\mathbf{x}_{m\setminus\{i\}}}$. In the following number of fitted variables m is a fixed integer. Note that as $\mathbf{X}\boldsymbol{\beta} = \mathbf{X}_t\boldsymbol{\beta}_t$, $\lambda(m)$ does not depend on the number of potential regressors p. The same observation applies to $T^2_{i,m}$. The following results have been proved in [1].

Theorem 1 *Let $i, j \in m$.*
(i) In the case of deterministic $\mathbf{X}$ assume that $\lambda(m \setminus \{i\})$ and $\lambda(m \setminus \{j\})$ exist. Then $T^2_{i,m} \geq T^2_{j,m}$ almost surely for sufficiently large n implies

$$\lambda(m \setminus \{i\}) \geq \lambda(m \setminus \{j\}). \tag{6}$$

Moreover, strict inequality in (6) implies $T^2_{i,m} > T^2_{j,m}$ almost surely for sufficiently large n.
(ii) In the case of random $\mathbf{X}$ assume that $\Sigma_{\mathbf{x}_m}$ is invertible and $\mathbf{E}x_j^4$ are finite for all $j \in m$. Then $T^2_{i,m} \geq T^2_{j,m}$ almost surely for sufficiently large n implies

$$\rho^2_{y,\mathbf{x}_{m\setminus\{j\}}} \geq \rho^2_{y,\mathbf{x}_{m\setminus\{i\}}}. \tag{7}$$

Moreover, strict inequality in (7) implies $T^2_{i,m} > T^2_{j,m}$ *almost surely for sufficiently large* n.

In the case of random $\mathbf{X}$ the explicit formula for almost sure limits in (6) can be obtained and condition (6) is simplified to (7). It is also easy to see that for m having two elements condition (7) is equivalent to $\rho^2(y, x_i) = \rho^2(y, \mathbf{x}_{m\setminus\{j\}}) > \rho^2(y, \mathbf{x}_{m\setminus\{i\}}) = \rho^2(y, x_j)$.

Proposition 1 *When* $\mathbf{X}$ *is deterministic consider the mean squared error of prediction for OLS estimation in model* m

$$MSEP_n(m) = \mathbf{E}(||\mathbf{Y}^* - \mathbf{X}_m\hat{\boldsymbol{\beta}}_m||^2) = \sigma^2(n + |m|) + ||\mathbf{X}\boldsymbol{\beta} - P_m\mathbf{X}\boldsymbol{\beta}||^2,$$

where $\mathbf{Y}^* = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}^*$ *with* $\boldsymbol{\varepsilon}^*$ *being an independent copy of* $\boldsymbol{\varepsilon}$. *Let*

$$MSEP(m) = lim_{n\to\infty} n^{-1} MSEP_n(m).$$

Thus the ordering in (6) is equivalent to ordering

$$MSEP(m \setminus \{i\}) \geq MSEP(m \setminus \{j\}).$$

Moreover, for random $\mathbf{X}$, *(7) is equivalent to*

$$\mathrm{var}(y - P_{m\setminus\{i\}}y) \geq \mathrm{var}(y - P_{m\setminus\{j\}}y).$$

Corollary 1 *Let* $m \supseteq t$.
(i) In the case of deterministic $\mathbf{X}$ *assume that* $\lambda(m\setminus\{i\})$ *is defined for any* i. *Then condition*

$$\lambda(m \setminus \{i\}) > 0, \tag{8}$$

for all $i \in t$ *implies that* $\min_{i\in t} T^2_{i,m} > \max_{i\in t^c\cap m} T^2_{i,m}$ *almost surely for sufficiently large* n.
(ii) In the case of random $\mathbf{X}$ *assume that* $\Sigma_{\mathbf{x}_m}$ *is invertible and* $\mathbf{E}x_j^4 < \infty$ *for all* $j \in m$. *Then* $\min_{i\in t} T^2_{i,m} > \max_{i\in t^c\cap m} T^2_{i,m}$ *almost surely for sufficiently large* n.

Corollary 1 asserts that when $m \supseteq t$ the relevant variables precede the spurious ones asymptotically provided that column $\mathbf{X}_i$ for any $i \in t$ is separated from the linear space spanned by other columns in m. Below we provide simple sufficient condition for (8) to hold.

Proposition 2 *For deterministic* X *and* $m \not\supseteq t$ *assume that* $n^{-1}\mathbf{X}'_{t\cup m}\mathbf{X}_{t\cup m} \to W$, *as* $n \to \infty$, W *is positive definite matrix. Then* $\lambda(m) > 0$.

The proof of Proposition 2 is relegated to Appendix. Various versions of condition (8) are used to prove asymptotic results of model selection for linear models (cf [8], [9], [10], [11]). E.g. in the last paper the condition equivalent to $\lambda(s) > 0$ for any s such that $t \not\subset s$ is used to prove consistency of Bayes selection method introduced there. Note the fact that (8) is automatically satisfied for random $\mathbf{X}$ which can be regarded as superior feature of random design when compared to fixed design modelling.

Corollary 2 *Assume that* $\Sigma_{\mathbf{x}_{m\cup t}}$ *is diagonal, invertible and* $\mathbf{E}x_j^4 < \infty$ *for all* $j \in m$ *(in the case of random* $\mathbf{X}$*) and* $\lim_{n\to\infty} n^{-1}\mathbf{X}'_{m\cup t}\mathbf{X}_{m\cup t}$ *is diagonal and invertible (in the case of deterministic* $\mathbf{X}$*). Then* $\min_{i\in t\cap m} T^2_{i,m} > \max_{i\in t^c\cap m} T^2_{i,m}$.

Corollaries 1 and 2 indicate that, when a model containing all significant variables is fitted or variables are uncorrelated, the ordering with respect to the squared t-statistics ensures that the coordinates corresponding to nonzero coefficients are placed ahead the spurious ones. In a general case when the fitted model is misspecified (i.e. at least one significant variable is omitted) and the variables are not independent it may happen that condition (6) or (7) is not satisfied for some $i \in t$, $j \notin t$ and irrelevant variable j is placed ahead relevant variable i when the ordering of variables is based on squared t-statistics. Example 1 explores such a situation.

Example 1 Consider random-design regression model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\beta} = (\beta_1, 0, \beta_3)'$, $\boldsymbol{\varepsilon}$ has $N(0, \mathbf{I})$ distribution and rows of $\mathbf{X}$ are normally distributed with covariance matrix

$$\Sigma_{\mathbf{x}} = (\sigma_{ij}) = \begin{bmatrix} 1 & b & 0 \\ b & 1 & a \\ 0 & a & 1 \end{bmatrix},$$

where $a, b \in (0, 1)$ are parameters. Thus the true variables are uncorrelated, correlation between true variable x_1 and spurious x_2 is equal b whereas correlation between true variable x_3 and spurious x_2 is equal a. A misspecified model $m = \{1, 2\}$ containing two variables only is fitted: x_1 (true) and x_2 (spurious). Theorem 1 (ii) states that $T^2_{1,m} > T^2_{2,m}$ for sufficiently large n with probability 1 i.e. the true variable will precede the spurious one in the ordering if and only if (7) is satisfied. It is easy to verify that in this case condition (7) yields

$$\sigma_{11}^{-1}(\beta_1\sigma_{11} + \beta_2\sigma_{12} + \beta_3\sigma_{13})^2 > \sigma_{22}^{-1}(\beta_1\sigma_{12} + \beta_2\sigma_{22} + \beta_3\sigma_{23})^2$$

or equivalently $\rho^2(x_1, y) > \rho^2(x_2, y)$. For $\beta_1 = \beta_3 = 1$ an easy calculation shows this is equivalent to $1 > b + a$. When the spurious variable x_2 is strongly correlated with true ones it takes over their roles in the misspecified model and in effect has more predictive power than variable x_1. For $\beta_1 = \beta_3 = 1$ we carried out $L = 500$ simulations for $n = 100, 200, 500$ and computed fraction of correct orderings for which $T^2_{1,m} > T^2_{2,m}$ with varying value of parameter a and for fixed $b = 0.5$. The results are presented in Figure 1. Note that to the left of the value $a = 0.5$ probability of correct ordering significantly increases in concordance with the condition $a + b < 1$. When the correlation a between spurious variable x_2 and true variable x_3 missing from the model is strong then the ordering of variables in m induced by t-statistics can be incorrect with high probability, i.e. it is likely that $T^2_{1,m} < T^2_{2,m}$. Note that when model $m = \{2, 3\}$ is fitted the condition for correct ordering is the same.

2.3 Properties of $\lambda_n(m)$

In this section we discuss some formal properties of term $\lambda_n(m)$ defined above in Theorem 1. Proposition 3 below gives an interpretation of $\lambda_n(m)$ in terms of the Kullback–Leibler divergence (KL). Let $\mathbf{f}_{\mathbf{X}_t\boldsymbol{\beta}_t}(s)$ be the probability density function (p.d.f.) of conditional distribution of $\mathbf{Y}$ given $\mathbf{X}$, i.e the p.d.f. of the multivariate

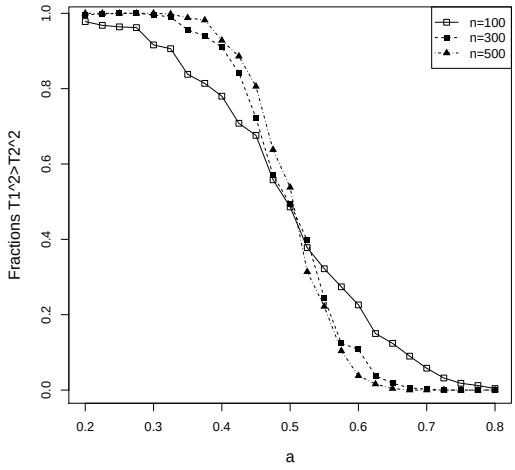


FIGURE 1: Estimated probabilities of $T^2_{1,m} > T^2_{2,m}$ with respect to a based on $N = 500$ trials.

normal distribution $N(\mathbf{X}_t\boldsymbol{\beta}_t, \sigma^2\mathbf{I})$. Let $\mathbf{f}_{\mathbf{X}_m\boldsymbol{\beta}_m}(s)$ be the p.d.f. corresponding to model m, i.e. the p.d.f. of $N(\mathbf{X}_m\boldsymbol{\beta}_m, \sigma^2\mathbf{I})$. Let $sp_{\mathbf{X}}(m)$ denote the space spanned by columns of $\mathbf{X}$ corresponding to model m. The following properties hold. They are proved in the appendix.

Proposition 3

$$KL(\mathbf{f}_{\mathbf{X}_t\boldsymbol{\beta}_t}, \mathbf{f}_{\mathbf{X}_m\boldsymbol{\beta}_m}) = \int_{-\infty}^{+\infty} \mathbf{f}_{\mathbf{X}_t\boldsymbol{\beta}_t}(s) \log \frac{\mathbf{f}_{\mathbf{X}_t\boldsymbol{\beta}_t}(s)}{\mathbf{f}_{\mathbf{X}_m\boldsymbol{\beta}_m}(s)} ds = \frac{||\mathbf{X}_t\boldsymbol{\beta}_t - \mathbf{X}_m\boldsymbol{\beta}_m||^2}{2\sigma^2} \quad (9)$$

and

$$KL(\mathbf{f}_{\mathbf{X}_t\boldsymbol{\beta}_t}, sp_{\mathbf{X}}(m)) = \inf_{\mathbf{X}_m\boldsymbol{\gamma} \in sp_{\mathbf{X}}(m)} KL(\mathbf{f}_{\mathbf{X}_t\boldsymbol{\beta}_t}, \mathbf{f}_{\mathbf{X}_m\boldsymbol{\gamma}}) = \frac{\lambda_n(m)}{2\sigma^2}. \quad (10)$$

It follows that $\lambda(m)$ is equal, up to multiplicative factor $(2\sigma^2)^{-1}$, to a limiting value of Kullback–Leibler divergence, averaged per observation, between probability density function corresponding to the true model and space spanned by columns of $\mathbf{X}$ corresponding to model m.

Proposition 4 *The following equality holds*

$$\lambda_n(m) = \boldsymbol{\beta}'_{t\setminus m}[\mathbf{X}'_{t\setminus m}\mathbf{X}_{t\setminus m} - \mathbf{X}'_{t\setminus m}\mathbf{X}_m(\mathbf{X}'_m\mathbf{X}_m)^{-1}\mathbf{X}'_m\mathbf{X}_{t\setminus m}]\boldsymbol{\beta}_{t\setminus m}.$$

It is seen that the matrix pertaining to quadratic form above is Schur complement (see e.g. [12], p. 95) of the block $\mathbf{X}'_m\mathbf{X}_m$ of the matrix $\mathbf{X}'_{t\cup m}\mathbf{X}_{t\cup m}$.

Proposition 5 *The following inequality holds*

$$\lambda_n(m) \geq \lambda_{\min}(\mathbf{X}'_{t\cup m}\mathbf{X}_{t\cup m})||\boldsymbol{\beta}_{t\setminus m}||^2.$$

It follows from Proposition 5 that $\lambda_{\min}(\mathbf{X}'_{t\cup m}\mathbf{X}_{t\cup m}) > 0$ implies that $\lambda_n(m)$ is also positive.

2.4 Modifications of the original approach

As time complexity of the calculation of final scores is linear in B it is worthwhile to consider variants of the method which would yield similar performance for smaller number of runs. Here we introduce two algorithms: Weighted RSM (WRSM) and Screened RSM (SRSM), which also can be combined together. First we will describe the WRSM procedure.

WRSM Algorithm

1. Input: observed data $(\mathbf{Y}, \mathbf{X})$, number of subset draws B, size of the subspace $|m| < \min(p, n)$.
2. For each variable i fit univariate model $y \sim x_i$ and compute weight of i-th variable $w_i^{(0)}$.
3. For each variable i compute $\pi_i = w_i^{(0)} / \sum_{l=1}^{p} w_l^{(0)}$.
4. Repeat the following procedure for $k = 1, \ldots, B = B_n$, where B_n is such that $B_n \to \infty$ when $n \to \infty$ and starting with $C_{i,0} = 0$ for any i.
   - Randomly draw a model $m^* = \{i_1^*, \ldots, i_{|m|}^*\}$ from the original feature space in such a way that probability of choosing i-th variable is equal π_i.
   - Fit model $y \sim \mathbf{x}_{m^*}$ and compute weight w_{i,m^*} for each $i \in m^*$. Set $w_{i,m^*} = 0$ if $i \notin m^*$.
   - Update the counter $C_{i,k} = C_{i,k-1} + I\{i \in m^*\}$.
5. For each variable i compute the final score FS_i^* defined as

$$FS_i^* = \frac{1}{C_{i,B}} \sum_{m^*: i \in m^*} w_{i,m^*}.$$

6. Sort the list of variables according to scores FS_i^*: $FS_{i_1}^* \geq FS_{i_2}^* \ldots \geq FS_{i_p}^*$.
7. Output: Ordered list of variables $\{i_1, \ldots, i_p\}$.

Actual point 4 of the above procedure uses a simplified scheme, namely probabilities π_i are applied sequentially, that is the probability of choosing the next variable is proportional to the probabilities amongst variables not chosen till that moment. Note that this does not match exactly the procedure given in the algorithm. Probability that the given i-th variable will be selected to a randomly drawn model m^* is

$$\begin{aligned} &P(i \in m^*) = P(i \text{ be selected in the first step}) + \\ &P(i \text{ be selected in the second step}) + \ldots + P(i \text{ be selected in the } m\text{-th step}) = \\ &\pi_i + \pi_i \sum_{j \neq i} \frac{\pi_j}{1 - \pi_j} + \ldots + \pi_i \sum_{j_1, \ldots, j_{|m^*|-1}} \frac{\pi_{j_1}}{1 - \pi_{j_1}} \cdot \frac{\pi_{j_1} \cdots \pi_{j_{|m^*|-1}}}{1 - \pi_{j_1} - \ldots \pi_{j_{|m^*|-1}}}. \end{aligned} \tag{11}$$

In the sampling literature the above probability is referred to as an inclusion probability. Observe that for large p the inclusion probability is approximately

proportional to π_i. When $|m^*|\pi_i < 1$ for all i, one can use unequal probabilities (UP) sampling techniques, e.g Systematic Sampling proposed in [13], to have $P(i \in m^*) = |m|\pi_i$. For further examples of UP sampling methods see [14]. To illustrate the issue consider a simple example. Let $p = 3$, $|m^*| = 2$, $\pi_1 = 0.4$ and $\pi_2 = \pi_3 = 0.3$. It is easy to verify that in this case inclusion probabilities calculated from (11) are equal 0.74, 0.62 and 0.62, respectively, whereas the desired values are 0.8, 0.6 and 0.6.

In the WRSM procedure variables whose individual influence on response is more significant, have larger probability of being chosen to any of the random subspaces. Since in WRSM the relevant variables are more likely to be selected, we can limit the number of repetitions B in the main loop and reduce the computational cost of the procedure.

Let $\hat{\beta}_{i,\{i\}}$ be a least squares estimator based on univariate model $y \sim x_i$ and $T_{i,\{i\}}$ be the corresponding t-statistic. In WRSM we take $w_i^{(0)} = |T_{i,\{i\}}|$.

In the following theorem we determine asymptotic final scores assigned by the above procedure. Let $\mathcal{M}_{|m|}$ be the family of all subsets $\{i_1, \ldots, i_{|m|}\}$ of $\{1, \ldots, p\}$ (models) of size $|m|$ and $|\mathcal{M}_{|m|}| = \binom{p}{|m|}$ be its cardinality. Analogously let $\mathcal{M}_{i,|m|}$ be the family of all subsets of size $|m|$ containing variable i and note that $|\mathcal{M}_{i,|m|}| = \binom{p-1}{|m|-1}$. Let P^* by a resampling measure on $\mathcal{M}_{|m|}$ determined by point 4 of the algorithm. Thus a probability of choosing model m is given by

$$P^*(m) = \sum_{S(j_1,\ldots,j_{|m|})} \pi_{j_1} \frac{\pi_{j_2}}{1-\pi_{j_1}} \cdots \frac{\pi_{j_{|m|}}}{1-\pi_{j_1}-\ldots-\pi_{j_{|m|-1}}},$$

where $S(j_1, \ldots, j_{|m|})$ is a set of all permutations of indices $\{j_1, \ldots, j_{|m|}\}$. The expected value with respect to this distribution will be denoted by $\mathbf{E}^*$. In the case of deterministic $\mathbf{X}$ let

$$t_{i,m} = \frac{\lambda_{m\setminus\{i\}} - \lambda_m}{\sigma^2 + \lambda_m} = \frac{MSEP(m \setminus \{i\}) - MSEP(m)}{MSEP(m)}.$$

and for the random $\mathbf{X}$

$$t_{i,m} = \frac{\rho^2_{y,\mathbf{x}_m} - \rho^2_{y,\mathbf{x}_{m\setminus\{i\}}}}{1-\rho^2_{y,\mathbf{x}_m}}.$$

It follows from the proof of Theorem 1 (see [1]) that under its assumptions in both cases $(n-|m|)^{-1}T^2_{i,m} \xrightarrow{a.s.} t_{i,m}$. Thus $t_{i,m}$ stands for asymptotic weight in RSM scheme. We state the result for WRSM procedure in the case when the number of predictors p is fixed and the initial weights $w_i^{(0)}$ are deterministic.

Theorem 2 *Let $(w_1^{(0)}, \ldots, w_p^{(0)})'$ be a deterministic vector. In the case of deterministic $\mathbf{X}$ assume that $\lambda(m)$ and $\lambda(m \setminus \{i\}), i \in m$, exist for all subsets of a given size $|m|$. In the case of random $\mathbf{X}$ assume that $\Sigma_{\mathbf{x}_m}$ is invertible for all subsets of a given size $|m|$ and $\mathbf{E}x_j^4 < \infty$ for all j. Then for almost any sequence $(\mathbf{Y}^{(n)}, \mathbf{X}^{(n)})_{n=1}^{\infty}$*

$$FS_i^* \xrightarrow{P^*} AFS_i := \frac{\sum_{m\in\mathcal{M}_{i,|m|}} t_{i,m}P^*(m)}{\sum_{m\in\mathcal{M}_{i,|m|}} P^*(m)}, \quad \text{as} \quad n \to \infty.$$

Thus FS_i^* is asymptotically equivalent to AFS_i, which is a weighted average of mean squared error of prediction $MSEP$, when the variable i is omitted from model m. The average is taken over all models m containing this variable. Note that the limiting value AFS_i equals conditional expectation $\mathbf{E}^*(t_{i,M}|i \in M)$, where $M = m$ is a random subset chosen by the procedure. Observe that in the original RSM we have $P^*(m) = 1/|\mathcal{M}_{|m|}|$ and then

$$AFS_i := \frac{1}{|\mathcal{M}_{i,|m|}|} \sum_{m \in \mathcal{M}_{i,|m|}} t_{i,m}.$$

Now we will discuss screening random subspace method (SRSM).

SRSM Algorithm

1. Input: observed data $(\mathbf{Y}, \mathbf{X})$, number of subset draws B, size of the subspace $|m| < \min(p, n)$.
2. For each variable i fit univariate model $y \sim x_i$ and compute weight of i-th variable $w_i^{(0)}$.
3. Let $\mathcal{M}_{screen} = \{i : w_i^{(0)} > \text{median}_{1 \le k \le p}(w_k^{(0)})\}$. RSM procedure is performed on data $(\mathbf{Y}, \mathbf{X}_{\mathcal{M}_{screen}})$.
4. Output: Ordered list of variables $\{i_1, \ldots, i_{\lfloor p/2 \rfloor}\}$.

In SRSM procedure the preliminary screening based on univariate models is performed. Variables corresponding to the smallest weights $w_i^{(0)}$ are discarded and the RSM is performed on the remaining variables. This step reduces data dimensionality. Here the choice of the median as the threshold in 3 is arbitrary, in general it may depend on preliminary knowledge of researcher. As in WRSM the number of repetitions B can be limited to reduce the computational cost. The choice of initial weights is $w_i^{(0)} = |T_{i,\{i\}}|$.

The following example shows a similar screening procedure which is based on thresholding of absolute values of t-statistics. We give the formal justification of such procedure under conditions given below. Define $\mathcal{M}_{screen} = \{i : |\hat{\beta}_{i,\{i\}}| > r_n/n\}$, where r_n is a threshold sequence such that $r_n/n \to r < \min_{i \in t} |\beta_i|$. Let $\gamma_i = \lim_{n \to \infty} n^{-1} \mathbf{X}_i' \mathbf{X}_t \boldsymbol{\beta}_t$, for $i \notin t$.

Proposition 6 *Assume that columns of* $\mathbf{X}$ *are standardised, i.e. their sample means are zero and* $n^{-1}||\mathbf{X}_i||^2 = 1$ *for all* i. *Assume also that* σ *is known,* $n^{-1}\mathbf{X}_{k_1}' \mathbf{X}_{k_2} \to 0$, *for all pairs of relevant variables* $k_1, k_2 \in t$ *and* $\log(p) = o(n)$. *Then*

$$P(\mathcal{M}_{screen} \supset t) \to 1. \tag{12}$$

If $\max_{i \notin t} |\gamma_i| < r$ *we have*

$$P(\mathcal{M}_{screen} = t) \to 1. \tag{13}$$

The proof of the above Proposition is relegated to the Appendix. Note that under assumptions of Proposition 6 ordering of variables with respect to $|T_{i,\{i\}}|$ is equivalent to ordering with respect to $|\hat{\beta}_{i,\{i\}}| = |n^{-1}\mathbf{X}_i'\mathbf{Y}|$. Convergence in (12)

indicates that with probability tending to one the true model t will be contained in the set of variables retained after the screening procedure. It follows from (13) that when dependence between spurious and relevant variables is not very strong, the true model t will be identified with probability tending to one, even when the number of all potential variables is large. In practise it is difficult to apply the above procedure since the proper choice of the threshold sequence r_n depends on an unknown parameter $\boldsymbol{\beta}_t$.

3 Model selection procedures

We briefly describe model selection procedure based on the RSM. In the following observed data $(\mathbf{Y}, \mathbf{X})$ is split into two subsets: training set $(\mathbf{Y}^t, \mathbf{X}^t)$ containing n_t observations and validation set $(\mathbf{Y}^v, \mathbf{X}^v)$ containing n_v observations. Let also $(\mathbf{Y}^{\text{test}}, \mathbf{X}^{\text{test}})$ containing n_{test} observations be a test set. The following two-stage model selection procedure is performed.

Step 1. RSM procedure is performed on set $(\mathbf{Y}^t, \mathbf{X}^t)$. The covariates $\{1, \ldots, p\}$ are ordered with respect to RSM final scores

$$FS_{i_1} \geq FS_{i_2} \geq \ldots \geq FS_{i_p}.$$

Step 2. From the nested family of models

$$\mathcal{M}_{nested} = \{\{i_1\}, \{i_1, i_2\}, \ldots, \{i_1, i_2, \ldots, i_{\min(p,n)-1}\}\}$$

we select model $m_{\text{opt}} = \{i_1, \ldots, i_{|m_{\text{opt}}|}\}$ for which the prediction error $n_v^{-1}||\mathbf{Y}^v - \mathbf{X}^v\hat{\boldsymbol{\beta}}_{m_{\text{opt}}}||^2$ is minimal. Here, $\hat{\boldsymbol{\beta}}_{m_{\text{opt}}}$ is a least squares estimator based on model m_{opt} computed on training data. The analogous model selection procedure is performed for WRSM and SRSM using in step 1 the ordering given by the respective procedure.

The score FS_i is a variable importance measure which shows the significance of the i-th variable and describes its predictive power. In the first step we obtain a ranking of variables, showing what is the contribution of each of them in explaining the response. It follows from the properties of QR decomposition that in the second step it suffices to fit only one model based on $\min(n, p) - 1$ variables sorted according to ranks of final scores. If only variable importance estimation is of interest there is no need to split data into training and validation sets– the RSM is performed based on all n observations.

As benchmarks we also consider two other methods. The first, is the lasso method proposed in [3]. For this method, the estimator is defined by

$$\hat{\boldsymbol{\beta}}_{\text{lasso}}(\alpha) = \arg\min_{\boldsymbol{\beta}} \left[||\mathbf{Y}^t - \mathbf{X}^t\boldsymbol{\beta}||^2 + \alpha||\boldsymbol{\beta}||_{l_1}\right],$$

where $||\cdot||_{l_1}$ denotes l_1 norm and α is a parameter. Because of the nature of the penalty choosing sufficiently large α will result in some of the coefficients to be exactly zero. Thus the lasso can be viewed as a variable selection method. The optimal value α (denoted by α_{opt}) is chosen by minimizing the prediction error on independent validation set, i.e. $n_v^{-1}||\mathbf{Y}^v - \mathbf{X}^v\hat{\boldsymbol{\beta}}_{\text{lasso}}(\alpha)||^2$ or by cross-validation. We

use the first option in our numerical experiments in order to make a comparison with the RSM more objective.

As the second benchmark, the univariate approach is considered. In this method informativeness and prediction strength of each feature is evaluated individually. Here, for each variable $i \in \{1, \ldots, p\}$ we compute squared value of its t-statistic $T^2_{i,\{i\}}$ based on simple regression model $y \sim x_i$. Then the covariates are ordered with respect to $T^2_{i,\{i\}}$ and the same procedure on hierarchical list of models as in the RSM is performed.

4 Numerical Experiments

In this section we study the performance of the proposed methods as prediction tools. We compare original RSM with WRSM and SRSM proposed here. We also used a hybrid method WSRSM in which in the first step screening is performed and then the WRSM is applied to remaining variables. As benchmarks we used the lasso and the univariate method. Recall that t denotes the set of coordinates which correspond to non-zero coefficients $\boldsymbol{\beta}_t$. The following linear models have been considered:

(M1) $t = (2k + 7 : k = 3, \ldots, 12)$, $\boldsymbol{\beta}_t = (1, \ldots, 1)'$,
(M2) $t = (k^2 : k = 1, \ldots, 5)$, $\boldsymbol{\beta}_t = (1, 1, 1, 1, 1)'$,
(M3) $t = (1, \ldots, 5, 11, \ldots, 15, 21, \ldots, 25)$,
$\boldsymbol{\beta}_t = (2.5, \ldots, 2.5, 1.5, \ldots, 1.5, 0.5, \ldots, 0.5)'$.

Number of potential regressors is $p = 1000$, and number of observations is $n = 200$. The rows of $\mathbf{X}$ were generated independently from the standard normal p-dimensional distribution with zero mean and the covariance matrix $\Sigma_{\mathbf{x}} = (\rho_{ij}) = \rho^{|i-j|}$. Three values of $\rho = 0, 0.5, 0.8$ were considered. The outcome is $\mathbf{Y} = \mathbf{X}_t\boldsymbol{\beta}_t + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ has zero-mean normal distribution with covariance matrix $\sigma^2\mathbf{I}$ and $\sigma^2 = 1$ (for models M1 and M2) and $\sigma^2 = 1.5$ (for model M3). Models M1 and M2 were used in [15] whereas model M3 is model 7 in [16]. Observe that for models M1 and M2 when $\rho > 0$ dependence between the relevant variables is much weaker than that between the relevant variables and the spurious ones adjacent to them. The simulation experiments were repeated $L = 500$ times. For each simulation trial, data $(\mathbf{Y}, \mathbf{X})$ is split into training set $(\mathbf{Y}^t, \mathbf{X}^t)$ and validation set $(\mathbf{Y}^v, \mathbf{X}^v)$ containing $n_v/2 = 100$ observations each and final model m_{opt} is selected as described in Section 3.

For all methods the prediction strength of the selected model is assessed by prediction error on independent test set using the average error

$$n_{\text{test}}^{-1}||\mathbf{Y}^{\text{test}} - \mathbf{X}^{\text{test}}\hat{\boldsymbol{\beta}}_{m_{\text{opt}}}||^2$$

with $\hat{\boldsymbol{\beta}}_{m_{\text{opt}}}$ being an estimator based on model m_{opt} computed on training data. For the RSM we considered $B = 5000$ choices of a random subspace consisting of $|m| = \min(n_t, p)/2 = 50$ attributes.

Figures 2, 3, 4 present prediction errors for models (M1), (M2) and (M3). It is seen that RSM works better than the lasso for model (M1) and (M2) when the

dependence is moderate ($\rho = 0.5$) or strong ($\rho = 0.8$). In the case of model (M3) lasso outperforms RSM. Using weighting in RSM improves the results for models (M1) and (M2) when the dependence is not very strong ($\rho \leq 0.5$). In the case of model (M3) WRSM outperforms RSM for all dependence structures. For model (M3), where the lasso outperforms RSM, it is in its turn outperformed by WRSM. It is interesting that screening (SRSM) does not improve the results of RSM for $M = 1000$ and WSRSM behaves comparably to WRSM. However, it should be pointed out that using WRSM and SRSM we can substantially reduce the number of repetitions B. Figure 5 presents the means of prediction errors with respect to B in the case of model (M3). In particular, figure 5 (a) indicates that the mean of prediction error for WRSM with $B = 50$ is smaller than the one for RSM with $B = 1000$.

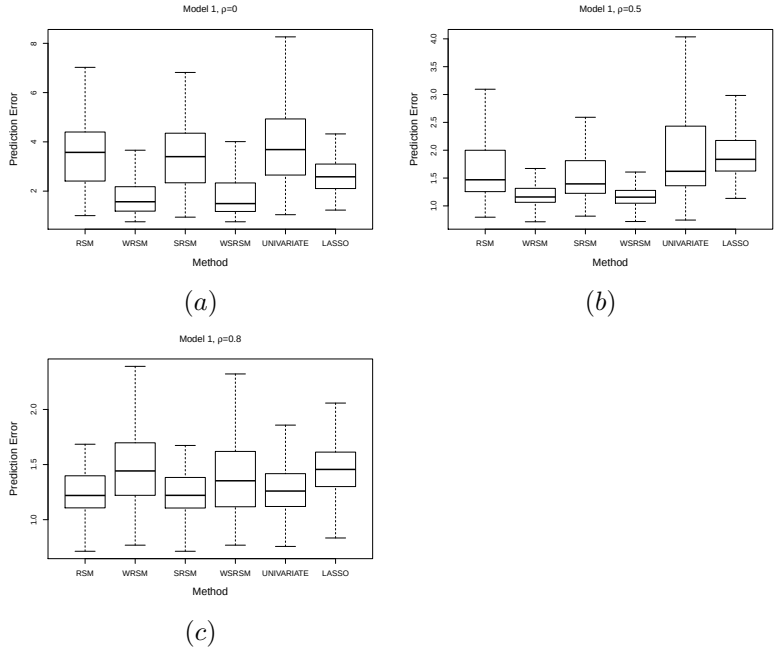


FIGURE 2: Prediction errors for model M1 with $M = 1000$ and $n = 200$ based on $L = 500$ simulation trials. Figure (a) corresponds to $\rho = 0$, figure (b) to $\rho = 0.5$ and (c) to $\rho = 0.8$.

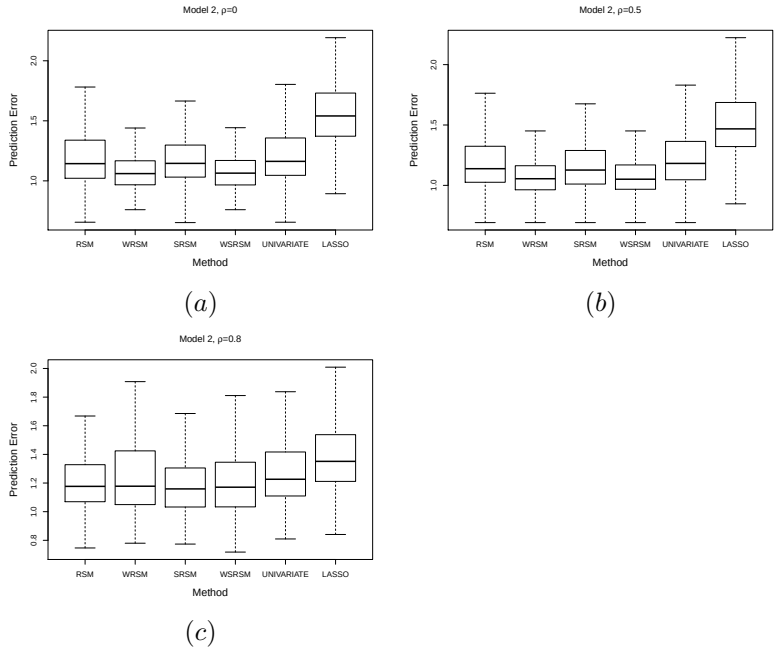


FIGURE 3: Prediction errors for model M2 with $M = 1000$ and $n = 200$ based on $L = 500$ simulation trials. Figure (a) corresponds to $\rho = 0$, figure (b) to $\rho = 0.5$ and (c) to $\rho = 0.8$.

5 Acknowledgements

Research of the second author was partially supported by Project *'Information technologies: research and their interdisciplinary applications'* POKL.04.01.01-00-051/10-00. We appreciate the comments of anonymous referee which helped to improve the original version of the manuscript.

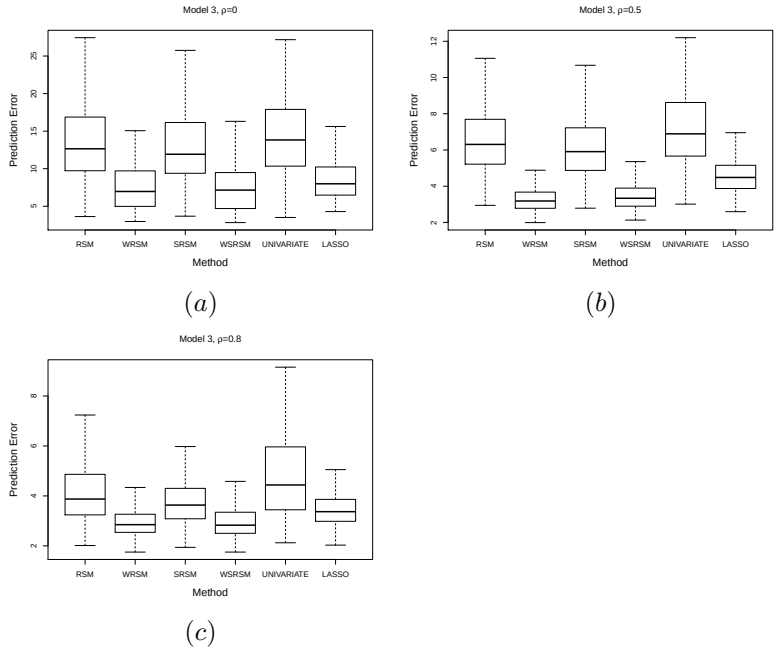


FIGURE 4: Prediction errors for model M3 with $M = 1000$ and $n = 200$ based on $L = 500$ simulation trials. Figure (a) corresponds to $\rho = 0$, figure (b) to $\rho = 0.5$ and (c) to $\rho = 0.8$.

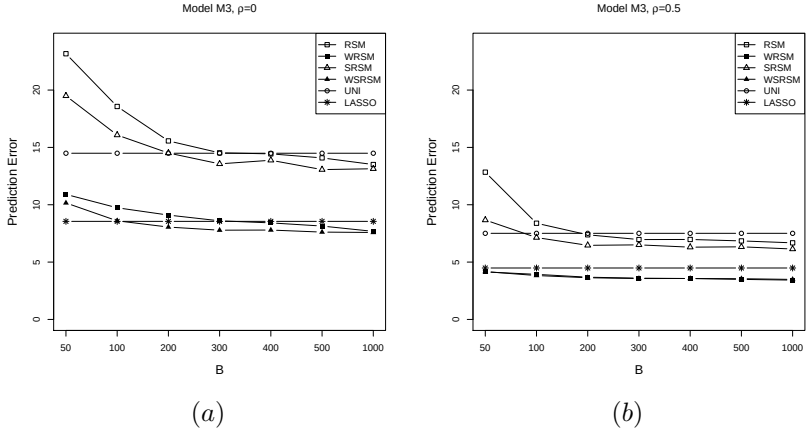


FIGURE 5: Means of prediction errors with respect to B for model M3 with $M = 1000$ and $n = 200$ based on $L = 500$ simulation trials. Figure (a) corresponds to $\rho = 0$ and figure (b) to $\rho = 0.5$.

A Proofs

A.1 Proof of Theorem 2

First note that

$$\mathbf{E}^* \frac{T^2_{i,m^*}}{n - |m|} = \sum_{m \in \mathcal{M}_{i,|m|}} \frac{T^2_{i,m}}{n - |m|} P^*(m) \tag{14}$$

and for almost any sequence $(\mathbf{Y}_n, \mathbf{X}_n)_{n=1}^{\infty}$

$$\mathrm{Var}^* \frac{T^2_{i,m^*}}{n-|m|} = \sum_{m\in\mathcal{M}_{i,|m|}} \frac{T^4_{i,m}}{(n-|m|)^2} P^*(m) - \left(\sum_{m\in\mathcal{M}_{i,|m|}} \frac{T^2_{i,m}}{n-|m|} P^*(m) \right)^2 \to$$

$$\sum_{m\in\mathcal{M}_{i,|m|}} t^2_{i,m} P^*(m) - \left(\sum_{m\in\mathcal{M}_{i,|m|}} t_{i,m} P^*(m) \right)^2 < \infty, \quad \text{as} \quad n\to\infty. \tag{15}$$

Using (14), (15) and Markov's inequality we have that

$$\frac{1}{B_n} \sum_{m^*: i\in m^*} \frac{T^2_{i,m^*}}{n-|m|} - \mathbf{E}^* \frac{T^2_{i,m^*}}{n-|m|} \xrightarrow{P^*} 0, \quad \text{as} \quad n\to\infty.$$

Thus, using the fact that $\frac{C_{i,B_n}}{B_n} \xrightarrow{P^*} \sum_{m\in\mathcal{M}_{i,|m|}} P^*(m)$ we obtain

$$TS^*_i - \frac{\sum_{m\in\mathcal{M}_{i,|m|}} \frac{T^2_{i,m}}{n-|m|} P^*(m)}{\sum_{m\in\mathcal{M}_{i,|m|}} P^*(m)} \xrightarrow{P^*} 0, \quad \text{as} \quad n\to\infty,$$

which, together with $(n-|m|)^{-1} T^2_{i,m} \to t_{i,m}$ for almost any sequence $(\mathbf{Y}^{(n)}, \mathbf{X}^{(n)})_{n=1}^{\infty}$, yields the assertion of the Theorem.

A.2 Proof of Proposition 2

Matrix W as a positive definite matrix can be decomposed as $W = W^{1/2}W^{1/2}$, where $W^{1/2} = US^{1/2}U'$, U is an orthogonal matrix and S is a diagonal matrix with positive diagonal. Let D_m be $(|t\cup m|)\times|m|$ matrix such that $\mathbf{X}_m = \mathbf{X}_{t\cup m} D_m$. We can write

$$\begin{aligned}
&n^{-1}||\mathbf{X}\boldsymbol{\beta} - P_m\mathbf{X}\boldsymbol{\beta}||^2 = \\
&n^{-1}\boldsymbol{\beta}'_t[\mathbf{X}'_{t\cup m}\mathbf{X}_{t\cup m} - \mathbf{X}'_{t\cup m}\mathbf{X}_m(\mathbf{X}'_m\mathbf{X}_m)^{-1}\mathbf{X}'_m\mathbf{X}_{t\cup m}]\boldsymbol{\beta}_t = \\
&n^{-1}\boldsymbol{\beta}'_t[\mathbf{X}'_{t\cup m}\mathbf{X}_{t\cup m} - \mathbf{X}'_{t\cup m}\mathbf{X}_{t\cup m}D_m(D'_m\mathbf{X}'_{t\cup m}\mathbf{X}_{t\cup m}D_m)^{-1}D'_m\mathbf{X}'_{t\cup m}\mathbf{X}_{t\cup m}]\boldsymbol{\beta}_t,
\end{aligned}$$

which converges to

$$\begin{aligned}
&\lambda(m) = \boldsymbol{\beta}'_t[W - WD_m(D'_mWD_m)^{-1}D'_mW]\boldsymbol{\beta}_t = \\
&(W^{1/2}\boldsymbol{\beta}_t)'[\mathbf{I} - W^{1/2}D_m[(W^{1/2}D_m)'(W^{1/2}D_m)]^{-1}D'_mW^{1/2}](W^{1/2}\boldsymbol{\beta}_t) = \\
&||(W^{1/2}\boldsymbol{\beta}_t) - H_m(W^{1/2}\boldsymbol{\beta}_t)||^2 > 0,
\end{aligned}$$

where H_m is a projection on the space spanned by columns of $W^{1/2}$. The last inequality follows from the fact that the columns of $W^{1/2}$ are linearly independent and model m does not contain at least one significant variable.

A.3 Proof of Proposition 3

Equality in (9) follows from

$$KL(\mathbf{f}_{\mathbf{X}_t\boldsymbol{\beta}_t}, \mathbf{f}_{\mathbf{X}_m\boldsymbol{\beta}_m}) = \int_{-\infty}^{+\infty} \mathbf{f}_{\mathbf{X}_t\boldsymbol{\beta}_t}(s) \log \frac{\mathbf{f}_{\mathbf{X}_t\boldsymbol{\beta}_t}(s)}{\mathbf{f}_{\mathbf{X}_m\boldsymbol{\beta}_m}(s)} ds =$$

$$\frac{2(\mathbf{X}_t\boldsymbol{\beta}_t)'(\mathbf{X}_t\boldsymbol{\beta}_t - \mathbf{X}_m\boldsymbol{\beta}_m)}{2\sigma^2} + \frac{(\mathbf{X}_m\boldsymbol{\beta}_m)'(\mathbf{X}_m\boldsymbol{\beta}_m) - (\mathbf{X}_t\boldsymbol{\beta}_t)'(\mathbf{X}_t\boldsymbol{\beta}_t)}{2\sigma^2} = \frac{||\mathbf{X}_t\boldsymbol{\beta}_t - \mathbf{X}_m\boldsymbol{\beta}_m||^2}{2\sigma^2}.$$

Equality (10) simply follows from

$$\inf_{\mathbf{X}_m\boldsymbol{\gamma}\in sp_{\mathbf{X}}(m)} KL(\mathbf{f}_{\mathbf{X}_t\boldsymbol{\beta}_t}, \mathbf{f}_{\mathbf{X}_m\boldsymbol{\gamma}}) = \inf_{\mathbf{X}_m\boldsymbol{\gamma}\in sp_{\mathbf{X}}(m)} \frac{||\mathbf{X}_t\boldsymbol{\beta}_t - \mathbf{X}_m\boldsymbol{\gamma}||^2}{2\sigma^2} = \frac{||\mathbf{X}_t\boldsymbol{\beta}_t - P_m\mathbf{X}_t\boldsymbol{\beta}_t||^2}{2\sigma^2} = \frac{\lambda_n(m)}{2\sigma^2}.$$

A.4 Proof of Proposition 4

The following equalities hold

$$\begin{aligned}
&\lambda_n(m) = ||\mathbf{X}_t\boldsymbol{\beta}_t - P_m\mathbf{X}_t\boldsymbol{\beta}_t||^2 = \\
&||\mathbf{X}_{t\setminus m}\boldsymbol{\beta}_{t\setminus m} - P_m\mathbf{X}_{t\setminus m}\boldsymbol{\beta}_{t\setminus m} + \mathbf{X}_{m\cap t}\boldsymbol{\beta}_{m\cap t} - P_m\mathbf{X}_{m\cap t}\boldsymbol{\beta}_{m\cap t}||^2 = \\
&||\mathbf{X}_{t\setminus m}\boldsymbol{\beta}_{t\setminus m} - P_m\mathbf{X}_{t\setminus m}\boldsymbol{\beta}_{t\setminus m}||^2 = \boldsymbol{\beta}'_{t\setminus m}\mathbf{X}'_{t\setminus m}[\mathbf{I} - P_m]\mathbf{X}_{t\setminus m}\boldsymbol{\beta}_{t\setminus m} = \\
&\boldsymbol{\beta}'_{t\setminus m}[\mathbf{X}'_{t\setminus m}\mathbf{X}_{t\setminus m} - \mathbf{X}'_{t\setminus m}\mathbf{X}_m(\mathbf{X}'_m\mathbf{X}_m)^{-1}\mathbf{X}'_m\mathbf{X}_{t\setminus m}]\boldsymbol{\beta}_{t\setminus m}.
\end{aligned}$$

The second equality follows from the fact that P_m is linear.

A.5 Proof of Proposition 5

The following inequality holds

$$\lambda_n(m) = ||\mathbf{X}_t\boldsymbol{\beta}_t - P_m\mathbf{X}_t\boldsymbol{\beta}_t||^2 = \inf_{\boldsymbol{\alpha}\in\mathbf{R}^{|m|}} ||\mathbf{X}_{t\setminus m}\boldsymbol{\beta}_{t\setminus m} - \mathbf{X}_m\boldsymbol{\alpha}||^2 = \inf_{\boldsymbol{\alpha}\in\mathbf{R}^{|m|}} [(\boldsymbol{\beta}_{t\setminus m}, \boldsymbol{\alpha})'\mathbf{X}'_{t\cup m}\mathbf{X}_{t\cup m}(\boldsymbol{\beta}_{t\setminus m}, \boldsymbol{\alpha})] \geq \lambda_{\min}(\mathbf{X}'_{t\cup m}\mathbf{X}_{t\cup m})||\boldsymbol{\beta}_{t\setminus m}||^2.$$

A.6 Proof of Proposition 6

In view of the assumptions and the fact that $n^{-1}\mathbf{X}'_i\boldsymbol{\varepsilon} \sim N(0, \sigma^2/n)$ we have

$$\hat{\beta}_{i,\{i\}} = n^{-1}\mathbf{X}'_i\mathbf{Y} = n^{-1}||\mathbf{X}_i||^2\beta_i + n^{-1}\mathbf{X}'_i\mathbf{X}_{t\setminus i}\boldsymbol{\beta}_{t\setminus i} + n^{-1}\mathbf{X}'_i\boldsymbol{\varepsilon} \to \beta_i, \tag{16}$$

for relevant variable $i \in t$. For spurious variable $i \notin t$ we have

$$\hat{\beta}_{i,\{i\}} = n^{-1}\mathbf{X}'_i\mathbf{Y} = n^{-1}\mathbf{X}'_i\mathbf{X}_t\boldsymbol{\beta}_t + n^{-1}\mathbf{X}'_i\boldsymbol{\varepsilon} \to \gamma_i.$$

The convergence in (12) follows from (16). In order to show (13) we have to prove that $P(\max_{i\notin t} |\hat{\beta}_{i,\{i\}}| > r_n/n) \to 0$. The following inequalities hold

$$\begin{aligned}
&P(\max_{i\notin t} |\hat{\beta}_{i,\{i\}}| > r_n/n) = P(\max_{i\notin t} |\mathbf{X}'_i\mathbf{Y}| > r_n) \leq (p - |t|) \max_{i\notin t} P(|\mathbf{X}'_i\mathbf{Y}| > r_n) \leq \\
&(p - |t|)[\max_{i\notin t} P(\mathbf{X}'_i\mathbf{Y} > r_n) + \max_{i\notin t} P(\mathbf{X}'_i\mathbf{Y} < -r_n)].
\end{aligned} \tag{17}$$

Now, using the assumption $\max_{i\notin t} |\gamma_i| < r$ and Mill's inequality (see in [17]), the first probability in (17) can be bounded from above by

$$p \max_{i\notin t} P(\mathbf{X}'_i\mathbf{X}_t\boldsymbol{\beta}_t + \mathbf{X}'_i\boldsymbol{\varepsilon} > r_n) =$$

$$p \max_{i \notin t} P(n^{-1/2}\mathbf{X}_i'\boldsymbol{\varepsilon} > n^{1/2}(r_n/n - \mathbf{X}_i'\mathbf{X}_t\boldsymbol{\beta}_t/n)) \leq$$
$$p \max_{i \notin t} \frac{1}{n^{1/2}(r_n/n - \mathbf{X}_i'\mathbf{X}_t\boldsymbol{\beta}_t/n)} \exp(-n(r_n/n - \mathbf{X}_i'\mathbf{X}_t\boldsymbol{\beta}_t/n)^2/2) \to 0,$$

under assumption $\log(p)/n \to 0$. The second probability in (17) is treated analogously.

References

[1] Mielniczuk, J., Teisseyre, P.: Using random subspace method for prediction and variable importance assesment in regression. Computational Statistics and Data Analysis (2012) in press, 10.1016/j.csda.2012.09.018.

[2] Donoho, D.L.: High-dimensional data analysis: the curses and blessings of dimensionality. Aide-Memoire of a Lecture at AMS Conference on Math Challenges of the 21st Century (2000)

[3] Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society B **58** (1996) 267–288

[4] Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society B **67**(2) (2005) 301–320

[5] Fan, J., Lv, J.: Sure independence screening for ultra-high dimensional feature space (with discussion). Journal of the Royal Statistical Society B **70** (2008) 849–911

[6] Chen, J., Chen, Z.: Extended Bayesian Information Criteria for model selection with large model spaces. Biometrika **95** (2008) 759–771

[7] Frommlet, F., Ruhaltinger, F., Twaróg, B., Bogdan, M.: Modified versions of Bayesian information criterion for genome-wide association studies. Computational Statistics and Data Analysis **56** (2012) 1038–1051

[8] Zhang, P.: On the distributional properties of model selection criteria. Journal of the American Statistical Association **87** (1992) 732–737

[9] Shao, J.: Linear model selection by cross-validation. Journal of the American Statistical Association **88** (1993) 486–494

[10] Zheng, X., Loh, W.Y.: Consistent variable selection in linear models. Journal of the American Statistical Association **90**(429) (1995) 151–156

[11] Casella, G., Giron, J., Martinez, M., Moreno, E.: Consistency of Bayesian procedures for variable selection. Annals of Statistics **37** (2009) 1207–1228

[12] Gentle, J.E.: Matrix Algebra: Theory, Computations, and Applications in Statistics. Springer, New York (2007)

[13] Madow, W.G.: On the theory of systematic sampling, II. Annals of Mathematical Statistics **20** (1949) 334–354

[14] Tillé, Y.: Sampling Algorithms. Springer, New York (2006)

[15] Zheng, X., Loh, W.Y.: A consistent variable selection criterion for linear models with high-dimensional covariates. Statistica Sinica **7** (1997) 311–325

[16] Huang, J., Ma, S., Zhang, C.H.: Adaptive lasso for high-dimensional regression models. Statistica Sinica **18** (2008) 1603–1618

[17] Birnbaum, Z.W.: An inequality for Mill's ratio. Annals of Mathematical Statistics **13**(2) (1942) 245–246

Issues of Polish Question Answering

Piotr Przybyła

Institute of Computer Science, Polish Academy of Sciences,
ul. Jana Kazimierza 5, 01-248 Warszawa, Poland,
P.Przybyla@phd.ipipan.waw.pl

Abstract

In this paper, perspectives for building a Polish question answering (QA) system are considered. It begins with an explanation of an increasing demand for such systems, along with inevitable difficulties and usage examples. Then, a question answering task is defined as a point of reference and for future evaluation. Existing solutions of similar tasks for English, evaluated during text processing conferences, are presented along with their common architectural scheme. To assess the applicability of those solutions to Polish, their language dependence is discussed, preceded by an outline of distinctive features of Slavonic languages, especially Polish, and their impact on text processing. Some already existing QA solutions for Polish are also enumerated.

Keywords: question answering, natural language processing, Polish, information extraction

1 Introduction

Question Answering (QA) is a challenge present in computer science since its beginnings. It is a natural human desire: to have a purely automatic tool that will help in solving problems with information just like mechanical tools solve problems with matter. The invention of first computers sparked off numerous visions of *talking machines*, playing important roles in popular culture. It seemed to be the most obvious direction of development for *electronic brains*. From that point of view, the following definition could be used:

Definition 1 ***Question Answering System*** *is a computer system, capable of addressing questions formulated by a human user in his natural language.*

Why this feature of computer systems is so important for users and so neglected by their designers? To understand it, one needs to realise that computers think in their own way, and the burden of translating the problem to a computer-understandable way usually lies on user's shoulders. The poorer are the computer skills of the user, the harder it becomes.

For example consider the task of searching the WWW in order to find a specific piece of information. This service is a basic tool for Internet users, including the least experienced ones. However, designing a successful query involves obeying certain rules. Some of them are counterintuitive (e.g. choosing words which will

not appear in web pages that will *not* contain desired information), some involve additional search tools. Unfortunately, most users are not aware of them, which leads to the following results:

- every day, 3 million queries to the *Yandex* (a Russian search engine) are plain questions [1] (*How to lose weight?* is the most popular),
- 80% of the users do not use boolean operators at all [2],
- other advanced features of search engines (wildcards, etc.) are also seldom used.

This type of problems are encountered whenever knowledge stored in a computer system needs to be accessed by a person, who is not qualified enough (although he may be an expert in his own field), e.g. in medical systems, customer services or libraries.

In fact, almost all applications could benefit from adding question answering elements. Even regarding those with low level of interaction with user, such as weather modelling or engineering calculations, an underlying problem could be formulated as simple questions in a natural language, like *Will it rain tomorrow?* and *Is this bridge able to hold the expected load?*, respectively. More about such modules could be found in [3].

1.1 Difficulties

Now, let us focus on another side of the problem: why so desirable feature is so rarely implemented in commercial products? That is because its realisation is very hard; Shapiro describes it as AI-complete. It means that *solving the problem of the area is equivalent to solving the entire AI problem - producing a generally intelligent computer program* [4]. Features that make the natural language processing (especially question answering) so challenging are the following:

- **contextuality**: The meaning of every language element (a word, a sentence) depends highly on its neighbourhood. It makes Information Retrieval (IR) and Information Extraction (IE) hard, as information obtained by a user may be different than expected because of its context. Usually, in those cases, systems provide the user with additional information about the context, such as text snippets in web search engines. The most problematic contextual elements i.e. ellipses[1] and anaphoric expressions[2] may be dealt with in QA systems by specialized modules, called anaphora resolution solutions [5].
- **ambiguity**: Sometimes even while knowing a context of a particular sentence, we still have a number of possible ways to understand it. It could be unintended, but also may be a deliberate act of an author in order to influence a recipient, e.g. by suggesting instead of stating explicitly. Usually, it is assumed that such situations should not appear in well-formed, informative texts.

[1] Ellipsis is an omission of a word (or more) in a text, which could be easily determined from a context.

[2] Anaphora is an expression with a certain word (or a whole group of words) replaced by another word, usually a pronoun.

- **imprecision**: It seems obvious that human utterances lack the precision of, say, mathematical formulae or numerical computations. However, it is not a problem as long as questions which are to be answered using this knowledge are of similar precision. Again, choosing a textual information source adequate to our needs is crucial.
- **implicitness**: In communication between humans, a message being transferred is not stated explicitly, but rather derived from all the knowledge common for its sender and receiver, which words are only a part of. Other relevant knowledge sources include gesticulation, facial expression and tone of voice. Unfortunately, even avoiding the above by working with digital texts, we still have to take into account another hidden source of knowledge: a common sense. For example, let us consider the following pair of sentences: *Elephants ate all the fruits yesterday. They seem very content with it.* To understand it, we need to resolve the anaphora *they* in the second sentence. Unfortunately, it is not possible on syntactic level; *elephants* and *fruits* are both acceptable candidates in this situation. Obviously, any human reader knows that he should choose the *elephants*, as the *fruits* are unable to have any feelings, including the contentness. However, this information is not provided by the context; it's also highly unlikely to appear in any conventional textual source, such as a newspaper archive or an encyclopaedia. It is expected to be a part of world understanding, usually called commonsense knowledge, including which in a computer system poses a great challenge.

The problems of contextuality and implicitness in a QA task are easier to deal with when using appropriate linguistic tools; some of them are discussed in section 4.4. A careful choice of source texts for QA helps to overcome difficulties with ambiguity and imprecision, see section 2.2.

1.2 Paper aim and organization

In section 2 a clear, language-independent problem definition is presented to focus on core problems, namely the natural language processing (NLP). Next, in section 3, solutions and approaches present in the literature are enumerated including a common architectural scheme with detailed explanation of its elements. Section 4 is devoted to discussing the relevancy of previous statements to the Slavonic languages, especially Polish. In particular, a few existing QA solutions are presented and the architectural scheme from section 3.3 is analysed to identify language-dependent elements and discuss their availability in Polish. Section 5 concludes the paper.

2 Problem definition

In the previous section, the motivation for building QA systems was outlined — a huge number of computer systems could improve their users' satisfaction by implementing natural language communication. The purpose of this section is to define a QA problem in order to focus our attention on its most important features and set aside problems related with a particular use. Generally, the task which is to

be defined here is called an **open domain factoid question answering using text corpus**. Let us state the following:

Definition 2 ***Question Answering*** *is a problem of automatic extraction of a concise answer in a* ***natural language****, in response to a question formulated in the* ***natural language****, based on knowledge gathered from collection of texts in the* ***natural language****.*

It is a much narrower formulation than in the Definition 1; in this case not only the questions, but also the answers and the sources are expressed in a natural language. However, as it includes the core language processing problem, its solutions could also be used in implementations of similar tasks, say, interface of a database.

2.1 Questions

Let us define a Question in our QA system as a sentence, being a question according to natural language rules, expected answer to which is a simple entity, i.e.:

- An answer to the question is present in the text collection, but it may be indirect, or preparing it may involve combining knowledge from several documents.
- Questions should not require any complex reasoning or computing, so the focus remains on text processing. For example the following question: *Which is the largest country of those that spend more than 10% of their budget on their army?* would not be accepted, unless there is a corresponding sentence in a source text.
- A target of question is a simple entity, regardless whether named (person, city, country, date, quantity, etc.) or unnamed (thing, concept, property, animal, event, action etc.). Such question are usually called **factoid questions**. Queries demanding a definition or an explanation, such as *What is a global warming?* do not fall into this category.
- The queries are grammatically correct questions in the natural language, which rules out requests like *Tell me the name of the first post-war president of France.*

A test set for a QA task is just a set of questions as defined above; no additional information, e.g. about context or domain, is provided, thus the questions need to be self-explaining and precise.

2.2 Texts

As stated in section 1.1, a level of difficulty of a QA task depends highly on texts to be processed. An ideal textual database for evaluating such systems would have the following properties:

- **linguistic correctness**: Usually texts that people describe as well-written and correct are also relatively easy for automatic processing.
- **precision**: To extract valuable information from the texts, we need them to be written precisely, without ambiguities or vague statements.

- **informativeness**: To be able to answer interesting, non-trivial and worth asking questions, we need the corresponding source to contain appropriate answers. That rules out all sort of literary work, albeit satisfying the remaining conditions.
- **abundance**: Modern NLP challenges are mostly connected with a need to deal with huge numbers of texts — to evaluate a QA system we want it to process a significant knowledge base, containing an overwhelming amount of information, mostly irrelevant to any query.
- **open-domain**: Closed-domain systems are being actively developed, but employ different techniques (such as manual creation of possible questions list or a knowledge base). Herein, only open-domain systems are considered, which needs to be reflected in the knowledge source.

Sources which are used for similar purposes in literature are newswire bases or encyclopaedias. They come from relatively reliable sources and satisfy all the conditions outlined above. On the other hand, they are very far from what may be found in the most important "battleground" of the modern text processing: the Internet. The content of web pages is usually imprecise, ambiguous, more persuasive or emotional than informative, often incorrect in terms of syntax or spelling. The *Wikipedia* may be some kind of a compromise in that matter: offers highly-informative articles, but written by internet users instead of professional journalists or scientists.

2.3 Answers

What exactly is an answer to our question? For example, let us consider the question *When did Albert Einstein marry his second wife?*. In fact, the four answer levels are possible:[3]

1. A document containing an answer: *Wikipedia:Albert_Einstein*. However, such an answer is rather typical for Information Retrieval systems, as it still requires a user to extract the entity from the text.
2. An answering sentence: *Einstein married Elsa Löwenthal (née Einstein) on 2 June 1919, after having had a relationship with her since 1912*. Here, there is less work for a user, but he still needs to find the dates in the sentence and decide, which corresponds to his question.
3. Only an entity demanded: *2 June 1919*. That seems to be the option most convenient for a user.
4. A full phrase based on a question: *Albert Einstein married his second wife on 2 June 1919*. Although that would be a natural way of responding, it is seldom implemented in existing open-domain QA systems.

Henceforth, in this paper by *answer* the entity demanded, as in the point 3, will be meant.

[3]Answers extracted from: `http://en.wikipedia.org/wiki/Albert_Einstein`.

2.4 Evaluation

Generally, an evaluation of a QA system is a process of measuring its outcome in a precisely defined environment. The environment consists of questions, sources and answers satisfying certain conditions, for example those outlined in sections 2.1, 2.2 and 2.3 of this document. What is more, one needs to define measures to be applied to gathered answers. Aside from obvious computing a percentage of questions answered correctly, worth mentioning are number of questions a tested system decided to answer (crucial in *Jeopardy!* competition, in which QA system named *Watson* took part [6]) and adequacy of a document returned as a support of an answer (e.g. TREC competition [7]).

3 Related work

The field of question answering attracts a lot of attention, resulting in many interesting solutions. However, while their authors use different testing environments or different correctness measures, they remain incomparable. One of the ways to overcome this problem are open competitions organised together with corresponding conferences.

3.1 Open competitions

The most important competition was *Question Answering track* at *Text REtrieval Conference (TREC)*, which was arranged in years 1999-2007 [7]. In 2007, a total of 515 questions was organised in 70 series, each related to a certain target (e.g. *Guiness Brewery*, *Jon Bon Jovi* or *March Madness 2011*). The following question types were present[4]:

- **FACTOID**: an expected result type is a single entity, being an answer to the question, e.g. *On what date did this earthquake strike?*,
- **LIST**: an expected result type is a list of entities satisfying the question, e.g. *What countries were affected by this earthquake?*,
- **OTHER**: an expected answer type is an interesting information *nugget* about the target, different from those present in the remaining questions from the series.

In 2008, the competition transformed into *Opinion Question Answering* at *First Text Analysis Conference (TAC 2008)*, where the task was to find opinions in a blog corpus [8]. The *Question Answering for Machine Reading Evaluation* has been offered at *Conference and Labs of the Evaluation Forum*. In this case answers were given (5 options for each of 160 questions), so the focus was on a deep text understanding, namely not only finding entities in text, but also reasoning and inferring based on extracted knowledge [9]. Tasks (i.e. source texts, questions and answers) were given in English, German, Italian, Romanian, Spanish, Arabic and Bulgarian. Another multilingual task was the *Cross-Lingual Question Answering (CLQA) Task* at *NTCIR (NII Test Collection for IR Systems) Workshop*. An aim

[4]Examples from the target *Pakistan earthquakes of October 2005*, published in [7].

was to answer questions with named entities extracted from corpora in Chinese, Japanese and English [10].

Another idea worth mentioning has been proposed by a team working on *Watson* [6] at IBM. In [11] they proposed to build so-called *Open Collaboration Framework* not only in order to compare results, but also to invigorate collaboration between academic and industrial organisations by using shared system model and open-source components and exchanging elements of QA solutions.

3.2 Solutions

The first solutions for problems formulated in a way similar to section 2 were proposed in the 60's (a detailed review may be found in [12]). However, they relied mostly on transforming a text and questions into some kind of a formal representation and then employing rapidly developing theorem-proving techniques to find an answer. Unfortunately, this approach did not lead to satisfying results not only because of the ambiguity and imprecision of English, but also because of a lack of reliable linguistic tools necessary for such a transformation. In this situation, researchers needed to base on few manually-prepared rules covering only a small part of the language. The beginnings of the Internet encouraged researchers to use similar approach again, this time to extract information from the WWW [13].

A large number of modern QA systems took part in the TREC QA competition in years 1999-2007 (overviews published in [14, 15, 16, 17, 18, 19, 20, 21, 7]). Most of participating solutions had a similar architecture, outlined in the next section, and used available linguistic tools and resources heavily.

The last breakthrough worth mentioning was made by a team from the IBM Research Division, who managed to build *Watson*, a complex QA system, which participated live in the U.S. TV quiz show called *Jeopardy!* [6]. The task differs from what is described above in three ways: first, queries are not formulated as pure questions, but *clues*, to which a participant is expected to respond, asking an appropriate question. Secondly, usually it is not enough to have pure factoid knowledge; one needs to combine distant relations, solve puzzles, understand word games. Finally, during competition, each of the participants needs to decide whether he wants to respond to a clue after seeing it; therefore, a precise confidence estimation becomes crucial.

3.3 Common architecture

As stated above, a majority of modern QA solutions share a common architectural scheme, which has been illustrated in Figure 1.

As we can see in the figure, a whole process starts with a pre-processing of a text base, which is to be executed offline, before the system is presented to users. When a question is available, it undergoes a similar processing stage, which also determines its type. The information is used in subsequent steps: searching for relevant documents using an index, choosing a sentence which matches the question best, and, finally, extracting a demanded entity from the sentence.

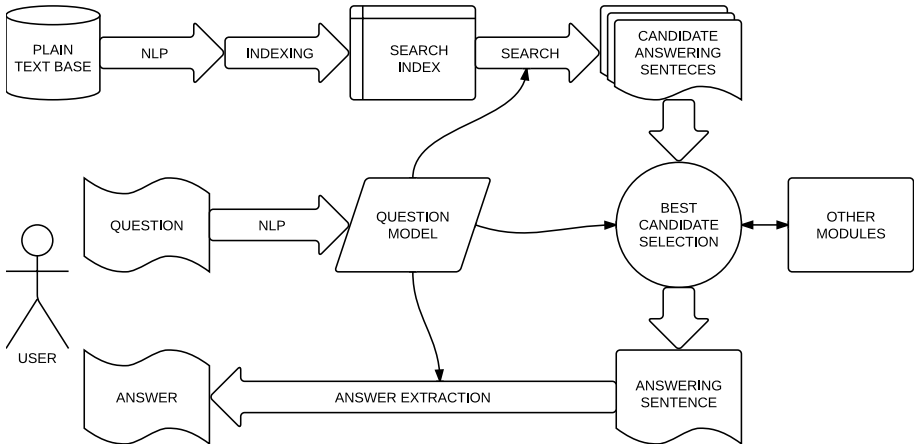


FIGURE 1: Outline of a typical QA system.

3.3.1 Text base processing

Before an evaluation begins, the text corpus is processed to facilitate its usage as a knowledge base for question answering. For English sources, usually a *stemming*[5] is applied, but in case of languages with the nominal inflection it may get more complicated (see section 4.1). Systems employing a deep semantic parsing may also execute it as a stage of the text preprocessing (e.g. [22]).

3.3.2 Index

An index is introduced because of performance reasons. Regardless of a sentence selection method, it is not possible to apply it to all of the documents from the corpus. Therefore, in the preparatory stage, the full-text search index is created, using one of powerful and freely available search engines, such as *Apache Lucene*. When the question is processed, it is transformed into a list of keywords, forming a search query. This transformation process usually includes a removal of stop-words[6] and using external resources for query expansion by adding synonyms [23] or more complex semantic transformations [22]. A certain number of top results is then passed on to the subsequent steps. Some of the systems, such as [24], include feedback loops, returning to this stage in case of extracted documents being irrelevant.

3.3.3 Question processing

There are three goals of this stage:

[5] *Stemming* is a process of assigning a pseudo-word (called a *stem*) to words in a text in a way that will guarantee (or, more likely, make very probable) that different forms of the same word will get equal stems.

[6] The term *stop-words* refers to words, which are very frequent in all documents in a particular language, such as (for English) *a*, *the* or *is*.

1. question type determination,
2. answer type determination,
3. transformation to a search query.

The first task is to find out what is a general type of the question: factoid, list, definition, yes-no or other. In the TREC QA task this information was being given explicitly [7], but it is not the case in real-world applications. The second stage is an interesting problem itself: how to determine what is the type of an entity, which is sought by the questioner? The interrogative pronouns (e.g. *who*, *where*, *when*) may be of some use (*who* usually refers to a person), but they are not enough (*who* question may also be answered by an entity being an organisation). Usually solutions include a number of possible target categories (e.g. PERSON, DATE, TITLE, QUANTITY, CITY, ...) or even complex hierarchies, like in [25]. Apart from rules created manually some systems use the WordNet[7] [24, 27] or machine learning techniques [25, 28] to find the answer type. Requiring the user to fill in a form instead of writing the question, like in [29], renders question processing unnecessary.

3.3.4 Sentence selection

The core element of every QA system is responsible for choosing a sentence which is most likely to contain the answer. Four main approaches to the problem may be identified:

1. Pattern matching,
2. Semantic matching,
3. Probabilistic matching,
4. Conversion into logic forms.

Pattern matching is an extremely simple and surprisingly effective technique for question answering which has been used extensively. For example, let us consider the following question: *When did Frederick I Barbarossa die?*. It is quite probable that the answering sentence will take the form: *Frederick I Barbarossa died on <DATE>*. We can then scan the sources very efficiently, just looking for the specified pattern. Of course, if this information is expressed differently, then the approach fails. However, if we use the WWW as a corpus, then we may exploit its redundancy; the same information appearing in different web pages, expressed in different ways, increases the probability of finding our pattern. Apart from creating those patterns by hand, we may also automatically acquire them from the Web [30, 31]. This approach has also been successively employed when focusing on a particular question type, namely definition questions [32].

The use of the pattern matching is always limited by a variety of the natural languages; that is why semantic matching is being employed nowadays. In this approach we still match the question to a candidate sentence, but the connection between (potentially) corresponding words is no longer a pure equality, but can be

[7] *Princeton WordNet* [26] is a lexical database containing words organized in synonymous sets and connected by semantic relations, such as hypernymy, hyponymy or meronymy (part-whole relation).

derived from a lexical database like the WordNet. For example, while answering the question mentioned previously (*When did Frederick I Barbarossa die?*) we should also accept *On <DATE> Frederick I Barbarossa passed away* as an answer because *to pass away* and *to die* are synonyms, but also *Frederick I Barbarossa drowned in the river on <DATE>* because *to drown* is a hyponym of *to die*. We could also take into account more complex relations to find sentences like *The death of Frederick I Barbarossa on <DATE> led to ...* by recognizing the connection between *to die* and *death*. Harabagiu et al. proposed and evaluated a semantic matching solution [24] based on the WordNet, while Moldovan and Novischi proposed an algorithm for computing the strength of a relation between different concepts in [33]. An interesting solution is presented in [22], where a whole sentence is transformed into a semantic network, called *MultiNet*, and then matched to a question. Besides the WordNet matching, other important semantic connections of the question and the sentence are outlined in [34], such as similar noun phrases and co-occurring named entities.

Probabilistic matching is based on gathering a set of *features*, characterizing a similarity between the question and a candidate sentence, but also including other evidence, such as presence of named entities or information from WordNet. To assess importance of particular features and to select the sentence which is most likely to contain the answer, tools from the statistics are used, such as the logistic regression [35] or the entropy [36].

The last approach (conversion into logic forms) is less popular, but also has its advantages. The source text is converted into a formal language (say, a set of predicates) by a specialised parser, and knowledge obtained in that way may be used to find an answer in a process similar to theorem proving. Of course, once we have valid knowledge represented as a set of predicates, it is guaranteed that the answers will be valid, too. The only problem is that converting an ambiguous, imprecise, contextual and implicit natural language statements into strict logics poses a great challenge (if is possible at all). Apart from the already mentioned solution by Katz [13], an interesting system has been presented in [37]. To cope with those unpleasant features of a natural language, they propose a set of sources of world knowledge (common sense), such as lexical chains between concepts in WordNet, mentioned previously.

3.3.5 Answer extraction

If it is enough to give a sentence as an answer, the result of the selection outlined above (i.e. the best sentence) is presented to the user. However, if we want to show only an entity demanded, as proposed in section 2.3, then an appropriate extraction is necessary. Usually it is easy to select words corresponding to an entity of an expected type (e.g. a date). If there are more than one, we need to know from the previous stage, which of the entities has been matched as the answer. Providing an answer in a way proposed in section 2.3 p. 4 would involve the Natural Language Generation (NLG) capabilities.

3.3.6 Other modules

Apart from the stages described above, a question answering system may include a number of additional tools, improving its performance. Usually they are employed in the sentence selection stage.

- **WordNet** lexical database, used in a variety of roles, described above.
- **Named Entity Recognition (NER)** solutions are used to scan the text efficiently and find fragments which correspond to named entities[8]. They are useful in QA to reduce a number of sentences taken into account by processing only those containing named entities of type compatible with the question target and to extract the desired entity from the sentence.
- **Anaphora resolution** - anaphora, as described previously, is a word (usually a pronoun) substituting an expression which has already been mentioned. The process of resolution, i.e. assigning a set of possible targets to each anaphoric expression is very helpful in question answering. For example, let us consider answering the question *When did Albert Einstein publish special relativity theory?* The Wikipedia article contains the following:

  *On 30 April **1905**, Einstein completed his thesis, with Alfred Kleiner, Professor of Experimental Physics, serving as pro-forma advisor. Einstein was awarded a PhD by the University of Zurich. His dissertation was entitled "A New Determination of Molecular Dimensions". **That same year**, which has been called Einstein's annus mirabilis (miracle year), he published four groundbreaking papers, on the (...), special relativity, and (...), which were to bring him to the notice of the academic world.*[9]

  As we can see, the question target (year 1905) is three sentences away from the mention of the special relativity publication. The only link between them is the anaphoric expression *That same year*. Moreover, it does not refer to the whole named entity *30 April 1905*, but only to the year, which has to be taken into account during resolution (special relativity was published in 1905, but not on April 30). An example of successful incorporation of an anaphora resolution module into a QA system is presented in [5].

4 Prospects for Polish

One of the goals of this paper is to outline the prospects for building a Polish question answering system. Unfortunately, all of the solutions described above were prepared for English or German, which differ substantially from Polish in their structure. The purpose of this section is to analyse which of the elements of the question answering environment are directly applicable to Polish, which require some modifications, and which are hardly usable.

[8]The meaning of a named entity term is somewhat vague. Usually it is supposed to include entities referred by proper names (persons, organisations, countries, cities, mountains, lakes, etc.), titles of works (books, pieces of music, paintings, journals etc.), temporal (time, date, year, century) and numerical expressions (count, quantity, percentage, money).

[9]Extracted from the article in Wikipedia: `http://en.wikipedia.org/wiki/Albert_Einstein`.

4.1 NLP of Slavonic languages

Polish is a Slavonic language and shares most of the features common for the group. The most important ones from the point of view of NLP are: the rich nominal inflection, the free word order and the complicated inflection of proper names (both Polish and foreign). Herein, they are briefly explained only; a detailed review may be found in [38].

The nominal inflection makes text processing substantially harder, as every noun or adjective may take a number of forms[10] depending on its role in a sentence. On the other hand, it allows words to appear in different order without a complete change of meaning, as information about their roles in the sentence (e.g. distinction between a subject and an object of a verb) is preserved in the nominal inflection. For example, let us consider the following sentence: *John ate a fish.* Its obvious translation to Polish would be: *Jan zjadł rybę.* However, we could also say *Rybę zjadł Jan* with the same meaning but a slightly different focus. The remaining options (*Zjadł rybę Jan*, *Zjadł Jan rybę*, *Jan rybę zjadł*, *Rybę Jan zjadł*) would seem awkward, but remain perfectly understandable. This is not the case in English: *A fish ate John* has a totally different meaning. Of course, for longer sentences only some of the orderings preserve its comprehensibility; that is why this phenomenon is called the **relatively** free word order.

The nominal inflection affects also proper names, altering their appearance. An inflected form depend on numerous factors, such as a source language, a name pronunciation and a language tradition. The nominal inflection of foreign proper names confuse even native speakers of Polish; it is a cause of very common errors. Polish proper nouns inflect differently than their common equivalents too, for details see [38]. That is why the NER task is much harder in the Slavonic languages than in English.

4.2 Existing solutions for Polish

Only a few attempts to build a QA system have been made for Polish. First one, from 1985, was a Polish interface [39] to the ORBIS database, containing information about the solar system planets encoded as PROLOG rules. A POLINT system, based on it, is still being developed in the Adam Mickiewicz University in Poznań [40]. In 2002 another system capable of handling questions in Polish operating in a restricted domain (business information) was presented by Duclaye et al. [41]. Some elements (a reasoning module responsible for spatial relations) of a Polish QA system under development (*Hipisek.pl*) have been published by Walas [42].

A solution closest to what is analysed in this paper is that proposed by Piechociński and Mykowiecka [43]. It answers questions in Polish (translated from the TREC QA task) by scanning the content of the Polish Wikipedia. Unfortunately, it relies heavily on its structure, so couldn't operate on an arbitrary text corpus.

[10] In Polish there exist 7 cases, but in each declension some of them correspond to identical forms.

To sum up, several QA systems for Polish exist, but their usage is limited either by a reliance on a specific knowledge base [43] or by a closed domain (the remaining ones). None of them would be able to participate in a TREC competition in Polish, i.e. answer natural language questions using a given corpus of texts.

4.3 Language-dependence of elements

To discuss the possibility of building a Polish QA system, solving TREC-like tasks, the elements of the common architecture outlined in section 3.3 are analysed here to show their dependence on a processed language.

4.3.1 Text base processing

The main objective of this stage is to analyse words in order to replace them by a stem or a base form, common for all inflected forms of a particular word. In English it is done by stemming, i.e. transforming each segment according to a set of predefined rules. Unfortunately, because of the rich nominal inflection, the mutability of Slavonic words is much higher. Although stemmers for Polish exist as well [44], their performance is not very good. It is more recommendable to use a full morphological analyser, which processes a text slower, but guarantees to find every possible base form for each recognized segment.

4.3.2 Index

Index could work equally well with English and the Slavonic languages, if implemented properly. For example, the *Lucene* works perfectly with Polish corpora.

4.3.3 Question processing

The question processing module clearly needs to be prepared for a particular language, as question syntaxes may differ a lot between languages.

4.3.4 Sentence selection

Sentence selection method deserves special attention. Some of the approaches may depend on a processed language, but not necessarily all of them.

Let us consider sentence selection by pattern matching. It may be successfully applied to some languages, but Slavonic unfortunately will lead to much lower accuracy. That is because of the relatively free word order, which makes a pattern match only a most common ordering and ignore remaining ones. For example, let us try to answer the question *Who ate a fish?*. It would be converted to the pattern {<PERSON> <eat>[11] <fish>} and result in a match after finding the sentence *John ate a fish*. For Polish, the same question *Kto zjadł rybę?* correspond to the pattern {<PERSON> <jeść> <ryba>}, which matches the sentence *Jan zjadł rybę*, but fails in case of *Rybę zjadł Jan*, which carries the same meaning. This feature of the Slavonic languages also makes conversion into logic forms harder.

[11] By <eat> I mean any word which becomes identical to a base form *eat* after stemming, i.e. *eat*, *ate*, *eaten*, etc.

4.3.5 Answer extraction

Answer extraction stage depends heavily on a sentence structure, which is language dependent. This is another element, which becomes harder to implement for languages with the free word order.

4.3.6 Other modules

- **WordNet** contains relations about words in a particular language. Some of them correspond to relations between real-world entities (*a horse* is a hyponym of *an animal* in most languages), but some do not.
- **Named Entity Recognition (NER)** is an example of a task which becomes substantially more complex when implemented to the Slavonic languages. A core part of most such solutions is a *gazetter* i.e. a large database of known proper names. For example, we could most likely find there the name *Nelson*. If it appears in a text, NER module spots it easily. Unfortunately, this is not the case in Polish because of the rich nominal inflection - this very same name could appear in 5 different singular forms (*Nelson*, *Nelsona*, *Nelsonowi*, *Nelsonem*, and *Nelsonie*) and 5 plural (e.g. while talking about Nelson family). Complicated and often incorrect inflection of foreign proper names also needs to be taken into account.
- **Anaphora resolution** clearly depends on a processed language.

4.4 Linguistic tools

In the previous section we've mentioned a lot of linguistic tools which are relevant when building a QA system, but could not be used in different languages. Let us check their availability for Polish:

- **stemming** - As stated previously, several stemmers for Polish, discussed in [44], exist.
- **morphological analysis** - There are several tools of that type, *Morfeusz SGJP* [45], *Morfologik* [46], *PoliMorf* [46], to name only a few.
- **tagging** - Several taggers, capable of selecting the most probable interpretation of these provided by morphological analysers, exist, such as *PANTERA* [47] (using transformation-based learning) or *TaKIPI* [48] (using decision trees),
- **shallow parsing** - *Spejd* [49] allows to obtain a shallow structure of a sentence.
- **deep parsing** - *Świgra* [50] creates deep parsing trees in the DCG formalism.
- **named entity recognition** - Two tools could be employed for this task: *NERF* [51] and *Liner2* [52].
- **WordNet** - Widely available are: *plWordNet* [53], containing over 110000 synsets and *PolNet* [54] with approximately 11700 synsets.
- **anaphora resolution** - Polish anaphora resolution module (*Project CORE*) is under development [55].

As visible above, there are plenty of linguistic tools available for Polish, making a good starting point for developing a Polish question answering system.

5 Conclusion

In this paper, the possibility of building a Polish question answering system was discussed. First, a general explanation of a QA problem and possible improvements of the user experience by implementing natural language communication modules were shown to justify an effort of tackling the task. To focus on the most important properties of the QA problem, its definition, also usable for preparing an evaluation, was specified. Numerous solutions for English were presented to show their common architecture and distinctive features. Presented description of properties of the Slavonic languages and an analysis of language dependence of the elements of the scheme substantiate the claim that we need to apply a different approach to several issues, when Polish is concerned. Its feasibility is further supported by an enumeration of existing implementations of the relevant linguistic tools.

I demonstrated that an open-domain text-based question answering system for Polish is not only desirable but also achievable given the current level of development of Polish NLP. This study is a first step of building such a system.

Acknowledgements

Critical reading of the manuscript by Agnieszka Mykowiecka is gratefully acknowledged. Study was supported by research fellowship within "Information technologies: research and their interdisciplinary applications" agreement number POKL.04.01.01-00-051/10-00.

References

[1] Yandex.ru: Report on Yandex.ru queries. Technical report (2010)

[2] Hook, D.: Study of Search Engine Transaction Logs Shows Little Change in How Users use Search Engines. Evidence Based Library and Information Practice **1**(3) (2006) 88–94

[3] Andrenucci, A., Sneiders, E.: Automated Question Answering: Review of the Main Approaches. In: Third International Conference on Information Technology and Applications (ICITA'05). (2005) 514–519

[4] Shapiro, S.C.: Encyclopedia of Artificial Intelligence. John Wiley & Sons, Inc., New York (1992)

[5] Ferrández, A., Vicedo, J.L.: Coreference In Q&A. In Strzalkowski, T., Harabagiu, S.M., eds.: Advances in Open Domain Question Answering (Text, Speech and Language Technology). Springer Netherlands (2007) 71–96

[6] Ferrucci, D.A., Brown, E., Chu-carroll, J., Fan, J., Gondek, D., Kalyanpur, A.A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J., Schlaefer, N., Welty, C.: Building Watson: An Overview of the DeepQA Project. AI Magazine **31**(3) (2010) 59–79

[7] Dang, H.T., Kelly, D., Lin, J.: Overview of the TREC 2007 Question Answering track. In: Proceedings of The Sixteenth Text REtrieval Conference, TREC 2007. (2007)

[8] Dang, H.T.: Overview of the TAC 2008 Opinion Question Answering and Summarization Tasks. In: Proceedings of Text Analysis Conference (TAC 2009). (2008)

[9] Peñas, A., Hovy, E.: QA4MRE: Question Answering for Machine Reading Evaluation at CLEF 2012 (2012)

[10] Sasaki, Y., Lin, C.j., Chen, K.h., Chen, H.h.: Overview of the NTCIR-6 Cross-Lingual Question Answering (CLQA) Task. In: Proceedings of NTCIR-6 Workshop Meeting. (2007)

[11] Ferrucci, D.A., Nyberg, E., Allan, J., Barker, K., Brown, E., Chu-Carroll, J., Ciccolo, A., Duboue, P., Fan, J., Gondek, D., Hovy, E., Katz, B., Lally, A., Mccord, M., Morarescu, P., Murdock, B., Porter, B., Prager, J., Heights, Y.: Towards the Open Advancement of Question Answering Systems. Technical report, IBM (2009)

[12] Simmons, R.F.: Natural Language Question-Answering Systems: 1969. Communications of the ACM **13**(1) (1970) 15–30

[13] Katz, B.: Annotating the World Wide Web Using Natural Language. In: Proceedings of the 5th RIAO Conference on Computer Assisted Information Searching on the Internet RIAO 97. (1997)

[14] Voorhees, E.M.: The TREC-8 Question Answering Track Report. In: Proceedings of The Eight Text REtrieval Conference (TREC 2000). Volume 7., National Institute of Standards and Technology (1999)

[15] Voorhees, E.M.: Overview of the TREC-9 Question Answering Track. In: Proceedings of The Ninth Text REtrieval Conference (TREC 2000). (2000)

[16] Voorhees, E.M.: Overview of the TREC 2001 Question Answering Track. In: Proceedings of the Tenth Text Retrieval Conference (TREC 2001). (2001)

[17] Voorhees, E.M.: Overview of the TREC 2002 Question Answering Track. In: Proceedings of The Eleventh Text REtrieval Conference (TREC 2002). (2002)

[18] Voorhees, E.M.: Overview of the TREC 2003 Question Answering Track. In: Proceedings of the Twelfth Text REtrieval Conference (TREC 2003). Volume 142. (2003)

[19] Voorhees, E.M.: Overview of the TREC 2004 Question Answering Track. In: Proceedings of The Thirteenth Text REtrieval Conference (TREC 2004). (2004)

[20] Voorhees, E.M., Dang, H.T.: Overview of the TREC 2005 Question Answering Track. In: Proceedings of The Fourteenth Text REtrieval Conference (TREC 2005). (2005)

[21] Dang, H.T., Lin, J., Kelly, D.: Overview of the TREC 2006 Question Answering Track. In: Proceedings of The Fifteenth Text REtrieval Conference (TREC 2006). Volume 2004. (2006)

[22] Hartrumpf, S.: Question Answering using Sentence Parsing and Semantic Network Matching. In: Proceedings of the 5th conference on Cross-Language Evaluation Forum: Multilingual Information Access for Text, Speech and Images CLEF'04ultilingual Information Access for Text, Speech and Images CLEF'04. (2004) 512–521

[23] Hovy, E., Hermjakob, U., Lin, C.Y.: The Use of External Knowledge in Factoid QA. In: Proceedings of the Tenth Text Retrieval Conference (TREC 2001). (2001)

[24] Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Gîrju, R., Rus, V., Morarescu, P.: The role of lexico-semantic feedback in open-domain textual question-answering. In: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics - ACL '01. (July 2001) 282–289

[25] Hovy, E., Gerber, L., Hermjakob, U., Lin, C.Y., Ravichandran, D.: Toward Semantics-Based Answer Pinpointing. In: Proceedings of the first international conference on Human Language Technology research (HLT). (2001)

[26] Fellbaum, C.: WordNet: An Electronic Lexical Database. The MIT Press (1998)

[27] Ahn, D., Jijkoun, V., Mishne, G., Müller, K., De Rijke, M., Schlobach, S.: Using Wikipedia at the TREC QA Track. In Voorhees, E.M., Buckland, L.P., eds.: Proceedings of The Thirteenth Text REtrieval Conference (TREC 2004). (2004)

[28] Li, X., Roth, D.: Learning Question Classifiers. In: Proceedings of the 19th International Conference on Computational Linguistics (COLING-2002). Volume 1 of COLING '02. (2002)

[29] Buchholz, S., Daelemans, W.: SHAPAQA: Shallow Parsing for Question Answering on the World Wide Web. In: Proceedings of Euroconference on Recent Advances in Natural Language Processing RANLP2001. (2001)

[30] Ravichandran, D., Hovy, E.: Learning surface text patterns for a Question Answering system. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02. (2002) 41–47

[31] Duclaye, F., Yvon, F., Collin, O.: Learning paraphrases to improve a question-answering system. In: Proceedings of the 10th Conference of EACL Workshop Natural Language Processing for Question-Answering. (2003)

[32] Miliaraki, S., Androutsopoulos, I.: Learning to identify single-snippet answers to definition questions. In: Proceedings of the 20th international conference on Computational Linguistics - COLING '04. (August 2004)

[33] Moldovan, D., Novischi, A.: Lexical chains for question answering. In: Proceedings of the 19th International Conference on Computational Linguistics (COLING-2002). (2002)

[34] Boni, M.D., Manandhar, S.: The Use of Sentence Similarity as a Semantic Relevance Metric for Question Answering. In: Proceedings of the AAAI Symposium on New Directions in Question Answering. (2003)

[35] Ko, J., Si, L., Nyberg, E.: A Probabilistic Framework for Answer Selection in Question Answering. In: Proceedings of the 2007 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics HLT-NAACL 2007, Association for Computational Linguistics (2007) 524–531

[36] Ittycheriah, A.: A statistical approach for open domain question answering. In Strzalkowski, T., Harabagiu, S.M., eds.: Advances in Open Domain Question Answering (Text, Speech and Language Technology). Volume 32 of Text, Speech and Language Technology. Springer Netherlands (2007) 35–69

[37] Moldovan, D., Paşca, M., Surdeanu, M.: Some Advanced Features of LCC's PowerAnswer. In Strzalkowski, T., Harabagiu, S.M., eds.: Advances in Open Domain Question Answering (Text, Speech and Language Technology). Volume 32. Springer Netherlands (2006) 3 – 34

[38] Przepiórkowski, A.: Slavonic information extraction and partial parsing. In: Proceedings of the Workshop on Balto-Slavonic Natural Language Processing Information Extraction and Enabling Technologies - ACL '07. (2007)

[39] Vetulani, Z.: PROLOG Implementation of an Access in Polish to a Data Base. In: Studia z automatyki, XII. (1988) 5–23

[40] Vetulani, Z.: Question answering system for Polish (POLINT) and its language resources. In: Proceedings of the Question Answering - Strategy and Resources Workshop, Las Palmas de Grand Ganaria, 28th May, Paris, ELRA (2002) 51–55

[41] Duclaye, F., Sitko, J., Filoche, P., Collin, O.: A Polish Question-Answering System for Business Information. In: BIS 2002, 5th International Conference on Business Information Systems, Poznań, Poland, 24-25 April 2002. (2002) 209–212

[42] Walas, M.: How to answer yes/no spatial questions using qualitative reasoning? In Gelbukh, A., ed.: 13th International Conference on Computational Linguistics and Intelligent Text Processing. (2012) 330–341

[43] Piechociński, D., Mykowiecka, A.: Question answering in Polish using shallow parsing. In Garabík, R., ed.: Computer Treatment of Slavic and East European Languages: Proceedings of the Third International Seminar, Bratislava, Slovakia, VEDA: Vydava- tel'stvo Slovenskej akadéme vied (2005) 167–173

[44] Weiss, D.: A Survey of Freely Available Polish Stemmers and Evaluation of Their Applicability in Information Retrieval. In: 2nd Language and Technology Conference, Poznań, Poland. (2005) 216–221

[45] Woliński, M.: Morfeusz — a Practical Tool for the Morphological Analysis of Polish. In Kłopotek, M., Wierzchoń, S., Trojanowski, K., eds.: Intelligent Information Processing and Web Mining. (2006) 511–520

[46] Woliński, M., Miłkowski, M., Ogrodniczuk, M., Przepiórkowski, A., Szałkiewicz, L.: PoliMorf: a (not so) new open morphological dictionary for Polish. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012. (2012) 860–864

[47] Acedański, S.: A morphosyntactic Brill Tagger for inflectional languages. In: Proceedings of the 7th international conference on Advances in Natural Language Processing (IceTAL'10). (August 2010) 3–14

[48] Piasecki, M.: Polish Tagger TaKIPI: Rule Based Construction and Optimisation. Task Quarterly **11**(1-2) (2007) 151–167

[49] Przepiórkowski, A.: Powierzchniowe przetwarzanie języka polskiego. Akademicka Oficyna Wydawnicza EXIT, Warszawa (2008)

[50] Woliński, M.: An efficient implementation of a large grammar of Polish. Archives of Control Sciences **15** (2005) 251–258

[51] Savary, A., Waszczuk, J.: Narzędzia do anotacji jednostek nazewniczych. In: Narodowy Korpus Języka Polskiego [Eng.: National Corpus of Polish]. (2012) 225–252

[52] Marcińczuk, M., Janicki, M.: Optimizing CRF-based Model for Proper Name Recognition in Polish Texts. In: CICLing 2012, Part I. (2012) 258–269

[53] Maziarz, M., Piasecki, M., Szpakowicz, S.: Approaching plWordNet 2.0. In: Proceedings of the 6th Global Wordnet Conference. (2012)

[54] Vetulani, Z., Kubis, M., Obrębski, T.: PolNet – Polish WordNet: Data and Tools. In: Proceedings of the seventh International conference on Language Resources and Evaluation (LREC 2010). (2010)

[55] Kopeć, M., Ogrodniczuk, M.: Creating a Coreference Resolution System for Polish. In: The eighth international conference on Language Resources and Evaluation (LREC). (2012)

Comparing SAT-based bounded model checking RTECTL and ECTL properties

Agnieszka M. Zbrzezny

IMCS, Jan Długosz University. Al. Armii Krajowej 13/15, 42-200 Częstochowa, Poland.

Abstract

We compare two SAT-based bounded model checking algorithms for properties expressed in the existential fragment of soft real time computation tree logic (RTECTL) and in the existential fragment of computation tree logic (ECTL). To this end we use a faulty train controller system (FTC) and the generic pipeline paradigm (GPP), the classic concurrency problems, which we formalise by means of a finite transition system. We consider several properties of the problems that can be expressed in both RTECTL and ECTL, and we present the performance evaluation of the mentioned BMC methods by means of the running time and the memory used.

1 Introduction

The problem of model checking [1] is to check automatically whether a structure M defines a model for a modal (temporal, epistemic, etc.) formula α. The practical applicability of model checking is strongly limited by the state explosion problem, which means that the number of model states grows exponentially in the size of the system representation. To avoid this problem a number of state reduction techniques and symbolic model checking approaches have been developed, among others, [2, 3, 4, 5].

The SAT-based bounded model checking (BMC) is one of the symbolic model checking technique designed for finding witnesses for existential properties or counterexamples for universal properties. Its main idea is to consider a model reduced to a specific depth. The first BMC method was proposed in [6], and it was designed for linear time properties. Next in [7] the method has been extended to handle branching time properties. Further extensions of the SAT-based BMC method for real-time systems and multi-agent systems can be found, among others, in [8, 9].

The existential fragment of the computation tree logic (ECTL) [7] is a formalism that allow for specification of properties such as "*there is a computation such that α will eventually request*", or "*there is a computation such that α will never be true*", but it is impossible to to directly express bounded properties like for example "*there is a computation such that α will be true in less than 15 unit time*", or "*there is a computation such that α will never be averted after 15 unit time*", or "*there is a computation such that α will always be averted between 10 and 20 unit time*". Note however that this bounded properties can be formalised in ECTL by using nested applications of the next state operators, but the resulting ECTL formulae can be very complicated and problematic to work with. An existential fragment of the soft real-time CTL (RTECTL) [10] defeats this restriction

by introducing time-bounded temporal operators, and it supplies a much more compact and convenient way of expressing time-bounded properties.

The purpose of this paper is to compare the SAT-based bounded model checking of RTECTL properties and the SAT-based bounded model checking of equivalent ECTL properties and to present a correct and complete translation from RTECTL to ECTL.

We have expected that BMC for RTECTL would give better performance than BMC for the ECTL formulae that result from the translation of RTECTL formulae. This is because the size of these ECTL formulae is exponential in the size of the original RTECTL formulae. However, our experiments have shown that for certain RTECTL formulae BMC can be less effective than BMC for their translation into ECTL formulae. These are the formulae for which the interval at the operator $\mathbf{EG}$ is finite: then after the translation to ECTL, $\mathbf{EG}$ operator disappears, and there are more paths, but they are much shorter.

The structure of the paper is as follows. In Section 2 we shortly recall definitions of transition systems and their parallel composition. Then, we present syntax and semantics of RTECTL and ECTL. In Section 3 we define a translation from RTECTL to ECTL, and show its correctness. In section 4 we shortly present BMC technique for RTECTL and ECTL. In Section 5 we present experimental evaluation of the SAT-based BMC for RTECTL [11] and ECTL [12] on equivalent formulae and for a faulty train controller system (FTC) and generic pipeline paradigm (GPP). In Section 6 we conclude the paper.

2 Preliminaries

In this section we first define Transition System (TS), then we recall syntax and semantics of two logics ECTL and RTECTL.

2.1 Transition System

A Transition System [13] (also called a *model*) is a tuple $\mathcal{M} = (S, Act, \longrightarrow, s^0, AP, L)$ where:

- S is a set of states,
- Act is a set of actions,
- $\longrightarrow \subseteq S \times Act \times S$ is a transition relation,
- $s^0 \in S$ is the initial state,
- AP is a set of atomic propositions, and
- $L : S \to 2^{AP}$ is a labelling function.

Transition system is called finite if S, Act, and AP are finite. For convenience, we write $s \xrightarrow{\sigma} s'$ instead of $(s, \sigma, s) \in \longrightarrow$. Moreover, we write $s \longrightarrow s'$ if $s \xrightarrow{\sigma} s'$, for some $\sigma \in Act$.

From now on we assume that a considered model has no terminal states, i.e. for every $s \in S$ there exist $s' \in S$ such that $s \longrightarrow s'$. The set of all natural numbers is denoted by $\mathbb{N}$, and the set of all positive natural numbers by $\mathbb{N}_+$. A *path* in $\mathcal{M}$ is an infinite sequence $\pi = (s_0, s_1, \ldots)$ of states such that $s_i \longrightarrow s_{i+1}$ for each $i \in \mathbb{N}$. For a path $\pi = (s_0, s_1, \ldots)$ and $i \in \mathbb{N}$, the i-th state of π is defined as $\pi(i) = s_i$. The i-th prefix of π, denoted by $\pi[..i]$ is defined as $\pi[..i] = (s_0, s_1, \ldots, s_i)$, and the i-th suffix of π, denoted by π^i, is defined as $\pi^i = (s_i, s_{i+1}, \ldots)$. Note that if π is a path in $\mathcal{M}$ then the suffix π^i is also

a path in $\mathcal{M}$. By $\Pi(s)$ we denote the set of all the paths starting at $s \in S$, and by $\Pi(\mathcal{M})$ we denote the set of all the paths in $\mathcal{M}$.

2.2 Models and parallel composition

The concurrent systems are designed as collections of interacting computational processes that may be executed in parallel. Therefore, we assume that a concurrent system is modelled as a network of models that run in parallel and communicate with each other via executing shared actions. There are several ways of defining a parallel composition. We adapt the standard definition, namely, in the parallel composition the transitions not corresponding to a shared action are interleaved, whereas the transitions labelled with a shared action are synchronised.

We formalise the above by the following definition of parallel composition. Let $\mathcal{M}_i = (S_i, Act_i, \longrightarrow_i, s_i^0, AP_i, L_i)$ be models such that for all $i, j \in \{1, \ldots, m\}$, if $i \neq j$, then $AP_i \cap AP_j = \emptyset$. We take $Act = \bigcup_{i=1}^{m} Act_i$, and for $\sigma \in Act$ we define a set $Act(\sigma) = \{1 \leq i \leq m \mid \sigma \in Act_i\}$ that gives the indices of the components that synchronise at σ. *A parallel composition of m models* $\mathcal{M}_i$ is the model $\mathcal{M} = (S, Act, \longrightarrow, s^0, AP, L)$, where $S = \prod_{i=1}^{m} S_i$, $Act = \bigcup_{i=1}^{m} Act_i$, a transition $((s_1, \ldots, s_m), \sigma, (s'_1, \ldots, s'_m)) \in \longrightarrow$ iff $(\forall j \in Act(\sigma))\ (s_j, \sigma, s'_j) \in \longrightarrow_j$, $(\forall i \in \{1, \ldots, m\} \setminus Act(\sigma))\ s'_i = s_i$, $s^0 = (s_1^0, \ldots, s_m^0)$, $AP = \bigcup_{i=1}^{m} AP_i$, and $L((s_1, \ldots, s_m)) = \bigcup_{i=1}^{m} L_i(s_i)$.

2.3 The ECTL and RTECTL logics

2.3.1 Syntax of ECTL

The syntax of ECTL formulae over the set AP of atomic propositions is defined by the following grammar:

$$\varphi ::= \mathbf{true} \mid \mathbf{false} \mid p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathbf{EX}\varphi \mid \mathbf{E}(\varphi \mathbf{U} \varphi) \mid \mathbf{EG}\varphi$$

where $p \in AP$ and φ is a formula. The symbols $\mathbf{X}$, $\mathbf{U}$ and $\mathbf{G}$ are the modal operators for "neXt time", "Until" and "Globally", respectively. The symbol $\mathbf{E}$ is the existential path quantifier.

The derived basic modalities are defined as follows:

$$\mathbf{EF}\alpha \stackrel{df}{=} \mathbf{E}(\mathbf{true}\,\mathbf{U}\,\alpha), \qquad \mathbf{E}(\alpha\,\mathbf{R}\,\beta) \stackrel{df}{=} \mathbf{E}(\beta\,\mathbf{U}\,(\alpha \wedge \beta)) \vee \mathbf{EG}\beta.$$

2.3.2 Semantics of ECTL

Let $\mathcal{M}$ be a model, and φ an ECTL formula. An ECTL formula φ is *true* in the model $\mathcal{M}$ (in symbols $\mathcal{M} \models \varphi$) iff $\mathcal{M}, s^0 \models \varphi$ (i.e., φ is true at the initial state of the model $\mathcal{M}$), where

$$\begin{aligned}
&\mathcal{M}, s \models \mathbf{true}, \\
&\mathcal{M}, s \not\models \mathbf{false}, \\
&\mathcal{M}, s \models p && \text{iff } p \in L(s), \\
&\mathcal{M}, s \models \neg p && \text{iff } p \notin L(s), \\
&\mathcal{M}, s \models \alpha \wedge \beta && \text{iff } \mathcal{M}, s \models \alpha \text{ and } \mathcal{M}, s \models \beta,
\end{aligned}$$

$$
\begin{array}{ll}
\mathcal{M}, s \models \alpha \vee \beta & \text{iff } \mathcal{M}, s \models \alpha \text{ or } \mathcal{M}, s \models \beta, \\
\mathcal{M}, s \models \mathbf{EX}\alpha & \text{iff } (\exists \pi \in \Pi(s))(\mathcal{M}, \pi(1) \models \alpha), \\
\mathcal{M}, s \models \mathbf{E}(\alpha \mathbf{U} \beta) & \text{iff } (\exists \pi \in \Pi(s))(\exists m \geq 0)(\mathcal{M}, \pi(m) \models \beta \text{ and} \\
 & \quad (\forall j < m)\mathcal{M}, \pi(j) \models \alpha), \\
\mathcal{M}, s \models \mathbf{EG}\alpha & \text{iff } (\exists \pi \in \Pi(s))(\forall m \geq 0)(\mathcal{M}, \pi(m) \models \alpha).
\end{array}
$$

2.3.3 Syntax of RTECTL

Let $p \in AP$ and I be an interval in $\mathbb{N} = \{0, 1, 2, \ldots\}$ of the form: $[a, b)$ and $[a, \infty)$, for $a, b \in \mathbb{N}$. Note that the remaining forms of intervals can be defined by means of $[a, b)$ and $[a, \infty)$.

The language RTECTL is defined by the following grammar:

$$\varphi ::= \mathbf{true} \mid \mathbf{false} \mid p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathbf{EX}\varphi \mid \mathbf{E}(\varphi \mathbf{U}_I \varphi) \mid \mathbf{EG}_I \varphi$$

The derived basic modalities are defined as follows:

$$\mathbf{EF}_I \alpha \stackrel{df}{=} \mathbf{E}(\mathbf{true}\, \mathbf{U}_I\, \alpha), \qquad \mathbf{E}(\alpha\, \mathbf{R}_I\, \beta) \stackrel{df}{=} \mathbf{E}(\beta\, \mathbf{U}_I\, (\alpha \wedge \beta)) \vee \mathbf{EG}_I \beta.$$

2.3.4 Semantics of RTECTL

Let $\mathcal{M}$ be a model and φ an RTECTL formula. An RTECTL formula φ is *true* in the model $\mathcal{M}$ (in symbols $\mathcal{M} \models_{rt} \varphi$) iff $\mathcal{M}, s^0 \models_{rt} \varphi$ (i.e., φ is true at the initial state of the model $\mathcal{M}$), where:

$$
\begin{array}{ll}
\mathcal{M}, s \models_{rt} \mathbf{true}, & \\
\mathcal{M}, s \not\models_{rt} \mathbf{false}, & \\
\mathcal{M}, s \models_{rt} p & \text{iff } p \in L(s), \\
\mathcal{M}, s \models_{rt} \neg p & \text{iff } p \notin L(s), \\
\mathcal{M}, s \models_{rt} \alpha \wedge \beta & \text{iff } \mathcal{M}, s \models_{rt} \alpha \text{ and } \mathcal{M}, s \models_{rt} \beta, \\
\mathcal{M}, s \models_{rt} \alpha \vee \beta & \text{iff } \mathcal{M}, s \models_{rt} \alpha \text{ or } \mathcal{M}, s \models_{rt} \beta, \\
\mathcal{M}, s \models_{rt} \mathbf{EX}\alpha & \text{iff } (\exists \pi \in \Pi(s))(\mathcal{M}, \pi(1) \models_{rt} \alpha), \\
\mathcal{M}, s \models_{rt} \mathbf{E}(\alpha \mathbf{U}_I \beta) & \text{iff } (\exists \pi \in \Pi(s))(\exists m \in I)(\mathcal{M}, \pi(m) \models_{rt} \beta \text{ and} \\
 & \quad (\forall i < m)\mathcal{M}, \pi(i) \models_{rt} \alpha), \\
\mathcal{M}, s \models_{rt} \mathbf{EG}_I \alpha & \text{iff } (\exists \pi \in \Pi(s))(\forall m \in I)(\mathcal{M}, \pi(m) \models_{rt} \alpha).
\end{array}
$$

3 Translation from RTECTL into ECTL

In this section we present a translation from the RTECTL language to the ECTL language that is based on the intuitive description of such a translation presented in [10, 14]. We focused on the existential part of RTCTL only, because we use the BMC method. Moreover, we would like to stress that our semantics of the operator $\mathbf{G}_I$ is different than the one presented in [10, 14]. Further, unlike in [10, 14], we present the translation for all types of intervals that can appear together with the temporal operators $\mathbf{U}_I$ and $\mathbf{G}_I$, and for both operators.

Let $p \in AP$, α and β be formulae of RTECTL, $a \in \mathbb{N}$, $b \in \mathbb{N} \cup \{\infty\}$ and $a < b$. We define the translation from RTECTL into ECTL as a function tr : RTECTL $\rightarrow$ ECTL in the following way:

$$
\begin{aligned}
tr(p) &= p \\
tr(\neg p) &= \neg p \\
tr(\alpha \wedge \beta) &= tr(\alpha) \wedge tr(\beta) \\
tr(\alpha \vee \beta) &= tr(\alpha) \vee tr(\beta)
\end{aligned}
$$

$$
tr(\mathbf{E}(\alpha \mathbf{U}_{[a,b)} \beta)) = \begin{cases}
tr(\alpha) \wedge \mathbf{EX}(tr(\mathbf{E}(\alpha \mathbf{U}_{[a-1,b-1)} \beta))) & \text{if } a > 0 \text{ and } 1 < b < \infty \\
tr(\beta) \vee tr(\alpha) \wedge \mathbf{EX}(tr(\mathbf{E}(\alpha \mathbf{U}_{[0,b-1)} \beta))) & \text{if } a = 0 \text{ and } 1 < b < \infty \\
tr(\alpha) \wedge \mathbf{EX}(tr(\mathbf{E}(\alpha \mathbf{U}_{[a-1,\infty)} \beta))) & \text{if } a > 0 \text{ and } b = \infty \\
\mathbf{E}(tr(\alpha) \mathbf{U} tr(\beta)) & \text{if } a = 0 \text{ and } b = \infty \\
tr(\beta) & \text{if } a = 0 \text{ and } b = 1
\end{cases}
$$

$$
tr(\mathbf{EG}_{[a,b)} \alpha) = \begin{cases}
\mathbf{EX}(tr(\mathbf{EG}_{[a-1,b-1)} \alpha)) & \text{if } a > 0 \text{ and } 1 < b < \infty \\
tr(\alpha) \wedge \mathbf{EX}(tr(\mathbf{EG}_{[a,b-1)} \alpha)) & \text{if } a = 0 \text{ and } 1 < b < \infty \\
\mathbf{EX}(tr(\mathbf{EG}_{[a-1,\infty)} \alpha)) & \text{if } a > 0 \text{ and } b = \infty \\
\mathbf{EG} tr(\alpha) & \text{if } a = 0 \text{ and } b = \infty \\
tr(\alpha) & \text{if } a = 0 \text{ and } b = 1
\end{cases}
$$

$$
tr(\mathbf{EX}\alpha) = \mathbf{EX} tr(\alpha)
$$

Because $\mathbf{EF}_{[a,b)} \alpha \stackrel{df}{=} \mathbf{E}(true \mathbf{U}_{[a,b)} \alpha)$, the translation for $\mathbf{EF}_{[a,b)} \alpha$ can be defined using the translation for $\mathbf{E}(\alpha \mathbf{U}_{[a,b)} \beta)$:

$$
tr(\mathbf{EF}_{[a,b)} \alpha) = \begin{cases}
\mathbf{EX}(tr(\mathbf{EF}_{[a-1,b-1)} \alpha)) & \text{if } a > 0 \text{ and } 1 < b < \infty \\
tr(\alpha) \vee \mathbf{EX}(tr(\mathbf{EF}_{[0,b-1)} \alpha)) & \text{if } a = 0 \text{ and } 1 < b < \infty \\
\mathbf{EX}(tr(\mathbf{EF}_{[a-1,\infty)} \alpha)) & \text{if } a > 0 \text{ and } b = \infty \\
\mathbf{EF} tr(\alpha) & \text{if } a = 0 \text{ and } b = \infty \\
tr(\alpha) & \text{if } a = 0 \text{ and } b = 1
\end{cases}
$$

The following theorem, which can be proven by induction on the length of an RTECTL formula, states correctness of the above translation.

Proposition 1. *Let α be an* RTECTL *formula, and $a \in \mathbb{N}$. Then,*

$$
tr(\mathbf{EG}_{[a,\infty)} \alpha)) = \underbrace{\mathbf{EX} \ldots \mathbf{EX}}_{a} (\mathbf{EG} tr(\alpha)).
$$

Example 1. $tr(\mathbf{EG}_{[3,\infty)}\alpha) = \mathbf{EXEXEX}(\mathbf{EG}tr(\alpha))$.

Proposition 2. *Let α be an* RTECTL *formula, and $a, b \in \mathbb{N}$. If $a < b < \infty$, then*

$$tr(\mathbf{EG}_{[a,b)}\alpha)) = \underbrace{\mathbf{EX}\ldots\mathbf{EX}}_{a}(tr(\alpha) \wedge \mathbf{EX}(tr(\alpha)) \wedge \ldots \wedge \underbrace{\mathbf{EX}\ldots\mathbf{EX}}_{b-a}(tr(\alpha))).$$

Example 2. $tr(\mathbf{EG}_{[3,6)}\alpha) = \mathbf{EXEXEX}(tr(\alpha) \wedge \mathbf{EX}(tr(\alpha) \wedge \mathbf{EX}(tr(\alpha))))$.

Proposition 3. *Let α and β be* RTECTL *formulae, and $a \in \mathbb{N}$. If $a > 0$, then*

$$tr(\mathbf{E}(\alpha\mathbf{U}_{[a,\infty)}\beta)) = tr(\alpha) \wedge \underbrace{\mathbf{EX}(tr(\alpha) \wedge \mathbf{EX}(tr(\alpha)\ldots}_{a-1} \wedge \mathbf{E}(tr(\alpha)\mathbf{U}tr(\beta)))).$$

Example 3. $tr(\mathbf{E}(\alpha\mathbf{U}_{[3,\infty)}\beta)) = tr(\alpha) \wedge \mathbf{EX}(tr(\alpha) \wedge \mathbf{EX}(tr(\alpha) \wedge \mathbf{E}(tr(\alpha)\mathbf{U}tr(\beta))))$.

Proposition 4. *Let α and β be* RTECTL *formulae, and $a, b \in \mathbb{N}$. If $a < b < \infty$, then*

$$tr(\mathbf{E}(\alpha\mathbf{U}_{[a,b)}\beta)) = tr(\alpha) \wedge \underbrace{\mathbf{EX}(tr(\alpha) \wedge \mathbf{EX}(tr(\alpha) \wedge \ldots}_{a}$$
$$\wedge \underbrace{\mathbf{EX}(tr(\beta) \vee tr(\alpha) \wedge \mathbf{EX}(tr(\beta) \vee tr(\alpha)\ldots}_{b-a-1} \wedge \mathbf{EX}(tr(\beta)))))))).$$

Example 4. $tr(\mathbf{E}(\alpha\mathbf{U}_{[3,6)}\beta)) = tr(\alpha) \wedge \mathbf{EX}(tr(\alpha) \wedge \mathbf{EX}(tr(\alpha) \wedge \mathbf{EX}(tr(\alpha) \wedge$
$\mathbf{EX}(tr(\beta) \vee tr(\alpha) \wedge \mathbf{EX}(tr(\beta) \vee tr(\alpha) \wedge \mathbf{EX}(tr(\beta)))))))$.

Lemma 1. *Let $\mathcal{M}$ be a model, and φ an* RTECTL *formula. Then* $(\forall s \in S)(\mathcal{M}, s \models_{rt} \varphi \Rightarrow \mathcal{M}, s \models tr(\varphi))$

Proof. We proceed by induction on the length of formulae. Let $s \in S$, and assume that $\mathcal{M}, s \models_{rt} \varphi$. Now consider the following cases:

1. $\varphi \in AP$. Then, since $tr(\varphi) = \varphi$, we have that $tr(\varphi) \in AP$. Therefore, $\mathcal{M}, s \models_{rt} \varphi \iff \varphi \in L(s) \iff tr(\varphi) \in L(s) \iff \mathcal{M}, s \models tr(\varphi)$.
2. $\varphi = \neg p$, where $p \in AP$. Then $tr(\varphi) = \varphi$. Therefore, $\mathcal{M}, s \models_{rt} \varphi \iff \mathcal{M}, s \models_{rt} \neg p \iff p \notin L(s) \iff \mathcal{M}, s \models \neg p \iff \mathcal{M}, s \models \varphi \iff \mathcal{M}, s \models tr(\varphi)$.
3. $\varphi = \alpha \wedge \beta$. By the definition of the satisfiability relation we have that $\mathcal{M}, s \models_{rt} \alpha$ and $\mathcal{M}, s \models_{rt} \beta$. By the inductive hypothesis, we get $\mathcal{M}, s \models tr(\alpha)$ and $\mathcal{M}, s \models tr(\beta)$. Thus, $\mathcal{M}, s \models tr(\alpha) \wedge tr(\beta)$, and therefore $\mathcal{M}, s \models tr(\alpha \wedge \beta)$.
4. $\varphi = \alpha \vee \beta$. By the definition of the satisfiability relation we have that $\mathcal{M}, s \models_{rt} \alpha$ or $\mathcal{M}, s \models_{rt} \beta$. By the inductive hypothesis, we get $\mathcal{M}, s \models tr(\alpha)$ or $\mathcal{M}, s \models tr(\beta)$. Thus, $\mathcal{M}, s \models tr(\alpha) \vee tr(\beta)$, and therefore $\mathcal{M}, s \models tr(\alpha \vee \beta)$.
5. $\varphi = \mathbf{EX}\alpha$. By the definition of the satisfiability relation we have that there exists $\pi \in \Pi(s)$ such that $\mathcal{M}, \pi(1) \models_{rt} \alpha$. By the inductive hypothesis, we conclude that $\mathcal{M}, \pi(1) \models tr(\alpha)$. Thus, $\mathcal{M} \models \mathbf{EX}tr(\alpha)$, since $\pi(0) = s$. Therefore $\mathcal{M} \models tr(\mathbf{EX}\alpha)$, since $tr(\mathbf{EX}\alpha) = \mathbf{EX}tr(\alpha)$.

6. $\varphi = \mathbf{EG}_I\alpha$. By the definition of the satisfiability relation we have that there exists $\pi \in \Pi(s)$ such that $(\forall m \in I)(\mathcal{M}, \pi(m) \models_{rt} \alpha)$. By the inductive hypothesis, we conclude that there exists $\pi \in \Pi(s)$ such that

$$(\forall m \in I)(\mathcal{M}, \pi(m) \models tr(\alpha)). \tag{1}$$

Now consider the following cases:

(a) $I = [a, \infty)$. From (1) we conclude that $\mathcal{M}, s \models \underbrace{\mathbf{EX} \ldots \mathbf{EX}}_{a}(\mathbf{EG}tr(\alpha))$. From this, by Proposition 1, it follows that $\mathcal{M}, s \models tr(\mathbf{EG}_{[a,\infty)}\alpha)$.

(b) $I = [a, b)$, where $a < b < \infty$. From (1) we conclude that

$$\mathcal{M}, s \models \underbrace{\mathbf{EX} \ldots \mathbf{EX}}_{a}(tr(\alpha) \wedge \mathbf{EX}(tr(\alpha)) \wedge \ldots \wedge \underbrace{\mathbf{EX} \ldots \mathbf{EX}}_{b-a}(tr(\alpha))).$$

From this, by Proposition 2, we get $\mathcal{M}, s \models tr(\mathbf{EG}_{[a,b)}\alpha))$.

7. $\varphi = \mathbf{E}(\alpha \mathbf{U}_I \beta)$. By the definition of the satisfiability relation we have that there exists $\pi \in \Pi(s)$ such that $(\exists m \in I)(\mathcal{M}, \pi(m) \models_{rt} \beta$ and $(\forall i < m)(\mathcal{M}, \pi(i) \models_{rt} \alpha)$. By the inductive hypothesis, we conclude that

$$(\exists m \in I)(\mathcal{M}, \pi(m) \models tr(\beta) \text{ and } (\forall i < m)(\mathcal{M}, \pi(i) \models tr(\alpha))) \tag{2}$$

Now consider the following cases:

(a) $I = [a, \infty)$. From (2) we conclude that

$$\mathcal{M}, s \models tr(\alpha) \wedge \underbrace{\mathbf{EX}(tr(\alpha) \wedge \mathbf{EX}(tr(\alpha) \ldots}_{a-1} \wedge \mathbf{E}(tr(\alpha) \mathbf{U} tr(\beta)))).$$

From this, by Proposition 3, it follows that $\mathcal{M}, s \models tr(\mathbf{E}(\alpha \mathbf{U}_{[a,\infty)}\beta))$.

(b) $I = [a, b)$, where $a < b < \infty$. From (2) we conclude that

$$\mathcal{M}, s \models tr(\alpha) \wedge \underbrace{\mathbf{EX}(tr(\alpha) \wedge \mathbf{EX}(tr(\alpha) \wedge \ldots}_{a}$$

$$\wedge \underbrace{\mathbf{EX}(tr(\beta) \vee tr(\alpha) \wedge \mathbf{EX}(tr(\beta) \vee tr(\alpha) \ldots}_{b-a-1} \wedge \mathbf{EX}(tr(\beta)))))))).$$

From this, by Proposition 4, it follows that $\mathcal{M}, s \models tr(\mathbf{E}(\alpha \mathbf{U}_{[a,b)}\beta))$.

□

Lemma 2. *Let $\mathcal{M}$ be a model, and φ an* RTECTL *formula. Then* $(\forall s \in S)(\mathcal{M}, s \models tr(\varphi) \Rightarrow \mathcal{M}, s \models_{rt} \varphi)$.

Proof. We proceed by induction on the length of formulae. Let $s \in S$, and assume that $\mathcal{M}, s \models tr(\varphi)$. Now consider the following cases:

1. $\varphi \in AP$. Then, since $\varphi = tr(\varphi)$, we have that $tr(\varphi) \in AP$. Therefore, $\mathcal{M}, s \models tr(\varphi) \iff tr(\varphi) \in L(s) \iff \varphi \in L(s) \iff \mathcal{M}, s \models_{rt} \varphi$.
2. $\varphi = \neg p$, where $p \in AP$. Then $tr(\varphi) = \varphi$. Therefore, $\mathcal{M}, s \models tr(\varphi) \iff \mathcal{M}, s \models tr(\neg p) \iff p \notin L(s) \iff \mathcal{M}, s \models_{rt} \neg p \iff \mathcal{M}, s \models_{rt} \varphi$.
3. $\varphi = \alpha \wedge \beta$. Then, $tr(\varphi) = tr(\alpha \wedge \beta) = tr(\alpha) \wedge tr(\beta)$. By the definition of the satisfiability relation for ECTL we have that $\mathcal{M}, s \models tr(\alpha)$ and $\mathcal{M}, s \models tr(\beta)$. By the inductive hypothesis, we get $\mathcal{M}, s \models_{rt} \alpha$ and $\mathcal{M}, s \models_{rt} \beta$, and therefore $\mathcal{M}, s \models_{rt} (\alpha \wedge \beta)$.

4. $\varphi = \alpha \vee \beta$. Then, $tr(\varphi) = tr(\alpha \vee \beta) = tr(\alpha) \vee tr(\beta)$. By the definition of the satisfiability relation for ECTL we have that $\mathcal{M}, s \models tr(\alpha)$ or $\mathcal{M}, s \models tr(\beta)$. By the inductive hypothesis, we get $\mathcal{M}, s \models_{rt} \alpha$ or $\mathcal{M}, s \models_{rt} \beta$, and therefore $\mathcal{M}, s \models_{rt} (\alpha \vee \beta)$.
5. $\varphi = \mathbf{EX}\alpha$. Then, $tr(\varphi) = tr(\mathbf{EX}\alpha) = \mathbf{EX}tr(\alpha)$. By the definition of the satisfiability relation we have that there exists $\pi \in \Pi(s)$ such that $\mathcal{M}, \pi(1) \models tr(\alpha)$. By the inductive hypothesis, we conclude that $\mathcal{M}, \pi(1) \models_{rt} \alpha$. Thus, $\mathcal{M}, s \models_{rt} \mathbf{EX}\alpha$, since $\pi(0) = s$.
6. $\varphi = \mathbf{EG}_I\alpha$. Consider the following cases:
   (a) $I = [a, \infty)$. From Proposition 1, we get that $\mathcal{M}, s \models \underbrace{\mathbf{EX} \ldots \mathbf{EX}}_{a}(\mathbf{EG}tr(\alpha))$.
   From this it follows that there exists $\pi \in \Pi(s)$ such that $(\forall m \geqslant a)$ $(\mathcal{M}, \pi(m) \models tr(\alpha))$. By the inductive hypothesis, we conclude that $(\forall m \geqslant a)(\mathcal{M}, \pi(m) \models_{rt} \alpha)$, and therefore $\mathcal{M}, s \models_{rt} \mathbf{EG}_I\alpha$.
   (b) $I = [a, b)$, where $a < b < \infty$. From Proposition 2, we get that
   $$\mathcal{M}, s \models \underbrace{\mathbf{EX} \ldots \mathbf{EX}}_{a}(tr(\alpha) \wedge \mathbf{EX}(tr(\alpha)) \wedge \ldots \wedge \underbrace{\mathbf{EX} \ldots \mathbf{EX}}_{b-a}(tr(\alpha))).$$
   From this it follows that there exists $\pi \in \Pi(s)$ such that $(\forall m \in [a, b))$ $(\mathcal{M}, \pi(m) \models tr(\alpha))$. By the inductive hypothesis, we conclude that $(\forall m \in [a, b))(\mathcal{M}, \pi(m) \models_{rt} \alpha)$. Therefore $\mathcal{M}, s \models_{rt} \mathbf{EG}_I\alpha$.
7. $\varphi = \mathbf{E}(\alpha \mathbf{U}_I \beta)$. Consider the following cases:
   (a) $I = [a, \infty)$. From Proposition 3, we get that $\mathcal{M}, s \models tr(\alpha) \wedge \underbrace{\mathbf{EX}(tr(\alpha) \wedge \mathbf{EX}(tr(\alpha) \ldots}_{a-1} \wedge \mathbf{E}(tr(\alpha)\mathbf{U}tr(\beta)))$. From this it follows that there exists $\pi \in \Pi(s)$ such that $(\forall m < a)(\mathcal{M}, \pi(m) \models tr(\alpha))$ and $(\exists m \geqslant a)(\mathcal{M}, \pi(m) \models \mathbf{E}(tr(\alpha)\mathbf{U}tr(\beta))$ By the inductive hypothesis, we conclude that $(\forall m < a)(\mathcal{M}, \pi(m) \models_{rt} \alpha)$ and $(\exists m \geqslant a)(\mathcal{M}, \pi(m) \models_{rt} \beta)$, and therefore $\mathcal{M}, s \models \mathbf{E}(\alpha \mathbf{U}_I \beta)$.
   (b) $I = [a, b)$, where $a < b < \infty$. From Proposition 4, we get that
   $$\mathcal{M}, s \models tr(\alpha) \wedge \underbrace{\mathbf{EX}(tr(\alpha) \wedge \mathbf{EX}(tr(\alpha) \wedge \ldots}_{a}$$
   $$\wedge \underbrace{\mathbf{EX}(tr(\beta) \vee tr(\alpha) \wedge \mathbf{EX}(tr(\beta) \vee tr(\alpha) \ldots}_{b-a-1} \wedge \mathbf{EX}(tr(\beta)))))))).$$
   From this it follows that there exists $\pi \in \Pi(s)$ such that $(\forall m < a)$ $(\mathcal{M}, \pi(m) \models tr(\alpha))$ and $(\exists b - a - 1 > m > a)(\mathcal{M}, \pi(m) \models tr(\alpha) \vee tr(\beta))$ and $(\exists m < b)(\mathcal{M}, \pi(m) \models tr(\beta))$ By the inductive hypothesis, we conclude that $(\forall m < a)(\mathcal{M}, \pi(m) \models_{rt} \alpha)$ and $(\exists b - a - 1 > m > a)(\mathcal{M}, \pi(m) \models_{rt} \alpha \vee \beta)$ and $(\exists m < b)(\mathcal{M}, \pi(m) \models_{rt} \beta)$, and therefore $\mathcal{M}, s \models \mathbf{E}(\alpha \mathbf{U}_I \beta)$.

□

From the Lemma 1 and Lemma 2 we get the following theorem.

Theorem 1 (Correctness of the translation)**.** *Let $\mathcal{M}$ be a model, and φ an* RTECTL *formula. Then $\mathcal{M} \models_{rt} \varphi$ if, and only if $\mathcal{M} \models tr(\varphi)$.*

4 Bounded Model Checking

The SAT-based Bounded Model Checking (BMC) is a popular model checking technique for the verification of concurrent systems. Given a model $\mathcal{M}$, an existential modal formula φ, and a non-negative bound k, the SAT-based BMC consists in searching for a non-empty set of paths of length k that constitute a witness for the checked property φ. In particular, the BMC algorithms generate a propositional formula which is satisfiable if and only if the mentioned set of paths exist. The propositional formula is usually obtained as a combination of a propositional encoding of the unfolding of the transition relation of the given model, and a propositional encoding of the property in question. If the generated propositional formula is not satisfiable, then k is incremented until either the problem becomes intractable due to the complexity of solving the corresponding SAT instance, or k reaches the upper bound of the bounded model checking problem for the language under consideration.

All the SAT-based BMC, so the one for ECTL and RTECTL as well, are based on so called bounded semantics, which are the base of translations of specifications to the SAT-problem. In definitions of the bounded semantics one needs to represent cycles in models in a special way. To this aim k-paths, i.e., finite paths of length k, and loops are defined. These definitions have evolved over the last decade, and they have had a major impact on the effectiveness of the BMC encodings.

The SAT-based BMC method for ECTL was introduced in [7], and then it was improved in [12]. In this paper we use the definition and implementation of the SAT-based BMC for ECTL that was presented in [12]. The SAT-based BMC method for RTECTL was introduced in [15], and then it was improved in [11]. In this paper we use the definition and implementation of the SAT-based BMC for RTECTL that was presented in [11].

5 Experimental Results

In this section we present a comparison of a performance evaluation of two SAT-based BMC algorithms: for RTECTL [11, 16], and for ECTL [12]. In order to evaluate the behaviour of the algorithms, we have tested it on several RTECTL properties and equivalent ECTL properties.

An evaluation of both BMC algorithms, which have been implemented in C++ is given by means of the running time, the memory used, and the number of generated variables and clauses.

5.1 A Faulty Train Controller System

To evaluate the BMC technique for RTECTL and ECTL we analyse a scalable concurrent system, which is a faulty train controller system (FTC) (adapted from [17]). The system consists of a controller, and n trains (for $n \geq 2$), and it is assumed that each train uses its own circular track for travelling in one direction. All trains have to pass through a tunnel, but because there is only one track in the tunnel, arriving trains cannot use it simultaneously. There are signal lights on both sides of the tunnel, which can be either red or green. All trains notify the controller when they request entry to the tunnel or when they leave the tunnel. The controller controls the colour of the signal lights, however it can

be faulty, and does not fulfil its task. The controller does not ensure the mutual exclusion property: two trains never occupy the tunnel at the same time.

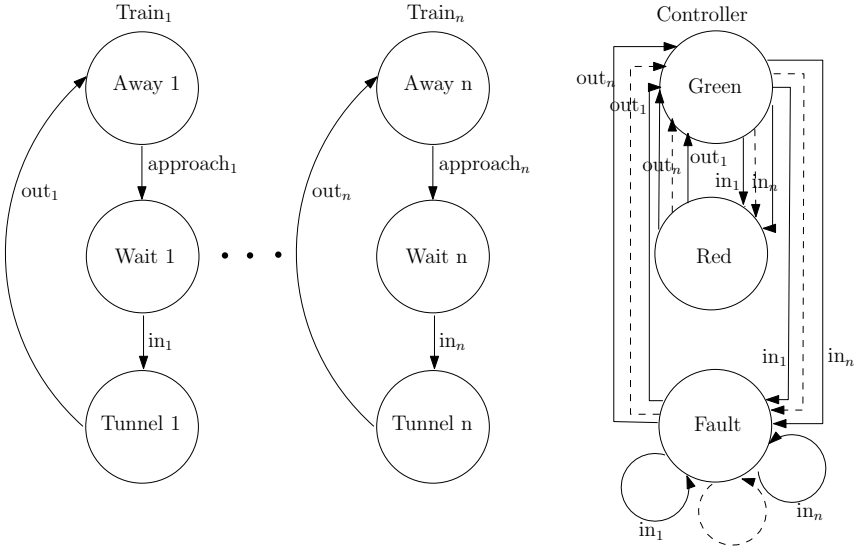


FIGURE 1: A network of automata for train controller system ([16])

An automata model of the FTC system is shown on Figure 1. The specifications for it are given in the existential form, i.e., they are expressed in the RTECTL language:

- $\varphi_1 = \mathbf{EF}_{[0,\infty)}\left(InTunnel_1 \wedge \mathbf{EG}_{[1,n+2)}(\bigwedge_{i=1}^{n} \neg InTunnel_i)\right)$, where n is the number of trains.
- $\varphi_2 = \mathbf{EF}_{[0,\infty)}\left(InTunnel_1 \vee \mathbf{EG}_{[1,n+2)}(\bigvee_{i=1}^{n} \neg InTunnel_i)\right)$, where n is the number of trains.

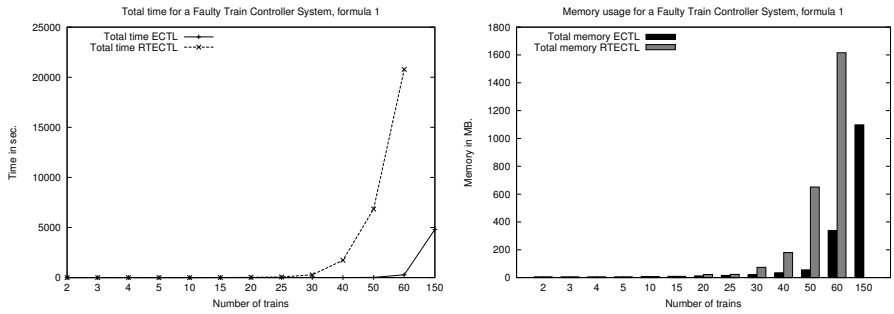
The equivalent ECTL formulae are:

- $tr(\varphi_1) = \mathbf{EF}(InTunnel_1 \wedge \mathbf{EX}((\neg InTunnel_1 \wedge \neg InTunnel_2) \wedge \mathbf{EX}((\neg InTunnel_1 \wedge \neg InTunnel_2) \wedge \mathbf{EX}(\neg InTunnel_1 \wedge \neg InTunnel_2)))$, for $n = 2$.
- $tr(\varphi_2) = \mathbf{EF}(InTunnel_1 \vee \mathbf{EX}((\neg InTunnel_1 \vee \neg InTunnel_2) \wedge \mathbf{EX}((\neg InTunnel_1 \vee \neg InTunnel_2) \wedge \mathbf{EX}(\neg InTunnel_1 \vee \neg InTunnel_2)))$, for $n = 2$.

The formula φ_1 states that there exists the case that Train 1 is in the tunnel and either it and other train will not be in the tunnel during the next $n + 1$ time units. The formula φ_2 expresses that there exists the case that Train 1 is in the tunnel or either it or other train will not be in the tunnel during the next $n + 1$ time units.

In all of the following tables, the amount of time used by BMC and SAT is given in the penultimate column, and maximum of memory usage of BMC and SAT is given in the

last column. In Tables 1 and 2 we present experimental results for the formulae φ_1 and φ_2, respectively. In Figures 2(a) and 2(b) we present a comparison of total time usage and total memory usage for the formulae φ_1.



(a) (b)

FIGURE 2: A comparison of total time usage and total memory usage for the formulae φ_1.

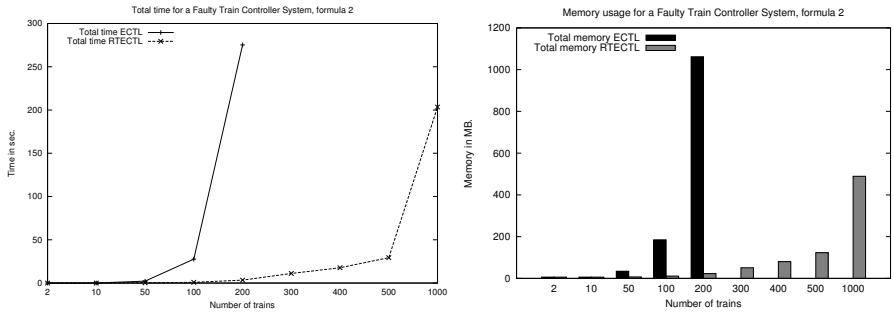
An analysis of experimental results for formula φ_1 leads to the conclusion that BMC for ECTL uses less time and memory comparing to BMC for RTECTL. The reason is that although BMC needs much more paths for verification in case of ECTL, but these paths are significantly shorter.

				BMC		**SAT**		**BMC + SAT**	
Logic	**n**	**k**	**LL**	**sec**	**MB**	**sec**	**MB**	**sec**	**MB**
RTECTL	2	3	2	0.01	1.70	0.00	6.00	0.01	6.00
ECTL	2	2	4	0.01	1.70	0.00	6.00	0.01	6.00
RTECTL	3	4	2	0.01	1.70	0.01	6.00	0.02	6.00
ECTL	3	2	5	0.00	1.70	0.00	6.00	0.00	6.00
RTECTL	4	5	2	0.02	1.83	0.00	6.00	0.03	6.00
ECTL	4	2	6	0.02	1.83	0.00	6.00	0.02	6.00
RTECTL	5	6	2	0.03	1.83	0.05	6.00	0.08	6.00
ECTL	5	2	7	0.02	1.96	0.01	6.00	0.03	6.00
RTECTL	10	11	2	0.28	2.47	0.52	7.00	0.80	7.00
ECTL	10	2	12	0.05	2.60	0.03	7.00	0.08	7.00
RTECTL	15	16	2	0.94	3.63	3.00	9.00	3.94	9.00
ECTL	15	2	17	0.21	3.76	0.12	9.00	0.33	9.00
RTECTL	20	21	2	2.93	5.31	23.78	22.00	26.71	22.00
ECTL	20	2	22	0.35	5.70	0.14	12.00	0.49	12.00
RTECTL	25	26	2	6.06	7.76	56.78	23.00	62.84	23.00
ECTL	25	2	27	0.90	8.26	0.69	16.00	1.60	16.00
RTECTL	30	31	2	11.56	10.98	269.14	74.00	280.70	74.00
ECTL	30	2	32	1.22	11.62	2.70	21.00	3.92	21.00

RTECTL	40	41	2	33.81	20.26	1698.67	180.00	1732.48	180.00
ECTL	40	2	42	2.87	21.16	5.88	35.00	8.74	35.00
RTECTL	50	51	2	102.58	33.93	6738.59	651.00	6841.17	651.00
ECTL	50	2	52	5.11	35.22	14.94	56.00	20.05	56.00
RTECTL	60	61	2	217.43	52.62	20563.46	1616.00	20780.89	1616.00
ECTL	100	2	102	29.58	199.40	247.63	339.00	277.21	339.00
ECTL	150	2	152	100.04	181.10	4709.48	1098.00	4809.52	1098.00

Table 1: Experimental results for formula φ_1 - RTECTL vs. ECTL. k is the bound, LL is the number of k-paths.

In Figures 3(a) and 3(b) we present a comparison of total time usage and total memory usage for the formulae φ_2.



(a) (b)

FIGURE 3: A comparison of total time usage and total memory usage for the formulae φ_2.

An observation of experimental results for formula φ_2 leads to the conclusion that BMC for RTECTL uses less time and memory comparing to BMC for ECTL. The reason is that BMC needs much more paths for verification in case of ECTL.

				BMC		**SAT**		**BMC + SAT**	
Logic	**n**	**k**	**LL**	**sec**	**MB**	**sec**	**MB**	**sec**	**MB**
RTECTL	2	2	2	0.00	1.70	0.00	6.00	0.00	6.00
ECTL	2	1	2	0.00	1.70	0.00	6.00	0.00	6.00
RTECTL	10	2	2	0.01	1.83	0.01	6.00	0.02	6.00
ECTL	10	1	10	0.05	2.21	0.02	6.00	0.07	6.00
RTECTL	50	2	2	0.16	3.12	0.06	7.00	0.23	7.00
ECTL	50	1	50	1.51	20.78	0.67	34.00	2.18	34.00

RTECTL	100	2	2	0.56	5.82	0.26	11.00	0.82	11.00
ECTL	100	1	100	11.20	109.00	16.42	185.00	27.61	185.00
RTECTL	200	2	2	2.36	15.11	0.99	23.00	3.36	23.00
ECTL	200	1	200	81.60	213.50	193.61	1062.00	275.21	1062.00
RTECTL	300	2	2	6.10	29.54	5.00	51.00	11.10	51.00
RTECTL	400	2	2	9.73	49.27	7.97	80.00	17.70	80.00
RTECTL	500	2	2	12.73	74.15	16.63	123.00	29.37	123.00
RTECTL	1000	2	2	54.94	276.40	148.54	489.00	203.47	489.00

Table 2: Experimental results for φ_2 - RTECTL vs. ECTL. k is the bound, LL is the number of k-paths.

5.1.1 Generic Pipeline Paradigm.

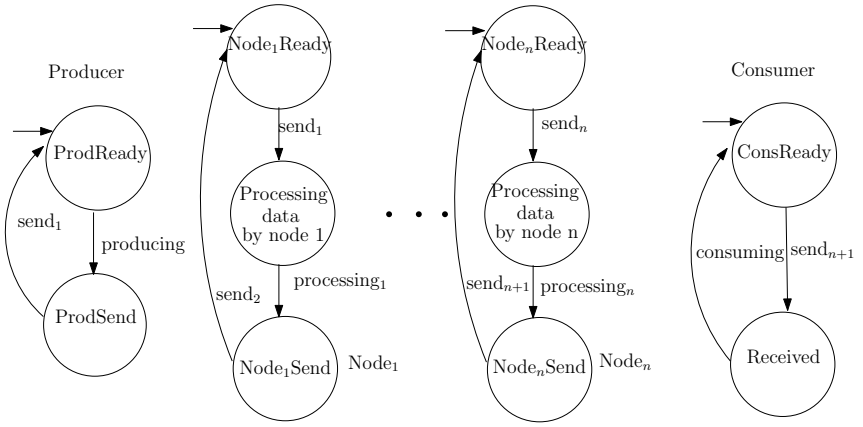


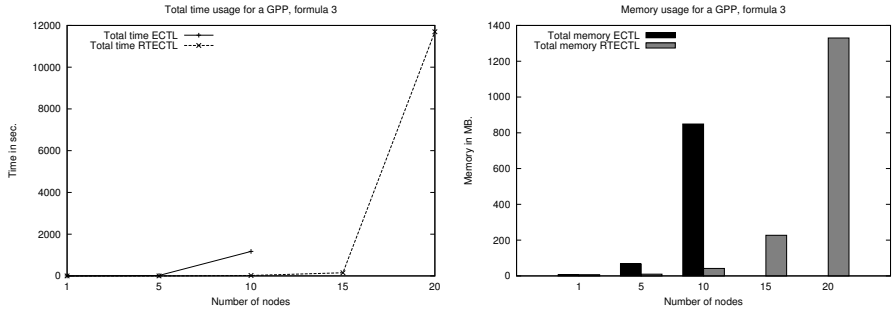
FIGURE 4: The GPP system [11]

The benchmark we consider is a generic pipeline paradigm (GPP) [18], which consists of three parts: Producer producing data, Consumer receiving data, and a chain of n intermediate Nodes that transmit data produced by Producer to Consumer. The local states for each component (Producer, Consumer, and intermediate Nodes), and their protocols are shown on Fig. 4. The comparison of both BMC algorithms for RTECTL and ECTL with respect to the GPP system has been done by means of the following RTECTL specification:
$\varphi_3 = \mathbf{EG}_{[0,\infty)}(\neg ProdSend \vee \mathbf{EF}_{[2n+1,2n+2)}(Received))$, where n is number of nodes.
The equivalent ECTL formula for $n = 2$ is:
$tr(\varphi_3) = \mathbf{EG}(\neg ProdSend \vee \mathbf{EX}(\mathbf{EX}(\mathbf{EX}(\mathbf{EX}(\mathbf{EX}(Received))))))$.
$\varphi_4 = \mathbf{EG}_{[0,n^2+2n+1)}(\neg Received))$, where n is number of nodes.

The equivalent ECTL formula for $n = 2$ is:
$tr(\varphi_4) = \mathbf{EX}(\neg Received \wedge \mathbf{EX}(\neg Received \wedge \mathbf{EX}(\neg Received \wedge \mathbf{EX}(\neg Received \wedge \mathbf{EX}(\neg Received \wedge \mathbf{EX}(\neg Received \wedge \mathbf{EX}(\neg Received \wedge \mathbf{EX}(\neg Received \wedge \neg Received)))))))))$

In Tables 3 and 4 we present experimental results for the formula φ_3 and φ_4, respectively. Also in the case of this benchmark we can observe that time and memory usage depends on number of paths.

In Figures 5(a) and 5(b) we present a comparison of total time usage and total memory usage for the formulae φ_3.



(a) (b)

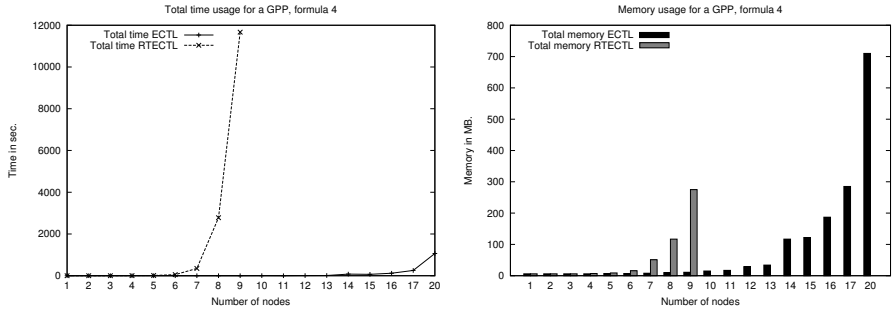
FIGURE 5: A comparison of total time usage and total memory usage for the formulae φ_3.

An observation of experimental results for formula φ_3 leads to the conclusion that BMC for RTECTL uses less time and memory comparing to BMC for ECTL. The reason is that although BMC needs much more paths for verification in case of ECTL.

				BMC		**SAT**		**BMC + SAT**	
Logic	**n**	**k**	**LL**	**sec**	**MB**	**sec**	**MB**	**sec**	**MB**
RTECTL	1	5	7	0.01	1.83	0.00	6.00	0.01	6.00
ECTL	1	5	19	0.04	2.09	0.01	6.00	0.05	6.00
RTECTL	5	13	15	0.89	4.92	0.56	10.00	1.44	10.00
ECTL	5	13	155	10.34	35.73	9.36	67.00	19.70	67.00
RTECTL	10	23	25	10.95	23.74	9.70	42.00	20.65	42.00
ECTL	10	23	505	258.28	445.00	916.79	849.00	1175.07	849.00
RTECTL	15	33	35	55.66	78.27	96.52	227.00	152.18	227.00
RTECTL	20	43	45	234.62	196.00	11461.90	1330.00	11696.52	1330.00

Table 3: Experimental results for formula φ_3 - RTECTL vs. ECTL. k is the bound, LL is the number of k-paths.

In Figures 6(a) and 6(b) we present a comparison of total time usage and total memory usage for the formulae φ_4.



(a) (b)

FIGURE 6: A comparison of total time usage and total memory usage for the formulae φ_4.

An analysis of experimental results for formula φ_4 leads to the conclusion that BMC for ECTL uses less time and memory comparing to BMC for RTECTL. The reason is that although BMC needs much more paths for verification in case of ECTL, but these paths are significantly shorter.

				BMC		**SAT**		**BMC + SAT**	
Logic	**n**	**k**	**LL**	**sec**	**MB**	**sec**	**MB**	**sec**	**MB**
RTECTL	1	4	1	0.00	1.70	0.00	6.00	0.00	6.00
ECTL	1	1	4	0.00	1.70	0.00	6.00	0.00	6.00
RTECTL	2	9	1	0.01	1.70	0.01	6.00	0.02	6.00
ECTL	2	1	9	0.00	1.70	0.00	6.00	0.00	6.00
RTECTL	3	16	1	0.02	1.83	0.28	6.00	0.29	6.00
ECTL	3	1	16	0.00	1.83	0.01	6.00	0.01	6.00
RTECTL	4	25	1	0.29	2.09	1.90	7.00	2.18	7.00
ECTL	4	1	25	0.02	2.09	0.03	6.00	0.05	6.00
RTECTL	5	36	1	0.58	2.34	10.63	9.00	11.20	9.00
ECTL	5	1	36	0.03	2.47	0.03	7.00	0.06	7.00
RTECTL	6	49	1	1.63	2.86	49.99	16.00	51.62	16.00
ECTL	6	1	49	0.07	2.99	0.08	7.00	0.15	7.00
RTECTL	7	64	1	2.97	3.50	346.04	51.00	349.01	51.00
ECTL	7	1	64	0.09	3.76	0.12	8.00	0.21	8.00
RTECTL	8	81	1	5.52	4.41	2775.31	117.00	2780.83	117.00
ECTL	8	1	81	0.15	4.66	0.79	10.00	0.94	10.00
RTECTL	9	100	1	11.55	5.44	11659.16	275.00	11670.71	275.00
ECTL	9	1	100	0.25	5.95	0.64	11.00	0.88	11.00

ECTL	10	1	121	0.32	7.62	1.92	15.00	2.24	15.00
ECTL	11	1	144	0.44	9.56	0.99	17.00	1.43	17.00
ECTL	12	1	169	0.52	12.01	6.97	29.00	7.49	29.00
ECTL	13	1	196	0.66	14.97	9.57	34.00	10.23	34.00
ECTL	14	1	225	0.90	18.46	75.31	117.00	76.21	117.00
ECTL	15	1	256	1.22	22.59	65.80	122.00	67.02	122.00
ECTL	16	1	289	1.46	27.50	116.33	187.00	117.80	187.00
ECTL	17	1	324	1.72	33.06	250.14	285.00	251.86	285.00
ECTL	20	1	441	3.07	55.17	1062.39	710.00	1065.46	710.00

Table 4: Experimental results for formula φ_4 - RTECTL vs. ECTL. k is the bound, LL is the number of k-paths.

6 Conclusions

For the tests we have used a computer equipped with AMD Phenom™ 9550 Quad-Core 2200 MHz processor and 8 GB of RAM, running Ubuntu Linux with kernel version 3.5.0-17-generic. We have used the state of the art SAT-solver MiniSat 2 [19, 20], which is one of the best SAT-solver. MiniSat 2 also has been used in the experimental results in many papers concerning the BMC method, for example [11, 16, 12], and many others.

In this paper we have presented a comparison between the BMC method for ECTL and the BMC method for RTECTL. Moreover, we have presented a correct translation for operator $\mathbf{EG}_I$, which was not defined in [10, 14]. Further, we have tested and compared with each other on to standard benchmarks the BMC translations for RTECTL and ECTL incremented in [11, 12].

Acknowledgements. The author wishes to thank Bożena Woźna-Szcześniak and Andrzej Zbrzezny for valuable comments and numerous suggestions which helped to improve this paper.

References

[1] Clarke, E., Grumberg, O., Peled, D.: Model Checking. MIT Press (1999)

[2] McMillan, K.: Symbolic model checking: An approach to the state explosion problem. Kluwer Academic Publishers (1993)

[3] Bryant, R.: Graph-based algorithms for boolean function manipulation. IEEE Transaction on Computers **35(8)** (1986) 677–691

[4] Lomuscio, A., Penczek, W., Qu, H.: Partial order reduction for model checking interleaved multi-agent systems. In: Proceedings of the 9th International Conference on Autonomous Agents and Multi-Agent systems (AAMAS'2010)., IFAAMAS Press (2010) 659–666

[5] Clarke, E., Emerson, E.A., Sistla, A.P.: Automatic verification of finite state concurrent systems using temporal logic specifications: A practical approach. ACM Transactions on Programming Languages and Systems **8(2)** (1986) 244–263

[6] Biere, A., Cimatti, A., Clarke, E., Zhu, Y.: Symbolic model checking without BDDs. In: Proceedings of the 5th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'99). Volume 1579 of LNCS., Springer-Verlag (1999) 193–207

[7] Penczek, W., Woźna, B., Zbrzezny, A.: Bounded model checking for the universal fragment of ctl. Fundamenta Informaticae **51**(1-2) (2002) 135–156

[8] Woźna, B., Zbrzezny, A.: Checking ACTL* properties of discrete timed automata via bounded model checking. In: Proceedings of the 1st Int. Workshop on Formal Analysis and Modeling of Timed Systems (FORMATS'03). Volume 2791 of LNCS., Springer-Verlag (2004) 18–33

[9] Penczek, W., Lomuscio, A.: Verifying epistemic properties of multi-agent systems via bounded model checking. In: Proceedings of the 2nd Int. Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS'03), ACM (2003) 209–216

[10] Emerson, E.A., Mok, A.K., Sistla, A.P., Srinivasan, J.: Quantitative temporal reasoning. Real-Time Systems **4**(4) (1992) 331–352

[11] Woźna-Szcześniak, B., Zbrzezny, A.M., Zbrzezny, A.: The BMC method for the existential part of RTCTLK and interleaved interpreted systems. In: Proceedings of the 15th Portuguese Conference on Artificial Intelligence (EPIA'2011). Volume 7026 of LNAI., Springer-Verlag (2011) 551–565

[12] Zbrzezny, A.: Improving the translation from ECTL to SAT. Fundamenta Informaticae **85(1-4)** (2008) 513–531

[13] Baier, C., Katoen, J.P.: Principles of model checking. MIT Press (2008)

[14] Campos, S.V.A.: A quantitative approach to the formal verification of real-time systems. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA (1996)

[15] Woźna-Szcześniak, B.: Bounded model checking for the existential part of real-time CTL and knowledge. In: Advances in Software Engineering Techniques - 4th IFIP TC 2 Central and East European Conference on Software Engineering Techniques, CEE-SET'2009. Revised Selected Papers. Volume 7054 of LNCS., Springer (2009) 164–178

[16] Woźna-Szcześniak, B., Zbrzezny, A.M., Zbrzezny, A.: Verifying RTECTL Properties of a Train Controller System. Scientific Issues, Jan Długosz University in Częstochowa, Mathematics **XV** (2011) 153–162

[17] Hoek, W., Wooldridge, M.: Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. Studia Logica **75**(1) (2003) 125–157

[18] Peled, D.: All from one, one for all: On model checking using representatives. In: Proceedings of the 5th Int. Conf. on Computer Aided Verification (CAV'93). Volume 697 of LNCS., Springer-Verlag (1993) 409–423

[19] Eén, N., Sörensson, N. MiniSat http://minisat.se/MiniSat.html.

[20] Eén, N., Sörensson, N.: MiniSat - A SAT Solver with Conflict-Clause Minimization. In: Proceedings of 8th International Conference on Theory and Applications of Satisfiability Testing(SAT'05). LNCS, Springer-Verlag (2005)

ITRIA 2012 | INFORMATION TECHNOLOGIES: RESEARCH AND THEIR INTERDISCIPLINARY APPLICATIONS

TECHNOLOGIE INFORMACYJNE: BADANIA I ICH INTERDYSCYPLINARNE ZASTOSOWANIA

Projekt POKL.04.01.01-00-051/10-00

Rodzaje wsparcia, jakie oferujemy:

Staże krajowe i zagraniczne dla doktorantów i postdoków
Stypendia dla młodych doktorów
Wykłady profesorów wizytujących z Polski i zagranicy

Dodatkowo prowadzimy bezpłatne kursy E-learning, które oferowane są na platformie edukacyjnej Moodle. **Ukończenie kursu gwarantuje dodatkowe punkty ECTS i wzbogacenie ścieżki awansu naukowego.**

Aktualnie trwa rekrutacja na 8 kursów e-learningowych:

Introduction to machine learning and data mining - dr hab. inż. Szymon Jaroszewicz

Introduction to discrete mathematics and algorithms - Dr Konstanty Junosza-Szaniawski, Mgr Paweł Rzążewski

Game and decision theory - Dr Marcin Malawski, Prof. dr hab. Andrzej Wieczorek

Biological inspirations in computational intelligence algorithms - Prof. dr hab. inż. Sławomir T. Wierzchoń

Monte Carlo methods: algorithms & applications

Discrete Optimization

Advanced statistical methods

Java for Science

Computer modeling

Informacje na temat kusrsów: **http://courses.ipipan.edu.pl**

Informacje dotyczące projektu: **http://phd.ipipan.waw.pl**

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

PhD STUDIES

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

The Project is co-financed by the European Union from resources of the European Social Found

ISBN 978-83-63159-04-7
e-ISBN 978-83-63159-05-4