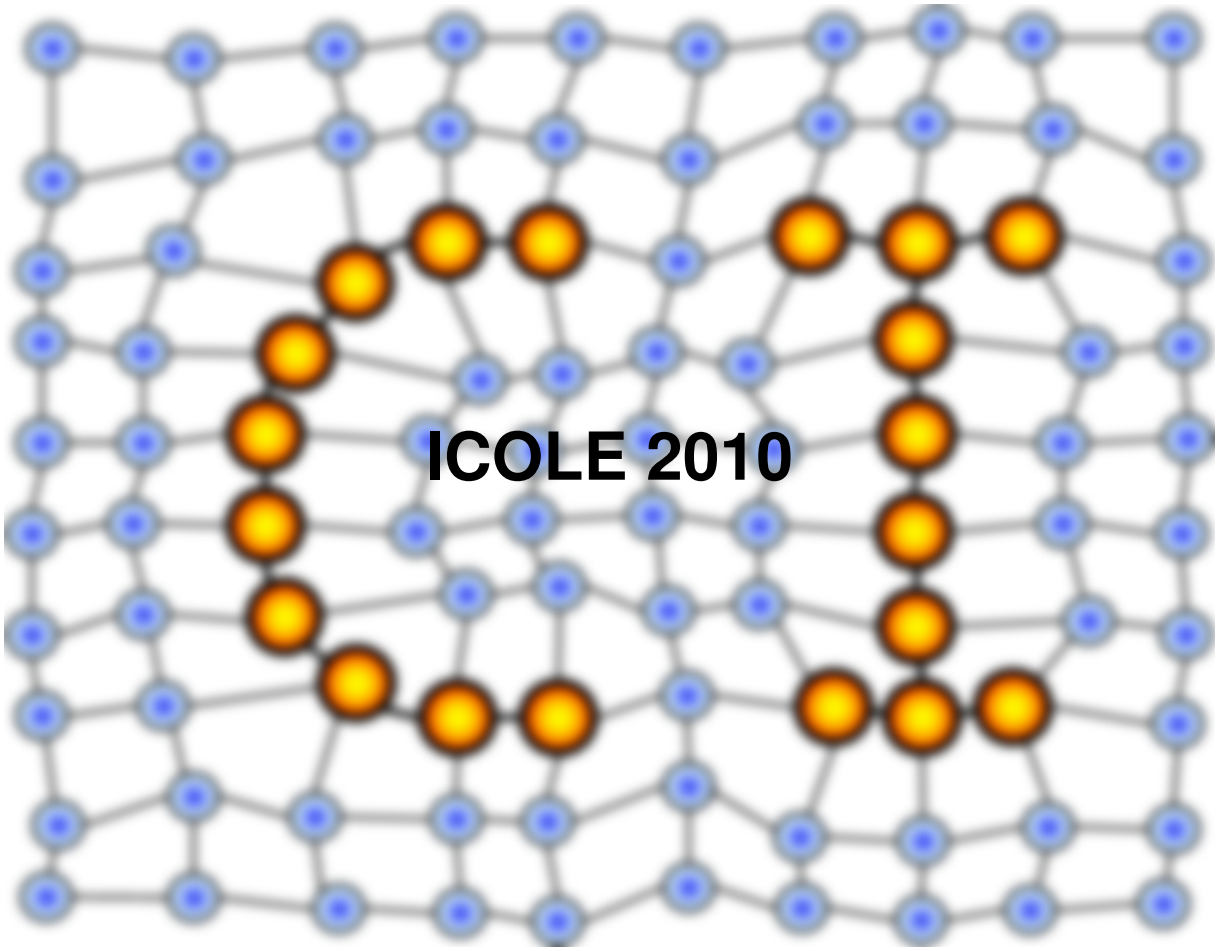
MACHINE LEARNING REPORTS



Report 01/2011
Submitted: 30.11.2010
Published: 31.01.2011

ICOLE 2010, Lessach, Austria

Table of contents

Preview

Jacek Blazewicz*, Klaus Ecker†, Barbara Hammer‡

The annual Polish-German workshop on Computational Biology, Scheduling, and Machine Learning, ICOLE'2010, took place in Lessach, Austria, from 27.9. – 1.10.2010, gathering together twenty-two scientists who are actively involved in the field from different universities including Poznan University, Clausthal University of Technology, University of Applied Sciences Mittweida, Ohio University, and Bielefeld University. The workshop continued the tradition of scientific presentations, vivid discussions, and exchange of novel ideas at the cutting edge of research connected to diverse topics in bioinformatics, scheduling, and machine learning, covering fundamental theoretical aspects, applications, as well as strategic developments in the fields.

This volume contains sixteen extended abstracts accompanying the presentations given at the workshop. The first eight papers deal with current problems in computational biology: The contribution 'A Computational Approach for Extracting Common Signals in Sets of CIS-Regulatory Modules' compares and extends search strategies in DNA sequences for CIS-regulatory motifs, which constitutes an important step to investigate and understand gene regulation, resulting in a promising and sensitive new hybrid search technique. In the contribution 'The exact algorithm and complexity analysis for RNA Partial Degradation Problem' RNA as a fundamental player not only for protein synthesis but also gene expression is investigated with respect to the crucial aspect of its degradation. An abstract computational model is proposed and its biological relevance as well as its theoretical properties e.g. concerning its complexity are investigated. RNA structure also plays a role in the article 'Comparison of RNA structures in torsional angle space' where a new subtle and powerful local structure representation and structure comparison technique for RNA structures is proposed and extensively evaluated using benchmark databases. The next two contributions also address the problem of structure prediction for proteins or RNA, respectively, as a fundamental step to understand the respective function. The approach 'Automated prediction of 3D structure of proteins based on descriptors approach' proposes de novo protein structure prediction techniques based on local structural descriptors. The article 'RNAComposer and the art of composing RNA structures' presents an advanced

*Institute of Computing Science, Poznan University of Technology, Poznan, Poland

†Department of Computer Science, Clausthal University of Technology, Clausthal-Zellerfeld, Germany

‡CITEC centre of excellence, Bielefeld University, Bielefeld, Germany

tool which allows a reliable prediction of RNA structure using a hybrid de novo and comparative approach. The next two contributions address the problem of efficiently assembling a DNA sequence from local structures in the presence of errors. The problem being NP hard, efficient approximation techniques have to be developed, such as hyper heuristics, as investigated in the contribution 'Hyper-heuristic study for the SBH problem'. Further, computational acceleration should be used as far as possible such as e.g parallelization on modern multi-core processors, as investigated in the approach 'Different approaches to parallel computing in the DNA assembly problem.'. As the final contribution in the field of computational biology, the article 'Modeling HCV infection using multi-agent simulation' investigates techniques to reliably model high level biological processes, more specifically viral infections are simulated by means of multi agent technology as compared to differential equations.

A second block of papers centers around simulation and scheduling techniques. Two contributions deal with an interesting project which presents a tool for traffic simulation in cooperation with a major car manufacturer in Poznan. In the first article, 'Effective data representation in traffic simulation and visualization' the basic data structures are discussed, representing the roadways by means of an elegant graphical model. In the second contribution, 'Researching the influence of changes in traffic organization on car factory production', model simulations and the possibilities to built strategic decisions of traffic routing based on simulation results are discussed. The article 'Survey of scheduling of coupled tasks with chains and in-tree precedence constraints' reviews the state of the art of an interesting scheduling problem, the scheduling of coupled tasks. Depending on the nature of the coupling, different complexity results can be derived.

A third block of papers centers around current developments connected to the field of machine learning. The contribution 'The Research Group Theoretical Computer Science at CITEC' gives an overview about typical machine learning problems and applications. One challenging problem of modern data inspection consists in the rapidly increasing size of data sets. This problem is particularly pronounced if data are represented indirectly in terms of dissimilarity matrices since these matrices scale quadratically with the number of data. The contributions 'Patch Affinity Propagation' and 'Relational generative topographic mapping for large data sets' propose two different techniques to get around this problem and investigate their suitability for two different unsupervised data inspection tools. Another challenging issue of modern data mining techniques relies in a reliable formal quantitative evaluation of mostly unsupervised approaches. A few general evaluation measures have been recently proposed in the context of data visualization, but their suitability to evaluate related tasks such as clustering has not been addressed so far. The paper 'Quality Assessment Measures for Dimensionality Reduction Applied on Clustering' presents first steps to investigate this problem. Finally, the contribution 'Functional Relevance Learning in Generalized Learning Vector Quantization' presents an interesting approach how modern data analysis techniques can benefit

from the incorporation of auxiliary structural information, in this case a functional nature of the given data.

Altogether, these contributions demonstrate the lively and fruitful scientific atmosphere caused by the interesting scientific range of the workshop, its international participants, and, last not least, the excellent possibilities offered by Daublebsky's wonderful house in Lessach and its surroundings.

A Computational Approach for Extracting Common Signals in Sets of *Cis*-Regulatory Modules

K. Ecker
Ohio University, Athens, Ohio
University of Technology, Clausthal, Germany

Abstract. In recent years it became apparent that gene regulation is a key issue for understanding the development and functioning of organisms. In a widely accepted regulatory mechanism, transcription factors are able to bind on special places of short length, mostly found up-stream in non-coding areas, thus mediating the gene expression apparatus. One purpose of this article is to analyze pros and cons of particular search strategies for cis-regulatory modules. From this we can learn to what extent the specificity of a search strategy influences the possible outcomes. Each known search method has its inherent limitation and covers specific search aspects. Tools based on different search strategies may nevertheless propose similar modules, which could then be interpreted as a stronger evidence of a practically relevant result. We consequently propose a "consolidation" strategy that analyzes the outputs for common signals with the objective selecting more reliable results than is possible by the individual tools.

1. Introduction

In recent years it became apparent that gene regulation is a key issue for understanding the development and functioning of organisms. In a widely accepted regulatory mechanism, transcription factors are able to bind on transcription factor binding sites (TFBS or motifs, i.e., on special places of short length, mostly found up-stream in non-coding areas), thus mediating the gene expression apparatus [1].

In the attempt to reveal regulation mechanisms, earlier research focused on detecting and identifying binding sites in non-coding areas situated closely to the genes, where it is conjectured that most of the information responsible for gene regulation is located. The basic idea behind is that functional regulatory elements should have been highly conserved during evolution. Also, genes that are expressed or suppressed in similar situations may have similar regulatory mechanisms. From Davidson's book on gene regulatory networks [2] and other sources we learn that it is expected that there exist about 10 times more transcription factor binding sites than genes. Numerous tools for discovering TFBSs have been developed during the last 20 years [3].

Particularly in higher organisms it is known that sets of binding sites often bind multiple transcription factors [4], mediating gene regulation by enhancing or silencing. These multiple TFBSs are called transcription factor binding modules, or *cis*-regulatory modules. In fact, only few modules are experimentally assured so far, and it must be assumed that many more are waiting to be discovered. Experiments for elucidating regulatory structures are costly and time consuming; hence discovering putative modules with the aid of computers is an important step to alleviate the work of the biologist.

The purpose of this article is to analyze pros and cons of particular search strategies for *cis*-regulatory modules. Each known search method has its inherent limitation and covers specific search aspects. We consequently propose a "*consolidation*" strategy that combines the tools in a way that commonly found structures are preferred and deficiencies of the component tools are cancelled out, and thus allow for more reliable results than by the individual tools.

Chapter 2 gives an overview on existing algorithmic approaches for module discovery. In Chapter 3 we propose a novel concept of combining different module discovery tools into a single consolidation tool. Chapter 4 refers to a case study with an application to a benchmark data set that indeed shows a higher success rate than with the single component tools.

2. Coarse Classification of Module Discovery Strategies

With the objective of understanding the gene regulation network, it is essential to separate *cis*-regulatory modules from the background data of the gene sequences. Unfortunately, because of the limited manpower, material, and financial resources, it is almost impossible to solve this problem solely by biological experiments. On the other hand, with the help of computational search methods putative modules can be provided with – hopefully – a high degree of trustworthiness, which can then be tested in biological experiments. Thus, for reducing cost and time it is important to develop computational methods that reliably allow detection and prediction of *cis*-regulatory modules.

A number of computational tools are already available. Some tools do not care about a detailed "fine structure" of the regulatory elements. Other tools try to elucidate the binding sites composing a regulatory module. In our understanding, a detailed view of modules is much better suited to reflect the basic structure underlying a regulatory network. Accordingly we define a module M in a non-coding sequence S as a list of words (or sites) from S, $M = ((p_1, l_1),\ldots, (p_k, l_k))$, where $p_1, \ldots, p_k$ are the – increasingly ordered – start positions of the module words, and $l_1, \ldots, l_k$ are the respective word lengths. The set of words defined in M is denoted by $W(M)$.

The search strategies behind the computational tools can be coarsely classified as follows.

Type (a): The perhaps oldest and most straight forward computational method starts from known TFBSs, chosen from a public TFBS data base such as TRANSFAC [8] or AGRIS [9]. These methods implement a search for clusters of motifs which may then be considered as *cis*-regulatory modules [10, 11 – 24, 27]. Unfortunately nobody knows how complete TFBS data bases are, and it has hence to be expected that many important modules will be missed.

Tybe (b): Alternatively one can start with putative motifs found by computational motif search tools. Many such tools are based on stochastic analyses. Words occurring with unexpectedly high or low frequencies are often involved in biologically functions, and accordingly module tools typically perform a search guided by probabilistic criteria such as log likelihood ratio, information content, *z*-score, or frequencies or probabilities of potential binding sites [6, 21, 26 – 31]. However, the quality of the results not only depends on the module search strategy, but also on the quality of the motif discovery tool. As it is known that motif discovery tools have an average success rate between 40 and 60 % [3], the chance of identifying a real module decreases with higher numbers of motifs in the module. For example, if the chance of a true positive motif is 50% , the chance that a module with four such words is true positive is 0.5^4, or $\approx 6.3\%$.

Type (c): A completely different approach is motivated by the hypothesis that the same or similar word combinations appearing at different places of the non-coding genome may have regulatory function. Such tools consequently perform a comparative search for common module structures in promoter sequences, without assessing known motif data bases or applying motif discovery tools [5, 6, 10, 31, 33, 34]. For example, if the same cluster of words is found in two or more places, with similar word order, they may be considered as rather unusual and are consequently assumed to be associated with regulatory functionality

Comparing the above methods, we see that each method has its obvious practical justification but also shows certain drawbacks. For instance, type (a) method may miss modules containing words not listed in a motif data base. Similarly, type (b) tools will miss modules containing words not identified by the applied motif discovery tool. For example, a module tool using frequency-based word selection will work poorly for modules containing words that are neither under- nor over-represented. Another concern arises in situations where a gene has more than one *cis*-regulatory module, with possibly interleaved word sequences. Though a module can be regarded as a cluster of motif words, the converse is not necessarily true. Methods searching for motif clusters have therefore to be regarded with care because the cluster words may belong to different modules. Modules obtained by comparative methods of type (c) may be more credible because, by definition, they occur at different places of a promoter or in different promoters. On the other hand, modules occurring only once cannot be detected by these methods.

3. Consolidation Strategy

As each search method has its own practical justification and shows advantages and drawbacks, it may be a good idea comparing the results of different tools and, if possible, bringing at least parts of them to an agreement. The motivation for a consolidation tool combining the advantages of different tools is therefore driven by the expectation of identifying a higher percentage of new, yet unknown, modules as is possible by the individual tools. This leads to the question of how to combine the results of the component tools. The problem hereby is, as can also be seen from Klepper's assessment [7] and from our own experiments of the kind mentioned later in Chapter 4, that the modules found by different tools often have little in common.

Suppose there are given two or more tools for discovering *cis*-regulatory modules, T_1 , T_2, … T_{k_0}, each implementing another search strategy. Let $\{S_1, \ldots, S_n\}$ be a set of promoter sequences expected to have some module in common. Each tool T_i, when applied to sequence S_j with particularly chosen search parameters, produces some set of putative modules $R_i(S_j)$. In order to reveal putative *cis*-regulatory modules we analyze the overlaps and differences of the outputs $R_1(S_j), \ldots, R_{k_0}(S_j)$ of the respective tools. When comparing the modules, say M_1 and M_2, we may encounter different situations such as completely different modules with no common word, or some words may overlap, or one module may be contained in the other module, or may even have exactly the same words in the same order. The question is how to treat such different possibilities. Accepting all presented modules for output would not make sense if the objective is to create a better tool. The other extreme, accepting only modules that are commonly found by all tools, will possibly lead to no new results at all. The reality should be somewhere between these extremes.

As supposedly already high quality modules are to be compared, we want to credit situations where modules, showing some degree of similarity, are identified by two or more tools or occur in different non-coding places. In the proposed consolidation approach we take into account the number of pair-wise word matches in the sets $W(M_1)$ and $W(M_2)$. The simplest solution would be by saying "count the common words in the modules M_1 and M_2." As it turns out, this definition is not well defined because there may be, for example, a word in M_1 that has more than one match in M_2 . How many words do they then have in common? The possibility that binding sites are similar but not necessarily identical complicates the situation.

In the following, let M_1 and M_2 be modules chosen from the set $\bigcup_{i=1}^{k_0} \bigcup_{j=1}^{n} R_i(S_j)$ of all modules. For convenience reason we use two different ways for capturing the similarity of words $v \in W(M_1)$ and $w \in W(M_2)$. The first uses the Hamming or edit distance $d(v,w)$. Words v, w are regarded *similar* if $d(v,w) \le d_{max}$, a given upper bound for word distance. To measure the degree of concordance of M_1 and M_2 we define the set of pairs of similar words in $W(M_1) \times W(M_2)$ by

$$M_1 \wedge_{\nu} M_2 := \{ (v,w) \in W(M_1) \times W(M_2) \mid d(v,w) \le d_{max} \} ,$$

The projection $pr_1(M_1 \wedge_{\nu} M_2)$ onto the first component contains all words of M_1 for which a similar word in M_2 exists. $pr_2(M_1 \wedge_{\nu} M_2)$ is defined analogously for M_2 . For later purposes we introduce the abbreviations

$$\nu_1(M_1, M_2) = pr_1(M_1 \wedge_{\nu} M_2) \quad \text{and} \quad \nu_2(M_1, M_2) = pr_2(M_1 \wedge_{\nu} M_2) .$$

Notice that the cardinalities of $\nu_1(M_1, M_2)$ and $\nu_2(M_1, M_2)$ can be different as a word in M_1 can have two or more matching words in M_2 . M_1 (respectively M_2) is considered *admissible for output* if the cardinality ν_1 (respectively ν_2) is not smaller than a given bound ν_{min} .

The second defines the *degree of overlap* of words v and w of respective lengths $l(v)$ and $l(w)$ as $\omega(v,w)/l(v)$ and $\omega(v,w)/l(w)$, where $\omega(v,w)$ is the number of overlapping characters, and introduce

$$OVL(M_1|M_2) = \{\, v \in W(M_1) \mid \exists\, w \in W(M_2)\ ,\ \omega(v,w)/l(v) \geq \omega_{min} \,\}\ ,$$

and, symmetrically,

$$OVL(M_2|M_1) = \{\, w \in W(M_2) \mid \exists\, v \in W(M_1)\ ,\ \omega(v,w)/l(w) \geq \omega_{min} \,\}$$

for the number of words in one set having an overlapping word in the other set. The consolidation method declares module M_1 [resp. M_2] *admissible for output* if $OVL(M_1|M_2)$ [resp. $OVL(M_2|M_1)$] has at least π_{min} elements, which is a given lower bound (an analogous definition is used for M_2).

In the following, we distinguish three scenarios with remarkable coincidences.

Case (a): Two different tools find modules M_1 and M_2 in a sequence S. Let M_1 and M_2 be modules chosen from the set $\bigcup_{i=1}^{k_0} R_i(S)$. In general, tools based on different search methods will find different modules. Therefore, modules with strong similarity may already give strong evidence of biological function. For admitting modules for output we have two criteria at hand: one uses similarity and the other overlap of binding sites. The number of similar words criterion is motivated by the observation that modules with many similar word pairs are rather unusual. Therefore, M_1 and M_2 are admitted for output if $|\nu_1(M_1, M_2)| \geq \nu_{min}^{(a)}$ and $|\nu_2(M_1, M_2)| \geq \nu_{min}^{(a)}$, where $\nu_{min}^{(a)}$ is some given lower bound. A sufficiently large number of overlapping words can also be considered as unusual, and correspondingly M_1 [resp. M_2] is as well admitted for output if $|OVL(M_1|M_2)| \geq \pi_{min}^{(a)}$ [resp. and $|OVL(M_2|M_1)| \geq \pi_{min}^{(a)}$] , where $\pi_{min}^{(a)}$ is a given lower bound. □

Case (b): A tool finds modules with similar words in different sequences. Let M_1 and M_2 be chosen from the set $\bigcup_{j=1}^{n} R_i(S_j)$ of modules found by tool T_i . After counting the number of similar words, M_1 and M_2 are admitted for output if $|\nu_1(M_1, M_2)| \geq \nu_{min}^{(b)}$ and $|\nu_2(M_1, M_2)| \geq \nu_{min}^{(b)}$. The lower bound $\nu_{min}^{(b)}$ can be chosen differently from that in case (a). As in case (a), the acceptance criterion may also use word overlaps, but with an independently chosen lower bound $\pi_{min}^{(b)}$. □

Case (c): Different tools find modules with similar words in different sequences. Finally, let M_1 and M_2 be modules chosen from the total module set, $\bigcup_{i=1}^{k_0} \bigcup_{j=1}^{n} R_i(S_j)$. As in case (b) we can use the similar words criterion or the criterion for accepting modules for output. For flexibility reason, lower bounds $\nu_{min}^{(c)}$ and $\pi_{min}^{(c)}$ are chosen independently of the previous bounds. □

The bounds for maximum word distance and minimum word overlap influence the choice of module candidates admitted for output. With large word distance or small word overlap pairs of admitted modules can have very different appearance. On the other hand, a required high degree of modules similarity implies a low upper bound for the word distance and a large overlap ratio. Regarding the sizes of the word sets $\nu_1(M_1, M_2)$ and $\nu_2(M_1, M_2)$ we can say that the larger they are, the higher the chance that their words belong to a *cis*-regulatory module. The same can be said for the word sets $OVL(M_1|M_2)$ and $OVL(M_1|M_2)$. The corresponding lower bounds of $\nu_{min}^{(a)}$, $\pi_{min}^{(a)}$, $\nu_{min}^{(b)}$, $\pi_{min}^{(b)}$, $\nu_{min}^{(c)}$, and $\pi_{min}^{(c)}$ hence define important criteria for the module selection.

4. Experience

In a first experimental study we considered three module discovery tools, Hierarchical Agglomerative Clustering (HAC), Enumeration Module Discovery (EMD), and CRMODULES (CRM), all developed at the Ohio University [5, 27]. The first two are of search types (a) and (b) and either use TFBS files of known motifs or apply the regulatory genomic analysis package WordSeeker [35]. The third tool performs comparative search of type (c) and is hence apt of any

motif discovery strategy. The tools were applied to the liver benchmark data set [36] which were also used by Klepper et al. in their tool assessment [7]. These data contain a list of experimentally verified *cis*-regulatory modules. For the selection conditions, the distance and overlap bounds are $d_{max} = 0$ and $\omega_{min} = 1$, and a module is accepted if either the same tool finds another module with at least three common words, i.e., $\nu_{min}^{(b)} = 3$ and $\pi_{min}^{(b)} = 3$, or if two different tools find modules with at least two common words, i.e., $\nu_{min}^{(a)} = 2$ and $\pi_{min}^{(a)} = 2$, $\nu_{min}^{(c)} = 2$, and $\pi_{min}^{(c)} = 2$.

Naturally, a tool being able to identify a larger number of known modules will be given higher confidence in its ability of identifying real, yet not experimentally confirmed, modules. In [37] the *success rate* of a tool is measured as the average ratio of correctly predicted and experimentally verified modules. For the liver data, the success rates of EMD, HAC, and CRM are shown in Table 1. For more details we refer to our new consolidation paper [38, in preparation].

Table 1: Success rates of CRM, EMD, HAC, and Consolidation Approach

Tool	CRM	EMD	HAC	Consolidation Approach
Success rate	0.67	0.47	0.45	0.86

References

1 T. Werner (1999). Models for Prediction and Recognition of Eukaryotic Promoters. Mammalian Genome 10, 168-175.

2 Davidson, E. H. (2006). The Regulatory Genome. Gene Regulatory Networks in Development and Evolution. Elsevier.

3 M. Tompa, N. Li, T. L. Bailey, G. M. Church, B. D. Moor, E. Eskin, A. V. Favorov, M. C. Frith, Y. Fu, W. J. Kent, V. J. Makeev, A. A. Mironov, W. S. Noble, G. Pavesi, G. Pesole, M. Regnier, N. Simonis, S. Sinha, G. Thijs, J. V. Helden, M. Vandenbogaert, Z. Wang, C. Workman, C. Ye, and Z. Zhu (2005). Assessing Computational Tools for the Discovery of Transcription Factor Binding Sites. Nature Biotechnology 23 No.1.

4 A. Wagner (1999). Genes Regulated Cooperatively by One or More Transcription Factors and Their Identification in Whole Eukaryotic Genome. Bioinformatics 15 no. 10, 776-784.

5 Ecker, K. and Welch, L. (2009), A concept for ab initio prediction of *cis*-regulatory modules, In Silico Biol. 9, 0024.

6 Li, L., Zhu, Q., He, X., Sinha, S. and H alfon, M. S. (2007). Large-scale analysis of transcriptional *cis*-regulatory modules reveals both common features and distinct subclasses, Genome Biology **8**, R101.

7 Klepper, K., Sandve, G. K., Abul, O., Johansen, J. and Drablos, F. (2008). Assessment of composite discovery methods, BMC Bioinformatics **9**, 123.

8 E. Wingender, E., Dietze, P., Karas H., and Knüppel, R. (1996), TRANSFAC: a database on transcription factors and their DNA binding sites, Nucleic Acid Res. 24, 238-241.

9 Davuluri, R. V., Sun, H., Palaniswamy, S. K., Matthews, N., Molina, C., Kurtz, M. and Grotewold, E. (2003). AGRIS: Arabidopsis Gene Regulatory Information Server, an information resource of Arabidopsis *cis*-regulatory elements and transcription factors, BMC Bioinformatics **4**, 25.

10 Kel, A., Konovalova, T., Waleev, T., Cheremushkin, E., Kel-Margoulis, O. and Wingender, E. (2006). Composite module analyst: a fitness-based tool for identification of transcription factor binding site combinations, Bioinformatics **22**, 1190-1197.

11 Johansson, Ö., Alkema, W., Wasserman, W. W. and Lagergren, J. (2003). Identification of functional clusters of transcription factor binding motifs in genome sequences: the MSCAN algorithm, Bioinformatics **19** Suppl 1, 169-176.

12 Aerts, S., Van Loo, P., Thijs, G., Moreau, Y. and De Moor, B. (2003). Computational detection of cis-regulatory modules, Bioinformatics **19** Suppl 2, 5-14.

13 Frith, M. C., Hansen, U. and Weng, Z. (2001). Detection of cis-element clusters in higher eukaryotic DNA, Bioinformatics **17**, 878-889.

14 Berman, B. P., Nibu, Y., Pfeiffer, B. D., Tomancak, P., Celniker, S. E., Levine, M., Rubin, G. M., and Eisen, M. B. (2002). Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the Drosophila genome. Proc. Natl Acad. Sci. USA **99**, 757-762.

15 Markstein, M., Markstein, P., Markstein, V. and Levine, M. S. (2002). Genome-wide analysis of clustered dorsal binding sites identifies putative target genes in the Drosophila embryo. Proc. Natl Acad. Sci. USA **99**, 763-768.

16 Sinha, S., van Nimwegen, E. and Siggia, E. D. (2003). A probabilistic method to detect regulatory modules. Bioinformatics **19** (Suppl.1), i292-i301.

17 Frith, M. C., Li, M. C. and Weng, Z. (2003). Cluster-Buster: finding dense clusters of motifs in DNA sequences, Nucleic Acids Res. **31**, 3666-3668.

18 Rajewsky, N., Vergassola, M., Gaul, U. and Siggia, E. D. (2002). Computational detection of genomic cis-regulatory modules applied to body patterning in the early Drosophila embryo. BMC Bioinformatics **3**, 30.

19 Rebeiz, M., Reeves, N. L. and Posakony, J. W. (2002). SCORE: a computational approach to the identification of cis-regulatory modules and target genes in whole-genome sequence data. Site clustering over random expectation. Proc. Natl Acad. Sci. USA **99**, 9888-9893.

20 Sharan, R., Ovcharenko, I., Ben-Hur, A. and Karp R. M. (2003). CREME: a framework for identifying cis-regulatory modules in human–mouse conserved segments. Bioinformatics **19** (Suppl. 1), i283-i291.

21 Sosinsky, A., Bonin, C. P., Mann, R. S. and Honig, B. (2003). Target Explorer: an automated tool for the identification of new target genes for a specified set of transcription factors. Nucleic Acids Res. **31**, 3589-3592.

22 Hallikas, O., Palin, K., Sinjushina, N., Rautiainen, R., Partanen, J., Ukkonen, E. and Taipale, J. (2006). Genome-wide prediction of mammalian enhancers based on analysis of transcription-factor binding affinity. Cell **124**, 47-59.

23 Halfon, M. S., Grad, Y., Church, G. M. and Michelson A. M. (2002). Computation-based discovery of related transcriptional regulatory modules and motifs using an experimentally validated combinatorial model. Genome Res. **12**(7), 1019-1028.

24 Philippakis, A. A., He, F. S. and Bulyk, M. L. (2005). Modulefinder: a tool for computational discovery of cis regulatory modules. Pacific Symposium on Biocomputing, pp. 519-530.

25 Bailey TL, Noble WS: Searching for statistically significant regulatory modules. Bioinformatics 2003, 19(Suppl. 2):ii16-ii25.

26 Grad, Y. H., Roth, F. P., Halfon, M. S. and Church, G. M. (2004). Prediction of similarly acting cis-regulatory modules by subsequence profiling and comparative genomics in *Drosophila melanogaster* and *D.pseudoobsuca*, Bioinformatics **20**, 2738-2750.

27 X. Liang (2010).Computational Methods for Cis-Regulatory Module Discovery. MS Thesis, Faculty of the Russ College of Engineering and Technology of Ohio University.

28 Thompson, W., Palumbo, M. J., Wasserman, W. W., Liu, J. S. and Lawrence, Ch. E. (2004). Decoding Human Regulatory Circuits, Genome Res. **14**, 1967-1974.

29 Zhou, Q. and Wong, W. H. (2004). CisModule: de novo discovery of cis-regulatory modules by hierarchical mixture modeling, PNAS **101**, 12114-12119.

30 Gupta, M. and Liu, J. S. (2005). De novo cis-regulatory module elicitation for eukaryotic genomes, PNAS **102**, 7079-7084.

31 Ivan, A., Halfon, M. S. and Sinha, A. (2008). Computational discovery of cis-regulatory modules in *Drosophila* without prior knowledge of motifs, Genome Biology **9**, R22.

32 Olga V. Kel-Margoulis, Alexander E. Kel, Ingmar Reuter, Igor V. Deineko, and Edgar Wingender (2001): TRANSCompel: a database on composite regulatory elements in eukaryotic genes. Nucleic Acids Research. 2002 January 1; 30(1): 332–334.

33 Kantorovitz, M. R., Robinson, G. E. and Sinha, S. (2007). A statistical method for alignment-free comparison of regulatory sequences, Bioinformatics **23** (ISMB/ECCB), i249-i255.

34 Pierstorff, N., Bergman, C. M. and Wiehe, T. (2006). Identifying cis-regulatory modules by combining comparative and compositional analysis of DNA. Bioinformatics **22**, 2858-2864.

35 J. Lichtenberg, M. Alam, T. Bitterman, F. Drews, K. Ecker, L. Elnitski, S. Evans, E.Grotewold, D.Gu, E.Jacox, K.Kurz, S.S.Lee, X.Liang, P.M.Majmudar, P.Morris, C.Nelson, E.Stockinger, J.D.Welch, S.Wyatt, A.Yilmaz, L.R.Welch (2009). Construction of Regulatory Encyclopedias of Genomes: Strategies and Case Studies, Ohio Collaborative Conference 2009, Cleveland, Ohio (USA)

36 W. Krivan and W. Wasserman (2001): A predictive model for regulatory sequences directing liver-specific transcription. Genome Research, 11:1559-1566, doi:10.1101/gr.180601.

37 K. Ecker, X. Liang, L. Welch (2010). A note on the evaluation of computational tools for discovering regulatory elements in genomes. In preparation.

38 R. Bunescu, K. Ecker, X. Liang, L. Welch (2010) A computational method for consolidating the outputs of *cis*-regulatory module discovery tools, in preparation.

The exact algorithm and complexity analysis for RNA Partial Degradation Problem.

Jacek Błażewicz[1,2], Marek Figlerowicz[1],
Marta Kasprzak[1,2], Agnieszka Rybarczyk[1,3]

Acknowledgements: The work has been partially supported by a Ministry of Science and Higher Education grant (N N519 314635).

In the last few years there has been observed the increasing interest in the ribonucleic acids research according to the discovery of the role that RNA molecules play in the biological systems. They do not only take part in the protein synthesis or serve as adaptors translating information encoded in nucleotide sequences but also influence and are involved in gene expression. It was demonstrated that most of them are produced from the larger molecules due to enzyme digestion or spontaneous degradation and play an essential role in the cellular processes. The involvement of RNA in many complex processes requires the existence of highly effective systems controlling its accumulation. In this context, it appears that the mechanisms of degradation are one of the most important factors influencing RNA activity. In this work, we would like to present our recent results concerning the spontaneous degradation of RNA molecules. We report our first attempt to describe this process using the bioinformatics methods. In our model studies we used the model RNA molecules designed in such a way that they should be very unstable, according to the rules developed by Kierzek and coworkers. On the basis of the results of their degradation we should be able to identify the regions of RNA molecules which are weak and the most susceptible to the cleavage. The undertaken biochemical and bioinformatics analyses confirmed the predicted and expected pattern of RNA degradation. Based on the obtained data, we would like also to propose a formulation of a new problem, called RNA Partial Degradation Problem (RNA PDP) and the exact algorithm based on the branch-and-cut idea, capable of reconstructing RNA molecule using results of biochemical analysis of its degradation. We present also laboratory and computational tests results in a case of real and randomly generated data. We also give the strong NP-completeness proof of the decision version of the RNA PDP problem which is equivalent to a non-existence of a polynomial-time exact algorithm for the analyzed problem in question.

[1]Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland
[2]Institute of Bioorganic Chemistry, Polish Academy of Sciences, Noskowskiego 12/14, 61-704 Poznan, Poland
[3]E-mail: arybarczyk@cs.put.poznan.pl

Comparison of RNA structures in torsional angle space

Tomasz Zok[1,2], Marta Szachniuk[1,3], Maciej Antczak[1]

Acknowledgements: This work has been supported by the Polish Ministry of Science and Higher Education [grant NN519314635]

1 Introduction

Structure comparison is a very important issue in many fields of bioscience. A well-known fact is that structural similarity often means functional resemblance. This means that comparing molecules structurally provides us with crucial information about their behaviour. Another fact is that well established measures are useful in the field of molecule structure prediction, where a need for a library of homologues arises. Therefore, it is essential to investigate and seek for measures, hopefully addressing needs specified above.

Structural similarity measures can be global or local. The first methodology aims to answer to what degree are two or more molecules similar. The second one provides methods and algorithms useful in all the situations where more detailed similarity information is needed. In local similarity the important factors are the fragments interesting from researcher's point of view. This may mean fragments of highest similarity, but also those with lowest resemblence. At best, a researcher is provided with a map of similarity regions for the whole structure.

In our previous work [1] we focused on global similarity measure MCQ based on trigonometric representation. The measure was found to be both reliable and fast. Thus it is a good alternative to well known global RMSD. Recently we focused on possible use of trigonometric representation in local similarity measures. The paper is a presentation of results we obtained in this research.

2 Methods

There are several possible representations of 3D structures. The basic one is algebraic representation, with Cartesian coordinates given for every atom in the structure. It is the most common one and widely used in the databases of molecular structures. However it has its limitations, among which are: dependence on structure's spatial features and large memory consumption. In consequence, all algorithms basing on algebraic representation are limited according to the

[1]Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-695 Poznan, Poland
[2]E-Mail: tzok@cs.put.poznan.pl
[3]Institute of Bioorganic Chemistry, Polish Academy of Sciences, Poznan, Poland

mentioned disadvantages. It is thus important to analyse equivalent, alternative representations.

A dihedral or torsional angle is an angle between two planes. These are defined by three successive chemical bonds – i.e. a chain of four atoms. In RNA there are several dihedral angles with high importance for biochemistry. Their values represent folding of structure backbone, each ribose ring and organic base binding. A set of torsion angles calculated for each residue of RNA is called its trigonometric representation.

In [1] we proposed an MCQ measure based on trigonometric representation of RNA. In this measure each residue was described by eight torsion angles: $\alpha, \beta, \gamma, \delta, \varepsilon, \zeta, \chi, P$. To calculate MCQ similarity between structures Q and R with K residues, the following algorithm was used:

1. for $\measuredangle$ in $\{\alpha, \beta, \gamma, \delta, \varepsilon, \zeta, \chi, P\}$ do:
2. for i := 1 to K do:
3. $\Delta := Q(\measuredangle_i) - R(\measuredangle_i)$
4. $\overline{x} := \overline{x} + \cos \triangle$
5. $\overline{y} := \overline{y} + \sin \triangle$
6. $\overline{x} := \frac{\overline{x}}{8K}$
7. $\overline{y} := \frac{\overline{y}}{8K}$
8. $MCQ = \arctan\left(\frac{\overline{y}}{\overline{x}}\right)$

In the above formulation, the third step needs additional clarifications. The angles are circular values, so the difference between them is defined differently then between real, integer or natural numbers. Before executing fourth and fifth steps of the algorithm, it is needed to normalise the difference Δ.

Another problem arises for missing values. It may happen that one structure was acquired incorrectly and some of its atoms' coordinates are not present in the dataset. In consequence some of dihedrals cannot be calculated and the difference between angles in the third step of the algorithm is undefined. In such situations, the difference Δ obtains one of the predefined values:

- π – as a penalty if one structure does not have angle information while the other has got it,
- 0 – as a reward if both structures do not have angle information.

MCQ was found to be good global similarity measure. Thus we decided to encompass its good features in a new measure of local similarity.

We based our research on trigonometric representation as well. Our goal was to provide information about local similarity in the context of each individual residue in two corresponding structures. Contrary to global measures, a process of comparing two structures locally needs to be interactive. These methods and algorithms provide large amounts of data and user input is required to correctly visualise the result. What we wanted to achieve was a map of similarity regions. Using trigonometric representation, we were able to provide such regions mapping for each of eight dihedral angle types and for local-MCQ. The latter

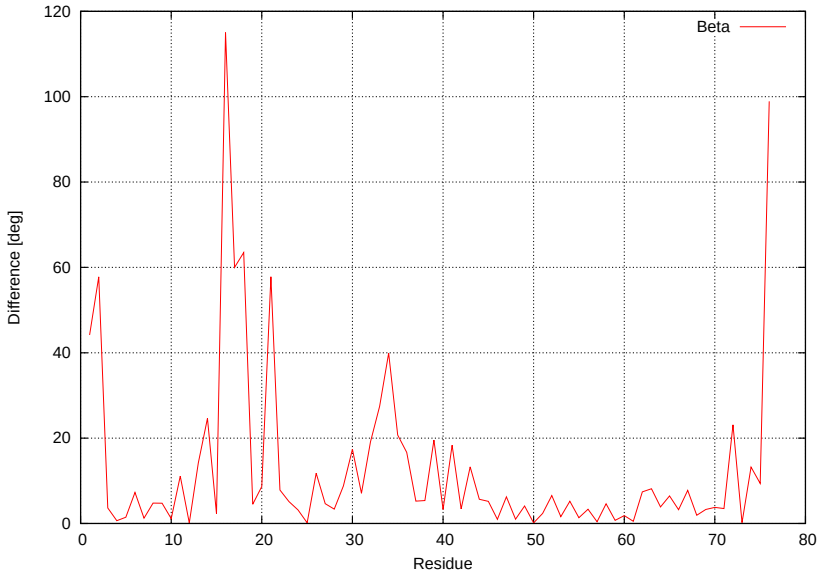


Figure 1: Differences of β dihedral angle for each residue in structures 1EHZ and 1EVV.

is a mean value of eight torsion angles defining a single residue. Thus, for two structures we obtained nine plots for values: $\alpha, \beta, \gamma, \delta, \varepsilon, \zeta, \chi, P, MCQ$. Each plot represents a difference in value of current dihedral angle for every residue. The closer the curve is to x-axis, the more similar are corresponding residues. It is thus a visual and qualitative evaluation of similarity. The plots provide information about:

- the most similiar residues,
- the least similiar residues,
- arguable residues for which there is no clear answer to the question of their resemblence.

In figure 1 we can see a plot for β dihedral angle for each residue in two example structures 1EHZ and 1EVV. Visually it is easy to distiniguish between residues with small difference (very similar) and big difference (very dissimilar). To the latter we can for sure include the beginning and ending residues, which are often dynamic and flexible fragments of RNA structures. However the highest peak on the plot corresponds to the region containing 16-18 residues, what clearly shows, that this is a very dissimilar fragment of the structures under consideration. Analogously to that, we can find that there is a long fragment of residues 45-61 which is very similar. Thanks to our local measure, we can easily point out candidates for deeper analysis.

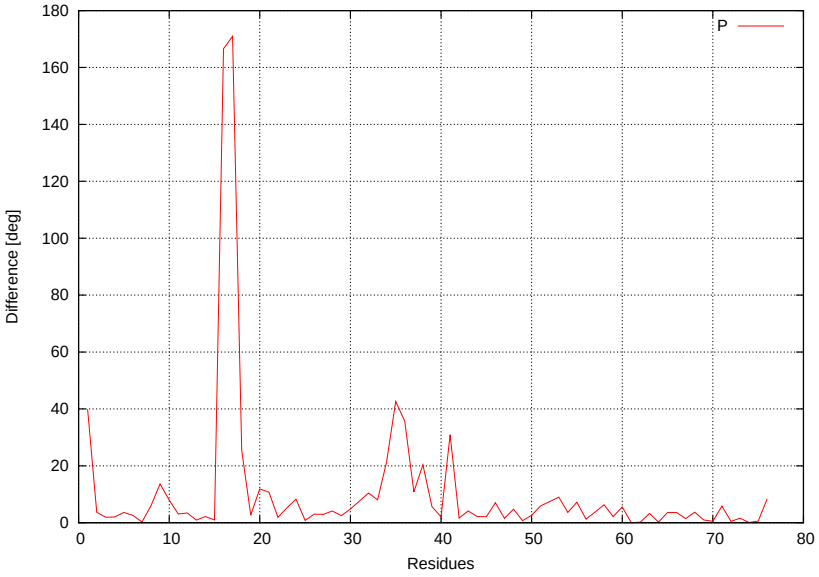


Figure 2: Differences of P dihedral angle for each residue in structures 1EHZ and 1EVV.

The plot for P angle (fig. 2) is more steady meaning that values for the neighbouring residues do not differ as much as for β. P is representing ribose ring folding and so it is less exposed to rapid changes. However this plot confirms our previous findings concerning 45-61 residues similarity and 16-18 residues dissimilarity. Even more, despite steadiness of the curve, the difference for the latter is bigger and completely outstanding in scope of whole plot. This reveals that listed residues are totally different and unaligned. Such conclusions can be drawn only by collation of results from multiple plots.

3 Conclusions

Mean of circular quantities can be applied to all dihedral angles in a structure or iteratively to a subset of them. This means that trigonometric representation is useful in both global and local structure comparison. Our experiments were successful and we created a visualisation tool for the new measure. It allows to plot maps of similarity regions for specified torsional angles which are interactive and helps to determine structure similarity in global as well as point out locations and fragments of low or high resemblance.

References

[1] T. Zok, M. Szachniuk, M. Popenda. Comparison of RNA structures – concepts and measures. *ICOLE: Perspectives of Bioinformatics, Operations Research and Machine Learning, Lessach, Austria.* 43–45, 2008.

Automated prediction of 3D structure of proteins based on descriptors approach

P. Lukasiak [1,2], K. Fidelis [3], M. Antczak [1]*,
A. Kryshtafovych[3], J. Blazewicz [1,2]

[1]Institute of Computing Science, Poznan University of Technology,
ul. Piotrowo 2, 60-965 Poznan, Poland
[2]Institute of Bioorganic Chemistry, Polish Academy of Sciences,
ul. Z. Noskowskiego 12/14, 61-704 Poznan, Poland
[3]Protein Structure Prediction Center, Genome and Biomedical Sciences
Facility, University of California, Davis, USA

Acknowledgements: The work has been partially supported by a Ministry of Science and Higher Education grant (N N519 314635).

1 Introduction

Understanding details of machinery of human organism has been a great challenge for humanity. Proteins are the machinery of life as they are involved in all important processes which occur in an organism. In the last decade the number of identified protein sequences gathered in databases increased tremendously but only for the fraction of them the three dimensional structure is known. Determination of a native folded structure of a particular protein is a key to understand its function. Such a determination is difficult and requires time and money consuming experiments such as crystallography or NMR techniques. Hence, prediction of the secrets of protein structure nature using efficient computer aided modeling techniques is of great interest because progress in that area can generate profits in medicine, chemistry.
Nowadays, homology modeling approaches are the most powerful protein structure prediction methods. One can assume that two proteins, with sufficient amino acid sequence similarity between them and similar function, can be usually considered as homologous. First for a given amino-acid sequence of the unknown protein called target one has to find homologous protein (or usually its part) called template. Next additional information about secondary structure of proteins are applied to improve alignments between target and template in order to obtain better prediction model. However, for new protein folds homologous can not be found in DB. In such case the approaches based on simulation of basic protein driving forces should be applied, in order to solve fold recognition problem, which are called *"ab initio"* or *"de novo methods"*. The native conformation of the protein is the one with significantly lower free energy than others, thus the protein folding process can be defined as the problem of energy function minimization. The energy function usually takes into

*Corresponding author: maciej.antczak@cs.put.poznan.pl

account hydrophobicity, electrostatic potential, non-bonding energy potentials (e.g. Lennard-Jones) and others. Due to the computational complexity of the problem, a protein structure is usually presented in a simplified manner and placed in a simplified space.

1.1 Local descriptors of proteins

The *Structural Classification of Proteins* (SCOP) (Murzin et al. 1995) database is a largely manual classification of protein structural domains based on similarities of their amino acid sequences and three-dimensional structures. It provides a comprehensive ordering of all proteins of known structure according to their evolutionary and structural relationships. The spatial structures of domains from SCOP are stored in *ASTRAL* database (Chandonia et al. 2004). A protein local substructure (descriptor) is a set of several short non-overlapping fragments of the polypeptide chain. Each substructure describes local environment of a particular residue and includes only those segments of the main chain that are located in the proximity of that residue (Kryshtafovych et al. 2003). A detailed description of descriptor construction can be found in (Hvidsten et al. 2003) and finally reorganizing these groups with respect to redundancy, we have created a library of popular geometrical substructures of proteins which is called *"descriptors library"*.

a)

Descriptor	Segment 1		Segment 2		Segment 3		Segment 4		Segment 5	
1qgoa_#8	4-10	ALLVVSF	39-43	FRAFT	63-67	LQALQ	77-83	VAIQSLH	91-95	EKIVR
1qo2a_#78	74-80	EHIQIGG	47-51	IHVVD	67-71	EKLSE	97-101	--RRQIV	89-93	EKLRK
1rpxa_#73	69-75	LPLDVHL	40-44	IHVDV	58-62	LVVDS	93-97	--DIVSV	85-89	PDFIK
1nsj__#82	78-84	NAVQLHG	58-62	GVFVN	66-70	EKILD	98-104	ILVIKAV	89-93	ELCRK
1mla_1#7	3-9	QFAFVFP	87-91	MMAGH	262-266	EYMAA	270-276	EHLYEVG	283-287	GLTKR
1qfja2#108	104-110	PMILIAG	134-138	TIYWG	183-187	TAVLQ	195-201	HDIYIAG	207-211	KIARD
1efvb1#8	4-10	LRVLVAV	119-123	LVLLG	47-51	EEAVR	59-65	KEVIAVS	76-80	RTALA
1iow_1#8	4-10	KIAVLLG	38-42	YFVDP	48-52	TQLKS	56-62	QKVFIAL	70-74	GTLQG
1yaca_#57	53-59	PTILTTS	80-84	PYIAR	97-101	VKAVK	14-20	AVLLVDH	120-124	AFPAL
1ig0a2#188	184-190	ISLLALG	40-44	TLLIL	128-132	TKCVN	216-222	FKLCYMT	200-204	VHSIT

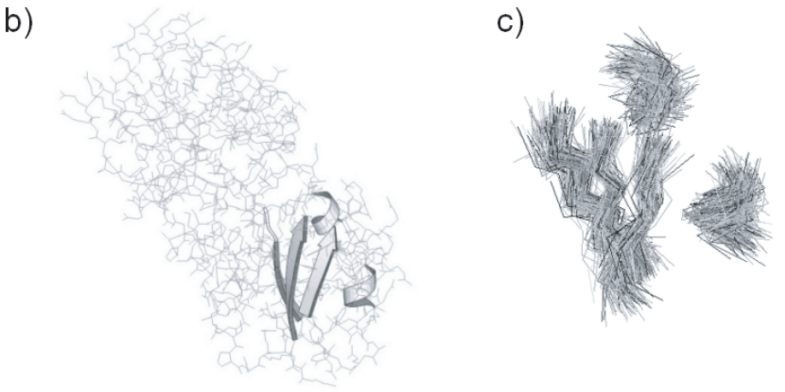


Figure 1: An example of a descriptor group: (a) The first 10 sequence fragment sets (i.e. descriptors) with sufficiently similar structure to the descriptor 1qgoa_#8 (descriptor name syntax: *protein domain_#central amino acid*). The group actually contains 233 descriptors. (b) A ribbon representation of descriptor 1qgoa_#8 showing its secondary structure. (c) The structure of the whole group (i.e. structure of all descriptors in the group).(Kryshtafovych et al. 2003)

A decision on the similarity of descriptors is made by comparing the following parameters: number and length of segments, shape of individual segments, number of geometrically similar segments and the overall fit quality in terms of the RMSD score of their superposition (Hvidsten et al. 2003). The library provides a set of building blocks for protein structures that are common to proteins independent of their global fold.

2 Problem Formulation

Protein tertiary structure prediction, including determination of protein folding pathways, is currently one of the most complex computational problem in the field of protein analysis which remains unsolved. In general the protein folding is a process of a building of a spatial structure from a linear structure of the polypeptide chain. Main aim of our research is to design and implement the automatic ab initio approach which can be used to protein tertiary structure prediction based on only unknown target sequence with using descriptors library. Using this information we can assign specific geometrical conformations to the target protein and, in principle, assemble the protein structure from the local substructures in descriptors library.

3 Method

In this paper, we present the automatic, computational methodology that can be used to protein tertiary structure prediction based on only unknown target sequence. In general the presented approach consists of three main phases:

- descriptor assignments based on target sequence,
- quality control of descriptor assignments,
- design of 3D structure based on descriptor assignments and global structural verification.

Research initialization: Local descriptors library includes descriptors representation based on residue serial number ranges and corresponding sequence of segments without the particular atoms coordinates. For each descriptor an identification tag which reflects information about the domain of its belonging, is assigned according to the ASTRAL nomenclature, as well as the number of the central residue (e.g. 1e43a2#231 is the descriptor from protein 1e43, chain a, domain 2 with origin at residue number 231). Designed database needed for our approach is a composition of the ASTRAL database in version 1.75 and the actual version of local descriptors library extended by additional entities which are used during machine learning discrimator designing.
Descriptors assignments: The main goal is to find the descriptors assignments to the unknown, target sequence defined as input. One can distinguished following solution components:

- sequence profiles generation based on the target sequence,
- sequence profiles or hidden markov models generation based on the aligned sequences of descriptors forming each descriptor group,
- descriptor group – target sequence assignments making,
- descriptor assignments confidence/probability evaluation.

For sequence similarity based assignment techniques one used FragHMMent (Bjrkholm et al. 2009) approach. As a result a confidence ranked list of descriptor to target sequence assignments was obtained. In order to minimize the complexity of profiles generation processing for groups with many descriptors included the most representative geometries within a group was defined.
Descriptors assignments filtering: During research following test sets were defined:

1. Native structures of descriptors.
2. Reconstruction of descriptors themselves. Strip a descriptor of its sidechains, reassign the same sidechains and compute the scoring function result. This should produce correct rotameric states.
3. Reconstruction of descriptor sidechains using backbones from other descriptors in the same group. Success in doing this could be reported as a function of the RMSD difference between the descriptor being reconstructed and the descriptor backbone being used.
4. Takes two descriptors from different descriptor groups. Verify that structures do not match (select according to high RMSD value of backbones superposition). Swap seqeunces between descriptors. Reconstruct the descriptor and compute the scoring function result.
5. Generate random strings of sequence from real proteins. Swap for desriptor sequences. Reconstruct the descriptor and compute the scoring function result.
6. Generate altogether random sequences from the amino-acids alphabet. Swap for descriptor sequences. Reconstruct the descriptor and compute the scoring function result.

In the test sets 1–3 descriptor reconstruction should produce a favorable scoring function result. In the test sets 4–6 descriptor reconstruction should produce an unfavorable scoring function result.

Additionally the side-chain refinement method was proposed with using Monte Carlo simulation (Metropolis algorithm (Metropolis et al. 1953)) in the dihedral angles space of particular descriptor rotameric configuration. We considered as energy function Lennard-Jones (LJ) (Brooks et al. 1983) potential which is also used in SCWRL (Canutescu et al. 2003) software.
Molecular mechanics (LJ, Coulomb's law), statistical knowledge-based (DFIRE) (Hongyi & Yaoqi 2002) potentials were used in order to rate and classify the results of test sets in two general classes (positive, negative examples). To improve the quality of classification a novel potential was proposed, called *congruency score*, which describes the atoms density (packing degree) of all superimposed descriptors in the particular group. The most important task was development of a machine learning discriminator which can be used to structural classification of descriptor assignments from corresponding pair of test sets (positive–negative examples). The cost function is induced based on results of classification experiment with using support vector machine classifier.

Final structure assembling & global structural verification: The final structure model of the unknown protein should be build with using structure consistency between descriptor to target sequence assignments stored on the confidence ranked list obtained from previous phase. The proposed solution consists of following steps:

- hierarchical construction method of putative extended local structures based on assignment confidences and overlaps with the starting descriptor assignment,
- structure extension approach based on the degree of segment overlap, backbone consistency between segments, and possibly rotameric consistency in the overlapping region of structure,

- consensus method to compare and choose the best extended structure from many structures obtained from multiple start points in the current level of assembling.

In other words the iterative local structure extension method should be designed which will proceed hierarchically starting with the strongest assignment. The overlap quality measure computed between the descriptor assignments may be based both on sequence and structure based features. It is necessary to devise assembly algorithms that cope with low quality assignments, including (a) elimination of erroneously assigned segments, (b) generation consensus alignments, (c) alignments based on contact consensus, and (d) Molecular Dynamics refinements with constraints that are generated with consensus alignments. Finally the Modeller can be used to extend the core model with loops. As a result the final protein structure will be obtained in PDB format.

4 Implementation and Tests

Data: We used the ASTRAL (version: ASTRAL SCOP 1.75) database with less than 40% sequence identity to each other have for generating descriptors and fold-oriented groups. Groups with fewer than ten descriptors or fewer than three segments were not used. The descriptors obtained from library which spatial coordinates based on ASTRAL SCOP 1.75 are subsequently referred to as the training set.
Test sets: To train the machine learning discriminator, we tested it on a set of descriptors assignments represented in the group (positive test) and a set of randomly drawn descriptors assignments not in the group (negative test) We also required that the negative test only included descriptors assignments from other folds than the fold associated with the group. Although the main rule was to have the same number of proteins in both sets, too few negative test proteins might result in signals that work well for these descriptors assignments, but do not generalize to unseen cases (is often referred to as overfitting).
Experiment & results performance: Experiment was conducted according to ten-fold cross validation approach and with using the WEKA machine learning software. We compute following statistical measures to evaluate the performance of machine learning approach.

- accuracy is the percentage of descriptor assignments obtained from possitive and negative test set class that are classified properly to all analysed cases,
- sensitivity is the percentage of descriptor assignements from positive test set class according to all descriptor assignments classified as correct,
- specificity is the percentage of descriptor assignements from negative test set class according to all descriptor assignments classified as incorrect.

The example of side-chain refinement simulation results based on 9 descriptors, are presented in table 1.

The partial results, describing the quality of machine learning classification for corresponding test sets, are presented in tables 2, 3, 4 (where LJ – Lennard-Jones; CL – Coulomb's law):

Descriptors	SCWRL	Metropolis
1a9xa6#a927.pdb	13.52	7.85
1dvpa1#a90.pdb	4.66	0.00
1h5qa_#a17.pdb	2.47	0.00
1k8kc_#c76.pdb	20.99	13.75
1muma_#a157.pdb	13.68	6.46
1pgua2#a429.pdb	4.89	0.00
1qqga2#a238.pdb	18.98	3.18
1wmza_#a129.pdb	4.69	0.00
2pbea1#a167.pdb	23.94	8.16

Table 1: Descriptors LJ energy comparison before (SCWRL) and after side-chain refinement simulation (Metropolis MC).

Potential	Quality	Accuracy	Sensitivity	Specificity
LJ	89.70%	0.90	0.89	0.90

Table 2: Classification quality between 1–6 test sets

Potential	Quality	Accuracy	Sensitivity	Specificity
LJ	88.61%	0.89	0.87	0.90
DFIRE	89.71%	0.90	0.89	0.90
CL	84.13%	0.84	0.83	0.86

Table 3: Classification quality between 2–5 test sets

Potential	Quality	Accuracy	Sensitivity	Specificity
LJ	88.61%	0.89	0.87	0.90
DFIRE	89.71%	0.90	0.89	0.90
CL	84.13%	0.84	0.83	0.86

Table 4: Classification quality between 3–4 test sets

5 Conclusion

In this paper a new method of protein tertiary structure prediction has been proposed. Partial results show the proposed approach to be promising but our research is still under construction. The solution of the protein folding problem requires an accurate potential that describes the interactions among different amino acid residues. The potential that would yield a complete understanding of the folding phenomena should be derived from the laws of physics. It can be seen that the side-chain refinement sampling is efficient. It quickly gets close to the target, and then remains in its attraction basin. The machine learning discrimiator achieved an average accuracy of 75% of correctly assigned descriptor assignements to the target sequence but the results can be improved when all test sets will be used during training phase. The conglomerate of introduced potentials should be taken into consideration during side-chain refinement simulation or machine learning dicriminator improvement. Finally assembly, global structural verification phase should be designed and developed in order to verify the quality of presented approach.

References

Bjrkholm, P., Daniluk, P., Kryshtafovych, A., Fidelis, K., Andersson, R. & Hvidsten, T. R. (2009), 'Using multi-data hidden markov models trained on local neighborhoods of protein structure to predict residue-residue contacts', *Boinformatics* **25**, 1264–1270.

Brooks, B., Bruccoleri, R., Olafson, B., States, D., Swaminathan, S. & Karplus, M. (1983), 'Charmm: a program for macromolecular energy, minimization, and dynamics calculations', *J. Comput. Chem.* **4**, 187–217.

Canutescu, A. A., Shelenkov, A. A. & Dunbrack, R. L. (2003), 'A graph-theory algorithm for rapid protein side-chain prediction', *Protein Science* pp. 2001–2014.

Chandonia, J., Hon, G., Walker, N., Conte, L. L., Koehl, P., Levitt, M. & Brenner, S. (2004), 'The astral compendium in 2004', *Nucleic Acids Research* **32**, 189–192.

Hongyi, Z. & Yaoqi, Z. (2002), 'Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction', *Protein Science* pp. 2714–2726.

Hvidsten, T. R., Kryshtafovych, A., Komorowski, J. & Fidelis, K. (2003), 'Anovel approach to fold recognition using sequence-derived properties from sets of structurally similar local fragments of proteins', *Bioinformatics* **19**, ii81–ii91.

Kryshtafovych, A., Hvidsten, T. R., Komorowski, J. & Fidelis, K. (2003), 'Fold recognition using sequence fingerprints of protein local substructures', *IEEE Computer Society Bioinformatics Conference* pp. 517–519.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. & Teller, E. (1953), 'Equation of state calculations by fast computing machines', *J. Chem. Phys.* **21**, 1087–1092.

Murzin, A. G., Brenner, S. E., Hubbard, T. & Chothia, C. (1995), 'Scop: a structural classification of proteins database for the investigation of sequences and structures', *J. Mol. Biol.* **247**, 536–540.

RNAComposer and the art of composing RNA structures

Marta Szachniuk[1,2,3] , Marisz Popenda[2], Maciej Antczak[3], Katarzyna J. Purzycka[2], Piotr Lukasiak[2,3], Natalia Bartol[3], Jacek Błażewicz[2,3], Ryszard W. Adamiak[2]

Acknowedgements: The work has been supported by the Polish Ministry of Science and Higher Education [grant PBZ-MNiSW-07/I/2007/01 to RWA, grant NN 519314635 to JB].

Introduction

The knowledge of three-dimensional structures of RNA molecules is the clue to understanding their biochemical functions. Thus, a great stress has been always imposed on the development of technologies which provide structural information. Among them, X-ray crystallography and NMR spectroscopy play a crucial role. These two have delivered the majority of RNA structures deposited in the Protein Data Bank [1] and the Nucleic Acid Database [2]. However, the amount of solved three dimensional structures of RNAs is far behind the increasing number of known RNA sequences. This fact in conjunction with the quest to answer perennial questions about structure–function relationship has given rise to the new, computational methods for molecular structure modeling.

There are two general approaches to the three-dimensional structure prediction by computational methods: *de novo* prediction and comparative (homology) modeling. The first one builds molecular models based on the thermodynamic hypothesis stating that the native structure of a molecule corresponds to the global minimum of its free energy. Following this physical principle, *de novo* predicting methods simulate the folding process by computing conformational changes and searching for the free-energy minimum. The second approach has resulted from an observation that evolutionarily related (homologous) RNA molecules usually adopt similar structure. Comparative methods following this approach construct the model of target RNA using template structure from homologous molecule [3].

The problem of RNA tertiary structure prediction has remained untouched until the late 90-s of the XX century. The first semi-automated systems for comparative modeling of RNAs have been released in 1998 [4][5]. The preliminary *de novo* solutions have appeared in 2006 [6]. To the best of our knowledge, the following systems for RNA structure prediction have been available as of October 2010: MANIP, PARADISE, ERNA-3D, NAST, C2A, MC-Fold/MC-Sym and ModeRNA for comparative modeling, YUP, NAB, RosettaRNA and iFoldRNA for *de novo* prediction.

[1] E-mail to the corresponding author: Marta.Szachniuk@cs.put.poznan.pl
[2] Institute of Bioorganic Chemistry, Polish Academy of Sciences, Poznan, Poland
[3] Institute of Computing Science, Poznan University of Technology, Poznan, Poland

Methods and tests

Here, we present RNAComposer - a new system for RNA tertiary structure modeling. Its general idea has been introduced in 2006 [7] and combines comparative modeling and *de novo* approach. RNAComposer has been designed for large-scale fully automated modeling of RNA structures. It is based on fragment matching and assembly and uses our own database of RNA fragments named RNA FRABASE [8][9]. The modeling process starts from the user-defined, preferentially experimentally adjusted, RNA secondary structures. In general, the tertiary model is composed in the following steps: (i) the secondary structure is cut into pieces, (ii) the database of RNA three-dimensional fragments is searched for fragments matching the pieces from the previous step, (iii) the best fragments are selected on the basis of the predefined criteria, (iv) *de novo* prediction algorithm is launched for the unmatched pieces, (v) three-dimensional fragments are merged to compose the whole structure, (vi) the model is optimized according to its energy and stereochemistry. RNAComposer can generate either one or a set of 3D models with atomic resolution. The number of generated models depends on the mode selected by the user. Two running modes are available in RNAComposer server: an interactive mode, in which user is provided with the single output structure and a batch mode, in which up to 1000 models can be predicted at a time. Theoretically, the length of the input RNA sequence for modeling is unlimited. However, in order to ensure the effectiveness of the computation and due to the optimization complexity, we have set the limit to 500 nucleotides.

RNAComposer can build high resolution models of large molecules in a very short time. Its performance has been compared to two other, fully automated tools: MC-Fold/MC-Sym and iFoldRNA. Selected results of computational tests are shown in Figure 1 and Figure 2. We have analyzed computation time (Figure 1) of the systems as well as the RMSD of the output models (Figure 2). Since the test set included the instances with already known tertiary structures, the global RMSD in each case has been computed between the predicted model and the original structure (PDB identifier is provided for each example – see Figures 1, 2).

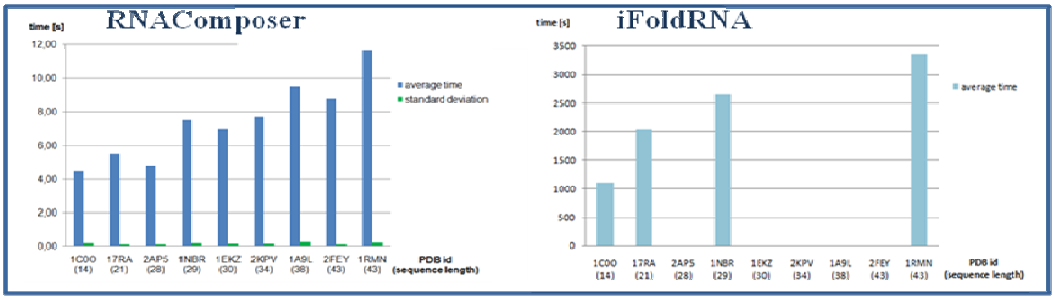


Figure 1. Time of 3D RNA structure computation by RNAComposer and iFoldRNA.

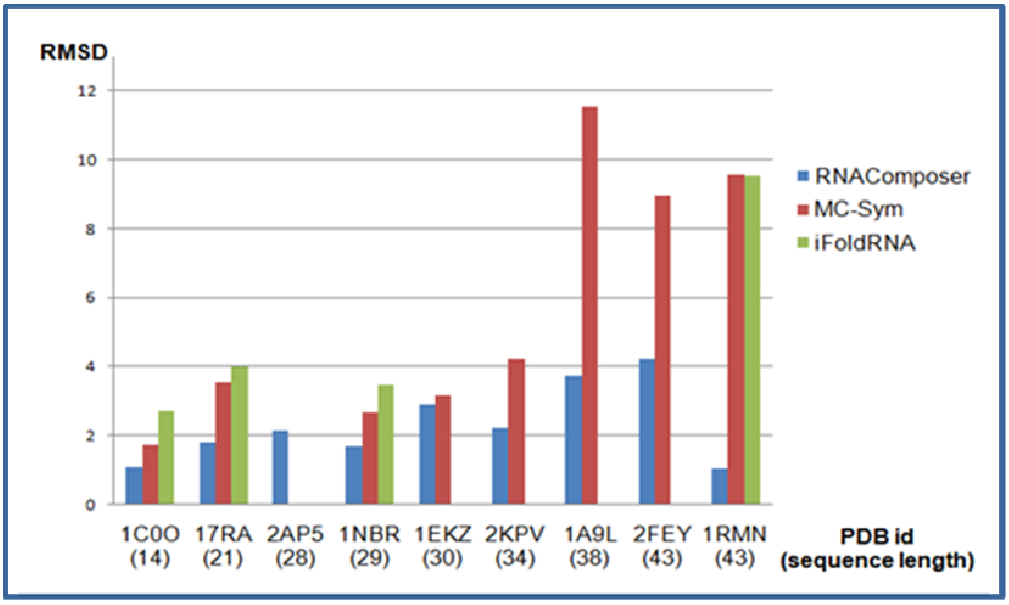


Figure 2. Global RMSD of the models predicted by RNAComposer, MC-Fold/MC-Sym and iFoldRNA.

Summary

We have presented RNAComposer, a new system for RNA tertiary structure modeling. The results of first computational tests have been performed to compare our tool with the existing ones. The tests show the superiority of RNAComposer over the other tools in both, time of computation and the quality of prediction. We believe that RNAComposer will greatly facilitate an analysis of RNA structures.

References

[1] H.M. Berman, J. Westbook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne. The Protein Data Bank. *Nucleic Acids Res*, 28:235-242, 2000.

[2] H.M. Berman, J. Westbook, Z. Feng, L. Iype, B. Schneider, C. Zardecki. The nucleic acid database. *Methods Biochem Anal*, 44:199-216, 2003.

[3] J.M. Bujnicki. Protein-Structure Prediction by Recombination of Fragments. *ChemBioChem*, 7:19-27, 2006.

[4] C. Massire, E. Westhof. MANIP: an interactive tool for modelling RNA. *J Mol Graph Model*, 16:197-205, 255–257, 1998.

[5] I. Tanaka, A. Nakagawa, H. Hosaka, S. Wakatsuki, F. Mueller, R. Brimacombe. Matching the crystallographic structure of ribosomal protein S7 to a three-dimensional model of the 16S ribosomal RNA. RNA. 4:542–550, 1998.

[6] R.K.Z. Tan, A.S. Petrov, S.C. Harvey. YUP: A Molecular Simulation Program for Coarse-Grained and Multiscaled Models. *J Chem Theory Comput*, 2:529–540, 2006.
[7] M. Popenda, L. Bielecki, R.W. Adamiak. High-throughput method for the prediction of low-resolution, three-dimensional RNA structures. *Nucleic Acids Symp Ser*, 50:67-68, 2006.
[8] M. Popenda, M. Blazewicz, M. Szachniuk, R.W. Adamiak. RNA FRABASE version 1.0: an engine with a database to search for the three-dimensional fragments within RNA structures. *Nucleic Acids Res*, 36:D386-D391, 2008.
[9] M. Popenda, M. Szachniuk, M. Blazewicz, S. Wasik, E.K. Burke, J. Blazewicz, R.W. Adamiak. RNA FRABASE 2.0: an advanced web-accessible database with the capacity to search the three-dimensional fragments within RNA structures. *BMC Bioinformatics*, 11: 231, 2010.

Hyper-heuristic study for the SBH problem

Aleksandra Świercz[0,1,2], Wojciech Mruczkiewicz[1], Jacek Blażewicz[1,2], Edmund K. Burke[3], Graham Kendall[3], Ceyda Oğuz [4]

Acknowledgements: The work has been partially supported by a Ministry of Science and Higher Education grant (N N519 314635).

1 Introduction

Sequencing by hybridization (SBH) is one of the methods of recognizing DNA sequences [5]. The method is composed of two phases: the first one, in which all the subfragments of an unknown sequence (the spectrum) are determined in the biochemical experiment on a microarray, and the computational phase, in which these subfragments are combined together with combinatorial algorithms in order to reconstruct the sequence. In the ideal biochemical experiment, all the subfragments of an unknown sequence are in the spectrum. In the real-life situation a spectrum may contain some additional subfragments, which do not occur in the sequence. They are called positive errors. On the other hand, some subfragments of a DNA sequence can be missing in spectrum and those errors are called negative. As the input to the computational phase one gets a spectrum of subfragments, and length n of an unknown sequence. The goal is to find a sequence not greater than n which is composed of the most number of spectrum elements. The process of reconstructing a DNA sequence in case of errors in spectrum is computationally hard.

In this paper we investigate the use of hyper-heuristic methodologies for solving the sequencing by hybridization problem. A hyper-heuristic approach operates on the heuristic search space, rather than operating on a direct representation of the problem. At each decision point, the hyper-heuristic decides which heuristic to execute from the set of those available [2]. A key aim of hyper-heuristic approaches is to enable the search methodology to operate across different problem instances, or even different problem domains, without having to manually adapt the search algorithm.

In the next section studied hyper-heuristics are described, while in the following section the designed set of low-level heuristics for the SBH problem is presented.

[0]E-mail: Aleksandra.Swiercz@cs.put.poznan.pl
[1]Institute of Computing Science, Poznań University of Technology, Poznań, Poland
[2]Institute of Bioorganic Chemistry, Polish Academy of Sciences, Poznań, Poland
[3]School of Computer Science, University of Nottingham, Nottingham, UK
[4]Department of Industrial Engineering, Koç University, Istanbul, Turkey

2 Hyper-heuristic study

The hyper-heuristics which were analyzed are derived from three different heuristic approaches – Choice Function, Tabu Search and Simulated Annealing. All of them are composed of two phases. The first phase is a selection of low-level heuristics. The second one is the acceptance phase, when it is decided whether the new solution obtained from the application of the selected low-level heuristic should be accepted or rejected. In the procedure the set of low-level heuristics and an initial solution is given on input. During each turn the hyper-heuristic selects one low-level heuristic and uses it to obtain new solution when accepted. If selected heuristic is rejected by the acceptance procedure then the current solution remains unchanged.

At the first phase (selection), the computation of a new solution can be very expensive and various techniques try to optimize the number of applied heuristics during each turn. The first hyper-heuristic was the *choice function*(CF) [3], which is a weighted sum of three functions f_1, f_2 and f_3 to measure and rank low-level heuristics. Function f_1 for any low-level heuristic h evaluates the change of the value of the old solution to a new solution value, obtained after applying h. Function f_2 evaluates the ratio of the change in the solution when applying a pair of heuristics. The goal of function f_2 is to find and promote a co-operational behavior of two low-level heuristics. Function f_3 is equal to the amount of time passed since heuristic h was used the last time. Functions f_1 and f_2 are designed to intensify search. Function f_3 introduces an element of diversification. The CF ranks all available low-level heuristics and selects one according to one of the selection methods: straight choice, ranked choice, roulette choice and decomp choice [3]. Every solution obtained here is accepted in the acceptance phase.

Tabu search hyper-heuristic is based on classical *tabu search* method. All low-level heuristics initially have a rank equal to zero. At every iteration, the low-level heuristic with the highest rank is selected and a new solution is obtained. If this solution brings any improvement to the objective, then the rank of h is increased by one. If the solution is not improved then the rank is decreased by one and h is put on the tabu list which prevents it from being used for k iterations. Every solution is accepted in the acceptance phase.

The last hyper-heuristic is based on the *simulated annealing*. It ranks low-level heuristics in exactly the same way as in tabu search selection but this rank is used as a probability of selecting low-level heuristic h. The hyper-heuristic selects h at random with weighted probability specified by the current rank. The acceptance method, different than for the other hyper-heuristics, is called *monte carlo*. It accepts improved solutions always. If a solution is worsened then it is rejected with some probability, which is an exponential function of the solution objective value change divided by the current temperature. At the beginning, the temperature is high and the algorithm is more likely to accept deteriorated solutions. As time passes, the temperature cools down and the method is less likely to accept worse solutions.

3 Low-level heuristics for SBH problem

The solution is encoded using two collections: an ordered set (*list*) of subfragments from which a DNA sequence is reconstructed and an unordered trash set which contains all the remained subfragments from the spectrum. The DNA sequence can be reconstructed by traversing the list and appending every subfragment at the closest position possible to the end of the constructed sequence. The reconstructed sequence might have length greater than n. Thus, feasible solutions are only those with length not greater than n.

The moves described in [1] are used as a template for the low-level heuristics. The moves operate on single subfragments and on clusters. A cluster of subfragments is a sequence of following subfragments that are shifted by only one nucleotide with the preceding subfragment. There are five different low-level heuristics.

- A single subfragment from the trash set is *inserted* into the solution.
- A subfragment is *shifted* from one position in the list to another. During the shift no cluster can be destroyed.
- *Shift of a cluster* to another position in the list. A cluster can be shifted only if it does not break another cluster.
- *Deletion* of the subfragment from the solution to the trash set. Only subfragments outside the cluster or being at one of the cluster ends may be deleted.
- *Deletion of the cluster* to the trash set. This low-level heuristic is quite invasive. It can break the solution and change it extensively.

4 Experimental results

The hyper-heuristics were tested with different subsets of low-level heuristics (some heuristics were also parametrized). All algorithms were tested on data coming from real DNA sequences obtained from GenBank [6]. The DNA sequences were of length 200-600 nucleotides. Spectra of these sequences were created and experimental (positive and negative) errors were artificially introduced. Errors are added to the spectra by removing some of the subfragments from the spectra (negative errors) and by adding some new random subfragments to the spectra (positive errors). The number of removed or added subfragments is determined by the error percentage. 5% of errors means that 5% of subfragments from the spectrum were removed and the same number of random subfragments were added to the spectrum.

In Table 1 the results of the best performing hyper-heuristics are presented: simulated annealing and roulette CF. Three measures evaluate each result. 'Avg usage' is the percentage of subfragments from the spectrum used to construct the solution. 'Optimal count' is the number of instances out of 40, which were solved with 100% of the usage. The above measure can evaluate the solution from the mathematical point of view, but for the biologists the most important is the similarity of the solution to the examined sequence. The last measure,

Instance	200		400		600	
	5%	20%	5%	20%	5%	20%
Hyper-heuristic Simulated Annealing						
Optimal count	37/40	34/40	35/40	22/40	30/40	14/40
Avg. usage [%]	99.81	99.64	99.83	98.93	99.66	98.20
Alignment [%]	98.06	91.48	95.69	82.00	93.25	74.18
Hyper-heuristic Roulette Choice Function						
Optimal count	40/40	40/40	36/40	37/40	23/40	5/40
Avg. usage [%]	100.00	100.00	99.89	99.92	98.94	97.20
Alignment [%]	99.52	98.74	95.61	94.68	92.92	86.03

Table 1: The results of the best performing hyper-heuristics

'alignment', calculates the number of the same letters in two sequences with the Needelman-Wunsch algorithm [4].

Analyzing the results in the table it can be observed that hyper-heuristics result in the solutions of very high usage. Simulated annealing appeared to be the best. Surprisingly, the roulette CF - the most random among CF selections - found the best solutions (considering similarity to the original sequence). For a bad configuration of low-level heuristics the roulette choice function could easily destroy solution by randomly using 'cluster delete' heuristic. But, for a good set of low-level heuristics all the hyper-heuristics were searching around local optimum, while roulette CF by destroying a good solution, could jump into different place in the solution space and resulted in finding better solutions.

Thus, both things are important in constructing a hyper-heuristic framework: to design a good learning mechanism (hyper-heuristic) and a good set of low-level heuristics: some heuristics which aim to intensify the search and some random which diversify the search.

References

[1] Blazewicz J, Formanowicz P, Kasprzak M, Markiewicz W, Weglarz J. Tabu search for DNA sequencing with false negative and false positives. *European Journal of Operational Research* 125:257–265, 2000.

[2] Burke E, Kendall G, Newall J, Hart E, Ross P, Schulenburg S. *chapter 16, Hyper-Heuristics: An Emerging Direction in Modern Search Technology.* In: Handbook of Metaheuristics, Kluwer Academic Publishers, 2003.

[3] Cowling P, Kendall G, Soubeiga E. A Hyperheuristic Approach to Scheduling a Sales Summit. In: *PATAT '00*, Springer-Verlag, London, UK, pp 176–190, 2001.

[4] Needleman SB, Wunsch CD. A general method applicable to the search for similarities of the amino acid sequence of two proteins. *J. Mol. Biol.* 48:443–453, 1970.

[5] Southern E. United Kingdom Patent Application GB8810400, 1988.

[6] http://www.ncbi.nlm.nih.gov/genbank/

Different approaches to parallel computing in the DNA assembly problem.

Piotr Gawron[1] (piotr.gawron@cs.put.poznan.pl),
Wojciech Frohmberg[1], Michał Kierzynka[1]
[1] *Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland*

Acknowledgements: This work has been supported by the Polish Ministry of Science and Higher Education [grant NN519314635]

1. Introduction

Progress in technology and science led to the development of a few different approaches to speed up the algorithms. For many years due to Moore's law CPU speed doubles every two years. However, recently processor makers designed multi-core chips instead of increasing the clock rate.

This situation implies that parallel computation has become necessary to take full advantage of modern computers. Authors described and compared usability of a few types of common used computation platforms:

- single processor computation and its extension to SMP (symmetric multiprocessing),
- computation in distributed systems using cluster and grid architecture,
- computation on SIMT (Single Instruction Multiple Thread) architecture on example of GPU computing.

2. Problem Formulation

In the DNA assembly problem the most time-consuming part is finding alignments between sequences. The most known algorithms which find alignment between two sequences are: Smith-Waterman algorithm [1] and Needleman-Wunsch algorithm [2]. The first finds global alignment and the second semiglobal alignment. In the DNA assembly problem Needleman-Wunsch algorithm gives results which are more appropriate than the above mentioned Smith-Waterman algorithm. Therefore algorithm for finding semiglobal alignment between two sequences was parallelised.

3. Methods and results

For the defined problem parallelism was introduced by parallel computing of many different alignments. Authors implemented algorithms according to three different approaches to parallel computation.

The first implementation involves SMP architecture (POSIX threads were used as a programming API). It is the easiest way to parallelise the algorithm – all threads have access to all data and the only one issue thing under consideration is the synchronisation in access to the memory. This solution has also some disadvantages. The scalability is far from linear (see Figure 1.) and what is more important increasing the number of processors in computer is not an easy or cheap task.

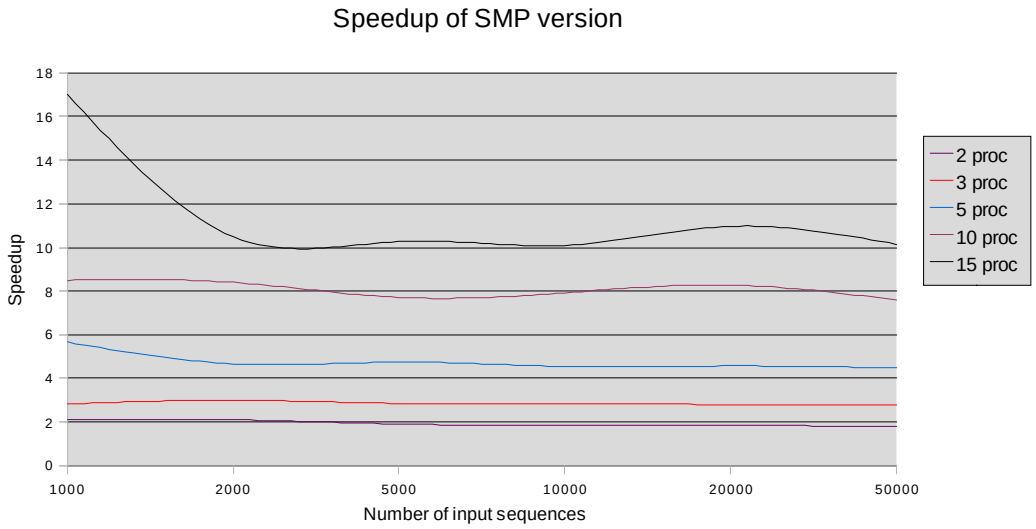


Figure 1Speedup of SMP version

The second implementation works in distributed system such as a grid or a computer cluster (OPENMPI was used as a programming API). This approach is harder to implement because of distributed memory and a need of passing data between nodes all the time. However results are more promising because of the scalability which is limited by the bandwidth of the network connection. In the Figure 3 and Figure 4 results for small number of nodes were presented. There is one thing which is worth of mention: for two processors speedup is equal to 1, because one processor is a host which manages the data.

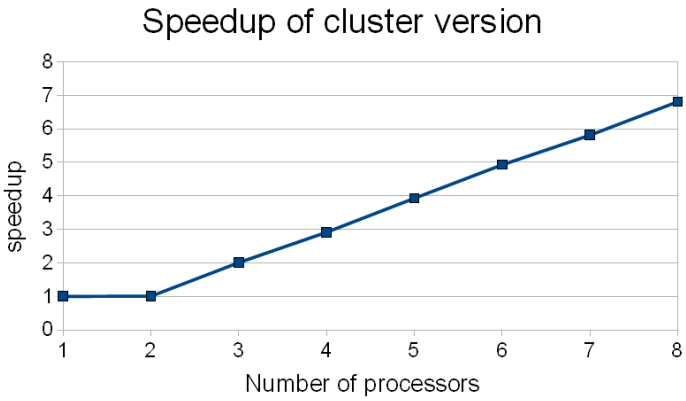


Figure 2 Speedup of cluster version

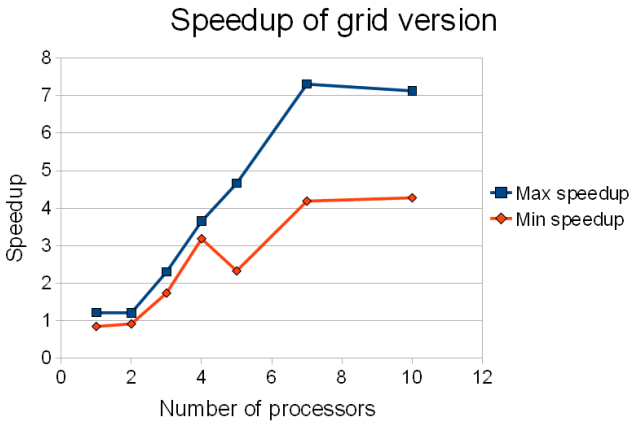


Figure 3 Speedup of gird version

The third version was developed for GPU (Graphics processing unit) computation. Although there are a few GPGPU (general-purpose computing on graphics processing units) technologies like ATI Stream or OpenCL on the market, one of them - CUDA [7], is a bit more established than others. Our implementation of alignment algorithms was done using this technology. There are a few substantial differences between CPU and GPU architectures that make GPU more powerful tool for executing parallel algorithms. Firstly, GPUs have much

more cores, which are the main computational units, e.g. NVIDIA GeForce 280 has 240 cores. Secondly, there is much less cache memory available on the GPU. Moreover, the cache memory on the graphics card is not managed automatically, but by a programmer. Such architecture gives opportunities to utilize the hardware more efficiently. On the other hand, writing parallel algorithms on GPU is more time-consuming. As a platform for comparison GPU and CPU version the following hardware was used:

- CPU: Intel Core 2 Quad Q8200, 2.33GHz,
- GPU: NVIDIA GeForce GTX 280 with 1GB of RAM,
- RAM: 8GB.

The GPU implementation was about 68 times faster than the CPU-based version. In this case also Smith-Waterman algorithm was implemented. The GPU version of SW algorithm was about 108 times faster.

4. Conclusion

All versions of the algorithm show significant speedup. However to achieve better results a mixed platform is taking into consideration (for instance a cluster of computers with powerful graphics card). Such platform would combine advantages of presented methods.

5. References

[1] S.B. Needlemana and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48 (3): 443-53, 1970.

[2] T.F. Smith and M.S. Waterman. Identification of Common Molecular Subsequences. Journal of Molecular Biology 147: 195-197, 1981.

Modeling HCV infection using multi-agent simulation

Szymon Wasik*, Paulina Jackowiak†, Marek Figlerowicz†, Jacek Blazewicz*

November 13, 2010

Abstract

Currently the approach most commonly used to model a viral infection is the system of differential equations. In this paper we compare it with the approach based on multi-agent systems. First we present the system designed to simulate the HCV infection and present its advantages. Then we propose method to determine values of parameters used in this model which is much more difficult that in case of differential equation. Finally we present some results obtained using this method.

1 Introduction

The approach most commonly used to model a viral infection is the system of differential equations. First time it was used to model the HIV infection in 1989 [3]. Since that time many models based on differential equations have been defined. Most of the models try to simulate HIV infection (see for example the review in [10]). However other viral infections are also analyzed using this type of model. The example can be HCV infection investigated in [4, 5, 2] or HBV infection in [6]. This is a very well known and well understood way of modeling but it can describe only some statistics about the population of cells (for example their amount). That is why other types of models are designed and analyzed, for example statistical models [8, 9]. Recently more popular become models based on multi-agent systems. They have many advantages in comparison with models based on differential equations [1].

In this paper we would like to present model based on multi-agent system that were designed by us and compare it with differential equations approach. First in sections 2.1 and 2.2 both types of models are presented and then in section 2.3 the comparison of them is made. In section 3 the method of determining values of parameters that appear in multi-agent model is investigated and section 4 presents some results.

2 Models of HCV infection

2.1 Differential equations model

Following equations describing HCV infection can be defined and this form of them is commonly used in literature:

*Institute of Computing Science, Poznan University of Technology, Poznan, POLAND
†Institute of Bioorganic Chemistry, Polish Academy of Sciences, Poznan, POLAND

$$\begin{aligned}
\frac{dU}{dt} &= s + r_U U\left(1 - \frac{U+I}{U_{max}}\right) - d_U U - (1-\eta)\beta V U + qI \\
\frac{dI}{dt} &= r_I I\left(1 - \frac{U+I}{U_{max}}\right) + (1-\eta)\beta V U - d_I I - qI \\
\frac{dV}{dt} &= (1-\epsilon)pI - cV
\end{aligned}$$

In above equations U denotes number of uninfected hepatocytes, I number of infected hepatocytes and V number of free virions. Uninfected hepatocytes are produced by differentiation of precursors at rate s and are infected at rate β proportional to the number of uninfected cells and free virions. Both uninfected and infected hepatocytes die at rate d_U and d_I respectively and proliferate at maximum rate r_U and r_I respectively until the maximal number of hepatocytes U_{max} is reached. Infected cells can be also cured through a noncytolytic process at rate q. Free virions are produced from infected hepatocytes at rate p and are cleared by immune system at rate c. Coefficients ϵ and η model the treatment with antiviral drugs (interferon and ribavirin) and when no treatment is set they equal 0. Time is measured in days and all quantities are measured in one milliliter of the tissue.

2.2 Multi-agent system

Another approach that can be used to model HCV infection is multi-agent system. In the base form of this method we can define two types of agents:

- **Hepatocytes** - liver cells that can be in infected or uninfected state. Hepatocytes can proliferate and after some time they die. Uninfected cells can become infected after interaction with virions and infected cells can produce free virions.
- **Virions** - free virus particles that exist in blood.

For agent types described above the simulation can be performed. In each step of simulation first the interactions between agents that are close to each other are analyzed and executed and then interactions between agents and environment are performed. During the simulation the number of cells of each type and other useful statistics are gathered and saved. The simulation in each step can utilize even hundreds of thousands of agents.

2.3 Comparison of models

In comparison with differential equations the approach based on multi-agent simulation has many advantages [1]. It describes each cell separately so it makes possible to add some attributes to them and easily distinguish and model different cells of the same type. It also allows incorporating space and more precisely describing the human body and its different parts. Multi-agent models make the modeling of interactions much easier and if needed model can be modified easily and quickly. This way of modeling is also intuitive and easy to understand by biologists which facilitates the collaboration between them and computer scientists or mathematicians and can make it more productive. The main problems with this type of models are larger demand for computing power and the way of finding parameters values. The recent development of efficient multi-core processors

and GPU devices makes it possible to simulate multi-agent systems in satisfactorily short time and solves the first problem. The further one can be solved using genetic algorithms [7].

3 Determining values of parameters

After gathering data in some clinical experiment the coefficients used in differential equations can be quite easily calculated using well known mathematical and numerical methods. Unfortunately determining values of parameters used in the multi-agent system is much more difficult. To solve this problem the reversed simulation method [7] was used. The comparison of this method with the classical, forward simulation is presented in the figure 1. In this method, as opposed to forward simulation, instead of setting values of parameters at the beginning the objective function is defined and the genetic algorithm is used to optimize the value of this function. After the simulation ends results of optimization are evaluated and if they are not satisfying the model can be redesigned.

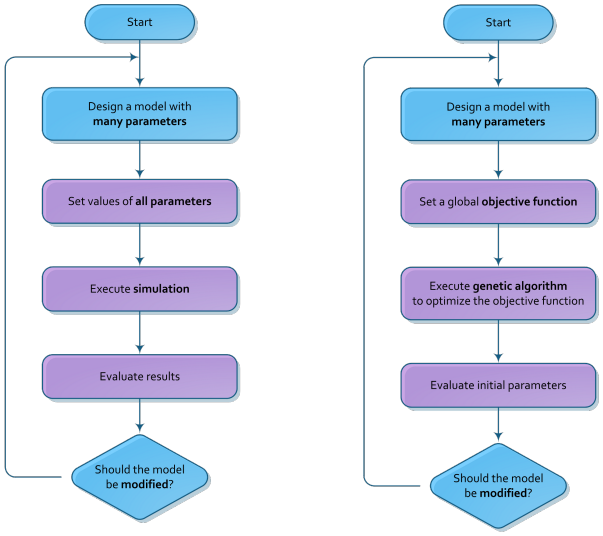


Figure 1: Operations flow in two simulation approaches - forward (left) and reversed (right) simulation.

4 Results

To verify the approach based on multi-agent simulation the input data set were defined using results of simulation of differential equations. The objective function tried to minimize the differences in values of both models at these points. The result is presented in the figure 2. It can be observed that results are similar which proves that the proposed method is correct and multi-agent simulation is at least as useful as differential equations approach.

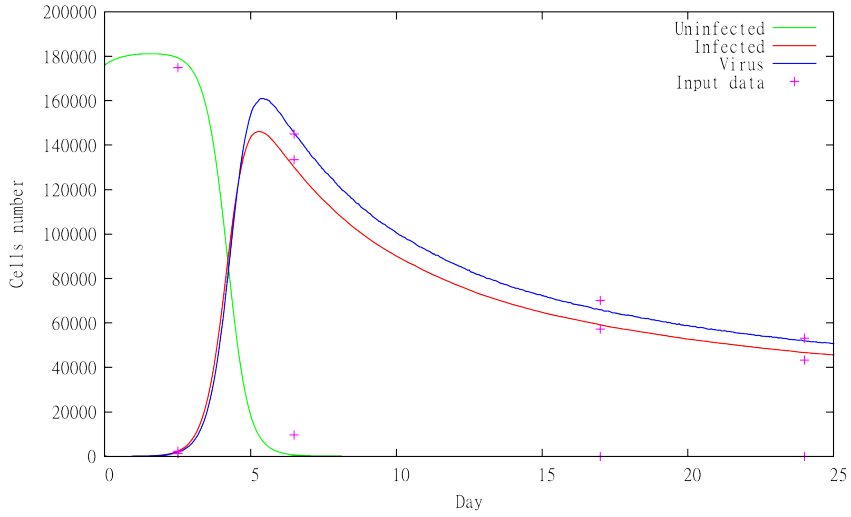


Figure 2: Comparison of model based on differential equations and multi-agent system using the default values of parameters defined for differential equations. Crosses present some data points taken from differential equations results.

Figure 3 presents the result of simulation when the objective was to maximize the number of uninfected hepatocytes but at the same time fit the model to the clinical data about level of virions in blood. This is an example of objective that can not be calculated using differential equation. As it can be observed from the plot the approach based on multi-agent simulation works and the chart presents the maximum possible number of uninfected liver cells according to the model presented in sections 2.1 and 2.2.

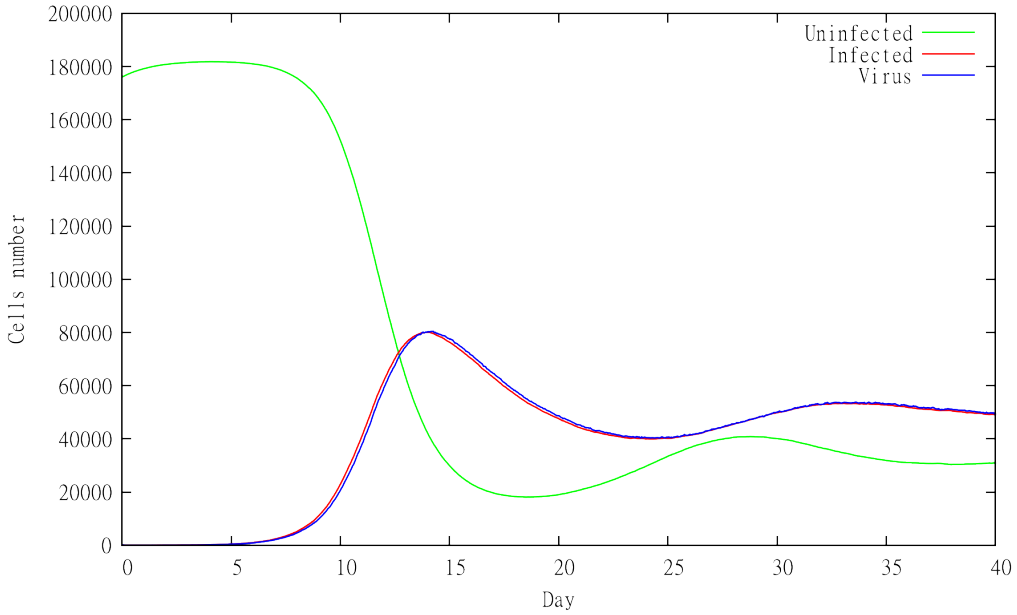


Figure 3: Maximization of the number of uninfected cells when number of virions fits the experimental data.

5 Summary

The method utilizing multi-agent simulation to analyze the HCV infection were proposed. There are some problems connected with this method but all of them can be solve using efficient personal computers or artificial intelligence methods (genetic algorithms). Presented results shows that the method has big potential and makes possible to perform analyzes that were not available with the use of differential equations.

References

[1] Gary An, Qi Mi, Joyeeta Dutta-Moscato, and Yoram Vodovotz. Agent-based models in translational systems biology. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 1(2):159–171, 2009.

[2] Harel Dahari, Ruy M. Ribeiro, and Alan S. Perelson. Triphasic decline of hepatitis c virus rna during antiviral therapy. *Hepatology*, 46(1):16–21, 2007.

[3] Alan S. Perelson. Modeling the interaction of the immune system with hiv. pages 350–370, 1989.

[4] Timothy C. Reluga, Harel Dahari, and Alan S. Perelson. Analysis of hepatitis c virus infection models with hepatocyte homeostasis. *SIAM Journal on Applied Mathematics*, 69(4):999–1023, 2009.

[5] Emi Shudo, Ruy M Ribeiro, and Alan S Perelson. Modeling hcv kinetics under therapy using pk and pd information. *Expert Opin Drug Metab Toxicol*, 5(3):321–32, 2009.

[6] Vana Sypsa and Angelos Hatzakis. Modelling of viral dynamics in hepatitis b and hepatitis c clinical trials. *Stat Med*, 27(30):6505–6521, Dec 2008.

[7] Takao Terano. Exploring the vast parameter space of multi-agent based simulation. In Luis Antunes and Keiki Takadama, editors, *Multi-Agent-Based Simulation VII*, volume 4442 of *Lecture Notes in Computer Science*, pages 1–14. Springer Berlin / Heidelberg, 2007.

[8] S. Wasik, P. Jackowiak, J. Krawczyk, P. Kedziora, P. Formanowicz, M. Figlerowicz, and J. Blazewicz. A certain model of hcv virus infection. Technical report, Institut fur Informatik, Technische Universitat Clausthal, November 2009. Proceedings of ICOLE'09: German-Polish Workshop on Computational Biology, Scheduling and Machine Learning, Lessach.

[9] S. Wasik, P. Jackowiak, J. B. Krawczyk, P. Kedziora, P. Formanowicz, M. Figlerowicz, and J. Blazewicz. Towards prediction of hcv therapy efficiency. *Computational and Mathematical Methods in Medicine*, 11(2):185–199, 2010.

[10] D. Wodarz and M. A. Nowak. Mathematical models of hiv pathogenesis and treatment. *Bioessays*, 24(12):1178–1187, 2002.

Effective data representation in traffic simulation and visualization

Mateusz Cichenski[1], Mateusz Jarus[1], and Grzegorz Pawlak[1]

[1]Institute of Computing Science, Poznan University of Technology, Poznan, Poland - e-mail: grzegorz.pawlak@cs.put.poznan.pl

1 Introduction

The significant role of traffic simulation is invaluable in the process of developing new traffic control strategies and improvement of roads infrastructure. The problem is vast, complicated and involves large amount of calculations, which makes the process hard to visualize in the real time. The representation of the real world and the way of how it changes with the flow of the time is important and is considered as a problem itself.

Each traffic simulation model has its own assumptions which restrict the model to simpler cases. In fact, the models differs each other in the way of how they work, but also how the representation of the data is computed during the simulation. The simulations run in the real-time manner and sometimes it is impossible to run the same simulation again. What is more, the simulations cannot be rewound in any way, which makes the analysis of results hard to accomplish.

2 Problem description

In the typical real-time approach one of the things that matters is the current state of the simulated fragment of reality. The representation of a time stamp T can be divided into two sub-representations: static components of the real world and dynamic one. The static components consist of the road positions, junctions and everything that is not changing during the simulation. Thus, this representation needs to be interpreted only once during the simulation process. Dynamic components of the world describe each object that can change in the time. The most common objects are vehicles and traffic control lights. This leads to the observation that in two given time stamps T_1 and T_2 the dynamic object can have different values of the properties such as position, speed or current state, to name a few. The ability to change state of dynamic objects makes them harder to track if one want to be able to rewind the simulation and replay it in the exactly the same way as before.

Before working out a solution, we define the requirements we want to meet with our representation of the real world. First of all, the simulation should be smooth and clear to interpret. Secondly, the amount of data should be as small

as possible. Otherwise, the process of re-creating of a state in time stamp T will take too much time. And lastly, the simulation should run with a reasonable time interval between two following states, so the result would look like real-time simulation.

3 Representation of the real world

To present our problem in a more formal way, we prepared a model for urban traffic simulation. In our model we treat the static components of the world as a directed graph. The nodes are representing the points in the real world where vehicle can change its direction or the roads crosses. The arcs represent the possible ways from one point to others and the direction of the arc is relevant to the traffic flow direction on the road [2]. The sample fragment of this representation was shown in Figure 1.

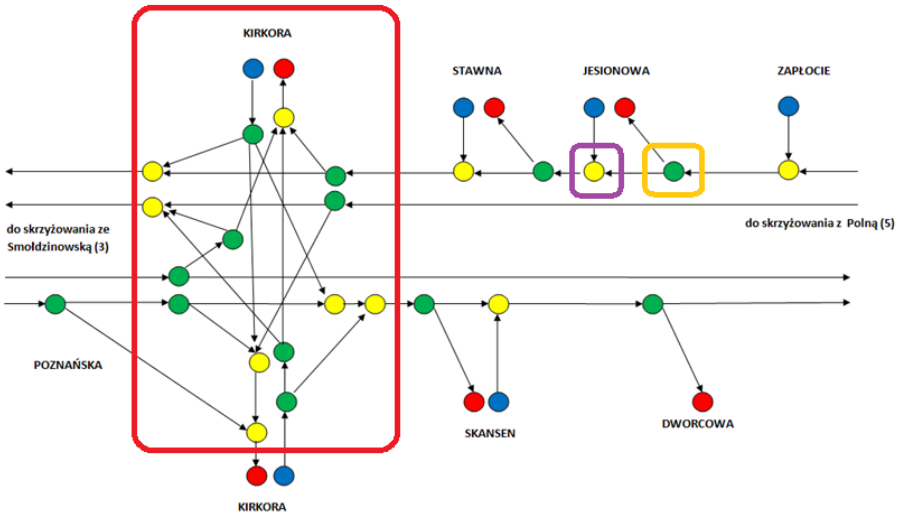


Figure 1: The graph representation example of consecutive crossroads

In Figure 1 nodes represent simple crossroads, i.e. orange rectangle represents road that splits into two roads and purple rectangle represents two roads merging into one. A group of nodes can represent larger crossroads, i.e. red rectangle represents a crossroad between double-lane road and single-lane road.

However, above representation is only the logical layer [1] of traffic architecture. Thus, we extended it to representation the real world scenario with the spatial coordinates included. We had to add simple nodes that were just used to pass the car from the input arc to the output arc without possibility of choosing from different paths. The nodes of the graph were placed in space according to the real satellite photo and the lengths of the arcs were preserved proportionally to reality. The sample crossroad was shown in Figure 2.

The arcs are called segments identified by ID and a group of consecutively connected segments, which means that from the first segment we can reach the

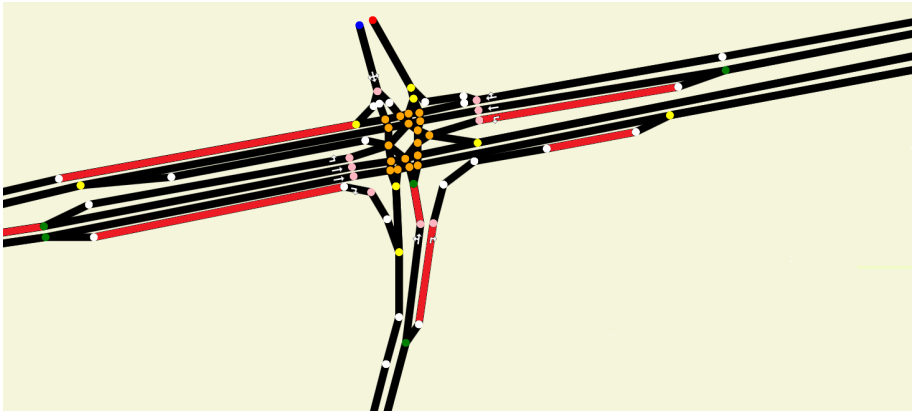


Figure 2: Example of the crossroad model

last segment going through the arcs, makes a road also identified by ID. The roads can join another road or a crossroad. The crossroad is a special group of segments that joins multiple roads. To make a double-lane road we put two segments in the same direction side by side. The segment can also be described by direction attribute, which is used to draw the arrows on the crossroads, but has no special meaning for simulation process.

We differentiate nodes by the role which they play in the system. The most common nodes have no special meaning and they are used to connect two segments. One segment goes into the node and the second one goes out of the node (**white**). If a node has two or more input arcs, then it serves as a road connection node (**yellow**). Green nodes are called decision points, have at least two output arcs and store the probability values [2, 3] for different car types for picking their further path from the possible output arcs (**green**). Those points can be grouped into a virtual decision point which spreads through more than one road. Another type of decision point are control points, which play the same role as decision point but they include the traffic lights state [2], before allowing the vehicle to move forward (**pink**). The crossroad nature makes it very difficult to model due to multiple road crossings. Each crossed road is connected with collision point which can have multiple input and output segments (**orange**). During simulation the vehicles will not collide with others when crossing those points. Lastly, we have a sink node(**red**) and a source (**blue**) nodes. The first one is used to remove vehicle from the system and the second one spawns new vehicles.

The nodes are very important for the simulation process and the properties assigned to them are used in modeling of real world. They provide distributions for the vehicles which determine the path of each vehicle and also the quantity of vehicles spawned in given period of time. The values can be specified with the precision bounded to minutes. The common types of vehicle move within the system based on the distributions in the control and decision points. The fixed tracks for special vehicles are modeled by using a 100% chance of picking the next segment of designated path.

4 Time line concept

In our dynamic component of the world representation we had to decide how to represent all moving objects. We introduce a common approach which uses time line with frames and key frames. However, the frames contain only parameters of the objects that changed their state and the key frames contain the whole set of the state parameters. It makes possible to restore the state using key frame, i.e. at a given time stamp T_1 the car moved forward on the segment, thus this information will be included in the following time stamp T_2. If the same car had to stop because there was another car in front of it, so it did not change its state thus providing this information in following time stamp T_3 is unnecessary.

To describe each possible situation in our world we created a list of possible state changes which will include only necessary data about the state which we call events. The vehicle can move along its current segment and the position of it is given in per mile value of segment length calculated from the beginning of the segment. Vehicle can change segments and the new segment ID is sufficient to determine the new assignment. In case of two segments of the same road lay side by side, which makes a double-lane road, the vehicle can also shift from one lane to another. To do this we need the progress of vehicle on the current segment just like in the move situation and also the progress of the shift move which is perpendicular to the vehicle move direction. We also provide special state for marking the cars in traffic including the state of being in traffic and leaving the traffic. Of course spawning new vehicle is also described in time line by providing the vehicle type, road ID and segment ID on which the car should be spawned. If car leaves our world the information is send in destroy message which only needs the ID of vehicle. Because those messages are set per vehicle, each of them contains information about the car they describe.

This event model on framed time line gives the possibility to run the simulation forward. To ensure that the previous states can be restored, we used special type of structure which we call extended structures. Basically, they provide the same amount of information as standard structures, but additionally they always describe exact position of the car within the static component of the world by providing information about road ID and a segment ID. The simulation writes the frames using standard structures described above with the given tick interval, however each Xth frame is written with the use of extended structures, so its called a key frame. To keep the simulation rewind smoothly we set X to 10 with the tick interval of 50 ms, which gives us a tick interval of 0.5 second in rewind mode.

5 Serialization

We designed the simulation and visualization as separate software modules. Thus, we had to provide sufficient data protocol to communicate between them. The first approach included generating a XML file which will describe each key frame as separate tag in which the presented events would be stored as inner tags. This lead to a problem of the size of output files which reduced the possible length of simulation. That is why we introduced the concept of serialization of data, which enabled us to drastically reduce the file size of simulation and lengthen the simulation duration up to required 24 hours.

However, by analyzing the data from the considered area we found out that we can reduce the size of resulting file even more. It can be done by operating on the bits instead of using standard numeric types of programming language, i.e. to store car ID we need only 12 bits instead of 16, because we will never have more than 4096 cars simultaneously in the system. By using such reductions we were able to keep the file size at approximately 25% of size of XML file describing the same simulation. Thus, we created special binary serializer which is responsible for packing and unpacking data structures presented in previous section. What is important, this reduction did not impact the time line rewind capabilities, which still working smoothly. Finally, in Table 1 we present sample file sizes and simulation generation times.

Description	File size	Generation time
1h, small traffic	9,9 MB	103 s
1h, large traffic	50,5 MB	163 s
3h, small traffic	27,1 MB	263 s
3h, large traffic	404 MB	763 s

Table 1: Results of the computational experiment. The file size depends on the traffic load and simulation duration.

6 Summary

Current traffic simulations are not capable of saving the simulation flow, which makes further analysis harder to accomplish. The approach with time line gives the opportunity to watch the scenario multiple times and share it with others. The used model can be extended to include more complex rules, but it will not affect the visualization. The data protocol can be slightly modified without additional effort in case of larger area investigation. By using serialization we can generate longer simulations to observe anomalies in traffic flow, for example traffic jams at certain hours.

References

[1] W.R. Blunden. *Wprowadzenie do teorii ruchu drogowego*. Wydawnictwa Komunikacji i Łączności, 1972.

[2] Janusz Chodur. *Funkcjonowanie skrzyzowan drogowych w warunkach zmiennosci ruchu*. Wydawnictwo PK, 2007.

[3] Wolfgang Mensebach. *Podstawy inzynierii ruchu drogowego*. Wydawnictwa Komunikacji i Łączności, 1978.

Researching the influence of changes in traffic organization on car factory production

Mateusz Cichenski[1], Mateusz Jarus[1], and Grzegorz Pawlak[1]

[1]Institute of Computing Science, Poznan University of Technology, Poznan, Poland - e-mail: grzegorz.pawlak@cs.put.poznan.pl

1 Introduction

The fluency of production in a car factory depends on many factors. A very important one is an undisrupted flow of parts in the transportation system between different plants. Car parts always need to be delivered to specific locations at certain time period. This is a crucial point where the manufacturing process will not suffer from slowdowns. However, even small disturbances in traffic organization may have negative effect on the cars flow on the roads. This, in turn, has a direct impact on the regularity and certainty of deliveries.

The problem of predicting traffic flow is very complex. On the other hand, it is essential to counteract the negative effects of congested roads. In that case the factory logistic may take some actions to keep the transportation system fluent. It could be for example: increasing the number of lorries, choosing different routes or even restructuring the plants logistic organization.

2 Problem description

Many factors have an influence on the traffic on urban roads. There are accidents, weather anomalies, road construction etc. Although, many of them are spontaneous, some of them (like the latter one) are predictable and planned. It means that it is possible to try to simulate the traffic flow on the roads before such events occur in reality. This, can help better prepare for factory managers by taking the appropriate actions. In case of a car factory the knowledge about future traffic flow may have a tremendous impact on the efficiency of parts deliveries. We would like to present a traffic simulation model that gives the ability to predict traffic flow on roads.

Our model should be very realistic and the the simulation must running fast at the same time. Sometimes, it is necessary to simulate a complex connection of roads. At peak hours hundreds of cars pass them every hour. Traffic control lights (often adaptive), subordinate roads, complicated junctions with many lanes - all of this needs to be perfectly mapped. To achieve this much input data is required. The number of cars, passing each crossroad is received from magnetic loops. From, at least, the most important crossroads are necessary. This historical data must be appropriately parsed before used in the simulation

software tool. Current traffic lights control programs and junctions schemes are necessary to realistically simulate crossroads. In case of roads reconstructions detailed information about the traffic organization during that time are necessary.

This input data analysis needs to be processed very carefully. Even minor errors in this phase may have a huge impact on the final results. Therefore, as much real data as possible should be acquired to minimize the approximations.

The software simulation tool should also provide some reports as the results. The visualization for the simulated process of predicted traffic flow is very important. It shows how the traffic flow changes during the time. However, a report with the most important parameters, summed up, is essential. It provides a fast way of detecting the sources of the traffic flow anomalies such as the length and liquidation time of traffic jams.

3 The traffic modeling problem

We created a model of real roads in the urban area. The roads are modeled as a directed graph. Each road consists of several segments connected by markers. There are different types of points. Decision points fork roads. Injection markers merge two of them into one. It is possible to put traffic control lights on supervised control points. Every collision point denotes a place where two or more roads are crossing. Source and sink markers create and destroy cars in the system, respectively. These six categories of points give the possibility to create any junction, from the simplest one, through very complex with many lanes and traffic control lights to even roundabouts.

Every junction was built based on real schemes. At first a logical graph was created by detecting specials point. An exemplary graph of real Baltycka-Lesna streets junction is presented in Figure 1. Using this graph a simulation model of the junction could be constructed. It consists of the same markers as the logical graph connected by segments of roads. To remove the angularity these straight sections were later replaced by Bézier curves in the visualization tool.

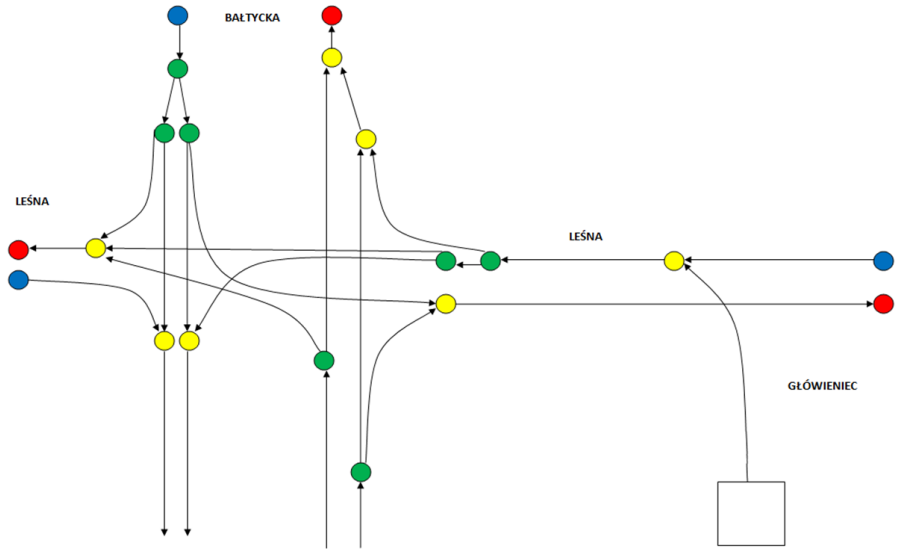


Figure 1: Exemplary logical graph presenting Baltycka/Lesna junction

Cars also are described by a few important parameters. There is a unique ID assigned to each of them. There are eleven types of cars - 7 factory cars, 2 city buses, 1 passenger car and a non-factory truck. Vehicles differ in the maximum speed they may achieve and the value of acceleration - trucks are always slower than passenger cars. The length of the cars we also take into consideration

4 Simulation process

One of the most important features of the simulation process is its ability to detect traffic jams. In the visualization software each car that is stuck in a traffic jam changes its status. This happens after such a vehicle is not active for a specified amount of time.

The simulation software also creates a report containing some additional information. There is every factory car type listed with the minimum, maximum and average time that it was stuck in a traffic jam. Bottlenecks in the traffic flow are also presented. There is every road marked where traffic jams used to generate with some details: the cumulative time of all traffic jams, number of traffic jams, average time of liquidation, average length and cumulative number of cars stuck in this traffic jam.

5 Input data analysis

5.1 Traffic control lights

A very important part of the simulation are traffic lights. In our simulation it is possible to assign many traffic lights to one marker. There could be one light for cars going straight on junction and a special right arrow light at the same time.

We received 24h traffic control lights programs from the City Roads Authority for all of the traffic lights in the simulation. They were implemented in our software tool accordingly. Their job confines to presenting specified light at certain time. It is also possible to assign the flashing yellow color - in such situation roads priorities take precedence.

Adaptive traffic lights are also available. The color that is currently presented by them depends on current traffic conditions.

Each car may pass any junction in one of three ways:

- with stoppage in case of red traffic light
- with speed reduction in case of yellow or red traffic light when the speed of car is too high to stop or in case of green traffic light on a curve
- without speed reduction [1]

5.2 Historical traffic data

Historical traffic data is the core on which depends the accuracy of prediction. It was accumulated on magnetic loops. Numbers of different types of vehicles were gathered. This allowed to create a more accurate model as cars differ in many parameters - maximum speed and acceleration value, length etc.

Unfortunately, this data does not differentiate between factory logistics and non-factory trucks. Internal company information about deliveries was used to fill this gap.

One hour granular data is used to supply the simulation software. This means that the number of different types of cars for every hour is calculated and assigned to the source markers. 24-hour traffic distribution is obtained in this process. Vehicles are then created in specified numbers at random time intervals.

The historical data also contains information about the distribution of cars on junctions in every direction. In our simulation we achieved this effect by specifying probability of choosing one or the other routing direction for each vehicle type. Therefore, it is easy to notice that non-factory routes are not deterministic. Every simulation may present a bit different situation on roads despite of the fact that the input parameters are identical.

There were lack of data from smaller junctions because no magnetic loops where there installed. In that case the calculations of the traffic density, combining the data from the main junctions, have been performed.

5.3 Roads reconstructions

In the presented simulation example, where the three phases of junction reconstruction in the considered urban area. In each of them, the different road traffic

organization was applied. Appropriate road models were created for those three stages. Figure 2 presents a first phase model. Brown areas determine places which are not available for traffic flow.

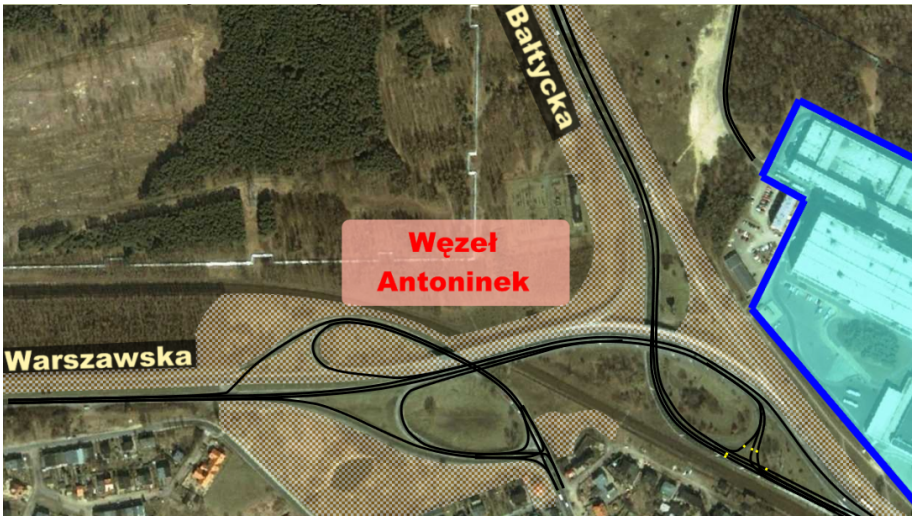


Figure 2: First phase of reconstructions modeled in the visualization software

6 Results

There are three variants corresponding to all reconstruction phases and one matching the state before the beginning of road constructions. The latter one could be used to verify the accuracy of the traffic flow prediction during the simulation, since it is easy to compare it with traffic conditions derived from the gathered data. Three types of simulations for each of these stages were generated - one for the morning peak (6:00 - 9:00 a.m.), one for the shift change (1:00 - 3:00 p.m.) and one for the afternoon peak (4:00 - 7:00 p.m.).

The average time of passing the route from Z1 to Z4 plant for all of the factory trucks was about 9 minutes in the morning peak, 12 minutes during the shift change and 11 minutes in the afternoon peak. In the reverse direction they needed 7.5, 11 and 7.5 minutes at the same time ranges, respectively. Table 1 presents detailed results.

First junction reconstruction phase introduced two significant changes in statistics. The average time of travel between Z1 and Z4 plants lengthened by almost 2 minutes during shift change and by 14 minutes in the afternoon peak. Second phase showed similar changes. Z1-Z4 route took 30 minutes during shit change and 18 minutes in the afternoon peak. The results were almost the same in the last stage of reconstructions.

7 Summary

Studies showed that this simulation model accurately predicts real traffic on roads. As a consequence it can be used to localize the sources of congestions

Car type	From	To	No of cars	Min	Max	Avg
Z4 truck	Z1 plant	Z4 plant	20	00:07:40	00:12:02	00:09:01
	Z4 plant	Z1 plant	24	00:06:46	00:09:40	00:07:48
JIT Swarzedz	Z1 plant	Z4 plant	13	00:07:45	00:10:53	00:09:02
	Z4 plant	Z1 plant	7	00:05:58	00:08:15	00:07:04
FBU	Z1 plant	Z4 plant	7	00:07:43	00:10:02	00:09:01
	Z4 plant	Z1 plant	10	00:06:11	00:10:04	00:07:28
Total	Z1 plant	Z4 plant	40	00:07:40	00:12:02	00:09:02
	Z4 plant	Z1 plant	41	00:05:58	00:10:04	00:07:36

Table 1: Times of deliveries of different types of factory cars between Z1 and Z4 plants

on roads. These results can then be used to improve the efficiency of deliveries in the factory logistic system. This, in turn, leads to increase car factory productivity and the reduce the logistic systems costs.

References

[1] Ryszard Krystek et al. *Symulacja ruchu potoku pojazdów.* Wydawnictwa Komunikacji i Łączności, 1980.

Survey of scheduling of coupled tasks with chains and in-tree precedence constraints

Michal Tanas*, Jacek Blazewicz†, Klaus Ecker‡

1 Introduction

A scheduling problem is, in general, a problem answering the question of how to allocate some resources over time in order to perform a given set of tasks, according to the definition given by Baker in [3]. In practical applications resources are processors, money, manpower, tools, etc. Tasks can be described by a wide range of parameters, like ready times, due dates, relative urgency factors, precedence constraints and many more. Different criteria can be applied to measure the quality of a schedule. The general formulation of scheduling problems and the commonly used notation can be found in books such as ones written by Brucker [8] or Blazewicz et al. [6]. A very interesting survey of the most important results is given in the handbook edited by J. Leung [13].

One branch of scheduling theory is concerned with scheduling of coupled tasks. A task is called *coupled* if it contains two operations where the second has to be processed some time after a completion of the first one. This variant of scheduling problem, described by Shapiro in [19] and by Orman et al. in [17], often appears in radar-like devices, where two subsequent radar pulses are used to calculate speed and trajectory of a moving object.

The complexity of various scheduling problems with coupled tasks has been deeply studied by Orman and Potts in [15]. Although they proved that most of the cases, like $1|(a_j, L_j, b_j)|C_{max}$, $1|(p_j, p_j, p_j)|C_{max}$, $1|(a, L_j, b)|C_{max}$ and $1|(a, L, b_j)|C_{max}$ are strongly NP-hard, they also found some important polynomial algorithms for $1|(p, p, b_j)|C_{max}$ and $1|(p, L, p)|C_{max}$in [16].

A coupled task scheduling problem with non-exact gap is surveyed by Gupta, who proved NP-hardness of $1|(a_j, [L_j, \infty], b_j)|C_{max}$ in [11]. NP-hardness of this case with unit processing times, i.e. the case of $1|(1, L_{ij}, 1), chains|C_{max}$ was proven by Wenci Yu in [22], where some interesting connections between coupled tasks and flow shops are also given. A similar problem of scheduling tasks with time-lags was studied by Brucker and Knust, and some important complexity results, i.e. NP-hardness of $1|p_j = 1; intree(L); r_j|C_{max}$ and polynomial solvability of $1|p_j = 1, outtree(L), r_j|\sum C_j$ and $1|p_j = 1; prec(L = 1)|\sum C_j$ were found [9].

In recent years the coupled tasks scheduling problem is widely studied, and

*Applied Computer Science Division, Physics Faculty, Adam Mickiewicz University, Poznan, Poland - E-Mail: michal.tanas@amu.edu.pl

†Poznan University of Technology, Poznan, Poland

‡Ohio University, Athens, USA

a lot of new important results have been achieved. In 2003 a polynomial algorithm for the problem $1|(a,L,b)|C_{max}$ was found by Ahr et al. [2]. Later Potts and Whitehead created new important heuristics to solve coupled tasks scheduling problems [18], while Li and Zhao analyzed heuristic algorithms applied to coupled tasks scheduling on a single machine [14]. Recently, new applications of coupled tasks scheduling in production systems in single machine no-wait robotic cells were introduced by Brauner, Finke, Lehoux-Lebacque, Potts and Whitehead [7], and also a cyclic case of the one machine coupled task problem was proven to be solvable in polynomial time by Lehoux-Lebacque, Brauner and Finke [12]. Approximation algorithms for coupled tasks problems with unit processing times were analyzed in [1] and [4] and coupled tasks problem with compatibility constraints was researched by Simonin et al in [20].

For this paper, especially interesting is the case of single machine scheduling of identical coupled tasks with unit processing time, i.e. the problem $1|(1,L,1)-coupled, exactgap|C_{max}$ with various types of precedence constraints. The strong NP-completeness of the case where the precedence constraints graph is a general graph is shown in [5]. On the other hand polynomial solvability of the $1|(1,2,1)-coupled, strict-tree|C_{max}$ case was shown in [21] and polynomial solvability of the $1|(1,2k,1)-coupled, strict-chains|C_{max}$ was shown in [10].

In this paper, we complement the above results by presenting polynomial time algorithm for the $1|(1,L,1)-coupled, strict\ chains, exact\ gap|C_{max}$ and $1|(1,L,1)-coupled, strict\ in-tree, exact\ gap|C_{max}$ problems. Moreover we state a hypotesis of equivalence of the $1|(1,L,1)-coupled, strict\ prec, exact\ gap|C_{max}$ and $P(L+1)|p_j=1, prec|C_{max}$ problems with the same type of graph of precedence constraints.

The organization of the paper is as follows. The problems are formulated in Section 2. The idea of polynomial time algorithm for the chains case is presented in Section 3.2. The idea of polynomial time algorithm for the in-tree case is presented in Section 3.3. The hypothesis of equivalence of $1|(1,L,1)-coupled, strict\ prec, exact\ gap|C_{max}$ and $P(L+1)|p_j=1, prec|C_{max}$ is presented in Section 4. We conclude in Section 5.

2 Problem formulation

Adapting the commonly accepted notation for scheduling problems, the scheduling problems considered here (see also [6]) can be denoted by
$1|(1,L,1), strict\ chains, exact\ gap|C_{max}$ and
$1|(1,L,1), strict\ in-tree, exact\ gap|C_{max}$ which means:

- There is a *single processor* in the system.
- There is a set of identical tasks, denoted by $T_1,\ldots,T_n$, to be scheduled.
- Each *task T_j is a pair of operations* with a gap between them.
- Every operation has unit processing time.
- Gaps are exact and have uniform constant length L, where L is a positive integer.

- The precedence constrains are strict, which means that the first operation of a subsequent task depends on the second operation of the preceding task.
- Precedence constraints graph has the form of chains or in-tree respectively.
- The optimization criterion is to minimize the schedule length $C_{max} = \max\{C_{j2}\}$ where C_{j2} is the completion time of the second operation of T_j.

Note that for any instance of the problem, provided the precedence constraints graph has no directed cycle, there is a trivial feasible (but not necessarily optimal) solution in which tasks are scheduled "one after another" in their topological order. If the precedence constraints graph has a directed cycle, there is no feasible solution for such instance.

3 A polynomial-time algorithms

3.1 General idea

In general, each coupled tasks scheduling problem contains in fact two separate subproblems, which both must be solved to obtain the final solution. These problems are:

- Which is the optimal permutation of tasks. To solve every scheduling problem the correct order of tasks must be found, so this problem is quite common amongst the scheduling problems.
- In which time units the machine must remain idle, despite there are free tasks to be processed. This problem is specific to the coupled tasks problems, in which the optimal permutation is not enough to obtain the optimal solution.

3.2 Chains case

As the problem $1|(1, L, 1) - coupled, strict\ chains, exact\ gap|C_{max}$ splits in fact into two separate sub-problems: how to find the optimal order of tasks and how to optimally schedule tasks in a given order.

The optimal order of tasks can be found through conversion to the problem $P(L+1)|pmtn|C_{max}$ which then can be solved by a McNaughton rule. The optimal order of tasks in the coupled tasks problem may be computed using such a parallel schedule.

The idea of the second stage is based on observation that any feasible schedule for the problem $1|(1, L, 1) - coupled, strict\ chains, exact\ gap|C_{max}$ can be decomposed into a sequence of partial schedules (called *segments*) and each segment contains an upper-bounded number of pairwise independent coupled tasks. Such decomposition limits the solution space to a polynomial size, which renders any algorithm working on such space (even the full search one) polynomial.

3.3 In-tree case

Similary, the problem $1|(1, L, 1) - coupled, strict\ in-tree, exact\ gap|C_{max}$ also splits into the same two separate sub-problems: how to find the optimal order of tasks and how to optimally schedule tasks in given order.

In this case, the optimal order of tasks can be found through conversion to the problem $P(L+1)|p_j = 1, in-tree|C_{max}$ which then can be solved by a Hu's algorithm. The optimal order of tasks in the coupled tasks problem can then be easily determined from such a parallel schedule.

The idea of the second stage is the same as in the chains case, and is again based on observation that any feasible schedule for the problem $1|(1, L, 1) - coupled, strict\ in-tree, exact\ gap|C_{max}$ can be decomposed into a sequence of partial schedules (called *segments*) and each segment contains an upper-bounded number of pairwise independent coupled tasks. Such decomposition again limits the solution space to a polynomial size, which renders any algorithm working on such space (even the full search one) polynomial.

4 Hypothesis of equivalence

The similarities between the two problems allows us to state the following hypothesis.

Hypothesis 1. *There are exists a bidirectional polynomial transformation between problem* $1|(1, L, 1), strict\ prec|C_{max}^{(dec)}$ *and* $P(L+1)|p_j = 1, prec|C_{max}^{(dec)}$ *with the same graph of precedence constraints. This means that both these problems are equivalnet in terms of theory of complexity.*

5 Conclusion

In this paper the complexity of coupled tasks scheduling problems are discussed. It is presented that the problem of scheduling of identical coupled tasks on a single machine is solvable in polynomial time if the precedence constraints graph has form of chains or in-tree. Moreover, a hypothesis is stated that the coupled tasks scheduling problems are equivalent to the corresponding problems of scheduling of unit processing time tasks on a parallel system. The confirmation or refutation of this hypothesis nees further research.

References

[1] Alexander A. Ageev and Alexei E. Baburin. Approximation algorithms for uet scheduling problems with exact delays. *Oper. Res. Lett.*, 35(4):533–540, 2007.

[2] D. Ahr, J. Bekesi, G. Galambos, and G. Reinelt M. Oswald. An exact algorithm for scheduling identical coupled tasks. *Mathematical Methods of Operations Research*, 59, No.2:193–203, 2004.

[3] K. Baker. *Introduction to Sequencing and Scheduling.* J. Wiley, New York, 1974.

[4] József Békési, Gábor Galambos, Marcus Oswald, and Gerhard Reinelt. Improved analysis of an algorithm for the coupled task problem with uet jobs. *Oper. Res. Lett.*, 37(2):93–96, 2009.

[5] J. Blazewicz, K. Ecker, T. Kis, C. N. Potts, M. Tanas, and J. Whitehead. Scheduling of coupled tasks with unit processing times. *Journal of Scheduling*, yet unknown:yet unknown, 2011.

[6] J. Blazewicz, K. Ecker, E. Pesch, G. Schmidt, and J. Weglarz. *Handbook of Scheduling. From Theory to Applications.* Springer, 2007.

[7] N. Brauner, G. Finke, V. Lehoux-Lebacque, C. Potts, and J. Whitehead. Scheduling of coupled tasks and one-machine no-wait robotic cells. *Computers and Operations Research*, 36 Issue 2:301–307, 2009.

[8] P. Brucker. *Scheduling Algorithms.* Springer, Berlin, third edition, 2001.

[9] P. Brucker and S. Knust. Complexity results for single-machine problems with positive finish-start time-lags. *Mathematik, Reihe P, Nr*, 202:299–316, 1998.

[10] K. Ecker and M. Tanas. Complexity of scheduling of coupled tasks with chains precedence constraints and constant even length of the gap. *Foundations of Computing and Decision Sciences*, 32, No.3:199–212, 2007.

[11] J. N. D. Gupta. *Single facility scheduling with two operations per job and time-lags.* preprint, 1994.

[12] V. Lehoux-Lebacque, N. Brauner, and G. Finke. Identical coupled tasks scheduling: polynomial complexity of the cyclic case. *Les cahiers Leibnitz*, 179, 2009.

[13] J. Leung, editor. *Handbook of Scheduling.* Chapman and Hall, 2004.

[14] H. Li and H. Zhao. Scheduling coupled tasks on a single machine. *IEEE Symposium on Computational Intelligence in Scheduling*, 1-5:137–142, 2007.

[15] A. J. Orman and C. N. Potts. On the complexity of coupled tasks scheduling. *Discrete Applied Mathematics*, 72:141–154, 1997.

[16] A. J. Orman, C. N. Potts, A. K. Shahani, and A. R. Moore. Scheduling for the control of a multifunctional radar system. *European Journal of Operational Research*, 90:13–25, 1996.

[17] A. J. Orman, A. K. Shahani, and A. R. Moore. Modelling for the control of a complex radar system. *Computers Ops Res.*, 25:239–249, 1998.

[18] C. N. Potts and J. D. Whitehead. Heuristics for a coupled-operation scheduling problem. *Journal of the Operational Research Society*, 58 (10):1375–1388, 2007.

[19] R. D. Shapiro. Scheduling coupled tasks. *Naval. Res. Logist. Quart.*, 27:477–481, 1980.

[20] G. Simonin, B. Darties, R. Giroundeau, and J. C. Konig. Isomorphic coupled-task scheduling problem with compatibility constraints on a single processor. In *yet unknown*, 2009.

[21] J. Whitehead. PhD thesis, University of Southampton, 2002.

[22] W. Yu. *The Two-machine Flow Shop Problem with Delays and the One-machine Total Tardiness Problem.* Technische Universiteit Eindhoven, 1996.

The Research Group Theoretical Computer Science at CITEC

Barbara Hammer*
Theroretical Computer Science,
CITEC centre of excellence, Bielefeld University,
Germany

December 15, 2010

Introduction

The research group 'Theoretical Computer Science for Cognitive Systems' (TCS) has been newly established at Bielefeld University starting from April, 1st, 2010 at the Faculty of Technology. It is located in the Centre of Excellence Cognitive Interaction Technologies (CITEC), which is funded within the frame of the excellence initiative of the German government (see http://www.cit-ec.de/). The mission of CITEC is to shape the command of technical systems into the ease and naturalness of human communication. As such, it boosts interdisciplinary research in diverse areas including robotics, sports sciences, natural language acquisition and understanding, human memory and learning, neuronal control strategies, or the investigation of animal behavior such as stick insects with the ultimate goal to understand how cognitive processes take place at different levels such that cognitive interaction with technical systems can be realized based on these findings. Computer Science constitutes a key enabling technology within this research environment to provide and implement functional algorithmic realizations of cognitive interaction, to establish a technical base e.g. by means of robotic platforms and to establish intuitive cognitive interfaces towards low-level functionalities, and to provide a formal mathematical background which substantiates the models by according guarantees and characterizations.

An ubiquitous feature in all these research areas is an increasing amount of electronically available data. This increases with respect to both, size and complexity due to improved sensor technologies and dedicated data formats and storage facilities. Hence automatic techniques which help humans to extract relevant information

*E-Mail: *bhammer@techfak.uni-bielefeld.de*

from the data are required. Typical data analysis tasks include data clustering, data visualization, the inference of data models for classification, regression, or density estimation, relevance learning and feature extraction, etc. The main focus of the TCS research group is the development, investigation, and application of cognitively inspired data analysis methods which provide an intuitive interface of digital data for humans. Within this line of research, different topics are currently investigated within the group.

Prototype based methods

Prototype based methods represent data in terms of prototypical representatives, and a clustering or classification is usually based on the distance of a data point to the given prototypes. Thus, prototype-based models provide very intuitive data analysis tools since they allow human insight by a direct inspection of the prototypes. A variety of intuitive training techniques for both, supervised and unsupervised settings exist, such as the popular self-organizing map (SOM), neural gas (NG), learning vector quantization (LVQ) and generalizations such as generalized LVQ (GLVQ), and statistical counterparts such as the generative topographic mapping (GTM) or soft robust learning vector quantization (SRLVQ).

Albeit very intuitive, one of the main drawbacks of the technologies is the crucial role of the Euclidean metric for data representation. Commonly, the simple Euclidean metric is not well suited for the given data due to very high dimensionality, inappropriate scaling, and correlations of the features. Within the TCS group, several approaches to get around this problem and to extend the techniques to incorporate more general metrics which are automatically adapted according to the given data have been developed, such as metric learning in supervised techniques [15, 13, 14, 17] or supervised or unsupervised metric adaptation for topographic mapping [8, 3]. These results can be accompanied by interesting theoretical investigations of the dynamics and generalization ability [4, 18, 5] as well as technologies to speed up and parallelize the models [1].

Dealing with dissimilarity data and structures

A further step abstracts even from vectorial representations and rather takes pairwise dissimilarities of data as inputs – this way, a wide applicability to modern data structures including discrete structures such as sequences, trees, or graphs becomes possible by means of dedicated dissimilarities such as information theoretic models, edit distances, or graph kernels. To apply prototype based techniques in such settings, an adequate representation of prototypes as well as efficient ways of how to adapt the prototypes have to be found. Within the TCS group, several techniques have been developed to extend the classical approaches towards general

dissimilarities by either median approaches or relational variants of the techniques [9, 7]. The excellent results in several practical applications can be accompanied by a thorough theoretical foundation embedding data in so-called Pseudo-Euclidean space, and techniques to speed up the systems to avoid the usually quadratic complexity caused by the size of the dissimilarity matrix.

Data visualization

Due to only vaguely specified objectives and more and more complex data, data visualization becomes more and more relevant to allow humans to rapidly scan through large volumes of complex data and to detect structures therein, relying on their astonishing cognitive capabilities as concerns visual perception. In consequence, fast and reliable nonlinear data visualization tools constitute a key technology in modern data analysis. Within the group, several different lines of research are taken in this frame including visualization models with high functionality such as an explicit mapping and its approximate inverse [2] or methods which help to shape the inherently ill-posed task of data visualization such as the incorporation of auxiliary information [6].

Biomedical applications

Apart from applications for technical systems, a large application area where the algorithms as developed and investigated in the TCS group concerns biomedical applications. Here complex settings and the necessity of human insight into the models make the methods developed in the TCS group ideal candidates for automatic data analysis tools. Application areas include, for example, the analysis of mass spectrometric data for proteomic profiling or rapid bacteria identification [16, 11, 12, 10].

References

[1] N. Alex, A. Hasenfuss, and B. Hammer. Patch clustering for massive data sets. *Neurocomputing*, 72(7-9):1455–1469, 2009.

[2] B. Arnonkijpanich, A. Hasenfuss, and B. Hammer. Local matrix adaptation in clustering and applications for manifold visualization. *Neural Networks*, 23(4):476–486, 2010.

[3] B. Arnonkijpanich, A. Hasenfuss, and B. Hammer. Local matrix adaptation in topographic neural maps. *Neurocomputing*, to appear.

[4] A.W.Witolaer, A.Ghosh, J. de Vries, B. Hammer, and M.Biehl. Window-based example selection in learning vector quantization. *Neural Computation*, 22(11):2924–2961, 2010.

[5] M. Biehl, A. Ghosh, and B. Hammer. Dynamics and generalization ability of LVQ algorithms. *Journal of Machine Learning Research*, 8:323–360, 2007.

[6] K. Bunte, B. Hammer, A. Wismueller, and M. Biehl. Adaptive local dissimilarity measures for discriminative dimension reduction of labeled data. *Neurocomputing*, 73(7-9):1074–1092, 2010.

[7] T. Geweniger, D. Zülke, B. Hammer, and T. Villmann. Median fuzzy-c-means for clustering dissimilarity data. *Neurocomputing*, 73(7-9):1109–1116, 2010.

[8] A. Gisbrecht and B. Hammer. Relevance learning in generative topographic mapping. *Neurocomputing*, to appear.

[9] B. Hammer and A. Hasenfuss. Topographic mapping of large dissimilarity datasets. *Neural Computation*, 22(9):2229–2284, 2010.

[10] F.-M. Schleif, B. Hammer, M. Kostrzewa, and T. Villmann. Exploration of mass-spectrometric data in clinical proteomics using learning vector quantization methods. *Briefings in Bioinformatics*, 9(2):129–143, 2008.

[11] F.-M. Schleif, B. Villmann, M. Kostrzewa, B. Hammer, and A. Gammerman. Cancer informatics by prototype networks in mass spectrometry. *Artificial Intelligence in Medicine*, 45(2-3):215–228, 2009.

[12] F.-M. Schleif, T. Villmann, and B. Hammer. Prototype based fuzzy classification in clinical proteomics. *International Journal of Approximate Reasoning*, 47(1):4–16, 2008.

[13] P. Schneider, M. Biehl, and B. Hammer. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21:3532–3561, 2009.

[14] P. Schneider, M. Biehl, and B. Hammer. Distance learning in discriminative vector quantization. *Neural Computation*, 21:2942–2969, 2009.

[15] P. Schneider, K. Bunte, H. Stiekema, B. Hammer, T. Villmann, and M. Biehl. Regularization in matrix relevance learning. *IEEE Transactions on Neural Networks*, 21:831–840, 2010.

[16] S. Simmuteit, F.-M. Schleif, T. Villmann, and B. Hammer. Evolving trees for the retrieval of mass spectrometry-based bacteria fingerprints. *Knowledge and Information Systems*, 25(2):327–343, 2010.

[17] T. Villmann, B. Hammer, F.-M. Schleif, W. Hermann, and M. Cottrell. Fuzzy classification using information theoretic learning vector quantization. *Neurocomputing*, 16-18:3070–3076, 2008.

[18] A. Witoelar, M. Biehl, A. Ghosh, and B. Hammer. Learning dynamics and robustness of vector quantization and neural gas. *Neurocomputing*, 71:1210–1219, 2008.

Patch Affinity Propagation

Xibin Zhu*, Barbara Hammer†
Theroretical Computer Science,
CITEC, Bielefeld University,
Germany

Acknowledgements: This work has been supported by the Cluster of Excellence 277 Cognitive Interaction Technology funded in the framework of the German Excellence Initiative.

Abstract

Affinity Propagation (AP) is a novel exemplar-based clustering algorithm, which is fast and finds more reliably clusters than other exemplar-based methods, e.g. k-centers [1]. But one constraint of AP is given by its quadratic space complexity due to its dependency on the similarity matrix. In consequence, data can often not be loaded in the main memory, and AP cannot work. In this paper we extend AP to patch clustering so that it can also work for large data sets. We test this extension of AP, called Patch Affinity Propagation (PAP) on benchmark data, e.g. 11clouds, and also on a huge real-life data set, the idw text data set. The results show that PAP works very well.

1 Affinity Propagation

Affinity Propagation (AP) is an exemplar-based clustering algorithm; it finds exemplars by passing messages between data points [1]. Affinity propagation takes pairwise similarities as input, where the similarity $s(i,k) = s_{ik}$ indicates how suitable data point k is as exemplar of data point i. The goal of affinity propagation is to maximize the cost function C given by the global similarity of data points to their exemplars:

$$\begin{aligned} C &= \frac{1}{2}\sum_{i=1}^{N} s(i, I(i)) + \sum_{i=1}^{N} \delta_i(I(i)) \\ I &: \{1, \cdots, N\} \to \{1, \cdots, N\} \\ \delta_i(I) &= \begin{cases} 0 & I(i) = i, I(j) = i \\ -\infty & \text{otherwise} \end{cases} \end{aligned} \tag{1}$$

N is the number of data points; the function $I(i)$ is the assignment of data point i to its exemplar; $\delta_i(I)$ is a punishment of this assignment: if the assignment is valid, which means data point j chooses point i as its exemplar, and i also chooses itself as exemplar, then the assignment will

*E-Mail: *xzhu@techfak.uni-bielefeld.de*
†E-Mail: *bhammer@techfak.uni-bielefeld.de*

not be punished, otherwise it will be punished by $-\infty$ [2]. AP optimizes this cost function by means of a factor graph formalism for which the max-sum algorithm yields to an approximate optimization. There are two types of messages that can be calculated and iteratively updated: *responsibility* and *availability*.

Update of responsibilities:

$$r_{ik} = s_{ik} - \max_{k' \neq k} \left\{ a_{ik'} + s_{ik'} \right\} \tag{2}$$

Update of availabilities:

$$\begin{aligned} a_{ik} &= \min \left\{ 0, r_{kk} + \sum_{i' \neq i,k} \max\{0, r_{i'k}\} \right\} \\ a_{kk} &= \sum_{i' \neq k} \max\{0, r_{i'k}\} \end{aligned} \tag{3}$$

The final assignments of data points to their exemplars are determined by searching the maximal sum of responsibilities and availabilities:

$$I(i) = \underset{k}{\operatorname{argmax}}(a_{ik} + r_{ik}) \tag{4}$$

The number of clusters is determined by the self-similarities s_{kk}, also called *preferences*, so it is not necessary to give the number of clusters in advance [1]. Bigger values of preferences lead to more clusters, and on the contrary small values lead to less clusters. As default the preferences of all data points are set to the median of the similarities, so that all data points have the same chance to become an exemplar.

2 AP for large data sets

The update rules of AP are simple, and can be easily implemented. AP works faster and finds clusters with lower error than other exemplar-based methods, e.g. k-centers[1]. As most similarity based clustering approaches, affinity propagation runs in the worst case in time $O(N^2)$, i.e. it is infeasible for huge data sets.

Patch clustering tries to get around this problem by means of an iterative consideration of the data. Basically, instead of taking all the data at once, the data are considered in patches of size P. First, data 1 to P are considered, then data $P+1$ to $2P$ and so on, until all data have been considered [4]. Data are clustered consecutively, whereby the exemplars of the previous patch serve as additional data points for the next patch. These exemplars represent the data points in their receptive field, and they are weighted with the number of points in their receptive field, also called *multiplicity*. Thus, affinity propagation is iteratively executed using points in the new patch and the exemplars of the previous patch, which are weighted with multiplicities, as inputs. The pseudocode of patch clustering is shown as Algorithm 1.

Instead of taking all the data points of the previous patch, only the exemplars and their multiplicities are taken for the next patch. Due to this compressed representation, information loss usually takes place. However, we will see in experiments that patch AP closely resembles the performance of AP, i.e. the information loss is usually not severe.

Algorithm 1 Pseudocode of patch clustering

function PATCH-CLUSTERING(*patchsize*)
  $P = patchsize$
  $l = 0$
  init (weighted) exemplars E as the empty set
  repeat
    read next patch of data $B = \{x_{l \cdot P+1}, ..., x_{(l+1) \cdot P}\}$
      i.e. the similarities $s(x_i, x_j)$, where $x_i, x_j \in B$
    read the similarities of data in B and the exemplars in E,
      i.e. $s(x_i, x_j)$ where $x_i \in B$, $x_j \in E$, vice versa
    read the similarities of the exemplars in E,
      i.e. $s(x_i, x_j)$ where $x_i, x_j \in E$
    read the preferences of data in B and the exemplars in E
      i.e. $p(x_i)$ where $x_i \in B$ or E
    call *affinity propagation for multiplicities* on these similarities and
        preferences, where points $x_i \in E$ are taken with multiplicities
    reset E as the exemplars found by AP for multiplicities, weight
        the points in E according to the number of points assigned
        to them, counted with multiplicities
    $l = l + 1$
  until all data are read or all similarities are read
end function

3 Affinity propagation for multiplicities

To apply patch clustering, it is necessary to extend affinity propagation so that it can take multiplicities of data points into account, because original AP works only for standard data points, which corresponds to multiplicities of data points equal to one ($m_i = 1$). One possibility to extend AP to multiple points is via the underlying cost function. We assume the multiplicity of data point i is denoted as m_i. Then, the cost function (overall similarity of data points to their exemplars) of affinity propagation becomes

$$C = \frac{1}{2} \sum_{i=1}^{N} m_i \cdot s(i, I(i)) + \sum_{i=1}^{N} \delta_i(I(i)) \tag{5}$$

Thus, the multiplicities can be taken into account by simply multiplying the similarities by the multiplicities m_i, i.e. we have to multiply the similarities corresponding to i by the multiplicity m_i. So the update formula of responsibilities 2 should be changed like follows

$$r_{ik} = m_i \cdot s_{ik} - \max_{k' \neq k} \left\{ a_{ik'} + m_i \cdot s_{ik'} \right\} \tag{6}$$

Because the availabilities only depend on the responsibilities, they are not changed. Furthermore the preferences should also take the multiplicities into account so that the data points with bigger multiplicities can be chosen as an exemplar with a higher probability than the data points with smaller multiplicities. For instance, if the similarity is a negative euclidian distance, then the preferences can be adapted like following

$$p_i = p_i / m_i \tag{7}$$

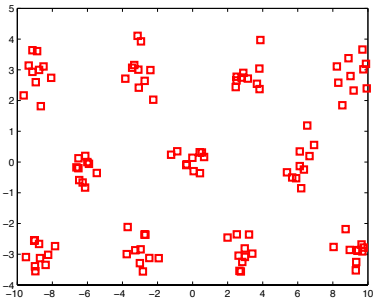


Figure 1: *11 Clouds* consists of 110 data points, which are separated into 11 apart sets.

4 Patch Affinity Propagation

With the new affinity propagation for multiplicities patch clustering can be realized and implemented, we call this integration of affinity propagation and the patch idea **Patch Affinity Propagation** (PAP). There are two methods of patch affinity propagation that are implemented which differ with respect to the heuristics according to which the preferences are set for every new patch:

PAP_K: by means of iteratively adjusting the preferences this method tries to find a given number of clusters (e.g. K) or a number of clusters in a given interval (e.g. between K-5 and K+5) for each patch run.

PAP_Heuristic: by means of adjusting the preferences this method tries to find more exemplars in the earlier patch runs so that the information loss of data points is low, and in the later patch runs the number of exemplars is decreased, until eventually a reasonable number of exemplars can be found.

Both methods work well in the tests (see chapter 5), but PAP_K takes more time than PAP_Heuristic, because it adjusts the preferences many times to find K exemplars for each patch run.

5 Tests and evaluation

First PAP is tested on the benchmark data set, *11 Clouds*, which contains 11 separate sets of 2D data points, each with 10 points (see figure 1), then PAP is tested on a large real-life data set, the idw text data set.

11Clouds

For different patch sizes both methods of PAP are tested on 11 Clouds. Tabel 1 shows the results of PAP_K, in this case we set an interval of number of clusters between 5 and 17 so that PAP_K can always find an appropriate number of clusters for each earlier patch run. Table 2 shows the results of the PAP_Heurisitic.

mQErrTrain is the average quantization error of the training data, *mQErrTest* is the average quantization error of the test data, *mStd* is the

PAP_K, ***11Clouds***, 10-fold cross validation, 50 runs according to 11 classes, prc=60%, $K \in [5, 17]$					
patch size	mTime[m.]	mQErrTrain(mStd)	mQErrTest(mStd)	mHitrateTrain(mStd)	mHitrateTest(mStd)
10	156.66	-51.024(0.551)	-6.869(0.281)	**1.000(0.000)**	**1.000(0.000)**
20	116.81	-49.030(0.488)	-6.336(0.366)	**1.000(0.000)**	**1.000(0.000)**
30	87.77	-46.824(0.402)	-5.829(0.342)	**1.000(0.000)**	**1.000(0.000)**
40	31.80	-47.517(0.402)	-6.389(0.334)	**1.000(0.000)**	**1.000(0.000)**
50	23,34	-46.160(0.351)	-5.924(0.306)	**1.000(0.000)**	**1.000(0.000)**

Table 1: evaluation of PAP_K (with an interval [5, 17]) for 11Clouds

PAP_Heuristic***11Clouds***, 10-fold cross validation, 50 runs					
patch size	mTime[m.]	mQErrTrain(mStd)	mQErrTest(mStd)	mHitrateTrain(mStd)	mHitrateTest(mStd)
10	**8.83**	-51.800(0.601)	-6.780(0.403)	**1.000(0.000)**	**1.000(0.000)**
20	**6.84**	-48.813(0.491)	-6.282(0.369)	**1.000(0.000)**	**1.000(0.000)**
30	**6.92**	-46.474(0.416)	-5.850(0.280)	**1.000(0.000)**	**1.000(0.000)**
40	**6.71**	-47.517(0.495)	-6.389(0.419)	**1.000(0.000)**	**1.000(0.000)**
50	**6.33**	-46.160(0.404)	-5.924(0.309)	**1.000(0.000)**	**1.000(0.000)**

Table 2: evaluation of PAP_Heuristic for 11Clouds

corresponding average standard deviation; *mHitrateTrain* is the average hit rate of training data, *mHitrateTest* is the average hit rate of test data, *mStd* is the corresponding average standard deviation; *mTime* is the average runtime.

For 11Clouds both methods work well, every data point (either in the training data or the test data) can be correctly classified into its class. But a significant difference is the runtime, PAP_K took much more time than PAP_Heuristic, because it adjusted the preferences many times to find an appropriate number of clusters for each patch.

idw data set

idw[1] is an abbreviation for "Informationsdienst Wissenschaft" (Information service for the sciences), it collects over 190,000 text documents (so far) in overall 33 areas, e.g. biology, medicine, politics, computer science, etc. For our tests we took nearly 80.000 texts, and preprocessed the texts by standard text preprocessing methods like *stopwords reduction*, *stemming*. We use a dimensionality reduction given by *random projection*. Because the idw texts are multi-labeled, we use the α-evaluation[5] to evaluate the results of PAP, which is a generalized version of the Jaccard Similarity Metric[3]. Tables 3 and 4 show the results of the two methods of PAP on the idw data set for different patch sizes and dimensions. After stopword reduction and stemming there are still 54553 words left, which means the dimensionality is still high, so we used random projection to reduce the high dimensions to 100, 500, and 1000 dimensions, respectively.

On the idw data both PAP methods work also well. For different patch sizes the accuracy in low-dimensional space is comparable with the accuracy of the original high-dimensional space, e.g. in 1000 dimensions the accuracy is almost the same as for 54553 dimensions. Although the average runtime in low dimensions is significantly shorter than in high dimensions, the accuracy is not significantly worse, just nearly 2%.

[1] `http://idw-online.de`

PAP_K, ***idw***(79810 documents, 54553 words/dimensions)
5-fold cross validation, 5 runs
according to 33 areas, $prc = 20\%, K \in [27, 39]$

α-evaluation: $\alpha = 1, \beta = 1, \gamma = 1/4$

Patch	Dim/RP	mTime [m.]	mCATrain_ML (*mStd*)	mCATest_ML (*mStd*)	mCATrain_ML_Simp (*mStd*)	mCATest_ML_Simp (*mStd*)
300	54553	85.59	**0.677(0.001)**	**0.677(0.001)**	0.767(0.001)	0.767(0.001)
300	100	29.96	0.666(0.002)	0.666(0.002)	**0.773(0.001)**	**0.773(0.001)**
300	500	27.82	0.667(0.001)	0.666(0.001)	0.764(0.002)	0.764(0.002)
300	1000	34.08	**0.669(0.001)**	**0.668(0.001)**	0.762(0.001)	0.762(0.002)
1000	54553	382.36	**0.680(0.001)**	**0.681(0.001)**	0.761(0.001)	0.763(0.001)
1000	100	219.98	0.663(0.001)	0.662(0.001)	**0.765(0.001)**	**0.765(0.001)**
1000	500	248.07	0.668(0.001)	0.668(0.001)	0.759(0.001)	0.759(0.001)
1000	1000	322.74	**0.680(0.001)**	**0.679(0.001)**	0.760(0.001)	0.759(0.002)

Table 3: Results of PAP_K on idw data

PAP_Heuristic, ***idw***(79810 documents, 54553 words/dimentions)
5-fold cross validation, 5 runs

α-evaluation: $\alpha = 1, \beta = 1, \gamma = 1/4$

Patch	Dim/RP	mTime [m.]	mCATrain_ML (*mStd*)	mCATest_ML (*mStd*)	mCATrain_ML_Simp (*mStd*)	mCATest_ML_Simp (*mStd*)
300	54553	74.90	**0.667(0.001)**	**0.668(0.002)**	0.765(0.001)	0.765(0.002)
300	100	**19.94**	0.667(0.002)	0.667(0.002)	**0.773(0.001)**	**0.777(0.002)**
300	500	32.04	0.667(0.001)	0.667(0.001)	0.765(0.001)	0.765(0.001)
300	1000	**22.98**	**0.670(0.003)**	**0.670(0.003)**	0.764(0.003)	0.764(0.003)
1000	54553	229.41	**0.680(0.002)**	**0.681(0.001)**	0.761(0.001)	0.763(0.001)
1000	100	48.91	0.664(0.000)	0.663(0.002)	0.763(0.001)	0.762(0.001)
1000	500	51.49	0.675(0.001)	0.674(0.001)	**0.764(0.001)**	**0.763(0.001)**
1000	1000	106.52	**0.680(0.001)**	**0.679(0.001)**	0.762(0.001)	0.761(0.001)

Table 4: Results of PAP_Heuristic on idw data

6 Conclusion

With the simple idea of patch clustering affinity propagation can be used on huge data sets. We demonstrated that it works well not just for the 11 Clouds benchmark, but also for the real data set given by the idw data set. Thus PAP offers a promising tool for clustering on huge data sets. In the future more more tests of PAP will be executed, e.g. on biological data sets.

References

[1] Frey, Brendan J. ; Dueck, Delbert: Clustering by Passing Messages between Data Points. In: *Science* 315 (2007), 972-976. `http://www.psi.toronto.edu/affinitypropagation/`

[2] Frey, Brendan J. ; Dueck, Delbert: Supporting Online Material for AP. (2007). `http://www.sciencemag.org/cgi/content/full/1136800/DC1`

[3] Gower, J.C. ; Legendre, P.: Metric and euclidiean properties of dissimilarity coefficients. In: *J. Classification 3* (1986), S. 5–48

[4] N. Alex, A. H. ; Hammer, B.: Patch clustering for massive data sets. In: *Neurocomputing* 72, Nr. 7-9, S. 1455–1469

[5] R.Boutell, Matthew: Learning multi-label scene classification. In: *Elsevier* Pattern Recognition (2004), Nr. 37, S. 1757–1771

Relational generative topographic mapping for large data sets

Andrej Gisbrecht[1,2], Bassam Mokbel[1], Barbara Hammer[1]

Acknowledgements: This work has been supported by the Cluster of Excellence 277 Cognitive Interaction Technology funded in the framework of the German Excellence Initiative.

1 Introduction

A wealth of dimensionality reduction methods exists, including t-SNE, LLE, MVU, Laplacian eigenmaps, or Isomap, for example [7, 9, 8]. In a nutshell, these technologies rely on the principle to map given data points to low dimensions such that important data characteristics are preserved as much as possible. The methods differ in the choice of the data characteristics (e.g. pairwise distances, locally linear relationships, pairwise probabilities, ...) and the way in which the objective is optimized (using e.g. spectral techniques or numerical optimization).

Unlike these approaches, the generative topographic mapping (GTM) takes a fundamentally different perspective [1]. It is based on a generative statistical model which explains the observed data distribution. The parameters are adapted such that the data log likelihood is maximized. The statistical model is constraint such that the single Gaussian modes can smoothly be associated with points in a low dimensional latent space. This way, data visualization is obtained as a by-product of the method. Being based on a Gaussian mixture model, it offers additional functionalities such as data clustering, neighborhood browsing, outlier detection, and direct out of sample extensions.

Original GTM has been proposed for Euclidean data. Often, data display a specific non-Euclidean structure: biological sequences and their alignment, graph structures and corresponding kernels, or text and corresponding information theoretic distances. In these cases, it is much more appropriate to work with the given dissimilarities rather than to enforce a feature representation of data. GTM has recently been extended to general dissimilarities, relational GTM (RGTM) [2, 3]. While the mapping and visualization obtained this way display a high quality for a number of benchmarks [2, 3], the method has a drawback: it has squared complexity and, thus, it is not feasible for large data sets - this problem is common for all methods which rely on pairwise dissimilarities.

In this contribution, we show that, for RGTM, due to its algebraic formulation of the training problem, the Nyström approximation of the dissimilarity matrix leads to a linear time approximation of the full procedure. One can prove that this approximation is exact if the approximation mirrors the embedding space of the given dissimilarity data. For smaller sizes, the approximation may lead to worse results. We test the method in two real world examples.

[1]Center of Excellence for Cognitive Interaction Technology, Bielefeld University, D-33594 Bielefeld, Germany
[2]E-mail: agisbrec@techfak.uni-bielefeld.de

2 The generative topographic mapping

GTM: We shortly review GTM and its extension to relational data. Given data $\mathbf{x} \in \mathbb{R}^D$, GTM defines a constraint mixture of Gaussians with centers induced by a regular lattice of points $\mathbf{w}$ in latent space. The prototypes are mapped to target vectors $\mathbf{w} \mapsto \mathbf{t} = y(\mathbf{w}, \mathbf{W})$ in the data space, where the function y is typically chosen as a generalized linear regression model $y : \mathbf{w} \mapsto \Phi(\mathbf{w}) \cdot \mathbf{W}$ induced by base functions Φ such as equally spaced Gaussians with bandwidth σ. Every latent point induces a Gaussian distribution

$$p(\mathbf{x}|\mathbf{w}, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left(-\frac{\beta}{2}\|\mathbf{x} - y(\mathbf{w}, \mathbf{W})\|^2\right) \tag{1}$$

with variance β^{-1}. In total, a mixture of K modes

$$p(\mathbf{x}|\mathbf{W}, \beta) = \sum_{k=1}^{K} \frac{1}{K} p(\mathbf{x}|\mathbf{w}_k, \mathbf{W}, \beta) \tag{2}$$

is generated. GTM training optimizes the data log-likelihood

$$\ln\left(\prod_{n=1}^{N}\left(\sum_{k=1}^{K} p(\mathbf{w}_k)p(\mathbf{x}_n|\mathbf{w}_k, \mathbf{W}, \beta)\right)\right) \tag{3}$$

with respect to $\mathbf{W}$ and β. This can be done by means of an EM approach which treats the generative mixture component $\mathbf{w}_k$ for a data point $\mathbf{x}_n$ as hidden parameter. In explicit formulas, responsibilities

$$R_{kn}(\mathbf{W}, \beta) = p(\mathbf{w}_k|\mathbf{x}_n, \mathbf{W}, \beta) = \frac{p(\mathbf{x}_n|\mathbf{w}_k, \mathbf{W}, \beta)p(\mathbf{w}_k)}{\sum_{k'} p(\mathbf{x}_n|\mathbf{w}_{k'}, \mathbf{W}, \beta)p(\mathbf{w}_{k'})} \tag{4}$$

of component k for point number n, and the model parameters by means of the formulas

$$\mathbf{\Phi}^T \mathbf{G}_{\text{old}} \mathbf{\Phi} \mathbf{W}_{\text{new}}^T = \mathbf{\Phi}^T \mathbf{R}_{\text{old}} \mathbf{X} \tag{5}$$

for $\mathbf{W}$ are subsequently computed until convergence, where $\mathbf{\Phi}$ refers to the matrix of base functions Φ evaluated at points $\mathbf{w}_k$, $\mathbf{X}$ to the data points, $\mathbf{R}$ to the responsibilities, and $\mathbf{G}$ is a diagonal matrix with accumulated responsibilities $G_{nn} = \sum_n R_{kn}(\mathbf{W}, \beta)$. The variance can be computed by solving

$$\frac{1}{\beta_{\text{new}}} = \frac{1}{ND} \sum_{k,n} R_{kn}(\mathbf{W}_{\text{old}}, \beta_{\text{old}}) \|\Phi(\mathbf{w}_k)\mathbf{W}_{\text{new}} - \mathbf{x}_n\|^2 \tag{6}$$

where D is the data dimensionality and N the number of data points. Usually, GTM is initialized referring to PCA to avoid convergence to local optima.

Relational GTM: We assume that data $\mathbf{x}$ are given by pairwise dissimilarities $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$ with corresponding dissimilarity matrix D, where the vector representation $\mathbf{x}$ of the data is unknown. As pointed out in [6, 4], if prototypes are restricted to linear combinations of data points of the form $\mathbf{t}_k = \sum_{n=1}^{N} \alpha_{kn}\mathbf{x}_n$ with $\sum_{n=1}^{N} \alpha_{kn} = 1$ the prototypes $\mathbf{t}_k$ can be represented indirectly by means of the coefficient vector α_k and, further, distances of data points and prototypes can be computed by

$$\|\mathbf{x}_n - \mathbf{t}_k\|^2 = [\mathbf{D}\alpha_k]_n - \frac{1}{2} \cdot \alpha_k^T \mathbf{D} \alpha_k \tag{7}$$

where $[\cdot]_i$ is component i of the vector.

As before, the targets $\mathbf{t}_k$ induce a Gaussian mixture distribution in the data space. They are obtained as images of points $\mathbf{w}$ in latent space via a generalized linear regression model. Since the embedding space of $\mathbf{t}_k$ is not known, we directly treat the mapping as a mapping to coefficients: $y : \mathbf{w}_k \mapsto \alpha_k = \Phi(\mathbf{w}_k) \cdot \mathbf{W}$ where, now, $\mathbf{W} \in \mathbb{R}^{d \times N}$. This corresponds to a generalized linear regression of the latent space into the (unknown) surrounding vector space due to the linear dependency of the targets and coefficients. In the α-space of linear combinations of data points, data points $\mathbf{x}_i$ itself are represented by unit vectors, in consequence, the data matrix $\mathbf{X}$ is now the unit matrix $\mathbf{I}$.

To apply (7), we put the restriction $\sum_n [\Phi(\mathbf{w}_k) \cdot \mathbf{W}]_n = 1$. This way, the likelihood function can be computed based on (1) where the distance computation can be performed indirectly using (7). As for GTM, we can use an EM optimization scheme to arrive at solutions for the parameters β and $\mathbf{W}$, where, again, the mode $\mathbf{w}_k$ responsible for data point $\mathbf{x}_n$ serves as hidden parameter. An EM algorithm in turn computes the responsibilities (4) using the alternative formula for the distances (7), and it optimizes the expectation $\sum_{k,n} R_{kn}(\mathbf{W}_{\text{old}}, \beta_{\text{old}}) \ln p(\mathbf{x}_n | \mathbf{w}_k, \mathbf{W}_{\text{new}}, \beta_{\text{new}})$ with respect to $\mathbf{W}$ and β under the above constraint on $\mathbf{W}$. Using Lagrange optimization one can see that the optimum automatically fulfills the constraints.

Hence the model parameters can be determined in analogy to (5,6) where, now, functions Φ map from the latent space to the space of coefficients α. We refer to this iterative update scheme as relational GTM (RGTM). Initialization of RGTM can take place by referring to the first MDS directions of the given dissimilarity matrix.

3 The Nyström method

We shortly review the Nyström technique as presented in [10]. By the Mercer theorem kernels $k(\mathbf{x}, \mathbf{y})$ can be expanded by orthonormal eigenfunctions ϕ_i and non negative eigenvalues λ_i in the form $k(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{y})$. If k is represented by a matrix, the number of non zero eigenvalues is given by the rank. The eigenfunctions and eigenvalues of a kernel are the solution of $\int k(\mathbf{y}, \mathbf{x}) \phi_i(\mathbf{x}) p(\mathbf{x}) dx = \lambda_i \phi_i(\mathbf{y})$, which can be approximated based on the Nyström method by sampling $\mathbf{x}_k$ i.i.d. according to p: $\frac{1}{m} \sum_{k=1}^{m} k(\mathbf{y}, \mathbf{x}_k) \phi_i(\mathbf{x}_k) \approx \lambda_i \phi_i(\mathbf{y})$. Using the matrix eigenproblem $\mathbf{K}^{(m)} \mathbf{U}^{(m)} = \mathbf{U}^{(m)} \mathbf{\Lambda}^{(m)}$ of the $m \times m$ Gram matrix $\mathbf{K}^{(m)}$ we can derive the approximations for the eigenfunctions and eigenvalues

$$\lambda_i \approx \frac{\lambda_i^{(m)}}{m}, \quad \phi_i(\mathbf{y}) \approx \frac{\sqrt{m}}{\lambda_i^{(m)}} \mathbf{k}_y \mathbf{u}_i^{(m)}, \tag{8}$$

where $\mathbf{u}_i^{(m)}$ is the ith column of $\mathbf{U}^{(m)}$. Thus, we can approximate ϕ_i at an arbitrary point $\mathbf{y}$ as long as we know the vector $\mathbf{k}_y = (k(\mathbf{x}_1, \mathbf{y}), ..., k(\mathbf{x}_m, \mathbf{y}))^T$.

One well known way to approximate a $n \times n$ Gram matrix, is to use a low-rank approximation. This can be done by computing the eigendecomposition of the kernel $\mathbf{K} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T$, where $\mathbf{U}$ is orthonormal and $\mathbf{\Lambda}$ is diagonal with $\mathbf{\Lambda}_{11} \geq \mathbf{\Lambda}_{22} \geq ... \geq 0$, and keeping only the m eigenspaces which correspond to the m largest eigenvalues of the matrix. The approximation is $\mathbf{K} \approx \mathbf{U}_{n,m} \mathbf{\Lambda}_{m,m} \mathbf{U}_{m,n}$, where $\mathbf{A}_{b,c}$ notes the matrix with the b first rows and c first columns of $\mathbf{A}$. The Nyström method can approximate a kernel in a similar way, without computing the eigendecomposition of the whole matrix, which is an $O(n^3)$ operation. For a given $n \times n$ Gram matrix $\mathbf{K}$ we randomly choose m rows and respective columns.

After permutation, we assume without loss of generality that these are the first m rows and columns. We denote these rows by $\mathbf{K}_{m,n}$ and columns by $\mathbf{K}_{n,m}$, which are transposes of each other, since the matrix is symmetric. Using the formulas (8) we obtain $\tilde{\mathbf{K}} = \sum_{i=1}^{m} 1/\lambda_i^{(m)} \cdot \mathbf{K}_{n,m}\mathbf{u}_i^{(m)}(\mathbf{u}_i^{(m)})^T\mathbf{K}_{m,n}$, where $\lambda_i^{(m)}$ and $\mathbf{u}_i^{(m)}$ correspond to the $m \times m$ eigenproblem. In the case that some $\lambda_i^{(m)}$ are zero, we replace the corresponding fractions with zero. Thus we get, $\mathbf{K}_{m,m}^{-1}$ denoting the Moore-Penrose Pseudoinverse,

$$\tilde{\mathbf{K}} = \mathbf{K}_{n,m}\mathbf{K}_{m,m}^{-1}\mathbf{K}_{m,n}. \tag{9}$$

For a given matrix $\mathbf{K}$ with rank m, this approximation is exact, if the m chosen m-dimensional points are linearly independent.

4 Nyström approximation for dissimilarities

Originally the Nyström method was presented for positive semidefinite Gram matrices. For dissimilarity data, a direct transfer is possible: A symmetric dissimilarity matrix $\mathbf{D}$ is a normal matrix and according to the spectral theorem can be diagonalized $\mathbf{D} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ with $\mathbf{U}$ an unitary matrix whose column vectors are the orthonormal eigenvectors of $\mathbf{D}$ and $\mathbf{\Lambda}$ a diagonal matrix with the eigenvalues of $\mathbf{D}$, which can be negative for non-Euclidean distances. Therefore the dissimilarity matrix can be seen as an operator $d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} \lambda_i \phi_i(\mathbf{x})\phi_i(\mathbf{y})$ where $\lambda_i \in \mathbb{R}$ correspond to the diagonal elements of $\mathbf{\Lambda}$ and ϕ_i denote the eigenfunctions. As we can see, the only difference to an expansion of a kernel is that the eigenvalues are allowed to be negative. All further mathematical manipulations can be applied in the same way.

Using the approximation (9) for the distance matrix, we can apply this result for RGTM. It allows to approximate (7) in the way

$$\|\mathbf{x}_n - \mathbf{t}_k\|^2 \approx \left[\mathbf{D}_{n,m}\left(\mathbf{D}_{m,m}^{-1}\left(\mathbf{D}_{m,n}\alpha_k\right)\right)\right]_n - \frac{1}{2} \cdot \left(\alpha_k^T\mathbf{D}_{n,m}\right) \cdot \left(\mathbf{D}_{m,m}^{-1}\left(\mathbf{D}_{m,n}\alpha_k\right)\right) \tag{10}$$

which is $\mathcal{O}(m^2 n)$ instead of $\mathcal{O}(n^2)$, i.e. it is linear in the number of data points n, assuming fixed approximation m. Again, the approximation is exact if m suits the rank of the matrix.

5 Experiments

We test the applicability of the Nyström technique for RGTM on two real life data sets: The Copenhagen chromosomes data set as presented in ([5]) contains 4200 data from 22 classes representing distances of grey valued images of chromosomes using a suitable dissimilarity measure. The idw data set contains 79810 articles with scientific news from the data base 'Informationsdienst Wissenschaft' (a German service organization gathering research news, see http://idw-online.de) which are multi-labeled with 8 categories. The articles are preprocessed by stop word reduction, stemming, and random projection from 54553 to 100 dimensions, using cosine dissimilarity afterwards.

Chromosomes: For the chromosomes data, we use 50 cycles, 10×10 base functions which standard deviation is set to the distance between two neighboring basis function centers, and 40×40 latent points. We report the classification rate obtained by a 10-fold cross-validation with 5 repeats and a different percentage of points $m \in \{2, 90\}$ for the Nyström approximation as depicted in

Fig. 1. The rightmost number corresponds to a standard RGTM setting without Nyström approximation. The red line denotes the time in seconds used for training.

The distance matrix of the chromosomes data set has a high rank and many large eigenvalues. Therefore, it cannot easily be approximated by a low rank matrix. This fact is mirrored by the graph shown in Fig. 1: while the approximation leads to a considerable speedup, it also causes a loss of information corresponding to a decrease of the accuracy from almost .9 to less than .6.

Idw data: The idw data set is trained for the same number of epochs, using 10×10 latent points and 3×3 base functions. Unlike for the Chromosomes data, a Nyström approximation using $m = 101$ leads to an exact reconstruction of the matrix due to the inherent low dimensionality of the data. Posterior labeling, where a relative cutoff of .8 is used, leads to a visualization of the represented classes as shown in Fig. 2. The resulting topographic mapping allows an intuitive inspection and retrieval of the main categories as present in the data set.

6 Conclusions

We investigated the suitability of the Nyström method to speed up relational topographic maps for large data sets. While the technique leads to linear effort, its suitability severely depends on the intrinsic dimensionality of the given dissimilarity matrix. As demonstrated in the experiments, there can be a considerable loss of information if the dimensionality is higher.

References

[1] C. Bishop, M. Svensen, and C. Williams. The generative topographic mapping. *Neural Computation* 10(1):215-234, 1998.

[2] A. Gisbrecht, B. Mokbel, and B. Hammer. Relational Generative Topographic Mapping. M. Verleysen (ed.), *ESANN 2010*: 277-282, 2010.

[3] A. Gisbrecht, B. Mokbel, A. Hasenfuss, and B. Hammer. Visualizing Dissimilarity Data using Generative Topographic Mapping. R. Dillmann, J. Beyerer, U.D. Hanebeck, T. Schultz (eds.), *KI 2010*: 227-237, 2010.

[4] A. Hasenfuss and B. Hammer. Relational Topographic Maps. M.R. Berthold, J. Shawe-Taylor, and N. Lavrac (eds.), *IDA 2007*: 93-105, 2007.

[5] B. Hammer and A. Hasenfuss. Topographic Mapping of Large Dissimilarity Data Sets. Neural Computation 22(9):2229-2284, 2010.

[6] R. J. Hathaway and J. C. Bezdek. Nerf c-means: Non-Euclidean relational fuzzy clustering. *Pattern Recognition* 27(3):429-437, 1994.

[7] L.J.P. van der Maaten and G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research* 9(Nov):2579-2605, 2008.

[8] L.J.P. van der Maaten, E.O. Postma, and H.J. van den Herik. *Dimensionality Reduction: A Comparative Review.* Tilburg University Technical Report, TiCC-TR 2009-005, 2009

[9] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *J. Mach. Learn. Res.*, 11:451–490, 2010.

[10] C. K. I. Williams, M. Seeger. Using the Nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems 13*: 682-688, 2001

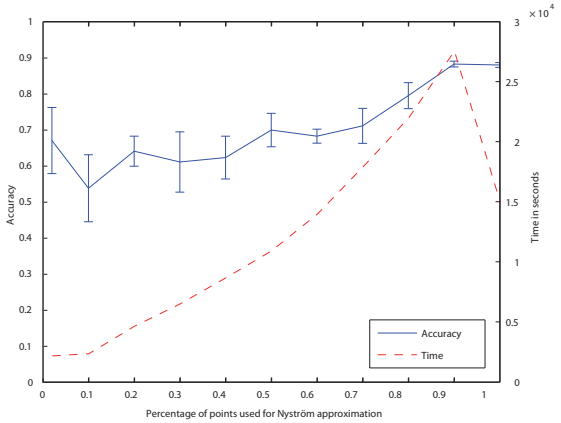


Figure 1: Accuracy and time when using the Nyström technique to speed up RGTM for Chromosomes

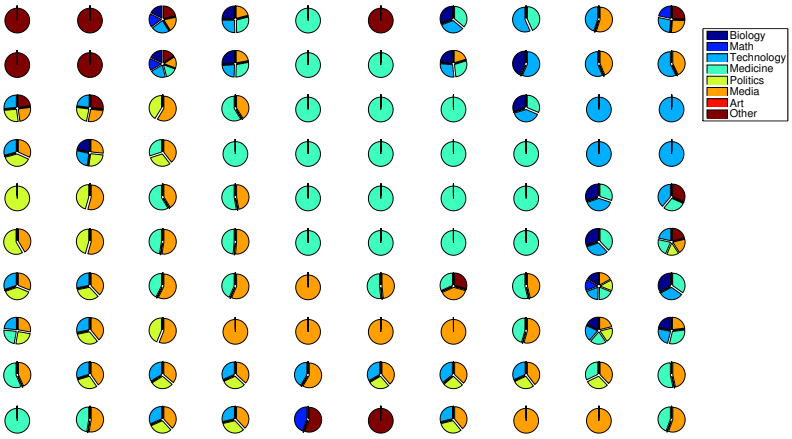


Figure 2: Visualization of the idw data set using RGTM and the Nyström approximation

Quality Assessment Measures for Dimensionality Reduction Applied on Clustering

Bassam Mokbel[1,2], Andrej Gisbrecht[1], Barbara Hammer[1]

Acknowledgements: This work has been supported by the Cluster of Excellence 277 Cognitive Interaction Technology funded in the framework of the German Excellence Initiative.

1 Introduction

The amount of electronic data available today increases rapidly in virtually all areas of life, such that visualization techniques constitute important interfaces between the digital information and the human user. In order to visualize high-dimensional data, there are numerous *dimensionality reduction (DR)* techniques available to map the data points to a low-dimensional space, e.g., the two-dimensional Euclidean plane, see [1, 2, 3] for an overview. As a general setting, original data are given as a set of N vectors $\mathbf{x}_i \in X \subset S^n,\ i \in \{1, \ldots, N\}$. Using DR, each data vector is mapped to a low-dimensional counterpart for visualization, called *target* $\mathbf{y}_k \in Y \subset \mathbb{R}^v,\ k \in \{1, \ldots, N\}$, where typically $n \gg v$ and $v = 2$. With an increasing number of such methods, a reliable assessment of the quality of produced visualizations becomes more and more important, in order to achieve comparability. One objective of DR is to preserve the available information as much as possible. In this sense, the reconstruction error $E_{\text{reconstr}} := \sum_i \|\mathbf{x}_i - f^{-1}(f(\mathbf{x}_i))\|^2$ where f denotes the DR mapping of the data, and f^{-1} its approximate inverse, could serve as a general quality measure. This has the drawback that, for most DR methods, no explicit mapping f is available and an approximate inverse f^{-1} is also not known. As an alternative, the existing quality assessment (QA) measures rely on statistics over input-versus-output discrepancies, which can be evaluated based solely on the given data points and their projections. Different QA approaches have been proposed in the last years, see [4] for an overview. These specialized measures represent a means to objectively assess an overall qualitative change under spatial transformation. Recently, two generalized approaches have been introduced that can serve as unifying frameworks, including some earlier QA concepts as special cases:

CRM: The *coranking matrix* and its derived quality measure, presented in [4].
IR: An *information retrieval* perspective measuring precision & recall for visualization, see [2].

These frameworks have been evaluated extensively in the context of DR tools,

[1]Center of Excellence for Cognitive Interaction Technology, Bielefeld University, D-33594 Bielefeld, Germany
[2]E-mail: bmokbel@techfak.uni-bielefeld.de

which map given data points to low-dimensional coordinates. The rapidly increasing size of modern data sets, however, causes the need to not only project single points, but to also priorly compress the available information. Hence, further steps such as, e.g., clustering become necessary. While the QA approaches can be used to evaluate DR methods, it is not clear in how far they can also reliably measure the quality of clustering. Conversely, typical QA measures for clustering such as the quantization error cannot be extended to DR methods, since this would lead to the (usually infeasible) reconstruction error. Hence, it is interesting to investigate if QA approaches for DR methods can be transferred to the clustering domain. This would open the way towards an integrated evaluation of the two steps.

2 Quality Assessment for Dimension Reduction

Coranking matrix (CRM) The CRM framework, presented in [4], offers a general approach to QA for DR. The coranking matrix is essentially a histogram over all rank errors in the given projection, see [4] for a detailed definition. In the originally proposed framework, ties of the ranks are broken deterministically, such that no two equal ranks occur. This has the advantage that several properties of the coranking matrix (such as constant row and column sum) hold, which are, however, not necessary for the evaluation measure. For our purposes, it is more suitable to allow equal ranks, e.g., if distances are identical. Based on the coranking matrix, various different quality measures can be computed. In our experiments, we only report the *overall quality* indicator, which is proposed as a reasonable objective, taking into account weighted averages of all intrusions and extrusions, see [4] for details.

Information retrieval (IR) In the IR framework, presented in [2], visualization is viewed as an information retrieval task. One data point $\mathbf{x}_i \in X$ is seen as a query of a user, which has a certain neighborhood A_i in the original data space, called *input neighborhood*. It represents the truthful, but unretrievable answer to the query. The retrieval result is based solely on the visualization which is presented to the user. There, the neighborhood of its respective target $\mathbf{y}_i \in Y$ is denoted by B_i, called the *output neighborhood*. If both neighborhoods are defined over corresponding notions of proximity, it becomes possible to evaluate, how truthful the query result is with respect to the given query. One can define the neighborhoods by a fixed distance radius α_d, valid in input, as well as in output space, so A_i and B_i consist of all data points (other than i itself), which have a smaller or equal distance to x_i and y_i respectively: Analogously, the neighborhoods can be defined by a fixed rank radius α_r, so the neighborhood sets A_i and B_i contain the α_r nearest neighbors. Note that A_i and B_i usually differ from each other due the projection of data to low dimensionality. These differences can be evaluated in terms of true positives, false positives, and misses for each query, i.e., data point i, which leads to the information retrieval measures *precision* and *recall*, see [2] for details. For a whole set X of data points, one can calculate the *mean precision* and *mean recall* by averaging over all data points $\mathbf{x}_i$, which we report for our experiments.

3 Quality Assessment for Clustering

Clustering aims at decomposing the given data $\mathbf{x}_i$ into homogeneous clusters, see [5]. Prototype-based clustering achieves this goal by specifying M prototypes $\mathbf{p}_u \in P \subset S^n,\ u \in \{1, \ldots, M\}$, which decompose the data by means of their receptive fields R_u, which are given by the points $\mathbf{x}_i$ closer to $\mathbf{p}_u$ than to any other prototype, breaking ties deterministically. Many prototype-based clustering algorithms exist, see, e.g., [6, 5]. If a large data set has to be visualized, a typical procedure is to first use the clustering algorithm to represent the dataset by a significantly smaller number of representative prototypes, and to visualize these prototypes in low dimensions, afterwards. In consequence, a formal evaluation of this procedure has to take into account both, the clustering step and the dimensionality reduction. To treat the two steps, clustering and visualization, within one common framework, we interpret clustering as a 'visualization' which maps data points to their closest prototype respectively: $\mathbf{x}_i \mapsto \mathbf{y}_i := \mathbf{p}_u$ such that $\mathbf{x}_i \in R_u$. In this case, the visualization space $\mathbb{R}^v$ coincides with the data space S^n. Obviously, by further projecting the prototypes, a 'proper' visualization could be obtained, which is equivalent to a classic dimensionality reduction problem, so it is not discussed here. The typical error measure for clustering is the quantization error $E_{\mathrm{qe}} := \sum_u \sum_{\mathbf{x}_i \in R_u} \|\mathbf{x}_i - \mathbf{p}_u\|^2$ which evaluates the averaged distance within clusters. Obviously, it coincides with the reconstruction error of visualization as introduced above. Hence, since the latter can usually not be evaluated for standard DR methods, the quantization error can not serve as evaluation for simultaneous clustering and visualization. As an alternative, one can investigate whether the QA tools for DR, as introduced above, give meaningful results for clustering algorithms. There exist some general properties of these measures which indicate that this leads to reasonable results: for fixed neighborhood radius α_r, an intrusion occurs only if distances between clusters are smaller than α_r; an extrusion occurs only if the diameter of a cluster is larger than α_r. Hence, the QA measures for DR punish small between-cluster-distances and large within-cluster-distances. Unlike the global quantization error, they take into account local relationships and they are parameterized by the considered neighborhood sizes.

In the following, we experimentally test in how far the QA measures for DR as introduced above lead to reasonable evaluations for typical clustering scenarios.

4 Experiments

We use two artificial 2-dimensional scenarios with randomly generated data points, where data are arranged in clusters (11 clouds), or data are distributed uniformly (random square), respectively. We use the *batch neural gas (BNG)* algorithm [6] for clustering as a robust classical prototype-based clustering algorithm. We use different numbers of prototypes per scenario, covering various 'resolutions' of the data.

11 clouds data This consists of 1100 random data vectors as shown in Fig. 1(a). We used 110, 11, and 5 prototypes, which lead to three respective situations: (I) $M = 110$ – each cloud is covered by ~ 10 prototypes, none of them located in-between the clouds, so one cluster consists of ~ 10 data

points; (II) $M = 11$ – there is one prototype approximately in the center of each cloud, so one cluster consists of ~ 100 data points; (III) $M = 5$ – there are not enough prototypes to cover each cloud separately, so only two of them are situated near cloud centers and three are spread in-between clouds. Cluster sizes vary between ~ 100 and ~ 300 data points, which includes more than one cloud on average.

The resulting prototypes are depicted in Fig. 1(a), the according QA results are shown in Fig. 2(a) to 2(f). The graphs show the different quality values, sampled over neighborhood radii from the smallest to the largest possible. The figures on the left hand side refer to distance-based neighborhoods, the ones on the right refer to rank-based neighborhoods.

In several graphs, especially in Fig. 2(c), 2(d), the grouped structure of the *11 clouds* data is resembled by wave or sawtooth patterns of the QA curves, showing that the total amount of rank or distance errors change rapidly as the neighborhood range coincides with cluster boundaries. Similarly, in all graphs there is a first peak in quality at the neighborhood radius where only a single cloud is approximately contained in each neighborhood. Within such neighborhoods, rank or distance errors are rare, even under the mapping of points to their closest prototypes. This effect is visible, e.g., in Fig. 2(e), 2(f). Interestingly, in Fig. 2(e), 2(f), there is a first peak where both, precision and recall are close to 1 corresponding to the 'perfect' cluster structure displayed in the model, while Fig. 2(a), 2(b) do not possess such value for small neighborhood corresponding to the structural mismatch because of the small number of prototypes. Unlike the IR measures, the CRM measure leads to smaller qualities for smaller numbers of prototypes in all cases. The structural match in the context of 11 prototypes can be observed in a comparably large increase of the absolute value, but the situation is less clear as compared to the IR measures. For the IR measures, the main difference in between the situations where a structural match can be observed (11 and 110 prototypes, respectively) is the smoothness of the curves, but not their absolute value.

Random square data Data and prototype locations for 10 and 100 prototypes are depicted, respectively, in Fig. 1(b). For $M = 100$, each cluster consists of ~ 10 data points, and with $M = 10$ the sizes were ~ 100. As expected, the QA graphs in Fig. 2(g) and 2(h) are continuously rising for the setting $M = 100$, whereas the curves are less stable but still following an upward trend for $M = 10$. This shows how the sparse quantization of the whole uniform data distribution leads to more topological mismatches over various neighborhood sizes.

5 Conclusions

In this contribution, we investigated the suitability of recent QA measures for DR to also evaluate clustering, such that visualization of large data sets, which commonly requires both, clustering and dimensionality reduction, could be evaluated based on one quality criterion only. While a formal transfer of the QA measures to clustering is possible, there exist qualitative differences between the IR and CRM evaluation criteria. It seems that IR based evaluation criteria allow to also detect appropriate cluster structures, corresponding to high precision and recall, while the situation is less pronounced for the CRM measures where a smooth transition of the measures corresponding to the cluster sizes can

be observed. One drawback of the IR evaluation is its dependency on the local scale as specified by the radii used for the evaluation. This makes it difficult to interpret the graphs due to the resulting sawtooth patterns even for data which possess only one global scale. If the scaling or density of the data varies locally, results are probably difficult to interpret. This question is subject of ongoing work.

References

[1] L.J.P. van der Maaten and G.E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

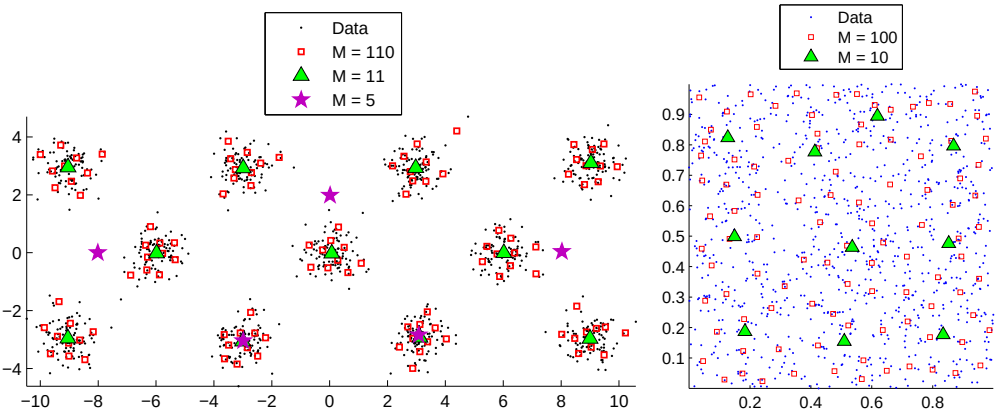
[2] Jarkko Venna, Jaakko Peltonen, Kristian Nybo, Helena Aidos, and Samuel Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *J. Mach. Learn. Res.*, 11:451–490, 2010.

[3] John A. Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Springer, 2007.

[4] John A. Lee and Michel Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomput.*, 72(7-9):1431–1443, 2009.

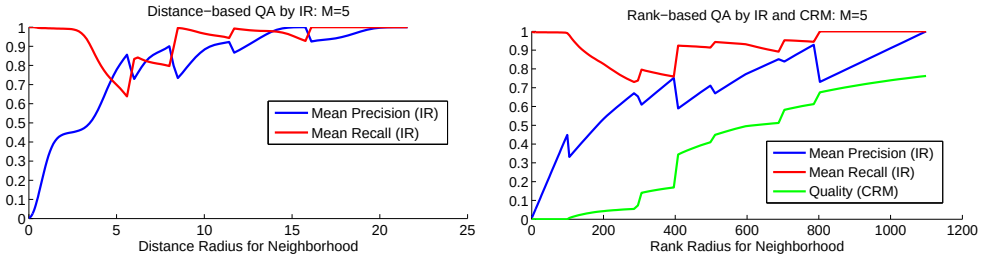
[5] Shi Zhong, Joydeep Ghosh, and Claire Cardie. A unified framework for model-based clustering. *Journal of Machine Learning Research*, 4:1001–1037, 2003.

[6] Marie Cottrell, Barbara Hammer, Alexander Hasenfuß, and Thomas Villmann. Batch and median neural gas. *Neural Networks*, 19:762–771, 2006.



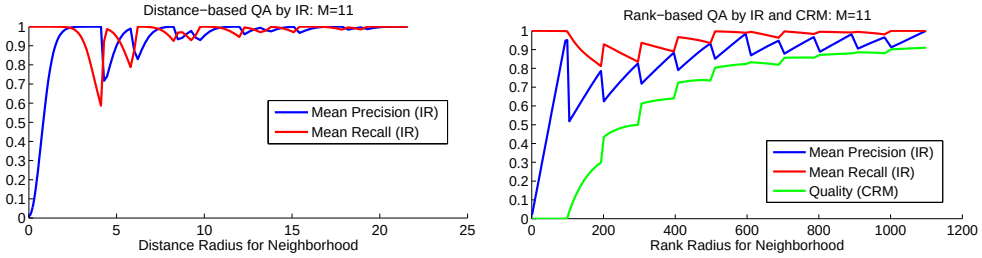
(a) *11 clouds* data (b) *random square* data

Figure 1: 1(a) *11 clouds* dataset and three independent prototype distributions of 110, 11, and 5 prototypes. 1(b) *random square* dataset with two independent prototype distributions of 100 and 10 prototypes.



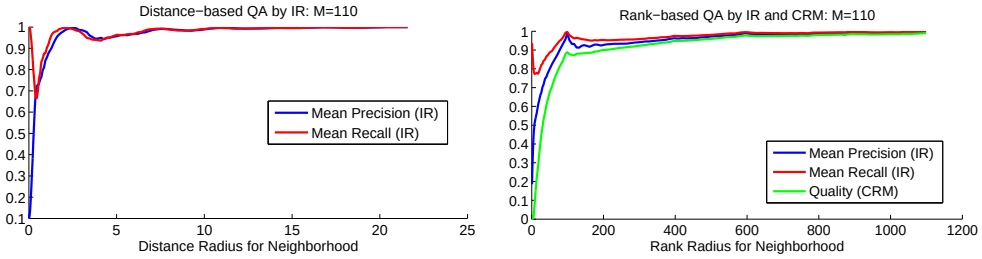
(a) (b)

QA for 11 clouds *dataset, clustered with 5 prototypes*



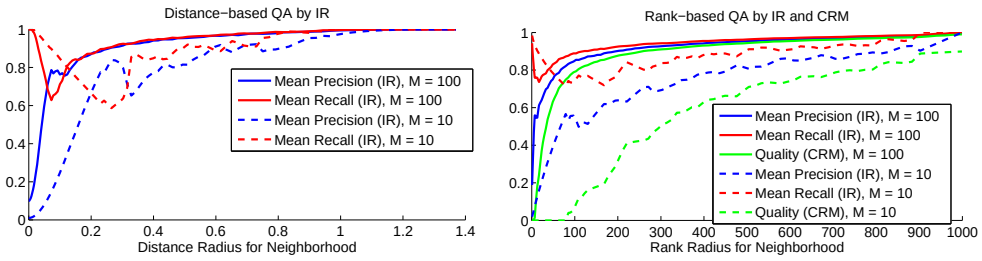
(c) (d)

QA for 11 clouds *dataset, clustered with 11 prototypes*



(e) (f)

QA for 11 clouds *dataset, clustered with 110 prototypes*



(g) (h)

QA for random square *data, clustered with 100 and 10 prototypes*

Figure 2: QA results from the IR and CRM frameworks for the two artificial clustering scenarios (*11 clouds* & *random square*) shown in Fig. 1. In the left column are the results with neighborhoods defined over distance radii; in the right column the neighborhoods were based on rank radii.

Functional Relevance Learning in Generalized Learning Vector Quantization

M. Kästner and T. Villmann
University of Applied Sciences Mittweida,
Technikumplatz 17, 09648 Mittweida, Germany
{kaestner,villmann}@hs-mittweida.de

Abstract

We propose a functional approach for relevance learning and matrix adaptation for learning vector quantization of high-dimensional functional data. We show, how parametrization of the functional relevance profile or functional matrix learning can be established for a reasonable number of parameters to be adapted.

Keywords: functional vector quantization, relevance learning, matrix learning, information theory

1 Introduction

During the last years prototype based models became one of the widely paradigms for clustering and classification. Thereby, different strategies are proposed in classification. Whereas support vector machines (SVMs) emphasize the class borders by the support vectors while maximizing the separation margin, the family of learning vector quantization (LVQ) algorithms is motivated by class representative prototypes and decision margin optimization to achieve high classification accuracy [2]. Based on the original but heuristically motivated standard LVQ introduced by KOHONEN [7] several more advanced methods were proposed. One key approach is the generalized LVQ (GLVQ) suggested by SATO&YAMADA [10] approximating the accuracy by a differentiable cost function to be minimized by stochastic gradient descent. This algorithm was extended to deal with metric adaptation to weight the data dimension according to their relevance for classification [4]. Usually, this relevance learning is based on weighting the Euclidean distance, and, hence, the data dimensions are treated independently leading to large number of weighting coefficients, the so-called relevance profile, to be adapted in

case of high-dimensional data. The extension of this approach is matrix learning where a parametric quadratic form is used [12].

If the data dimension is huge, as it is frequently the case for spectral data or time series, the relevance determination and the parameter adaptation may become crucially or instable. However, functional data have in common that the vectors can be seen as discrete realizations of functions, i.e. the vectors are so-called functional data. For those data the index of the vector dimensions is a representative of the respective independent function variable, i.e. frequency, time or position etc. In this sense the data dimensions are not longer uncorrelated.

The aim of the new relevance and matrix learning methods proposed here is to make use of this interpretation. Then, the relevance profile can be also assumed as a discrete representation of a one-dimensional relevance function. For the parameter matrix of the quadratic form in matrix learning a two-dimensional function description is assumed. We suggest to approximate these functions as a superpositions of only a few basis functions depending on a drastically decreased number of parameters compared to the huge number of independent relevance weights or matrix elements. We call the resulting algorithms *Generalized Functional Relevance LVQ* (GFRLVQ) and *Generalized Functional Matrix LVQ* (GFMLVQ). Further, we propose the integration of a sparseness criterion for minimizing the number of needed basis functions based on an entropy criterion resulting in *Sparse GFRLVQ* (S-GFRLVQ) and *Sparse GFMLVQ* (S-GFMLVQ).

2 Relevance and Matrix Learning in GLVQ – GRLVQ

As mentioned before, GLVQ is an extension of standard LVQ based on energy function E approximating the accuracy. Given a set $V \subseteq \mathbb{R}^D$ of data vectors $\mathbf{v}$ with class labels $x_{\mathbf{v}} \in \mathcal{C} = \{1, 2, \dots C\}$, the prototypes $\mathbf{w} \in W \subset \mathbb{R}^D$ with class labels y_j $(j = 1, \dots, N)$ should be distributed in such a way that they represent the data classes as accurate as possible. In particular, the following cost function is minimized

$$E(W) = \frac{1}{2} \sum_{\mathbf{v} \in V} f(\mu(\mathbf{v})) \text{ with } \mu(\mathbf{v}) = \frac{d^+(\mathbf{v}) - d^-(\mathbf{v})}{d^+(\mathbf{v}) + d^-(\mathbf{v})} \tag{1}$$

where f is a monotonically increasing function usually chosen as sigmoidal or the identity function. The function $\mu(\mathbf{v})$ is the classifier function where $d^+(\mathbf{v}) = d(\mathbf{v}, \mathbf{w}^+)$ denotes the distance between the data vector $\mathbf{v}$ and the closest prototype $\mathbf{w}^+$ with the same class label $y_{\mathbf{w}^+} = x_v$, and $d^-(\mathbf{v}) = d(\mathbf{v}, \mathbf{w}^-)$ is the distance to the best matching prototype $\mathbf{w}^-$ with a class label $y_{\mathbf{w}^-}$ different from $x_{\mathbf{v}}$. The

similarity measure $d(\mathbf{v}, \mathbf{w})$ is supposed differentiable with respect to the second argument but not necessarily to be a mathematical distance. More general similarity measure could be in consideration. One possible choices are the standard Euclidean distance or their weighted counterpart

$$d_{\boldsymbol{\lambda}}(\mathbf{v}, \mathbf{w}) = \sum_{i=1}^{D} \lambda_i (v_i - w_i)^2 \tag{2}$$

with relevance weights $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$. The vector $\boldsymbol{\lambda}$ is called relevance profile.

Learning in GLVQ of $\mathbf{w}^+$ and $\mathbf{w}^-$ is done by stochastic gradient descent learning with respect to the cost function $E(W)$ according to

$$\frac{\partial_S E(W)}{\partial \mathbf{w}^+} = \xi^+ \cdot \frac{\partial d^+}{\partial \mathbf{w}^+} \text{ and } \frac{\partial_S E(W)}{\partial \mathbf{w}^-} = \xi^- \cdot \frac{\partial d^-}{\partial \mathbf{w}^-}$$

with $\xi^+ = f' \cdot \frac{2 \cdot d^-(\mathbf{v})}{(d^+(\mathbf{v}) + d^-(\mathbf{v}))^2}$ and $\xi^- = -f' \cdot \frac{2 \cdot d^+(\mathbf{v})}{(d^+(\mathbf{v}) + d^-(\mathbf{v}))^2}$. Relevance learning in this model can be performed by adaptation of the relevance weights again by gradient descent learning:

$$\frac{\partial E_S(W)}{\partial \lambda_j} = \xi^+ \cdot \frac{\partial d^+_{\boldsymbol{\lambda}}}{\partial \lambda_j} + \xi^- \cdot \frac{\partial d^-_{\boldsymbol{\lambda}}}{\partial \lambda_j} \ . \tag{3}$$

The respective algorithm is named Generalized Relevance LVQ – GRLVQ [4], which still is a decision margin optimizer [3]. Yet, in this model the relevance weights as well as the vector components are treated independently as it is the natural way in the Euclidean distance or their weighted variant.

Matrix learning generalizes the idea of relevance learning [13, 12]. Instead of the weighted Euclidean distance (2), a positive definite bilinear form is used:

$$d_{\boldsymbol{\Lambda}}(\mathbf{v}, \mathbf{w}) = (\mathbf{v} - \mathbf{w})^T \boldsymbol{\Lambda} (\mathbf{v} - \mathbf{w}) \tag{4}$$

with a quadratic positive definite matrix $\boldsymbol{\Lambda}$. If the matrix $\boldsymbol{\Lambda}$ is decomposed into $\boldsymbol{\Lambda} = \boldsymbol{\Omega}^T \boldsymbol{\Omega}$, where $\boldsymbol{\Omega} \in \mathbb{R}^{D \times m}$ and $m > 0$ an arbitrary positive integer [1], then (4) can be rewritten as

$$d_{\boldsymbol{\Lambda}}(\mathbf{v}, \mathbf{w}) = \left(\boldsymbol{\Omega}^T (\mathbf{v} - \mathbf{w})^T \right)^2 \tag{5}$$

Accordingly to the relevance learning, we get

$$\frac{\partial E_S(W)}{\partial \Omega_{ij}} = \xi^+ \cdot \frac{\partial d^+_{\boldsymbol{\Lambda}}}{\partial \Omega_{ij}} + \xi^- \cdot \frac{\partial d^-_{\boldsymbol{\Lambda}}}{\partial \Omega_{ij}} \tag{6}$$

for the matrix learning vector quantization algorithm (GMLVQ).

3 Functional Relevance and Matrix Learning for GLVQ

As we have seen, the data dimensions are handled independently in both, GRLVQ and GMLVQ. This leads to a huge number of relevance weights to be adjusted, if the data vector are really high-dimensional as it is the case in many applications. For example, processing of hyperspectral data frequently requires the consideration of hundreds or thousands of spectral bands; time series may consist of a huge number of time steps. This huge dimensionality may lead to instable behavior of relevance learning in GRLVQ. For GMLVQ the problem is similar although a self-regularizing mechanism leads to the fact that the number of free parameters are in principle the same as in GRLVQ [11].

Yet, if the data vector are discrete representations of functions, both relevance and matrix learning can make use of this functional property to reduce the number of parameters in relevance learning. More precisely, we assume in the following that data vectors $\mathbf{v} = (v_1, \ldots, v_D)^T$ are representations of functions $v_i = v(t_i)$.

3.1 Functional Relevance Learning

For *functional relevance learning* the relevance profile can be interpreted as a function $\lambda(t)$ with $\lambda_j = \lambda(t_j)$, too. In the recently proposed *generalized functional relevance* LVQ (GFRLVQ) [5], the relevance function $\lambda(t)$ is supposed to be a superposition

$$\lambda(t) = \sum_{l=1}^{K} \beta_l \mathcal{K}_l(t) \tag{7}$$

of simple basis functions $\mathcal{K}_l$ depending on only a few parameters with the restriction $\sum_{l=1}^{K} \beta_l = 1$. Famous examples are standard Gaussians or Lorentzians:

$$\mathcal{K}_l(t) = \frac{1}{\sigma_l \sqrt{2\pi}} \exp\left(-\frac{(t-\Theta_l)^2}{2\sigma_l^2}\right) \tag{8}$$

and

$$\mathcal{K}_l(t) = \frac{1}{\eta_l \pi} \frac{\eta_l^2}{\eta_l^2 + (t-\Theta_l)^2}\ , \tag{9}$$

respectively. Now, relevance learning takes place by adaptation of the parameters β_l, Θ_l,σ_l and η_l, respectively. For this purpose, again a stochastic gradient scheme is applied. For an arbitrary parameter ϑ_l of the dissimilarity measure d we have

$$\frac{\partial_S E}{\partial \vartheta_l} = \xi^+ \cdot \frac{\partial d^+}{\partial \vartheta_l} + \xi^- \cdot \frac{\partial d^-}{\partial \vartheta_l}$$

Using the convention $t_j = j$ we get in the case of Gaussians for the weighting coefficient β_l, the center Θ_l and the width σ_l for

$$\frac{\partial d(\mathbf{v},\mathbf{w})}{\partial \beta_l} = \frac{1}{\sigma_l\sqrt{2\pi}} \sum_{j=1}^{D} \exp\left(-\frac{(j-\Theta_l)^2}{2\sigma_l^2}\right)(v_j - w_j)^2 \tag{10}$$

$$\frac{\partial d(\mathbf{v},\mathbf{w})}{\partial \Theta_l} = \frac{\beta_l}{\sigma_l^3\sqrt{2\pi}} \sum_{j=1}^{D} (j-\Theta_l) \exp\left(-\frac{(j-\Theta_l)^2}{2\sigma_l^2}\right)(v_j - w_j)^2 \tag{11}$$

$$\frac{\partial d(\mathbf{v},\mathbf{w})}{\partial \sigma_l} = \frac{\beta_l}{\sigma_l^2\sqrt{2\pi}} \sum_{j=1}^{D} \left(\frac{(j-\Theta_l)^2}{\sigma_l^2} - 1\right) \exp\left(-\frac{(j-\Theta_l)^2}{2\sigma_l^2}\right)(v_j - w_j)^2 \tag{12}$$

whereas for the Lorentzian we obtain

$$\frac{\partial d(\mathbf{v},\mathbf{w})}{\partial \beta_l} = \frac{1}{\pi} \sum_{j=1}^{D} \frac{\eta_l}{\eta_l^2 + (j-\Theta_l)^2}(v_j - w_j)^2 \tag{13}$$

$$\frac{\partial d(\mathbf{v},\mathbf{w})}{\partial \Theta_l} = \frac{\beta_l}{\pi} \sum_{j=1}^{D} \frac{2\eta_l (j-\Theta_l)}{\left(\eta_l^2 + (j-\Theta_l)^2\right)^2}(v_j - w_j)^2 \tag{14}$$

$$\frac{\partial d(\mathbf{v},\mathbf{w})}{\partial \eta_l} = \frac{\beta_l}{\pi} \sum_{j=1}^{D} \frac{(j-\Theta_l)^2 - \eta_l^2}{\left(\eta_l^2 + (j-\Theta_l)^2\right)^2}(v_j - w_j)^2 \tag{15}$$

Instabilities may occur if the center locations Θ_l, Θ_k become very similar for $l \neq k$. To avoid this phenomenon a weighted penalty term

$$P_R = \sum_{l=1}^{K} \sum_{m=1}^{K} \exp\left(-\frac{(\Theta_m - \Theta_l)^2}{2\xi_l\xi_m}\right) \tag{16}$$

is added to the cost function (1) according to the used basis functions. The resulting new cost function is

$$E_{GFRLVQ} = E(W) + \varepsilon_R P_R \tag{17}$$

with a properly chosen penalty weight $\varepsilon_R > 0$. For Gaussian basis functions we set $\xi_k = \sigma_k$, and for the Lorentzians we take $\xi_k = \eta_k$. The penalty can be interpreted as a repulsion with an influence range determined by the local correlations $\xi_l\xi_m$. The resulting additional update term for Θ_l-learning is

$$\frac{\partial P_R}{\partial \Theta_l} = \frac{1}{2} \sum_{m=1}^{K} \frac{(\Theta_l - \Theta_m)}{\xi_l\xi_m} \exp\left(-\frac{(\Theta_m - \Theta_l)^2}{2\xi_l\xi_m}\right)$$

leading to a minimum spreading of the basis function centers Θ_l. Analogously, an additional term occurs for the adjustemts of the ξ_l according to $\frac{\partial P_R}{\partial \xi_l}$, which has to be taken into account for the update of σ_k and η_k for Gaussians and Lorentzians, respectively.

3.2 Functional Matrix Learning

For *functional matrix learning vector quantization* (GFMLVQ) we assume in complete analogy to the functional relevance learning approach that the matrix $\mathbf{\Omega}$ in (5) is described in terms of a superposition

$$\mathbf{\Omega}(t_1, t_2) = \sum_{l=1}^{K} \beta_l \mathbf{K}_l(t_1, t_2) \tag{18}$$

of now two-dimensional basis functions $\mathbf{K}_l(t_1, t_2)$. For the Gaussian example we have

$$\mathbf{K}_l(t_1, t_2) = \frac{1}{\sigma_{1,l} \cdot \sigma_{2,l} \cdot 2\pi} \exp\left(-\left(\frac{(t_1 - \Theta_{1,l})^2}{2\sigma_{1,l}^2} + \frac{(t_2 - \Theta_{2,l})^2}{2\sigma_{2,l}^2}\right)\right) \tag{19}$$

whereas for the Lorentzian we get

$$\mathbf{K}_l(t_1, t_2) = \frac{1}{\eta_{1,l} \cdot \eta_{2,l} \cdot \pi^2} \left(\frac{\eta_{1,l}^2}{\eta_{1,l}^2 + (t_1 - \Theta_{1,l})^2} \cdot \frac{\eta_{2,l}^2}{\eta_{2,l}^2 + (t_2 - \Theta_{2,l})^2}\right) \tag{20}$$

and the derivatives have to be performed accordingly.

The penalty term (16) known from GFRLVQ avoiding there the total overlap of different basis functions $\mathbf{K}_l$ and $\mathbf{K}_k$ for $k \neq l$ has also to be adapted and reads now as

$$P_M = \sum_{l=1}^{K} \sum_{m=1}^{K} \exp\left(-\left(\frac{(\Theta_{1,m} - \Theta_{1,l})^2}{2\xi_{1,m}\xi_{1,l}} + \frac{(\Theta_{2,m} - \Theta_{2,l})^2}{2\xi_{2,m}\xi_{2,l}}\right)\right) \tag{21}$$

again with the settings $\xi_{i,k} = \sigma_{i,k}$ and $\xi_{i,k} = \eta_{i,k}$ for Gaussians and Lorentzians, respectively. Thus the full cost function

$$E_{GFMLVQ} = E(W) + \varepsilon_M P_M \tag{22}$$

is finally obtained for GFMLVQ with the penalty weight $\varepsilon_M > 0$.

4 Sparse GFRLVQ and GFMLVQ

In the GFRLVQ model the number K of basis functions to be used is free of choice so far. Obviously, if the K-value is too small, an appropriate relevance weighting is impossible. Otherwise, a K-value too large complicates the problem more than necessarily. Hence, a good adjustment is demanded. This problem can be seen as sparseness in functional relevance learning. A common methodology to judge

sparsity is the information theory. In particular, the Shannon entropy H of the weighting coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_K)$ can be applied. Maximum sparseness, i.e. minimum entropy, is obtained, iff $\beta_l = 1$ for exactly one certain l whereas the other β_m are equal to zero. However, maximum sparseness may be accompanied by a decrease of accuracy in classification and/or increased cost function value E_{GFRLVQ}.

To achieve an optimal balancing, we propose the following strategy: The cost function E_{GFRLVQ} is extended to

$$E_{S-GFRLVQ} = E_{GFRLVQ} + \gamma(\tau) \cdot H(\boldsymbol{\beta}) \tag{23}$$

with τ counting the adaptation steps. Let τ_0 be the final time step of the usual GFRLVQ-learning. Then $\gamma(\tau) = 0$ for $\tau < \tau_0$ holds. Thereafter, $\gamma(\tau)$ is slowly increased in an adiabatic manner [6], such that all parameters can immediately follow the drift of the system. An additional term for β_l-adaptation occurs for non-vanishing $\gamma(\tau)$-values according to this new cost function (23):

$$\frac{\partial E_{S-GFRLVQ}}{\partial \beta_l} = \frac{\partial E_{GFRLVQ}}{\partial \beta_l} + \gamma(\tau) \frac{\partial H}{\partial \beta_l} \tag{24}$$

with $\frac{\partial H}{\partial \beta_l} = -(\log(\beta_l) + 1)$. This term triggers the $\boldsymbol{\beta}$-vector to become sparse. The adaptation process is stopped if the E_{GFRLVQ}-value or the classification error shows a significant increase compared to the time τ_0.

Obviously, this optimization scheme can also be applied to GFMLVQ yielding *Sparse* GFMLVQ (S-GFMLVQ) with

$$E_{S-GFMLVQ} = E_{GFMLVQ} + \gamma(\tau) \cdot H(\boldsymbol{\beta}) \tag{25}$$

as cost function.

5 Conclusion

In this paper we propose the *functional* relevance and matrix learning for generalized learning vector quantization. This functional learning supposes that the data vectors are representations of functions such that the relevance profile or the parameter matrix can be assumed as a superposition of one- or two-dimensional basis functions, respectively. These basis functions depend on only a few parameters to be adapted during learning compared to the huge number of free parameters to be adjusted in usual relevance or matrix learning. To obtain an optimal number of basis function for the superposition a sparsity constraint is suggested. Thereby, sparsity is judged in terms of the entropy of the respective weighting coefficients in the superposition. The approach is introduced exemplarily for the weighted

Euclidean distance and a bilinear form also based on the Euclidean norm, for simplicity. Obviously, the Euclidean distance is not based on a functional norm. Yet, the transfer to real functional norms and distances like Sobolev norms [16], the Lee-norm [8, 9], kernel based LVQ-approaches [15] or divergence based similarity measures [14], which carry the functional aspect inherently, is straight forward and topic of future investigations.

References

[1] K. Bunte, B. Hammer, A. Wismüller, and M. Biehl. Adaptive local dissimilarity measures for discriminative dimension reduction of labeled data. *Neurocomputing*, 73:1074–1092, 2010.

[2] B. Hammer, M. Strickert, and T. Villmann. Relevance LVQ versus SVM. In L. Rutkowski, J. Siekmann, R. Tadeusiewicz, and L. Zadeh, editors, *Artificial Intelligence and Soft Computing (ICAISC 2004)*, Lecture Notes in Artificial Intelligence 3070, pages 592–597. Springer Verlag, Berlin-Heidelberg, 2004.

[3] B. Hammer, M. Strickert, and T. Villmann. On the generalization ability of GRLVQ networks. *Neural Processing Letters*, 21(2):109–120, 2005.

[4] B. Hammer and T. Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15(8-9):1059–1068, 2002.

[5] M. Kästner, B. Hammer, and T. Villmann. Generalized functional relevance learning vector quantization. In M. Verleysen, editor, *Proc. of European Symposium on Artificial Neural Networks (ESANN'2011)*, page in press, Evere, Belgium, 2011. d-side publications.

[6] T. Kato. On the adiabatic theorem of quantum mechanics. *Journal of the Physical Society of Japan*, 5(6):435–439, 1950.

[7] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).

[8] J. Lee and M. Verleysen. Generalization of the l_p norm for time series and its application to self-organizing maps. In M. Cottrell, editor, *Proc. of Workshop on Self-Organizing Maps (WSOM) 2005*, pages 733–740, Paris, Sorbonne, 2005.

[9] J. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Information Sciences and Statistics. Springer Science+Business Media, New York, 2007.

[10] A. Sato and K. Yamada. Generalized learning vector quantization. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*, pages 423–9. MIT Press, Cambridge, MA, USA, 1996.

[11] P. Schneider, K. Bunte, H. Stiekema, B. Hammer, T. Villmann, and M. Biehl. Regularization in matrix relevanz learning. *IEEE Transactions on Neural Networks*, 21(5):831–840, 2010.

[12] P. Schneider, B. Hammer, and M. Biehl. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21:3532–3561, 2009.

[13] P. Schneider, B. Hammer, and M. Biehl. Distance learning in discriminative vector quantization. *Neural Computation*, 21:2942–2969, 2009.

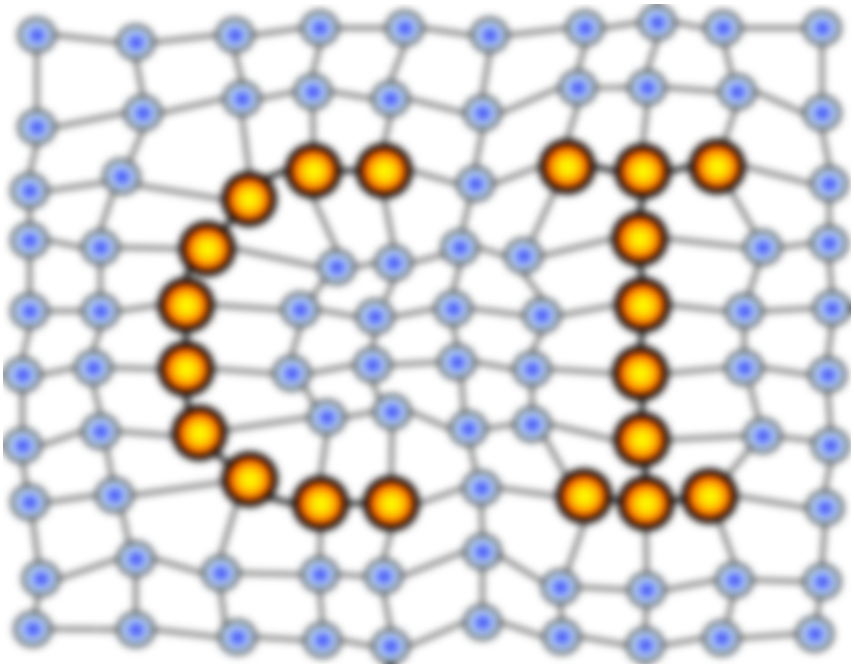
[14] T. Villmann and S. Haase. Divergence based vector quantization. *Neural Computation*, page in press, 2011.

[15] T. Villmann and B. Hammer. Theoretical aspects of kernel GLVQ with differentiable kernel. *IfI Technical Report Series*, (IfI-09-12):133–141, 2009.

[16] T. Villmann and F.-M. Schleif. Functional vector quantization by neural maps. In J. Chanussot, editor, *Proceedings of First Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS 2009)*, pages 1–4. IEEE Press, 2009. ISBN 978-1-4244-4948-4.

MACHINE LEARNING REPORTS

Report 01/2011



Impressum

Machine Learning Reports ISSN: 1865-3960

▽ Publisher/Editors

Prof. Dr. rer. nat. Thomas Villmann
University of Applied Sciences Mittweida
Technikumplatz 17, 09648 Mittweida, Germany
• http://www.mni.hs-mittweida.de/

Dr. rer. nat. Frank-Michael Schleif
University of Bielefeld
Universitätsstrasse 21-23, 33615 Bielefeld, Germany
• http://www.cit-ec.de/tcs/about

▽ Copyright & Licence

Copyright of the articles remains to the authors.

▽ Acknowledgments

We would like to thank the reviewers for their time and patience.