INSTITUTE OF MATHEMATICS AND INFORMATICS
VYTAUTAS MAGNUS UNIVERSITY

Romanas Tumasonis

MINING FREQUENT SEQUENCES IN LARGE DATA ARRAYS

Summary of Doctoral Dissertation

Physical Sciences (P 000)
Informatics (09 P)
Informatics, System theory (P 175)

Vilnius, 2007

This research was accomplished in the period from 2002 to 2006 at the Institute of Mathematics and Informatics.

The right for the doctoral studies in informatics was granted to the Institute of Mathematics and Informatics together with Vytautas Magnus University by Government of the Republic of Lithuania, Decree No. 1285, issued on the 13th of December 2004.

Scientific Advisor:

Prof. Dr. Habil. Gintautas Dzemyda (Institute of Mathematics and Informatics, Physical Sciences, Informatics, 09 P)

Council of Scientific field:

Chairman:

Prof. Dr. Habil. Vytautas Kaminskas (Vytautas Magnus University, Physical Sciences, Informatics, 09 P)

Members:

Assoc. Prof. Dr. Regina Kulvietienė (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering, 07 T)

Prof. Dr. Habil. Vydūnas Šaltenis (Institute of Mathematics and Informatics, Physical Sciences, Informatics, 09 P)

Prof. Dr. Habil. Rimantas Šeinauskas (Kaunas University of Technology, Technological Sciences, Informatics Engineering, 07 T)

Prof. Dr. Habil. Laimutis Telksnys (Institute of Mathematics and Informatics, Physical Sciences, Informatics, 09 P)

Opponents:

Assoc. Prof. Dr. Vitalijus Denisovas (Klaipėda University, Physical Sciences, Informatics, 09 P)

Prof. Dr. Habil. Leonidas Sakalauskas (Institute of Mathematics and Informatics, Physical Sciences, Informatics, 09 P)

The dissertation will be defended at the public meeting of the Scientific Council in the field of Informatics in the Conference and Seminars Centre of the Institute of Mathematics and Informatics on June 29, 2007.
Address: Goštauto str. 12, LT- 01108, Vilnius, Lithuania.

The summary of the dissertation was sent-out on May 29, 2007.
The dissertation is available at the M. Mažvydas National Library of Lithuania, the Library of the Institute of Mathematics and Informatics and the Library of Vytautas Magnus University.

MATEMATIKOS IR INFORMATIKOS INSTITUTAS
VYTAUTO DIDŽIOJO UNIVERSITETAS

Romanas Tumasonis

DAŽNŲ SEKŲ PAIEŠKA DIDELIUOSE DUOMENŲ MASYVUOSE

Daktaro disertacijos santrauka

Fiziniai mokslai (P 000)
Informatika (09 P)
Informatika, Sistemų teorija (P 175)

Vilnius, 2007

Disertacija rengta 2002–2006 metais Matematikos ir informatikos institute.

Doktorantūros teisė suteikta kartu su Vytauto Didžiojo universitetu 2004 m. gruodžio 13 d. Lietuvos Respublikos Vyriausybės nutarimu Nr. 1285.

Mokslinis vadovas:

prof. habil. dr. Gintautas Dzemyda (Matematikos ir informatikos institutas, fiziniai mokslai, informatika, 09 P)

Disertacija ginama Informatikos mokslo krypties taryboje:

Pirmininkas:

prof. habil. dr. Vytautas Kaminskas (Vytauto Didžiojo universitetas, fiziniai mokslai, informatika, 09 P).

Nariai:

doc. dr. Regina Kulvietienė (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija, 07 T),

prof. habil. dr. Vydūnas Šaltenis (Matematikos ir informatikos institutas, fiziniai mokslai, informatika, 09 P),

prof. habil. dr. Rimantas Šeinauskas (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija, 07 T),

prof. habil. dr. Laimutis Telksnys (Matematikos ir informatikos institutas, fiziniai mokslai, informatika, 09 P).

Oponentai:

doc. dr. Vitalijus Denisovas (Klaipėdos universitetas, fiziniai mokslai, informatika, 09 P),

prof. habil. dr. Leonidas Sakalauskas (Matematikos ir informatikos institutas, fiziniai mokslai, informatika, 09 P).

Disertacija bus ginama viešame Informatikos mokslo krypties tarybos posėdyje 2007 m. birželio 29 d. Matematikos ir informatikos instituto konferencijų ir seminarų centre.
Adresas: Goštauto g. 12, LT-01108, Vilnius, Lietuva.

Disertacijos santrauka išsiuntinėta 2007 m. gegužės 29 d.
Disertaciją galima peržiūrėti M. Mažvydo nacionalinėje bibliotekoje, Matematikos ir informatikos instituto ir Vytauto Didžiojo universiteto bibliotekose.

GENERAL DESCRIPTION

Topicality of the problem

Data mining is the process of automatic extraction of novel, useful and understandable patterns in very large databases. Data mining refers to the overall process of discovering new patterns or building models from a given dataset. There are many steps involved in the KDD (knowledge data discovering) enterprise which include data selection, data cleaning and preprocessing, data transformation and reduction, data mining task and algorithm selection, and, finally, post-processing and interpretation of discovered knowledge.

Discovering associations is one of the fundamental tasks of data mining. Its aim is to automatically seek dependencies from vast amounts of data. The task results in so-called association rules, which are of the form: if *A* occurs in the data then *B* occurs too. Only those rules that occur in the data frequently enough are generated.

However, various information sources generate data with an inherently sequential nature; i.e., it is composed of discrete events which have a temporal/spatial ordering. This kind of data can be obtained from, e.g., telecommunications networks, electronic commerce, www-servers of Internet, and various scientific sources, such as gene databases. The sequential nature of the data is totally ignored in the generation of the association rules. Thus, a part of the useful information included in the data is discarded. Therefore, since the mid of 90's the interest in discovering also the sequential associations in the data has arisen in the data mining community.

The sequential associations or *sequential patterns* can be presented in the form: when *A* occurs, *B* occurs within some certain time. So, the difference from traditional association rules is such that here the time information is included both in the rule itself and in the mining process in the form of timing constraints. Nowadays several highly efficient methods for mining these kind of patterns exist. The problem is that the input data are assumed to be sequences of discrete events including only the information of the ordering, usually according to the time. Often, however, the events are associated with some additional attributes. The existing methods cannot take into account this multi-dimensionality of data, and so they lose the additional information it involves. Furthermore, the methods are designed for some specific problem and are not, as such, applicable to different types of sequential data.

Aims and tasks of the work

Sequential pattern mining that finds the set of frequent subsequences in sequence databases is an important data mining task and has broad applications, such as business analysis, web mining, security, and bio-sequences analysis. We will examine just plain text information. Text mining is a variation on a field called data mining that tries to find interesting patterns from large databases. The difference between regular data mining and text mining is that in text mining patterns are extracted from natural language text rather than from structured databases of facts. Databases are designed for programs to process automatically; text is written for people to read. Text mining methods can be used in bioinformatics for analysis of DNA sequences.

The general tasks are as follows:

- Research on one data mining algorithm;

- Modification of this algorithm;
- Proposal of a new probability algorithm;
- Research on these algorithms with real and synthetics data.

Scientific novelty

The subject of the dissertation is to analyze the problem of the frequency of subsequences in large volume sequences (texts, DNA sequences, databases, etc.). A new algorithm ProMFS developed for mining frequent sequences with the matrix of frequent distances between elements has been proposed. It is based on the estimated probabilistic-statistical characteristics of the appearance of elements of the sequence and their order. The distinguishing feature of both algorithm modifications is that the primary modification employs the matrix of average distances and the newly presented modification employs the matrix of frequent distances. The algorithms build a much shorter new sequence and makes decisions on the main sequence in accordance with the results of analysis of the shorter one. The new modification of this algorithm has been compared with other algorithms.

Approbation and publication of research work

The main results of the work were published in these papers:

R. Tumasonis, G. Dzemyda. *Analysis of Statistical Characteristics in Mining of Frequent Sequences*. Intelligent Information Processing and Data Mining. Proceedings of the International IIS: IIPWM'05 Conference, ISSN 1615-3871, Springer Berlin Heidelberg New York, 2005, p. 377–387.

R. Tumasonis, G. Dzemyda. *The Probabilistic Algorithm for Mining Frequent Sequences*. Proceedings ADBIS'04 Eight East-European

Conference on Advances in Databases and Information Systems, ISBN 963311358X, 2004, p. 89–98.

R. Tumasonis, G. Dzemyda. *The Statistical Characteristics in Probabilistic Algorithms for Mining Frequent Sequence*s. Datarzinatne. Serija 5, Sejums 22. Scientific Proceedings of the Riga Technical University, ISSN 1407-7493, Izdevnieciba „RTU”, Riga 2005, p. 85–94.

R. Tumasonis, G. Dzemyda. *Atmintis problema ieškant dažnų sekų didelėse duomenų bazėse*. Informacijos mokslai, Vilnius, 2003, p. 193–199.

R. Tumasonis, G. Dzemyda. *Analitiniai duomenų gavimo būdai šiuolaikinėse informacinėse sistemose*. Informacinės technologijos 2005: Aktualijos ir perspektyvos. IV mokslinė praktinė konferencija, ISBN 9955-9779-0-9, Alytus, 2005, p. 180–186.

R. Tumasonis, G. Dzemyda. *Statistinių charakteristikų analizė dažnų sekų paieškoje*. INFORMACINĖS TECHNOLOGIJOS 2005, Kauno technologijos universitetas, Kaunas, 2005, p. 108–114.

R. Tumasonis, G. Dzemyda. *Dažnų sekų paieškos tikimybinis algoritmas*. INFORMACINĖS TECHNOLOGIJOS 2004, Kauno technologijos universitetas, Kaunas, 2004, II 2-8, p. 14–20.

R. Tumasonis, G. Dzemyda. *Dažnų sekų nustatymo didelėse duomenų bazėse algoritmų analizė*. INFORMACINĖS TECHNOLOGIJOS 2003, ISBN 9955-09-335-8, Kauno technologijos universitetas, Kaunas, 2003, p. II-6-II-13.

The results were also presented in these conferences:

R. Tumasonis, G. Dzemyda. *The Probablistic Algorihm for Mining Frequent Sequences.* ADBIS’04 Eight East-European Conference on Advances in Databases and Information Systems. September 22–25, 2004, Budapest, Hungary.

R. Tumasonis, G. Dzemyda. *Analysis of the Staistical Charakteristics in Mining Frequent Sequences.* Intelligent Information Processing and Web Mining IIPWM'05 Conference. June 13-16, 2005, Gdansk, Poland.

R. Tumasonis, G. Dzemyda. *The Statistical Characteristics in Probabilistic Algorithms for Mining Frequent Sequence*s. Thirteenth International Conference on Information Systems Development Methods and Tools, Theory and Practice. September 9-11, 2004, Vilnius, Lithuania.

R. Tumasonis, G. Dzemyda. *Tikimybinių charakteristikų panaudojimas dažnų sekų paieškoje*. Lietuvos matematikų draugijos XLV konferencija. June 17-18, 2003, Kaunas, LŽŪU.

R. Tumasonis, G. Dzemyda. *Analitiniai duomenų gavimo būdai šiuolaikinėse informacinėse sistemose*. Informacinės technologijos 2005: aktualijos ir perspektyvos. IV mokslinė praktinė konferencija. April 15-17, 2005, Alytus.

R. Tumasonis, G. Dzemyda. *Statistinių charakteristikų analizė dažnų sekų paieškoje*. INFORMACINĖS TECHNOLOGIJOS 2005. January 27–29, 2005, Kaunas, KTU

R. Tumasonis, G. Dzemyda. *Dažnų sekų paieškos tikimybinis algoritmas*. INFORMACINĖS TECHNOLOGIJOS 2004. January 27–29, 2004, Kaunas, KTU.

R. Tumasonis, G. Dzemyda. *Dažnų sekų nustatymo didelėse duomenų bazėse algoritmų analizė*. INFORMACINĖS TECHNOLOGIJOS 2003. January 26–28, 2004, Kaunas, KTU.

R. Tumasonis, G. Dzemyda. *Atmintis problema ieškant dažnų sekų didelėse duomenų bazėse*. Vienuoliktoji mokslinė kompiuterininkų konferencija. August 28–30, 2003, Vilnius.

CONTENT

The doctoral dissertation consists of four chapters, the general conclusions, the dictionary, the list of abbreviations and the list of references. The language of the dissertation is Lithuanian.

In the introduction the topicality of the problem discussed is defined, the goals and the tasks of the research are formulated, and the scientific novelty of the dissertation and the practical value of the work are substantiated.

Chapter 1. Mining frequent sets

The task of association mining is to discover a set of attributes shared among a large number of objects in a given database. For example, consider the sales database of a bookstore, where the objects represent customers and the attributes represent authors or books. The discovered patterns are the set of books most frequently bought together by the customers. An example could be that “40% of the people who buy Jane Austen's *Pride and Prejudice* also buy *Sense and Sensibility*”. The store can use this knowledge for promotions, shelf placement, etc. There are many potential application areas for association rule technology, which include catalogue design, store layout, customer segmentation, telecommunication alarm diagnosis, and so on.

The task of discovering all frequent associations in very large databases is quite challenging. The search space is exponential in the number of database attributes, and with millions of database objects the problem of I/O minimization becomes paramount. However, most current approaches are iterative in nature, requiring multiple database scans, which is clearly very expensive. Some of the methods, especially those using some form of sampling, can be sensitive to the data-skew, which can adversely affect performance. Furthermore, most approaches use very complicated internal data structures which have poor locality and add additional space and computation overheads. Our goal is to overcome all of these limitations.

In this chapter we present new algorithms for discovering the set of frequent attributes (also called itemsets). The key features of our approach are as follows:

1. We use a *vertical tid-list* database format, where we associate with each itemset a list of transactions in which it occurs. We show that all frequent itemsets can be enumerated via simple tid-list intersections.
2. We use a lattice-theoretic approach to decompose the original search space (lattice) into smaller pieces (sub-lattices), which can be processed independently in main-memory. We investigate two techniques for achieving the decomposition: prefix-based and maximal-clique-based partition.
3. We decouple the problem decomposition from the pattern search. We investigate three search strategies for enumerating the frequent itemsets within each sub-lattice: bottom-up, top-down and hybrid search.
4. Our approach requires roughly only a single database scan (with some pre-processed information) minimizing the I/O costs.

We present six algorithms combining the features listed above, depending on the database format, the decomposition technique, and the search procedure used. These include *Eclat* (Equivalence CLAss Transformation), *MaxEclat, Clique, MaxClique, Top-Down,* and

AprClique. Our algorithms not only minimize I/O costs by making only one database scan, but also minimize computation costs by using efficient search schemes. The algorithms are particularly effective when the discovered frequent itemsets are long. Our tid-list based approach is also insensitive to data-skew.

Chapter 2. Mining frequent sequences

The task of discovering all frequent sequences in large databases is quite challenging. The search space is extremely large. For example, with *m* attributes there are *O(m k)* potentially frequent sequences of length *k*. With millions of objects in the database the problem of I/O minimization becomes paramount. However, most current algorithms are iterative in nature, requiring as many full database scans as the longest frequent sequence, which is clearly very expensive. Some of the methods, especially those using some form of sampling, can be sensitive to the data-skew, which can adversely effect performance. Furthermore, most approaches use very complicated internal data structures which have poor locality and add additional space and computation overheads.

In this chapter we present the Partition and Appriori algorithm for discovering a set of all frequent sequences. The key features of our approach are as follows:

1. We use a *vertical id-list* database format, where we associate with each sequence a list of objects in which it occurs, along with the time-stamps. We show that all frequent sequences can be enumerated via simple id-list intersections.

2. We use a lattice-theoretic approach to decompose the original search space (lattice) into smaller pieces (sub-lattices), which can be processed independently in main-memory. Our approach usually

requires three database scans, or only a single scan with some preprocessed information, thus minimizing the I/O costs.

3. We decouple the problem decomposition from the pattern search. We propose two different search strategies for enumerating the frequent sequences within each sub-lattice: breadth-first and depth-first search.

The SPADE algorithm not only minimizes I/O costs by reducing database scans, but also minimizes computational costs by using efficient search schemes. The vertical id-list based approach is also insensitive to data-skew. An extensive set of experiments shows that SPADE outperforms previous approaches by a factor of two and by an order of magnitude, if we have some additional off-line information. Furthermore, SPADE scales linearly in the database size and a number of other database parameters.

Chapter 3. Data mining algorithms

In this chapter we have presented two standard GSP algorithm implementations (one with saving memory and another without saving memory), a recursive algorithm and the new ProMFS (probabilistic algorithm for mining frequent sequences) algorithm.

Assume that we have a set $L = \{i_1, i_2, ..., i_m\}$ consisting of *m* distinct elements, also called *items*. An *itemset* is a nonempty unordered collection of items. A *sequence* is an ordered list of itemsets. A *sequence* α is denoted as $(\alpha_1 \mapsto \alpha_2 \mapsto ... \mapsto \alpha_q)$, where the sequence *element* α_j is an itemset. An item can occur only once in an itemset, but it can occur multiple times in different item sets of a sequence. We solve a partial problem, where itemset consists of one item only. A sequence $\alpha = (\alpha_1 \mapsto \alpha_2 \mapsto ... \mapsto \alpha_n)$ is a *subsequence* of another sequence

$\beta = (\beta_1 \mapsto \beta_2 \mapsto ... \mapsto \beta_m)$, if there exist such numbers $t_1, t_2, ..., t_n$, where $t_{j+1} = t_j + 1,\ j = 1,...,n$ and $\alpha_j = \beta_{t_j}$ for all a_j. Here, β_{t_j} are elements of the set L. We analyze the sequence (the main sequence) S that is formed from single elements of L (not their sets, as in the classical formulation of the problem). In general, the number of elements in S is much larger than that in L. We have to find the most frequent subsequences in S. The problem is to find subsequences whose appearance frequency is more than some threshold called *minimum support*, i.e. the subsequence is frequent iff it occurs in the main sequence no less frequently than the minimum support.

The main problem is as follows: it is necessary to define only potentially useful subsequences, check up and prolong them gradually. Two algorithms are analyzed that analytically eliminate such subsequences that actually cannot be frequent. Two implementations of Generated-Sequence-Pattern algorithm (GSP) with and without economizing memory are tested. A recursive approach to sequence mining is suggested. It enables saving memory too. The algorithms are compared, and the cases are determined, when one algorithm works better than another one.

The new algorithm for mining frequent sequences is based on the estimation of the statistical characteristics of the main sequence:

- the probability of an element in the sequence;
- the probability for one element to appear after another one;
- the average distance between different elements of the sequence.

The main idea of the algorithm is the following:

1) some characteristics of the position and interposition of elements are determined in the main sequence;
2) the much shorter new model sequence $\tilde{C}$ is generated according to these characteristics;
3) the new sequence is analyzed with the GSP algorithm (or any similar one);
4) the frequency of subsequences in the main sequence is estimated by the results of the GSP algorithm applied on the new sequence.

Let:

1) $P(i_j) = \frac{V(i_j)}{VS}$ be the probability of occurrence of element i_j in the main sequence, where $i_j \in L,\ j = 1,...,m$. Here, $L=\{i_1,\ i_2,\ ...\ ,\ i_m\}$ is the set consisting of *m* distinct elements. $V(i_j)$ is the number of elements i_j in the main sequence *S*; *VS* is the length of the sequence. Note that $\sum_{j=1}^{m} P(i_j) = 1$.

2) $P(i_j \mid i_v)$ be the probability of appearance of element i_v after element i_j, where $i_j, i_v \in L,\ j,v = 1,...,m$. Note that $\sum_{v=1}^{m} P(i_j \mid i_v) = 1$ for all $j = 1,...,m$.

3) $D(i_j \mid i_v)$ be the distance between elements i_j and i_v, where $i_j, i_v \in L,\ j, v = 1,...,m$. In other words, the distance $D(i_j \mid i_v)$ is the number of elements that are between i_j and the first found i_v from i_j to the end of the main sequence (i_v is included). The distance between two adjacent elements of the sequence is equal to one.

4) $\overset{\frown}{A}$ be the matrix of average distances. Elements of the matrix are as follows: $a_{jv} = Average\ (D(i_j \mid i_v)),\ \ i_j, i_v \in L),\ \ j, v = 1, ..., m.$ All these characteristics can be obtained during one search through the main

sequence. According to these characteristics a much shorter model sequence $\tilde{C}$ is generated whose length is *l*. Denote its elements by c_r, $r = 1,...,l$. The model sequence $\tilde{C}$ will contain elements from *L*: $i_j \in L,\ j = 1,...,m$. For the elements c_r, a numeric characteristic $Q(i_j, c_r)$, $r = 1,...,l$, $j = 1,...,m$, is defined. Initially, $Q(i_j, c_r)$ is the matrix with zero values that are specified after the statistical analysis of the main sequence. The complementary function $\rho(c_r, a_{rj})$ is introduced that increases the value of characteristics $Q(i_j, c_r)$ by one. The first element c_1 of the model sequence $\tilde{C}$ is that from *L* corresponding to $\max(P(i_j))$, $i_j \in L$. According to c_1 the function $\rho(c_1, a_{1j}) \Rightarrow Q(i_j, 1 + a_{1j}) = Q(i_j, 1 + a_{1j}) + 1$, $j = 1,...,m$, is activated. Remaining elements c_r, $r = 2,...,l$, are chosen in the way described below. Consider the *r*-th element c_r of the model sequence $\tilde{C}$. The decision, which symbol from *L* should be chosen as c_r, will be made after calculating $\max(Q(i_j, c_r))$, $i_j \in L$. If for some *p* and *t* we obtain that $Q(i_p, c_r) = Q(i_t, c_r)$, then element c_r is chosen by maximal value of conditional probabilities, i.e. by $\max(P(c_{(r-1)} \mid i_p), P(c_{(r-1)} \mid i_t))$: $c_r = i_p$ if $P(c_{(r-1)} \mid i_p) > P(c_{(r-1)} \mid i_t)$, and $c_r = i_t$ if $P(c_{(r-1)} \mid i_p) < P(c_{(r-1)} \mid i_t)$. If these values are equal, i.e. $P(c_{(r-1)} \mid i_p) = P(c_{(r-1)} \mid i_t)$, then c_r is chosen depending on $\max(P(i_p), P(\mid i_t))$. After choosing the value of c_r, the function $\rho(c_r, a_{rj}) \Rightarrow Q(i_j, r + a_{rj}) = Q(i_j, r + a_{rj}) + 1$ is activated. All these actions are performed consecutively for every $r = 2,...,l$. This way we get the model sequence $\tilde{C}$ that is much shorter than the main one and that may be analyzed by the GSP algorithm with much less computational efforts.

We have changed one characteristic (average distance between different elements of the sequence) to the frequent distance between different elements of the sequence. $\widehat{F}$ is the matrix of these frequent distances. Elements of the matrix are as follows: $f_{jv} = Frequent\ (D(i_j \mid i_v),\ \ i_j, i_v \in L),\ \ j, v = 1, ..., m.$

Chapter 4. Results of the research

Compare these two different implementations and recursive algorithm by a special example. Suppose we have a text file with 90,000 A and B symbols. Our goal is to find all the frequent sequences. We have compared the times expended and memory use. The result are shown in Fig. 1 and Fig. 2

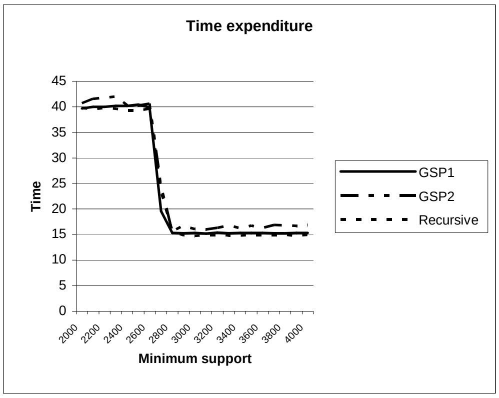

Fig. 1. Time expenditure of GSP implementations and recursive algorithm

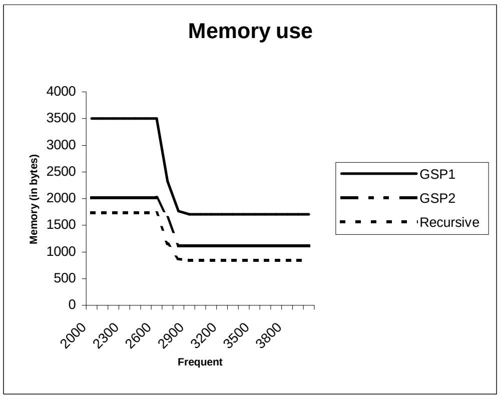

Fig. 2. Memory use of GSP implementations and recursive algorithm

The probabilistic mining of frequent sequences was compared with the GSP algorithm. We have generated a text file of 100,000 letters (1000 lines and 100 symbols in one line). L={A, B, C}, i.e. m=3, $i_1 = A, i_2 = B, i_3 = C$. In this text we have included one very frequent sequence *ABBC*. This sequence is repeated 20 times in one line. The remaining 20 symbols of the line are selected at random. First of all, the main sequence (100,000 symbols) was tested with the GSP algorithm. The results are presented in Fig. 3 and Fig. 4. They will be discussed in more detail together with the results of ProMFS. ProMFS generated the model sequence $\tilde{C}$ of length *l*=40.

This model sequence was examined with the GSP algorithm using the following minimum supports: 8, 9, 10, 11, 12, 13, and 14. The results are presented in Fig. 3 and 4. Fig. 3 shows the number of frequent sequences found by both GSP and ProMFS. Fig. 4 illustrates the expenditure of computation time used by both GSP and ProMFS to obtain the results of Fig. 3 (the minimum support in ProMFS is *Ms*=8; the results are similar for larger *Ms*). The results in Fig. 3 indicate that, if the minimum support in GSP analyzing the main sequence is comparatively small (less than 1500 with the examined data set), GSP finds much more frequent sequences than ProMFS. When the minimum support in GSP grows from 2500 to 6000, the number of frequent sequences by GSP decreases and that by ProMFS increases. In the range of [2500, 6000], the number of frequent sequences found by both GSP and ProMFS is rather similar. As the minimum support in GSP continues growing, the number of frequent sequences found by both algorithms becomes identical. When comparing the computing time of both algorithms (see Fig. 4), we can conclude that the ProMFS operates much faster. In the range of the minimum support in GSP [2500, 6000] ProMFS needs approximately 20 times less of computation time as compared with GSP to obtain the similar result.

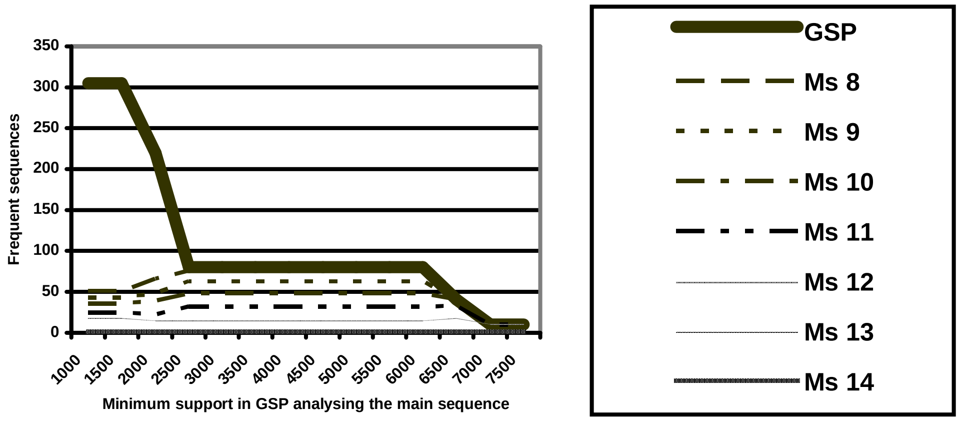

Fig. 3. The number of frequent sequences found by GSP and ProMFS (the minimum support in ProMFS is *Ms*=8, ... ,14)

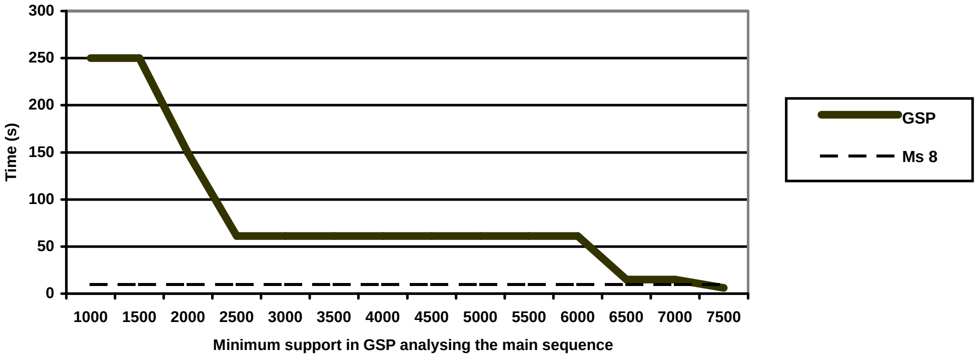

Fig. 4. The computing time by GSP and ProMFS (the minimum support in ProMFS is *Ms*=8)

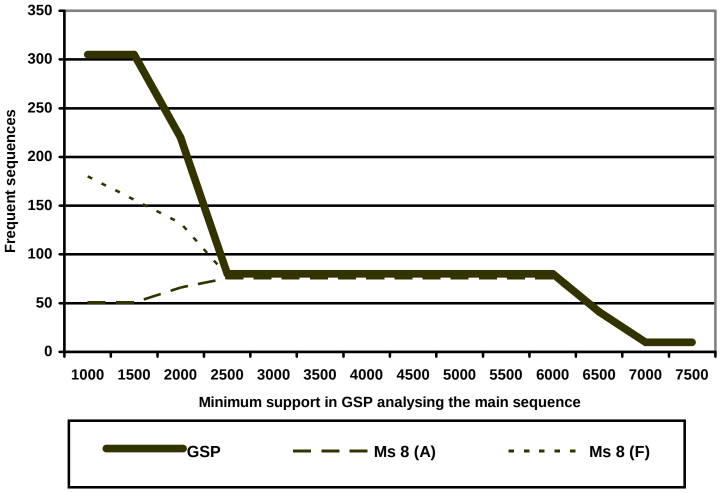

Fig. 5. The number of frequent sequences found by GSP and ProMFS with the average matrix (Ms 8 (A)) and the frequent matrix (Ms 8 (F)) (the support in ProMFS is *Ms*=8)

Fig. 5 illustrates the expenditure of computation time by both GSP and ProMFS with two different matrixes to obtain the results of Fig. 3 (the minimum support in ProMFS is *Ms*=8; the results are similar for larger *Ms*). The results in Fig. 3 indicate that if the minimum support in GSP analyzing the main sequence is comparatively small (less than 1500 with the examined data set), GSP finds many more frequent sequences than ProMFS. But the results of ProMFS with the matrix of frequent distance are better than with the matrix of average distance.

Databases are designed for programs to process automatically; text is written for people to read. Text mining methods can be used in bioinformatics for analysis of DNA sequences. A DNA sequence (sometimes called a genetic sequence) is a succession of letters representing the primary structure of a real or hypothetical DNA molecule or strand, The possible of sequence letters are A, C, G, and T, representing the four nucleotide subunits of a DNA strand (adenine, cytosine, guanine, thymine), and these are typically printed with no spaces between them, as in the sequence AAAGTCTGAC.

We have worked with real DNA data from ftp://ftp.ncbi.nih.gov/genbank/ (*34565 Homo sapiens chromosome 1 genomic contig).* Size is 250 Mb.
The results are shown in Fig. 6 and Fig. 7.

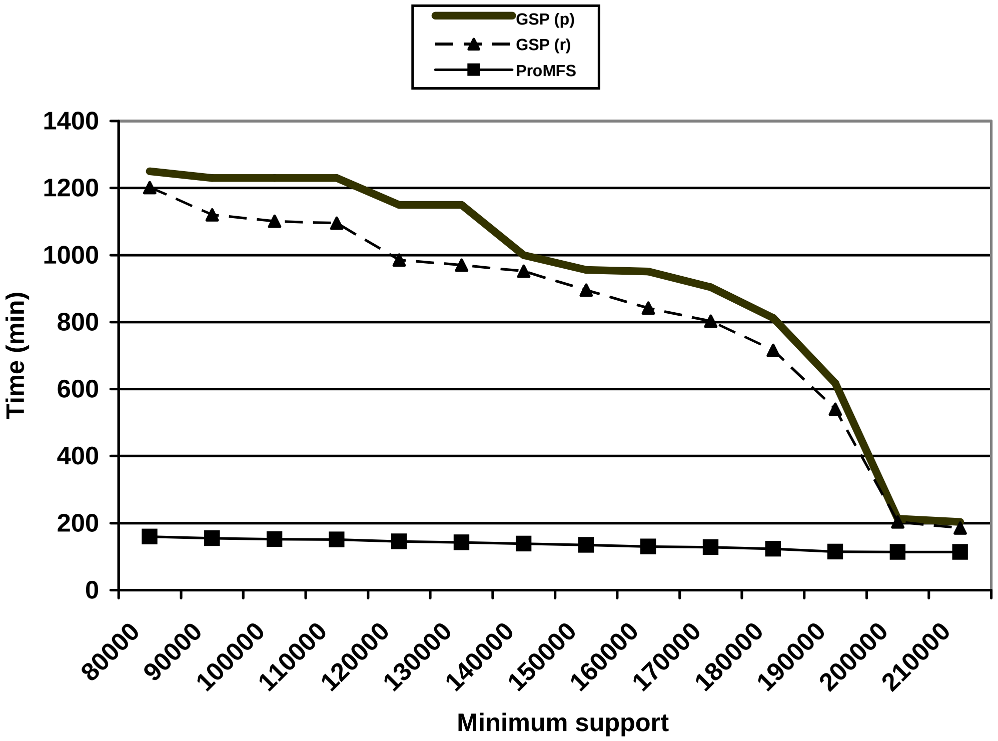

Fig. 6. The computation time by both GSP and ProMFS in Homo sapiens chromosome 1 genomic contig

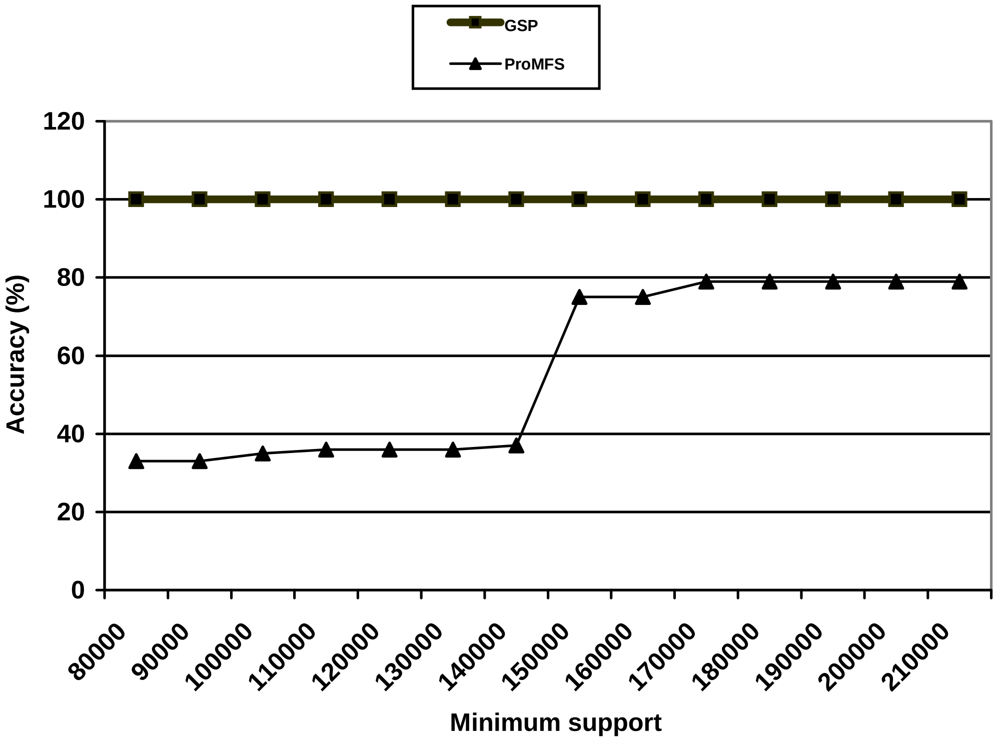

Fig. 7. The accuracy by both GSP and ProMFS in Homo sapiens chromosome 1 genomic contig

General conclusions

Two implementations of the Generated sequence pattern (GSP) algorithm and a recursive algorithm have been examined. The first implementation disregards saving memory, while the second one minimizes the memory consumption. Both implementations and a recursive algorithm are time intensive. Therefore, intense memory use requires relatively little time and is observed in case of large data sets. The best memory saving algorithm is recursive algorithm, because this algorithm uses memory just for frequent sequences, but not for all generated candidates.

The new algorithm ProMFS for mining frequent sequences with matrix of frequent distance has been proposed. It is based on the estimated probabilistic-statistical characteristics of the appearance of elements of the sequence and their order: the probability of an element in the sequence, the probability for one element to appear after another one, and the frequent distance between different elements of the sequence. The algorithm builds a much shorter new model sequence and makes the decision on the main sequence in accordance to the results of analysis of the shorter one. The model sequence may be analyzed by the GSP or other algorithm for mining frequent sequences: the frequency of subsequences in the main sequence is estimated by the results of the model sequence analysis.

The experimental research indicates that the new algorithm modification enables saving the computation time to a large extent and loses fewer sequences compared with the older algorithm modification. It is very important when analyzing very large data sequences.

DAŽNŲ SEKŲ PAIEŠKA DIDELIUOSE DUOMENŲ MASYVUOSE

(daktaro disertacijos santrauka)

Aktualumas

Pastaruoju metu, sparčiai vystantis informacinėms technologijoms bei greitoms ir efektyvioms duomenų saugojimo bei įrašymo formoms, visuomenę „užplūdo“ didžiuliai informacijos bei duomenų srautai iš pačių įvairiausių sričių. Dabar kiekvienos įmonės (komercinės, gamybinės, medicininės, mokslinės ir t.t.) veikloje vyksta įvairiausių įrašų registracija ir įvairiapusės informacijos apie įmonės veiklą saugojimas bei kaupimas. Be produktyvaus šios informacijos apdorojimo, šie duomenys su laiku gali pavirsti į visiškai beverčią „šiukšlių“ krūvą. Visi šie duomenys gali pasižymėti šiomis savybėmis:

- duomenys gali turėti neribotą dydį;
- duomenys gali būti įvairių tipų (loginiai, skaitiniai, tekstiniai);
- rezultatai turi būti konkretūs ir aiškūs;
- programiniai įrankiai, apdorojantys šiuos duomenis, turi būti paprasti naudojime.

Tokiu būdu labai aktualu rasti tarp šių didelių duomenų masyvų mums svarbią informaciją, kurią būtų galima panaudoti ateityje. Kitais žodžiais tariant, iš didelės masės duomenų reikia atrinkti informacijos „perlus“. Tai ir yra pagrindinis duomenų gavybos tikslas. Vienas iš svarbiausių jos tikslų yra dažnų pasikartojamumų radimas. Dar visai neseniai bet kuriai informacijos apdorojimo sistemai pakakdavo spręsti įvairius paieškos (surasti, kur ir kiek kartų pasikartoja nurodytas įrašas) arba statistinius uždavinius: koks yra vidutinis avaringumas (gimstamumas, nusikalstamumas) respublikoje, duotame rajone, per kažkokį laikotarpį ir t.t.. Duomenų gavybos technologijos nagrinėja šiuos duomenys žymiai sudėtingiau ir pateikia gana detalius šios analizės rezultatus. Duomenų gavyba leidžia atsakyti į klausimus, kokie žodžiai dažniausiai pasikartoja tekste, kokių kriterijų visuma turi įtaką avaringumui (gimstamumui, nusikalstamumui) respublikoje, duotame rajone ar per kažkokį laiko tarpą. Ne ką mažesnę svarbą duomenų gavyba turi medicinoje, nustatant žmogaus geno kodą ar kriterijus, pagal kuriuos žmonės serga viena ar kita liga. Taip pat duomenų gavyba yra populiari bankininkystėje, nustatant kokius kriterijus atitinka žmonės, kurie negali grąžinti paskolų. Prekybininkai irgi savo veikloje naudoja (arba gali naudoti) duomenų gavybos metodus, nustatant populiariausių prekių „krepšelius“, kurie buvo įsigyjami pirkėjų vieno apsipirkimo metu.

Anglišką terminą *data mining* galima būtų išversti kaip „duomenų gavyba“ arba „duomenų kasyba“. Tačiau nei vienas iš šių terminų nėra pakankamai išsamus ir iki galo neatspindi sąvokos prasmės. Dažnai su minėtu terminu yra siejamas terminas *žinių (iš)gavimas iš duomenų bazės* (angl. *knowledge discovery in databases*) ir *intelektinė duomenų analizė.* Šiuos terminus galima laikyti *data mining* sinonimais. Šių terminų atsiradimą sąlygojo nauji duomenų analizės metodai.

Tyrimo objektas

Disertacijos tyrimo objektas yra duomenų gavybos (angl. *data mining*) technologijos algoritmai, skirti nustatyti dažnus fragmentus. Nagrinėjamas atskiras duomenų gavybos atvejis, kai duomenys yra paprasti simboliai. Disertacijoje aprašytuose tyrimuose duomenys yra didelės apimties tekstiniai failai, kuriuose bus ieškoma dažnų sekų.

Mokslinis naujumas

Disertacijoje sukurtas, išnagrinėtas ir realizuotas naujas tikimybinis algoritmas, kuris remiasi elementų pasirodymo duomenyse tikimybinėmis charakteristikomis. Šis algoritmas yra apytikslis, tačiau veikia žymiai greičiau nei tikslieji algoritmai. Tokiu būdu jis yra naudojamas tais atvejais, kai dėl ženkliai didesnio pakankamai gerų rezultatų gavimo greičio, galime paaukoti tikslumą.

Tyrimo tikslai ir uždaviniai

Pagrindinis mokslinio tyrimo tikslas yra išnagrinėti kelis populiariausius dažnų sekų nustatymo algoritmus ir juos modifikuoti, o taip pat sukurti naują algoritmą specialiems uždaviniams spręsti. Tam tikrais atvejais tenka „aukoti“ tikslumą tam, kad gautume kuo greičiau pakankamai gerus rezultatus. Tokiu būdu pagrindinis disertacijos tikslas yra pasiūlyti apytikslį, bet greitą algoritmą, kurio pagalba galima būtų surasti dažnas sekas bei nustatyti tų sekų išsamumą.

Siekiant įgyvendinti suformuluotus tikslus, reikia išspręsti tokius uždavinius:

1) išnagrinėti dažniausiai naudojamus dažnų sekų nustatymo būdus bei algoritmus;

2) realizuoti GSP (*Generate Sequence Pattern*) algoritmą ir išanalizuoti jo veikimą su testiniais duomenimis;

3) modifikuoti realizuotą GSP algoritmą. Išanalizuoti šias modifikacijas su testiniais duomenimis;

4) patikrinti algoritmo modifikacijų greičio priklausomybę nuo duomenų dydžio, simbolių aibės duomenyse, o taip pat priklausomybę nuo tam tikrų dažnas sekas apsprendžiamų parametrų (pvz. minimalaus dažnumo);

5) sukurti ir realizuoti naują apytikslį algoritmą, kuris remiasi elementų pasirodymo duomenyse tikimybinėmis charakteristikomis;

6) palyginti šį algoritmą su tiksliuoju algoritmu (mūsų atveju modifikuotu GSP algoritmu);

7) išnagrinėti šio algoritmo tikslumo priklausomybę nuo įvairių pačio algoritmo ir duomenų charakteristikų;

8) realizuoti kitą naujojo algoritmo variantą ir išnagrinėti jo privalumus bei trukumus;

9) išnagrinėti realius duomenis su naujai pasiūlytu ir modifikuotu GSP algoritmu. Palyginti nagrinėjimo rezultatus.

Rezultatai

Disertacijoje buvo nagrinėjama dažnų sekų paieška dideliuose duomenų masyvuose. Buvo atlikti tokie darbai:

- Modifikuotas GSP dažnų sekų nustatymo algoritmas.
- Išnagrinėtas GSP algoritmo realizavimo būdas, panaudojant rekursinį dažnų sekų nustatymo būdą „gilyn“.
- Algoritmai palyginti pagal laiko ir atminties sąnaudas.
- Algoritmai palyginti pagal nagrinėjamos duomenų bazės pobūdį.
- Realizuotas naujas tikimybinis algoritmas ProMFS su dviem jo modifikacijomis: naudojant vidurkių matricą bei dažniausių atstumų matricą.
- Pateikti tikimybinio algoritmo su dviem modifikacijomis ir tikslaus GSP algoritmo palyginimai pagal laiko sąnaudas, tikslumą ir šių charakteristikų priklausomybę nuo failų dydžio bei elementų kiekio nagrinėjamame faile.
- Buvo išnagrinėti realūs duomenis su naujai pasiūlytu ir modifikuotu GSP algoritmu ir pateikti nagrinėjimo rezultatai.

Trumpos žinios apie autorių

Romanas Tumasonis studijavo Vilniaus universitete Matematikos ir informatikos fakultete, kurį baigė 1992 metais. Po baigimo pradėjo dirbti Vilniaus kolegijoje Elektronikos ir informatikos fakultete (buv. Aukštesnioji elektronikos mokykla) Programinės įrangos katedros dėstytoju. Vilniaus kolegijoje docento pareigose dirba iki šiol. Dėstomi dalykai yra operacinės sistemos, struktūrinis programavimas ir algoritmai, UNIX šeimos operacinės sistemos.

Vedęs. Turi 3 vaikus.

r.tumasonis@eif.viko.lt