VILNIUS UNIVERSITY

LORETA SAVULIONIENĖ

ASSOCIATION RULES SEARCH IN LARGE DATA BASES

Summary of Doctoral Dissertation

Technological Sciences, Informatics Engineering (07 T)

Vilnius, 2014

The doctoral dissertation was accomplished at Vilnius University Institute of Mathematics and Informatics in the period from 2008 to 2013.

Scientific Supervisor
Prof. Dr. Habil. Leonidas Sakalauskas (Vilnius University, Technological Sciences, Informatics Engineering – 07 T).

The dissertation will be defended at the Council of the Scientific Field of Informatics Engineering at the Institute of Mathematics and Informatics of Vilnius University:

Chairman
Prof. Dr. Habil. Gintautas Dzemyda (Vilnius University, Technological Sciences, Informatics Engineering – 07 T).

Members:
Prof. Dr. Habil. Juozas Augutis (Vytautas Magnus University, Physical Sciences, Informatics – 09 P),
Prof. Dr. Habil. Antanas Čenys (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering – 07 T),
Assoc. Prof. Dr. Vitalijus Denisovas (Klaipėda University, Technological Sciences, Informatics Engineering – 07 T),
Prof. Dr. Julius Žilinskas (Vilnius University, Technological Sciences, Informatics Engineering – 07 T).

Opponents:
Prof. Dr. Robertas Damaševičius (Kaunas University of Technology, Technological Sciences, Informatics Engineering – 07 T),
Assoc. Prof. Dr. Olga Kurasova (Vilnius University, Physical Sciences, Informatics – 09 P).

The dissertation will be defended at the public session of the Scientific Council of the Scientific Field of Informatics Engineering in the auditorium number 203 at Vilnius University Institute of Mathematics and Informatics, at 1 p. m. on the 12th of May, 2014.

Address: Akademijos st. 4, LT-08663 Vilnius, Lithuania.

The summary of the doctoral dissertation was distributed on the 11th of April 2014.

A copy of the doctoral dissertation is available for review at the Library of Vilnius University.

VILNIAUS UNIVERSITETAS

LORETA SAVULIONIENĖ

SUSIETUMO TAISYKLIŲ PAIEŠKA DIDELĖSE DUOMENŲ BAZĖSE

Daktaro disertacija

Technologijos mokslai, informatikos inžinerija (07T)

Vilnius, 2014

Disertacija parengta 2008 – 2013 metais Vilniaus universiteto Matematikos ir informatikos institute.

Mokslinis vadovas:
prof. habil. dr. Leonidas Sakalauskas (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07 T).

Disertacija ginama Vilniaus universiteto Matematikos ir informatikos instituto Informatikos inžinerijos mokslo krypties taryboje:

Pirmininkas
prof. habil. dr. Gintautas Dzemyda (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07 T).

Nariai:
prof. habil. dr. Juozas Augutis (Vytauto Didžiojo universitetas, fiziniai mokslai, informatika – 09 P),
prof. habil. dr. Antanas Čenys (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija – 07 T),
doc. dr. Vitalijus Denisovas (Klaipėdos universitetas, technologijos mokslai, informatikos inžinerija – 07 T),
prof. dr. Julius Žilinskas (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07 T).

Oponentai:
prof. dr. Robertas Damaševičius (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – 07 T);
doc. dr. Olga Kurasova (Vilniaus universitetas, fiziniai mokslai, informatika – 09 P).

Disertacija bus ginama viešame Informatikos inžinerijos mokslo krypties tarybos posėdyje 2014 m. gegužės 12 d. 13 val. Vilniaus universiteto Matematikos ir informatikos institute, 203 auditorijoje.
Adresas: Akademijos g. 4, LT-08663 Vilnius, Lietuva.

Disertacijos santrauka išsiuntinėta 2014 m. balandžio 11 d.

Disertaciją galima peržiūrėti Vilniaus universiteto bibliotekoje.

1. Introduction

Information is the currency of our era, but what is its value?
Information is not knowledge, and it surely is not wisdom.
John Lippman

Relevance of the problem

Nowadays activity of any company is based on large amounts of information and data. Both important and null information is hidden in large amounts of data. Effective discovery and usage of information hidden in data are the most important criteria to increase competitive ability in modern dynamic business environment. To solve these problems data mining is applied.

The basis of the task of mining frequent sub-sequence and discovery of association rules is the concept of patterns mapping data relationships. These patterns discover internal data structure and consistent patterns common to data subsets, which are presented in the form understandable to a user i.e. as association rules.

The discovery of association rules is applied in business, financial institutions, medicine, distance learning and in other spheres where large amounts of information should be processed and relationships between data should be discovered.

The discovered data relationships help analysts to faster and in a more accurate way make decisions, therefore, the discovery of association rules is an important task.

The dissertation proposes a new approximate algorithm for mining frequent sub-sequences, its modifications and the stochastic algorithm for discovering association rules and presents the evaluation of the algorithm errors. The results of the algorithms have been compared with other exact and approximate algorithms.

The object of the research

The object of the dissertation research is data mining algorithms and methods for solving the tasks of mining frequent sub-sequences and association rules. Simulated and real databases have been used for the described research of the dissertation.

The aim and objectives

The aim of the dissertation is to propose a new approximate algorithm for mining frequent sub-sequences and association rules and its modifications, and to present the evaluation of the algorithm errors.

To achieve the overall aim, the following objectives have been set:

- Analyse most frequently used data mining methods and algorithms for mining frequent sub-sequences and association rules.
- Design a new approximate algorithm for mining frequent sub-sequences.
- Evaluate accuracy, speed and statistical characteristics of the algorithm.
- Compare the designed algorithm with Apriori, GSP, SPADE, recursive and probabilistic ProMFS algorithms for mining frequent sequences.
- Implement modifications of the designed algorithm to increase accuracy and speed.
- Implement the modification of the designed algorithm to discover association rules.
- Design software for experiments.
- Carry out experiments with simulated and real data and compare with other exact and approximate algorithms.

Scientific novelty

The relevant task of mining association rules is researched in the dissertation. In order to solve the task a new stochastic algorithm for mining frequent sub-sequences, its modifications and the stochastic algorithm for discovering association rules have been proposed and probabilistic characteristics of the algorithms have been estimated. These algorithms for mining frequent sub-sequences and association rules are approximate and scanning a database once discover frequent sub-sequences and association rules. The evaluation of new proposed algorithms errors has been performed using statistical methods. The performance of these algorithms is faster comparing with exact and analysed approximate algorithms for mining frequent sub-sequences.

Research methods

The main research methods applied in the dissertation are: search for information, data simulation, and systemization of information, analysis, comparative analysis, summing-up, statistical analysis, exploratory research and experimental research. Analysing other authors' scientific and experimental achievements in the field of mining

frequent sub-sequences and association rules the methods of search for information, data simulation, systemization, analysis, comparative analysis, exploratory research and summing-up have been used. To evaluate the designed algorithms experimental research method and statistical analysis have been used.

Practical significance

During the research the stochastic algorithm for mining frequent sub-sequences and its modifications SDPA1, SDPA2 and the stochastic algorithm for discovering association rules have been designed. Software, which implements Apriori, GSP, SPADE, recursive, ProMFS, the stochastic for mining frequent sub-sequences, SDPA1, SPDA2 and the stochastic for discovering association rules algorithms, was designed. This software has been used to carry out research and is prepared for real usage.

Defended statements

- In the dissertation designed stochastic algorithm for mining frequent sub-sequences, SDPA1, SDPA2 and stochastic algorithm for discovery of association rules are approximate but fairly accurate and fast.
- SDPA1 algorithm is the modification of the stochastic algorithm for mining frequent sub-sequences, which increases the accuracy of the algorithm when parameter $g \in [0,6; 1]$ is chosen.
- SDPA2 algorithm is the modification of the stochastic algorithm for mining frequent sub-sequences, which uses frequent one-item sub-sequences discovered by chosen exact algorithm for mining frequent sub-sequences. SDPA2 is more accurate than the stochastic algorithm for mining frequent sub-sequences and SDPA1.

Approbation and publications of the research

The main results of the research have been published in 7 scientific publications, the results have been presented in 2 international scientific conferences and 5 national conferences.

The scope of the dissertation work

The dissertation is written in Lithuanian. The dissertation consists of five chapters, the list of references and appendices. The chapters of the dissertation are: Introduction, Discovery of association rules, Algorithms for mining frequent sub-sequences, Stochastic algorithms for mining frequent sub-sequences, Results of the research, General conclusions. The scope of the dissertation: 125 pages, 10 tables, 21 pictures, 4 appendices. Reference to 108 resources has been made in the dissertation.

2. Discovery of association rules

Association rules enable us to estimate the relationship among consistent patterns of events or processes or in other words relate various facts of events. Discovery of association rules is one of the most important and widely researched task in data mining, which was first introduced and researched by R. Agrawal. The aim of the task is to extract interesting relationships among data, common patterns, association and randomness of data structures from databases and other data warehouses.

Let, a set $I=\{i_1, i_2, \ldots, i_n\}$ contain n items, D be a database of transactions, where each transaction T consists of a set of items, which belong to I, i.e. $T \subseteq I$. The set of items $X \subseteq I$ belongs to transaction T only then if $X \subseteq T$. The set of items X is called an itemset. The itemset which contains k items is called k – itemset. The support of the itemset X, marked *supp X* is the number of transactions, where this itemset is the subset of transactions. The itemset is called frequent if repetition is not less than indicated minimum support *min_supp*.

Definition 1. An association rule is an implication of the form $X \Rightarrow Y$, where itemsets: $X \subseteq I$, $Y \subseteq I$ and $X \cap Y = \varnothing$.

Definition 2. Support of the association rule is called a value $supp(X \Rightarrow Y) = \frac{supp(X \cup Y)}{|I|}$.

Definition 3. Confidence of the association rule $X \Rightarrow Y$ is called a value $conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)}$, i.e. conditional probability that the itemset Y is in the transaction on condition that, the itemset X is in it.

Discovering association rules these values are defined: minimum support of the association rule *min_supp* and minimum confidence *min_conf*.

Discovering association rules, it is considered that all the analysed items are the same (homogeneous). If information on the attribution of an item to a specific group of items is added to the transaction, extended transactions could be analysed.

Analysing extended transactions association rules, which relate item groups and separate items with groups of items, etc., could be discovered. The association rules which include items from different hierarchy levels are called generalized association rules.

There are a lot of algorithms designed for discovering association rules. Estimation of the number of association rules is the essential problem. Big number of association rules, which are obvious or are absolutely useless i.e. uninteresting, is often discovered, therefore, the interest parameter of generalized association rules can be used.

Discovery of association rules refers to the anti-monotonous property, i.e. any k – item itemset is frequent only then if all contained (k-1) – item itemsets are frequent.

In general the task of the discovery of interesting association rules is defined as follows:

Let D be the set of transactions, I be the set of items, which have hierarchical relationships. We need to find expressions $X \Rightarrow Y$, which are generalized association rules, support of which is not less than defined minimal support value *min_supp*, confidence is not less than defined minimal confidence value *min_conf* and the rule $X \Rightarrow Y$ is interesting.

Discovery of association rules consists of two steps:

1. Discovery of the set of frequent items or sub-sequences.
2. Creating an association rule according to defined set of frequent items or sub-sequences.

Association rules are new knowledge from stored data, which are understandable and useful for consumers. Discovery of association rules is widely applied in practice for different tasks in genetics, medicine, marketing, finance, etc.

3. Algorithm for mining frequent sub-sequences

Algorithms for mining frequent sub-sequences and association rules are divided into two groups: exact algorithms and approximate algorithms. Exact algorithms for mining frequent sub-sequences started to be developed since 1995. These algorithms are acting thoroughly scanning the original database several times, which leads to time-consuming, but exact algorithms are irreplaceable, when it is necessary to provide accurate results. These algorithms are applied to solving genetic, biological, medical and similar tasks in which the main goal is accuracy, not time consumption. Exact algorithms are designed using one of the following methodologies: Apriori; FP - Growth; Eclat.

The database of transactions consists of many different items, which can create a large amount of sub-sequences (itemsets). Agrawal and Srikant applied downward principle for designing sub-sequences, i.e. k – item itemset is common, if it consists of frequent (k-1) – item sub-sequences. This frequent itemset design principle is called the Apriori principle. Frequent one-item sub-sequences are determined by scanning the database, then two-item sub-sequence candidates are generated and the database is scanned to determine the support of candidates. This process is repeated until no longer frequent (k+1) – item sub-sequence candidates are generated from frequent k – item sub-sequences. This methodology is based on Apriori, GSP and other algorithms.

Frequent pattern – growth methodology for mining frequent sub-sequences does not apply sequence-candidates generating principle. This methodology is based on divide – and – conquer principle. First time scanning the database a list of frequent items is generated, the items in it are arranged in descending order of support. Based on the list of frequent items, the database is transformed into frequent sub-sequence tree, i.e. FP – tree which stores information about association of items. This methodology is based on PrefixSpan, LAPIN and other algorithms.

Zaki proposed Eclat methodology examines data stored in a vertical format. First scanning the database all frequent one – item sub-sequences are identified.

(*k+1)* – length sub-sequences are created using *k* – length frequent sub-sequences, i.e. the Apriori principle, however, depth – search methodology is used which is similar to FP – growth. Generating (*k*+1) – length sub-sequences subordination in the transaction is taken into account. This methodology is based on SPADE, SPAM, PRISM and other algorithms.

The number of tasks of mining frequent sub-sequences, which accept errors made by algorithms, is growing. Often, the main test of the task is time, therefore, this condition is fulfilled only by approximate algorithms. Many activities seek to get a quick answer to the question "What object (or objects) is (are) the most frequent?" but they do not need the exact number of frequent object (or objects) in the database, therefore, accuracy can often be sacrificed for significantly higher speed in obtaining results.

Approximate algorithms for mining frequent sub-sequences are faster than the exact mining algorithms, because they are usually applied not to the entire database, but to smaller sample of the database. The strategies of approximate algorithms are based on MRA (*Multi Resolution Analysis*), PAC (*Probably Approximate Correct*) and Shannon's sampling theorems. Solving tasks of business, financial markets, insurance, consumer or customer behaviour, telecommunications, etc., errors made by approximate algorithms are acceptable, as often the main test of the task is time and the answer to the question "What object (or objects) is the most frequent?" is more important than to get the exact number of frequent object (or objects) in the database, therefore accuracy can often be sacrificed for significantly higher speed in obtaining results.

ApproxMAP (*Approximate Multiple Alignment Pattern Mining*) algorithm, instead of the discovery of exact frequent sub-sequences identifies sub-sequences which are often used in many other sub-sequences. This algorithm scans the original database several times searching for approximate sub-sequences.

ProMFS (*Probabilistic Algorithm for Mining Frequent Sequences*) algorithm is based on probabilistic characteristics which define item positions in the main sequence, and generate new considerably shorter model sequence which is analysed by GSP algorithm and estimates frequent sub-sequences in the sequence and indicates frequent sub-sequences in the original sequence of the database. ProMFS algorithm is based only

on empirical experiments and observations on how the algorithm acts in different databases, but it does not have theoretical estimate of errors made by the algorithm.

RSM (*Random Sampling Method*) analyses not the entire original database but much shorter random sample of the database. Random size sequence with the same probability is generated. Errors are estimated by standard methods of mathematical statistics, i.e. based on the properties of binomial distribution for a sample with replacement and hyper-geometric distribution for a sample without replacement and by the central limit theorem.

4. Stochastic algorithms for mining frequent sub-sequences

New proposed stochastic algorithms are approximate. The aim of the algorithms is to discover frequent sub-sequences in large databases and distinguish association rules. The advantage of these algorithms is that the database is scanned once and random length sub-sequences are randomly selected or omitted. The lengths of selected and omitted sub-sequences are distributed according to the uniform distribution. These algorithms allow us to combine two important tests, i.e. time and accuracy. Error probabilities of the designed stochastic algorithms are estimated using standard statistical methods.

New proposed stochastic algorithm for mining frequent sub-sequences

The database D is analysed. To discover frequent sub-sequences randomly chosen random length sub-sequences are analysed. As there is no information that some items occur more often than others in the database, so it is obvious that any item can be in the frequent sub-sequence with the same probability for all items. This probability is defined as q. It is noticed that the number of analysed sub-sequences is distributed according to the uniform distribution with parameter q, and the spacing lengths between the two analysed sub-sequences are also distributed according to the uniform distribution with parameter g.

The support of the sub-sequence is equal to the support among all selected sub-sequences. Relational support of the sub-sequence is equal to the support among all selected same length sub-sequences.

Let, when analysing the database *D*, randomly choose *N* (number of samples) various length sub-sequences s_k which are grouped according to the length of sub-sequences. The support of the sub-sequences *supp* (s_k) of particular length *k* is calculated according to the formula:

$$supp(s_k) = \frac{N_k}{N}, \text{ where } k = 1, 2, \ldots, n, \quad (1)$$

N_k the number of sub-sequences of particular length, *N* is the number of all sub-sequences, *k* is the length of sub-sequence, *n* is maximum length of sub-sequence.

The sub-sequence refers to the set of frequent sub-sequences, if the support exceeds the defined minimum support value *min_supp*, i.e. $supp(s_k) \geq min\_supp$.

The stochastic algorithm for mining frequent sub-sequences is approximate; therefore, the first and second type errors are possible.

The first type error is when the sub-sequence is frequent, but the stochastic algorithm does not recognize and refer to the set of frequent sub-sequences.

The second type error is when the sub-sequence is not frequent, and the stochastic algorithm refers it to the set of frequent sub-sequences.

The statistics p_1, p_2, which satisfy the inequality: $P(p_1 \leq p \leq p_2) = \gamma$ are chosen. The interval $[p_1; p_2]$ is called a confidence interval of the parameter *p*. The number γ is called a confidence level. The test of accuracy of the stochastic algorithm is the bound of confidence interval of the discovery of the sub-sequence.

The bound of confidence intervals are estimated according to the formulas:

$$p_1 = 1 - BetaInv\left(\frac{1-\gamma}{2}, n-k, k+1\right); \quad (2)$$

$$p_2 = 1 - BetaInv\left(\left(1-\frac{1-\gamma}{2}\right), n-k+1, k\right); \quad (3)$$

where p_1 and p_2 are the bound of confidence intervals, *n* is the number of all sub-sequences, *k* is the number of the particular sub-sequence appearance, *BetaInv* is quintile of the beta distribution, γ is confidence level.

Probability of the first type error is when $p_2 < \gamma$.

Probability of the second type error is when $p_1 > \gamma$.

General scheme of the stochastic algorithm for mining frequent sub-sequences:

Step 1. The database file is scanned.

Step 2. Maximum value n of the sample sub-sequence is input. The value defines maximum length of the sub-sequence, which could be taken for further analysis.

Step 3. The initial value q is input. The value of the variable q is in the interval [0; 1].

Step 4. The empty file of results is created. All sub-sequences selected processing the algorithm will be stored in the file. In the file all sub-sequences are stored ranged by length and support which is calculated according to formula (1).

Step 5. The value of the variable g is generated. The value is distributed according to uniform distribution and is in the interval [0; 1].

Step 6. It is checked if value q is greater than value g, i.e. if inequality $q > g$ is true.

Step 7. If inequality $q>g$ is true, then the length l of sample sub-sequence is calculated by the formula $l = round\ (q \cdot n)$, i.e. the number which indicates what length sub-sequence should be taken. The length of the sample is in the interval [1; n]. If the number of items in the transaction is less than generated length of sample sub-sequence, then all items of the transaction are taken and then another transaction is proceeded.

Step 8. The length t of the omitted sub-sequence is calculated by the formula $t = round\ (g \cdot n)$. This number is in the interval [1; n]. This value is used to estimate the length of the omitted sub-sequence before other iteration. If the number of left items in the transaction is less than the calculated omitted length sub-sequence, then all left items of the transaction are omitted and then another transaction is proceeded.

Step 9. If value g is greater than the value of parameter q, i.e. inequality $q > g$ is false, then none item is omitted. In this case, the length of the omitted sub-sequence equals zero.

Step 10. The sub-sequence s_k of calculated length is taken. The length of this sub-sequence was calculated in Step 7.

Step 11. The list of the results is scanned for selected sub-sequence s_k.

Step 12. If the sub-sequence s_k is in the list of the results, then sub-sequence number indicator is increased by one.

Step 13. If sub-sequence s_k is not in the list of the results, then a new sub-sequence is added to the list and sub-sequence number indicator equals to one.

Step 14. It is checked if the end of the database file is not reached. If the end is not reached, the calculated length sub-sequence is omitted. The length of the omitted sub-sequence was calculated in Step 8 or equated to zero in Step 9.

Step 15. After the sub-sequence is omitted new values q and g are generated, the values are in the interval [0, 1]. Then it is reverted to Step 6 and all steps of the algorithm are repeated till Step 14.

Step 16. If the end of the database file is reached, then the obtained file of the results is arranged where sub-sequences are ranged by the length and support which is calculated according to formula (1). The number of particular length sub-sequences equals to the value of sub-sequence indicator described in Step 12 and Step 13.

Step 17. The result of stochastic frequent sub-sequence algorithm is sub-sequences, which are ranged by length and support, i.e. *Result:* $= \cup_k s_k$, where k =1, 2, ..., n.

These modifications have been accomplished to stochastic algorithm for mining frequent sub-sequences:

1. Parameters q or g of the stochastic algorithm for mining frequent sub-sequences are input and are consistent throughout the process of the algorithm. If both parameters are captured, the inequality $g>q$ is always true and the sub-sequence, the length of which is equal to zero, is never omitted throughout the process of the algorithm, or the inequality $g>q$ is always false, and no sub-sequence is omitted, i.e. the entire database, where all items are divided into certain length sub-sequences, is omitted. This modification of the stochastic algorithm for mining frequent sub-sequences is marked as SDPA1.

2. Randomly selected random length sub-sequences l, containing at least one frequent one – item sub-sequence identified by the exact algorithm for mining frequent sub-sequences, are analysed by the stochastic algorithm for mining frequent sub-sequences. This modification of the stochastic algorithm for mining frequent sub-sequences is marked as SDPA2.

For estimating errors in modified stochastic algorithms SDPA1, SDPA2 the same methodology is used as in the stochastic algorithm for mining frequent sub-sequences.

The scheme of the stochastic modified SDPA1 algorithm for mining frequent sub-sequences (when q value is fixed/ when g value is fixed):

Steps 1 – 4. Do not change.

Step 5. Does not change / The value of the variable g is input, which does not change throughout the process of the algorithm. The value is in the interval [0; 1].

Step 6. Does not change / It is checked if value g is greater than value q, i.e. if inequality $g>q$ is true.

Steps 7 – 14. Do not change.

Steps 15. After the sub-sequence is omitted the same value q, in the interval [0; 1], is used. Then it is reverted to 5 and all steps of the algorithm are repeated till Step 14. / After the sub-sequence is omitted the new value q, in the interval [0; 1], is generated. Then it is reverted to Step 5 and all steps of the algorithm are repeated till Step 14.

Steps 16 – 17. Do not change.

The scheme of the stochastic modified SDPA2 algorithm for mining frequent sub-sequences:

Step 1. Does not change.

Step 2. The initial value q is input. The value of the variable q is in the interval [0; 1]. Then frequent one-item sub-sequences, which are identified by another exact algorithm for mining frequent sub-sequences, are input.

Steps 3-9. Do not change.

Step 10. It is checked if in the sub-sequence s_k, intended for selecting, there is at least one frequent one-item sub-sequence, which was identified in Step 2. If the condition is true, then the sub-sequence of the calculated length is selected. The length of the sub-sequence was calculated in Step 7. In other case, Step 14 is processed.

Steps 11 – 16. Do not change.

Step 17. The result of the stochastic algorithm SDPA2 for mining frequent sub-sequences is sub-sequences with at least one frequent one-item sub-sequence, identified by the exact algorithm for mining frequent sub-sequences. In the file of the results these sub-sequences are ranged by length and support, i.e. $Result := \cup_k s_k$, where k=1, 2, ..., n.

The new proposed stochastic algorithm for discovering association rules

The discovering of association rules consists of two steps: mining of frequent sub-sequences and discovering association rules from the set of frequent sub-sequences.

Features, that design association rules from frequent sub-sequences, are added to the stochastic algorithm for mining frequent sub-sequences. This algorithm is called the stochastic algorithm for discovering association rules.

The scheme of the stochastic algorithm for discovering association rules:

Step 1. The database file is scanned.

Step 2. Maximum value n of the sample sub-sequence is input. The value defines maximum length of the sub-sequence, which could be taken for further analysis.

Step 3.The initial value q is input. The value of the variable q is in the interval [0; 1].

Step 4. Minimum sub-sequence support value *min_supp* is input.

Step 5. The empty file of results is created. All sub-sequences selected processing the algorithm will be stored in the file. In the file all sub-sequences are stored ranged by length and support which is calculated according to formula (1).

Step 6. The file of the results of the association rules is created·. Designed association rules will be stored in the file.

Step 7. The value of the variable g is generated. The number is in the interval [0; 1].

Step 8. It is checked if value q is greater than value g, i.e. if inequality $q > g$ is true.

Step 9. If inequality $q > g$ is true, then the length l of sample sub-sequence is calculated by the formula $l = round\ (q \cdot n)$, i.e. the number which indicates what length sub-sequence should be taken. The length of the sample is in the interval [1; n]. If the number of items in the transaction is less than calculated length of sample sub-sequence, then all items of the transaction are taken and then another transaction is proceeded.

Step 10. Random value, which indicates the item – length sub-sequence to be omitted, is calculated by the formula $t = round\ (g \cdot n)$. This number is in the interval [1; n]. The value is used to estimate the length of the omitted sub-sequence before other iteration. If the number of left items in the transaction is less than the length of the calculated omitted sub-sequence, then all left items of the transaction are omitted and then another transaction is proceeded.

Step 11. If value g is greater than the value of parameter q, i.e. inequality $q > g$ is false, then none item is omitted. In this case, the length of the omitted sub-sequence equals zero.

Step 12. The process of taking calculated length sub-sequence s_k is performed. The length of this sub-sequence was calculated in Step 9.

Step 13. The list of the results is scanned for selected sub-sequence s_k.

Step 14. If the sub-sequence s_k is in the list of the results, then sub-sequence number indicator is increased by one.

Step 15. If sub-sequence s_k is not in the list of the results, then a new sub-sequence is added to the list and sub-sequence number indicator equals to one.

Step 16. It is checked if the end of the database file is not reached. If the end is not reached, the calculated length sub-sequence is omitted. The length of the omitted sub-sequence was calculated in Steps 10 – 11.

Step 17. After the sub-sequence is omitted new values q and g are generated, the values are in the interval [0, 1]. Then it is reverted to Step 8 and all steps of the algorithm are repeated till Step 16.

Step 18. If the end of the scanned database file is reached, then the obtained file of the results is arranged where sub-sequences are ranged by the length and support which is calculated according to formula (1).

Step 19. The sub-sequences, which are ranged by length and support, are saved in the file, i.e. *Result* : = $\cup_k s_k$, where k=1, 2, ..., n.

Step 20. All sub-sequences with the support not less than indicated support value *min_supp* and the number of items in the sub-sequence not less than two items, are selected.

Step 21. Association rules are created from every sub-sequence.

Step 22. The support (percentage) is calculated for every created association rule by the formula: $supp(X \Rightarrow Y) = \frac{supp(X \cup Y)}{|I|} \cdot 100\%$.

Step 23. The confidence (percentage) is calculated for every created association rule by the formula: $conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)} \cdot 100\%$.

Step 24. The result i.e. association rules, support and confidence of the association rules are saved in the file of the results created in Step 6.

Statistical characteristics

Statistical hypothesis is any inference about the distribution or parameters of the distribution of all the features of the analysed set.

The statistical hypothesis test is the rule that specifies the hypothesis rejections based on values of random sample.

The main inference about the distribution or its parameters is called null hypothesis H_0. Alternative inference, opposite to null hypothesis is marked H_1.

In order to determine which of the hypotheses is probative, a variety of statistical methods is used. In this case, evaluation of statistical tests is chosen.

The set of supports of sub-sequences is created. Then the change moment of statistical characteristics is defined.

There are two independent samples, the values of which are n_1 and n_2. In the first sample there are k_1 items with particular value of the feature found, and in the second there are k_2 items found.

The null hypothesis states that the proportions of the most frequent sub-sequences are identical in the database from which the samples are taken and the alternative hypothesis states that the probabilities are unequal:

$$H_0: r_1 = r_2;$$

$$H_1: r_1 \neq r_2.$$

From the rule of anti-monotonous it is obvious that if H_0 is true then analysed sub-sequences are frequent and the alternative hypothesis states that from analysed sub-sequences greater length sub-sequence is not frequent.

In the test hypothesis H_0 statistics of feature detection probability equality in samples can be evaluated in various ways. The test statistics u constructed in such a way that if the hypothesis H_0 is true, it would be distributed according to the standard normal distribution. The test statistics u is calculated by the formula:

$$u = \frac{d_1 - d_2}{\sqrt{\left(\frac{k_1 + k_2}{n_1 + n_2}\right) \cdot \left(1 - \frac{k_1 + k_2}{n_1 + n_2}\right) \cdot \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

where $d_1 = k_1/n_1$ and $d_2 = k_2/n_2$.

Let's indicate $d = (k_1+k_2) / (n_1+n_2)$, then the formula is:

$$u = \frac{d_1 - d_2}{\sqrt{d \cdot (1-d) \cdot \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} .$$

The test statistics z can also be evaluated this way:

$$z = \left(2 \arcsin \sqrt{d_1} - 2 \arcsin \sqrt{d_2}\right) \cdot \sqrt{\frac{n_1 \cdot n_2}{n_1 + n_2}} .$$

Evaluation of assumptions

To determine the moment of the change of sub-sequence characteristics a model in which the most frequent sub-sequence (e.g. market basket) is determined by the maximum likelihood method. The length of the frequent sub-sequences is determined by the monotony rule i.e. the subset of frequent sub-sequences is frequent sub-sequence. In the model for determining the moment of change of sequence characteristics the support of the most frequent sub-sequent is calculated, according to the length. Next these are applied:

- the length of the analysed sub-sequence is less than the particular length, then the support of the analysed sub-sequence is almost constant;
- the length of the analysed sub-sequence is greater than the particular length, then the probability of the analysed most frequent sub-sequence starts to decrease.

In this case, a binary process is designed, there two parallel particular length sub-sequences l_i and l_{i+1}, $i \in [1, N]$ are analysed. The value of the binary process is:

- equal to one, if the statistical test does not contradict the hypothesis of the probability of appearance of two parallel length sub-sequences match;
- equal to zero, if the probability of appearance of two parallel length sub-sequences significantly differ.

The moment of the change of characteristics of the binary process allows us to determine the length of the most frequent sub-sequences (the size of the market basket, i.e. how many items make up the most frequent market basket). As the number of appearance of two parallel sub-sequences in the samples is distributed according to the binomial law, logarithmic likelihood function can be created to determine the change of the probability of appearance of the most frequent sub-sequence:

$$f(k) = \frac{k!}{k_1! \cdot (k-k_1)!} \cdot \frac{(N-k)!}{(K-k_1)! \cdot (N-k-K+k_1)!} \cdot p_1^{k_1} \cdot (1-p_1)^{k-k_1} \cdot p_2^{K-k_1} \cdot \cdot (1-p_2)^{N-k-K+k_1}.$$

Logarithmic probability function is derived:

$$\ln\big(f(k)\big) = \sum_{i=1}^{k} \ln i - \sum_{i=1}^{k_1} \ln i - \sum_{i=1}^{k-k_1} \ln i + \sum_{i=1}^{N-k} \ln i - \sum_{i=1}^{K-k_1} \ln i - \sum_{i=1}^{N-k-K+k_1} \ln i + + \sum_{i=1}^{k_1} \ln p_1 + \sum_{i=1}^{k-k_1} \ln(1-p_1) + \sum_{i=1}^{K-k_1} \ln p_2 + \sum_{i=1}^{N-k-K+k_1} \ln(1-p_2),$$

where k – the moment of the change of characteristics of the binary process, k_1 – the number of matching of probability support till the change moment, k_2 – the number of matching of probability support after the change moment, N – maximum length of sub-sequence, K – the length of the analysed sub-sequence. The length of the most frequent sub-sequence corresponds the minimum value of logarithmic probability function. It is convenient to introduce minimizing function as the difference of two adjacent values of this function, which is equal to:

$$\ln(f(k)/f(k-1)) =$$

$$= \ln k - \ln(k-k_1) - \ln(N-k+1) + \ln(N-k-k_1+1) + \ln(1-p_1) + +\ln(1-p_2),$$

if k^{th} value of the binary process equals to 0.

If k^{th} value of the binary process equals to 1, then the difference of adjacent values of probability function is equal to:

$$\ln(f(k)/f(k-1)) = = \ln k - \ln k_1 - \ln(N-k+1) + \ln(K-k_1+1) + \ln p_1 + \ln p_2,$$

where $p_1 = \frac{k_1}{k}, p_2 = \frac{k_2}{k}$.

The minimum of the probability function coincides with the value of the first variable k, where the difference of two adjacent values of this function is positive. Calculations are started with the initial value $k = 0$. It is easy to notice, that the initial value of the probability function is:

$$\ln\big(f(0)\big) = \sum_{i=1}^{N} \ln i - \sum_{i=1}^{K} \ln i - \sum_{i=1}^{N-K} \ln i + K \cdot \ln p_2 + \ln(1-p_2) \cdot (N-K).$$

For calculating values of the logarithmic likelihood function recursive formulas can be used:

- if k^{th} value of the binary process equals to zero, then:

$$k_1(k+1) = k_1(k), \qquad k_2(k+1) = k_2(k);$$

- if k^{th} value of the binary process equals to one, then:

$$k_1(k+1) = k_1(k) + 1, \qquad k_2(k+1) = k_2(k) - 1.$$

After evaluation of the test statistics and the values of the likelihood function, the assessment of probability assumptions is carried out. When the alternative hypothesis is two-sided, the obtained value *u* corresponding to value *p* is calculated as follows:

$$p = 2 \cdot \Big(1 - NORMSDIST\big(ABS(u)\big)\Big).$$

p – value determines probability risk, that omitting H_0 the first type error will be made, therefore, H_0 can reasonably be omitted only if value *p* is not big, insignificant, less than common, traditional levels of significance (0.1; 0.05; 0.01 or 0.001). Value *p* – defines the likelihood of hypothesis H_0, i.e. the probability that the statement corresponds the reality, therefore, the greater the value *p* the more confident null hypothesis is.

5. Experimental research

Experimental research with the stochastic algorithms designed in the dissertation has been carried out using artificially generated and real data.

Computer running at 2.5GHz with Intel(R) Core(TM) i5-3210 processor, 4GB RAM memory has been used for experimental research. Apriori, GSP, recursive, SPADE, ProMFS algorithms, the stochastic algorithm for mining frequent sub-sequences, its modifications SDPA1, SDPA2 and the stochastic algorithm for discovering association rules have been implemented using Object Pascal programming language.

Experimental databases

Experimental research has been carried out using real database of purchase transactions, real database of the topics of final projects and generated databases.

Designed software has been used for generating databases.

These databases have been used for experimental research:

1. Generated databases.

2. Real database of the topics of final projects of Vilniaus kolegija / University of Applied Sciences Faculty of Electronics and Informatics Computer Programming study programme during 2000 – 2013. The database consists of 1,030 topics.
3. Real database of UAB "Arsuna" one – year purchase transactions. The database consists of 400,000 transactions.

Accuracy research of new proposed stochastic algorithm for mining frequent sub-sequences and SDPA1, SDPA2 algorithms

100 files of databases have been generated for the experiment, the size of which is 200,000 symbols, i.e. 2,000 lines, 100 symbols per line. The database consists of symbols *A*, *E*, *I*, *N*, *S*, i.e. $I = \{A, E, I, N, S\}$. The sub-sequence *SIENA* has been inserted among these symbols. The files of the database have been generated with these probabilities: sub-sequence *SIENA* insertion – 0.2; symbol *A* – 0.15; symbol *E* – 0.15; symbol *I* – 0.15; symbol *N* – 0.15; symbol *S* – 0.2.

These files are processed by new proposed stochastic algorithm for mining frequent sub-sequences, SDPA1, SDPA2 algorithms 100 times every. Maximum length of sub-sequence – 5 symbols – has been chosen in all experiments. In new proposed stochastic algorithms parameter q is used generating the value of the sub-sequence sample and determining if random length sub-sequence will be omitted after the sample. After the experiments, average time needed to process one file was estimated when values q are different. During the experiments the reliance of the results of SDPA1 algorithm on the value of parameter $g \in (0; 1)$. Average time of one file process by SDPA1 algorithm is presented in Figure 1.

Experimental results show that the higher the value of the parameter g, the faster the time of process of the algorithm increases, as frequently random length sub-sequence is not omitted. File processing time begins to increase when $g \geq 0.6$. When $g = 1$, file processing time increases as sub-sequence is not omitted, and the entire database is divided into sub-sequences.

In the experiment frequent sub-sequences have been determined by GSP, stochastic algorithm for mining frequent sub-sequences, SDPA1 and SDPA2 algorithms when the length of the sub-sequence is 5. Processing database files by GSP algorithm minimum support *min_supp*=1,000 has been chosen, and by SDPA1 algorithm parameter

g=0.5 has been chosen. All algorithms have identified the same frequent sub-sequences, only the number of sub-sequences differs. The number of frequent sub-sequences is approximately 4 times less of SDPA1 algorithm, when the chosen value of the parameter *g* is 0.5. The number of frequent sub-sequences is approximately 2 times less of SDPA2 algorithm.

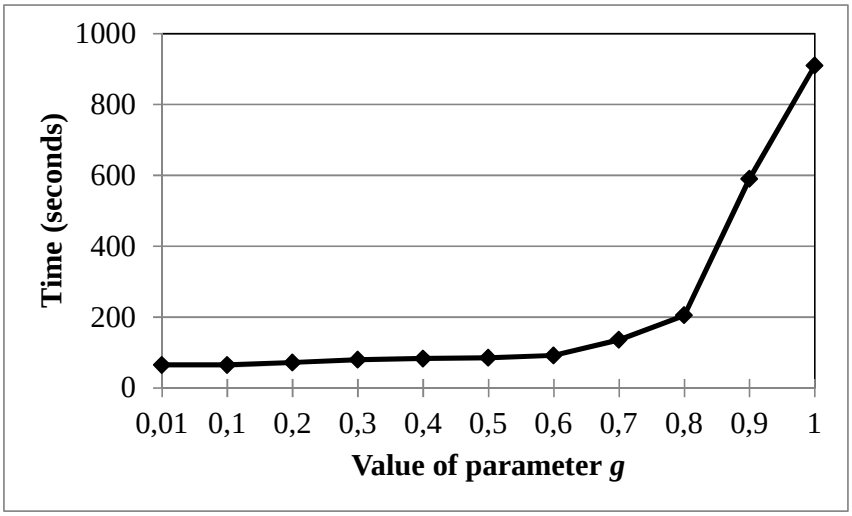

Fig. 1. SDPA1 algorithm process duration reliance on value of parameter *g*.

Comparison of the total number of frequent sub-sequences for different values of the parameter *g* is given in Figure 2.

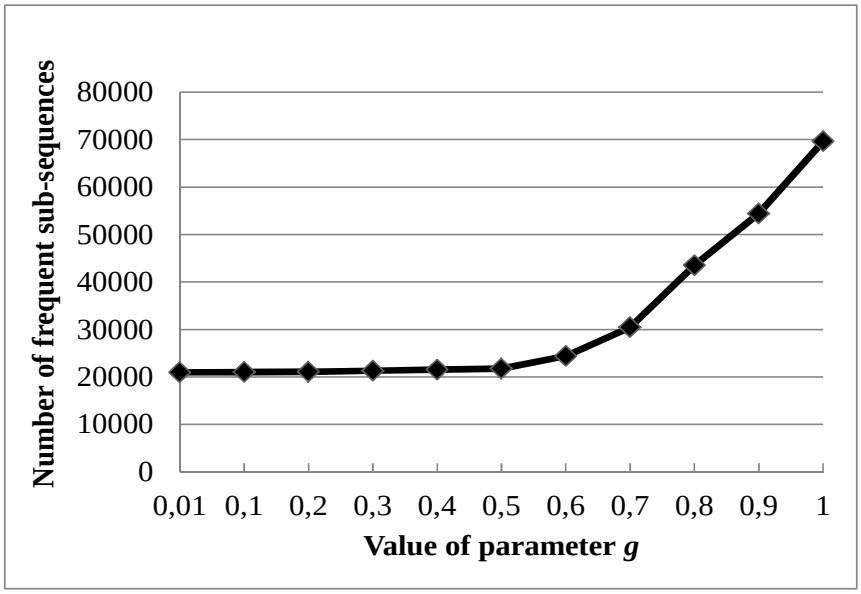

Fig. 2. Total number of frequent sub-sequences reliance on the value of the parameter *g* of SDPA1 algorithm

These databases have been processed by Apriori, GSP, SPADE, recursive, stochastic, SDPA1, SDPA2 and ProMFS algorithms. In Apriori, GSP, SPADE and recursive algorithms selected minimal sub-sequence support is *min_supp*=200,

min_supp=500, *min_supp*=1,000. Model set designed by ProMFS algorithm has been analysed by GSP algorithm, when *min_supp*=6, *min_supp*=15, *min_supp*=30. The value of the parameter *g* for SDPA1 algorithm: *q*=0.3. Average time for processing databases by the algorithms is presented in Figure 3.

The results of the experiment showed that the least time consumption is by SDPA1 algorithm, and the greatest time for processing databases is of recursive algorithm.

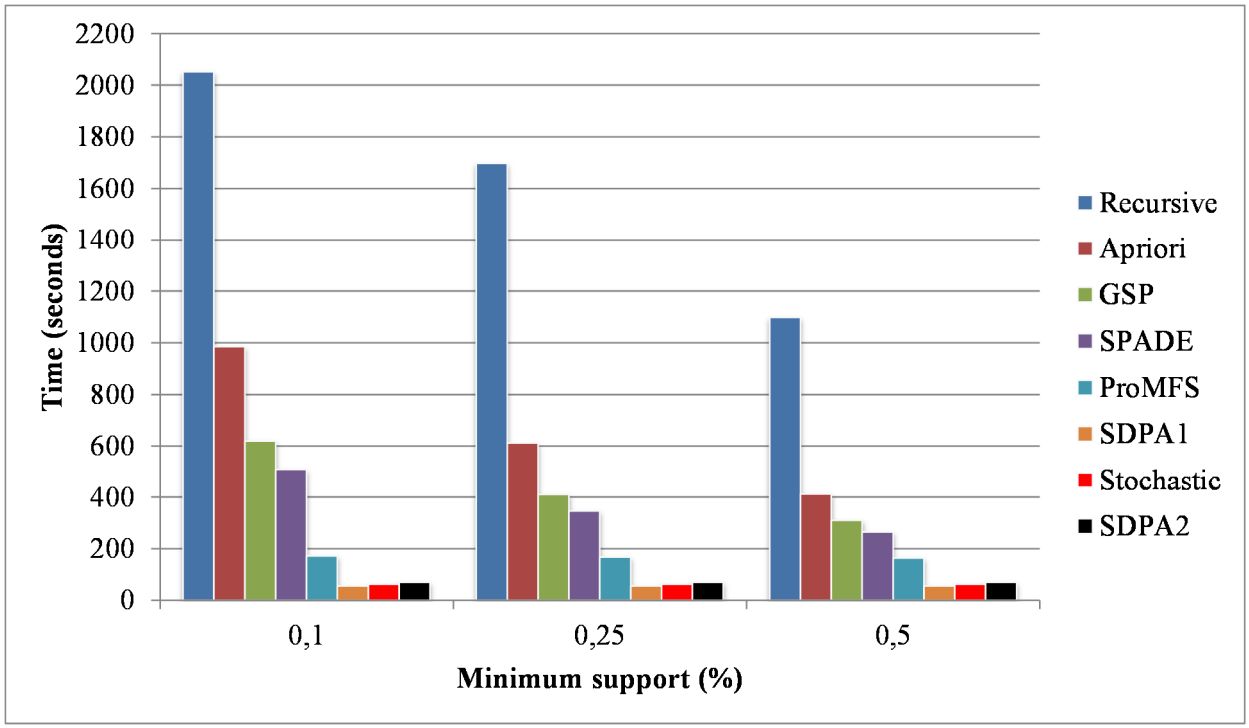

Fig. 3. Average processing time of Apriori, GSP, SPADE, recursive, ProMFS, stochastic, SDPA2 and SDPA1 algorithms.

The research of the accuracy of stochastic frequent sub-sequences algorithms SDPA1 and SDPA2

During the experiment 19 groups of databases have been generated. Each file group consists of 100 files. All 1,900 files have been processed by the stochastic algorithms for mining frequent sub-sequences, SDPA1 and SDPA2 algorithms 80 times every.

During the experiment first and second type errors of stochastic algorithms have been estimated

A sub-sequence refers to the set of frequent sub-sequences if its minimum support *min_supp* ≥ 0.07, when γ=0.95.

Confidence intervals p_1 and p_2 have been calculated according to the formulas (2) and (3) in every database. The confidence interval of the stochastic algorithm for mining frequent sub-sequence is [0.95; 0.99].

Confidence intervals p_1 and p_2 of SDPA1 algorithm have been calculated in every database with different values of the parameter g of the stochastic algorithm. Estimating the results of the experiment the determined confidence interval is [0.95; 0.99].

Confidence intervals p_1 and p_2 of SDPA2 algorithm have been calculated in every database. The confidence interval of SDPA2 algorithm is [0.97; 0.99].

The average first type error of stochastic algorithm for mining frequent sub-sequences is 2.38 %, the average second type error is 5.58 %. The average first type error of SDPA1 algorithm is 2.4 %, the second type error is 5.6 %. The average first type error of SDPA1 algorithm is 1.3 %, the average second type error is 3.12 %.

The research of the database of the topics of final projects

During the experiment the database of the topics of final projects of Vilniaus kolegija/ University of Applied Sciences Faculty of Electronics and Informatics Computer Programming study programme during 2000 – 2013 (13 academic years) has been researched. The database consists of 1,030 topics.

The aim of the experiment is to estimate the most frequent words in the topics of final projects and to discover association rules among the most frequent words.

During the experiment, one word is regarded as a single item. For example, the topic is “UAB “Skado medis” interneto svetainė”, then i_1=UAB, i_2=Skado, i_3=medis, i_4=interneto, i_5=svetainė.

During the experiment the database of the topics of final projects is analysed by exact SPADE algorithm and approximate ProMFS algorithm, the stochastic algorithm for mining frequent sub-sequences, SDPA1 and the stochastic algorithm for discovering association rules. During the experiment the value of the parameter g of SDPA1 algorithm equals 0.5.

During the experiment all algorithms (SPADE, ProMFS, stochastic algorithm for mining frequent sub-sequences, SDPA1 and the stochastic algorithm for discovering association rules) determined these most frequent items (words): Programa; UAB; Sistema; Svetainė; Tvarkymo; Modulis; Apdorojimo; Žaidimas.

Speed comparison of approximate ProMFS and stochastic algorithm for discovery of association rules SDPA1 in the database of the topics of final projects of Computer Programming study programme depending on the defined minimum support *min_supp* is presented in Figure 4.

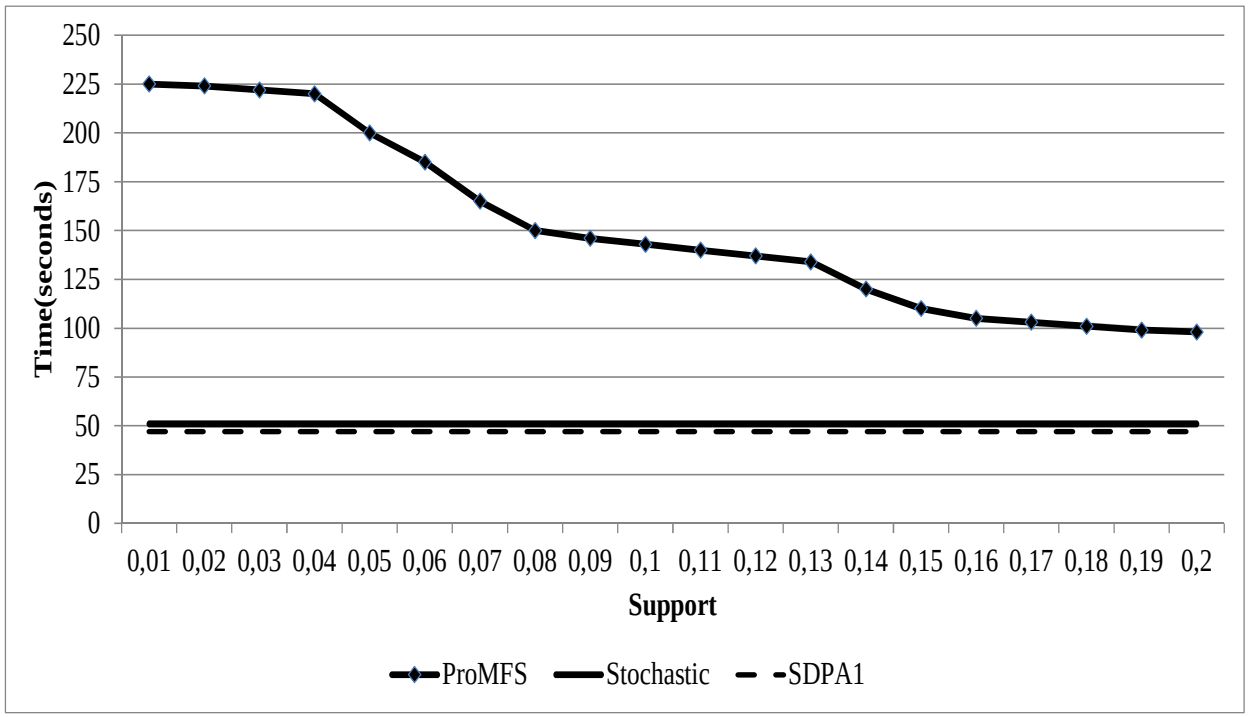

Fig. 4. Comparison of the speed of approximate methods.

The results of the experiment proved the conclusion of the ProMFS algorithm author dr. R. Tumasonis who states that *ProMFS algorithm is effective (much shorter time consumption), when minimum support is comparatively large*. The defined minimum support does not influence the efficiency of stochastic algorithm because frequent sub-sequences are chosen randomly. The value of minimum support influences only the number of created association rules.

The most frequent words in the topics of final projects during 2000 – 2013 are: *UAB*, *Programa*, *Sistema*, *Svetainė*, *Tvarkymo*, *Modulis*, *Apdorojimo*, *Žaidimas*. During the experiment chosen *min_supp* = 0.01.

Frequent 2–item sub-sequences and their support, when *min_supp* = 0.01 are presented in table 1.

The stochastic algorithm for discovery of association rules has determined association rules and calculated the support and confidence of association rules Association rules with minimum confidence *min_conf* = 2% are presented in table 2.

Table 1. 2 – item sub-sequences and the support.

Sub-sequence	**Support**
{*UAB, Programa*}	0.11
{*UAB, Sistema*}	0.08
{*UAB, Svetainė*}	0.04
{*UAB, Tvarkymo*}	0.06
{*UAB, Modulis*}	0.02
{*UAB, Apdorojimo*}	0.03
{*Programa, Tvarkymo*}	0.01
{*Programa, Apdorojimo*}	0.02
{*Sistema, Tvarkymo*}	0.01
{*Sistema, Apdorojimo*}	0.01
{*Tvarkymo, Modulis*}	0.01
{*Apdorojimo, Modulis*}	0.01

Table 2. Association rules.

Sub-sequence	**Support**	**Confidence, %**
UAB $\Rightarrow$ *Programa*	0.11	39.41
UAB $\Rightarrow$ *Sistema*	0.08	18.53
UAB $\Rightarrow$ *Svetainė*	0.04	5.88
UAB $\Rightarrow$ *Tvarkymo*	0.06	5.29
UAB $\Rightarrow$ *Modulis*	0.02	2.35
UAB $\Rightarrow$ *Apdorojimo*	0.03	3.24
Programa $\Rightarrow$ *Tvarkymo*	0.01	3.39
Sistema $\Rightarrow$ *Tvarkymo*	0.01	5.21
Sistema $\Rightarrow$ *Apdorojimo*	0.01	4.35
Tvarkymo $\Rightarrow$ *Modulis*	0.01	6.67
Apdorojimo $\Rightarrow$ *Modulis*	0.01	11.25

Final projects of Computer Programming study programme have practical value, as they were intended for particular companies. In general, programs or systems are designed for a particular customer(s), but a large part of the final projects includes website design. Less often module or game development are chosen.

In the period 2000 - 2013 among all final projects 52.23% were different applications (*Programa)*, 22.04% different systems (*Sistema)*, 13.69% Websites (*Svetainė)*, 6.02 % different models (*Modulis)* and 2.62% games (*Žaidimas)*, 3.4 % were other topics.

The research of the database of transactions

The database of transactions, which consists of 400,000 transactions has been analysed. All analysed items in the database are homogenous. The market basket contains items with the same attributes, except the title. The database contains 25 different title items.

The database of transactions has been processed by the stochastic algorithm for mining frequent sub-sequences, algorithms SDPA1, SDPA2, SPADE and probabilistic algorithm ProMFS for mining frequent sequences. During the experiment the value of minimum support has been chosen: *min_supp* = 0.02; 0.04; 0.06; 0.08; 0.1; 0.2; 0.3; 0.4; 0.5 and the length of the sub-sequence is in the interval [1; 10], i.e. the determined market basket is not greater than 10 items. The chosen value of the parameter *g* of the algorithm SDPA1 is 0.5, and in the SDPA2 algorithm one-item frequent sub-sequences have been determined by GSP algorithm. The length of the model sequence of the algorithm ProMFS is 1,342 items, in order to process the sequence GSP algorithm and *min_supp* = 4; 6; 8; 10; 12; 14; 16; 18; 20 have been applied. Time consumption of the algorithms, change of the number of defined frequent sub-sequences depending on defined minimum support has been compared. The association rules of the database of transactions have been discovered by the stochastic algorithm for discovery of association rules. The comparison of time consumption of the algorithms depending on defined minimum support is presented in Figure 5.

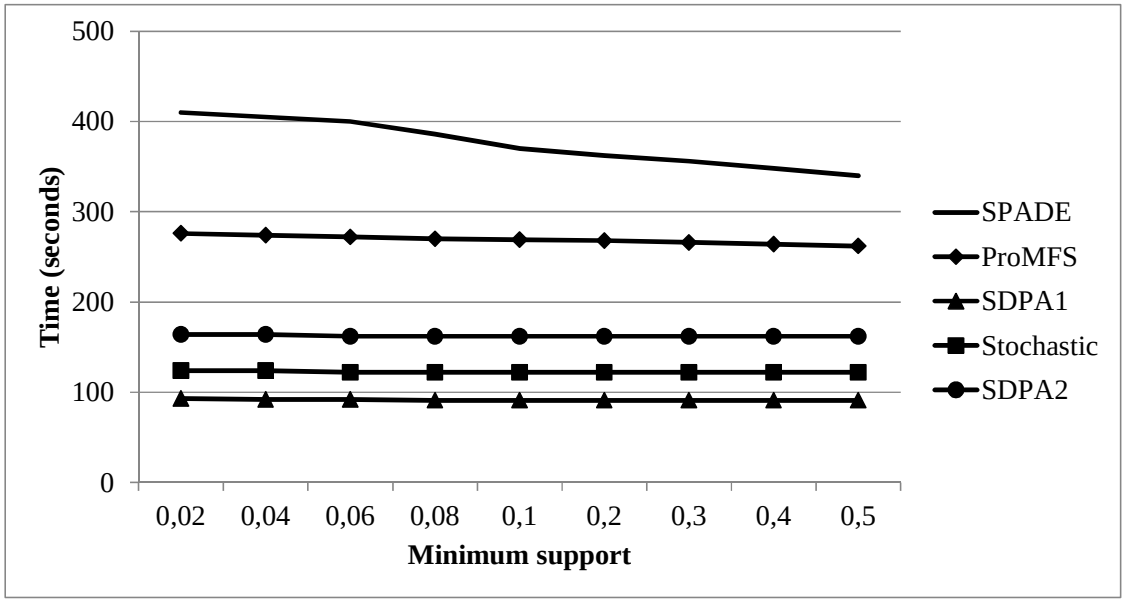

Fig. 5. The comparison of time consumption of the algorithms.

The comparison of the number of frequent sub-sequences of the algorithms depending on defined minimum support is presented in Figure 6.

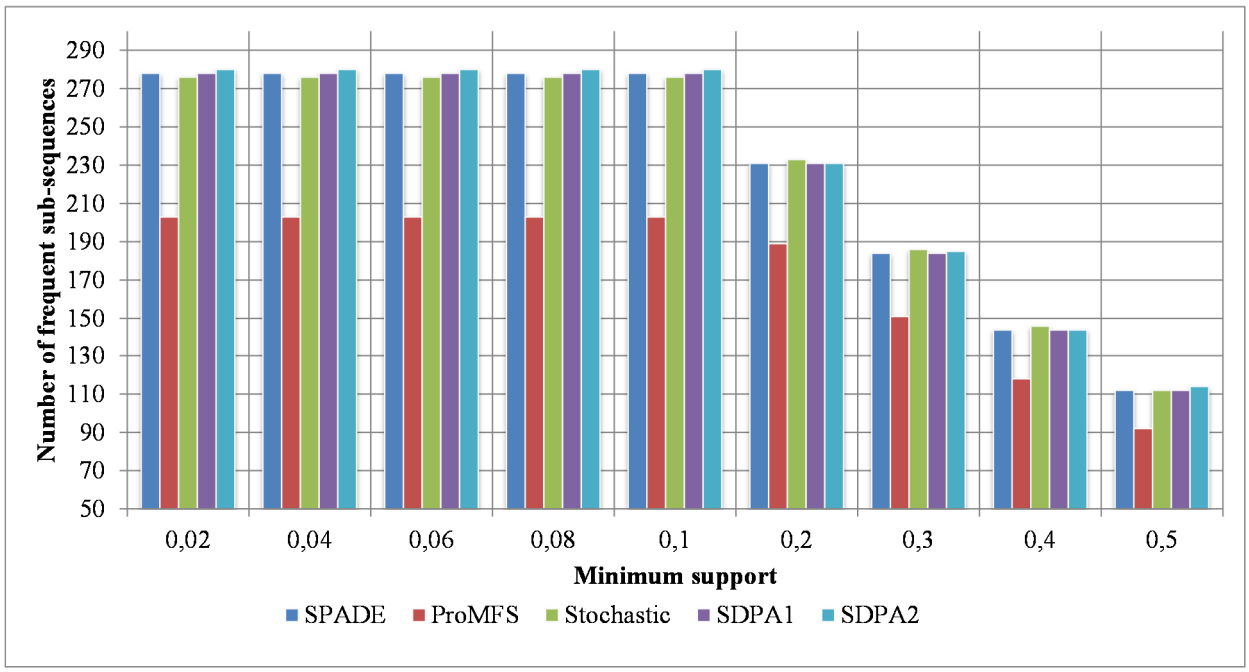

Fig. 6. The comparison of the number of frequent sub-sequences of the algorithms.

The number of frequent sub-sequences starts to decrease when *min_supp* = 0.1; 0.2; 0.3; 0.4; 0.5.

The comparison of the number of particular length frequent sub-sequences of the algorithms depending on defined minimum support is presented in Figures 7 – 10.

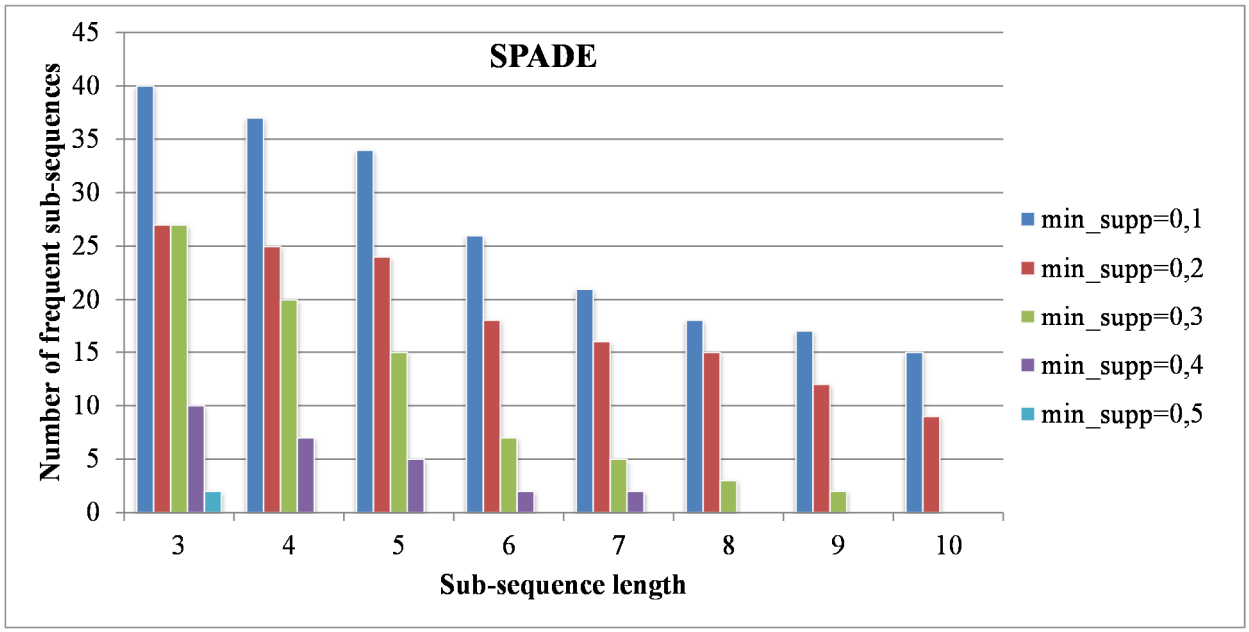

Fig. 7. The comparison of the number of frequent sub-sequences of the algorithm SPADE depending on the sub-sequence length.

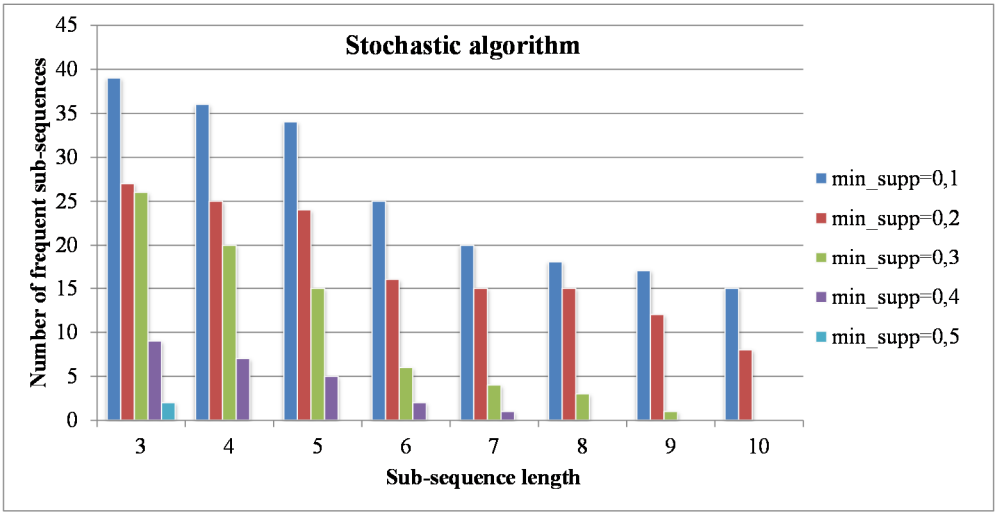

Fig. 8. The comparison of the number of frequent sub-sequences of the stochastic algorithm depending on the sub-sequence length.

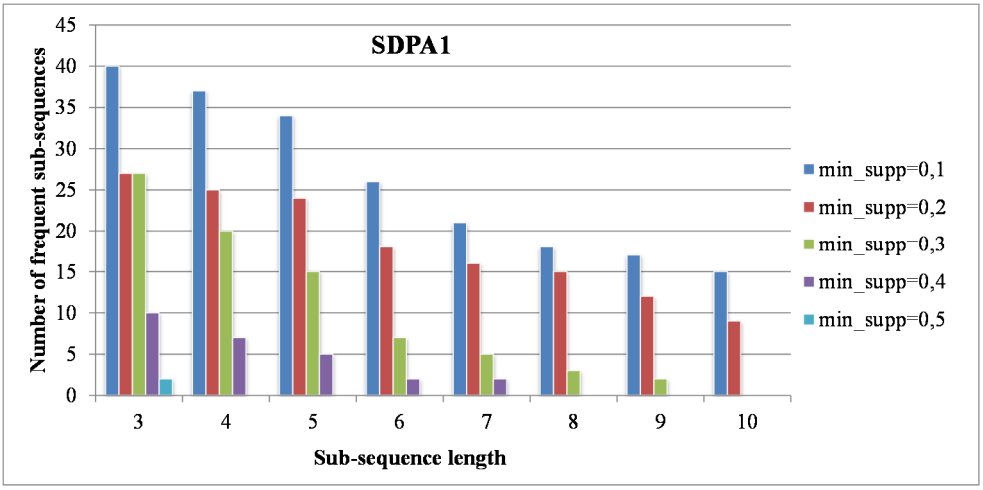

Fig. 9. The comparison of the number of frequent sub-sequences of the algorithm SDPA1 depending on the sub-sequence length.

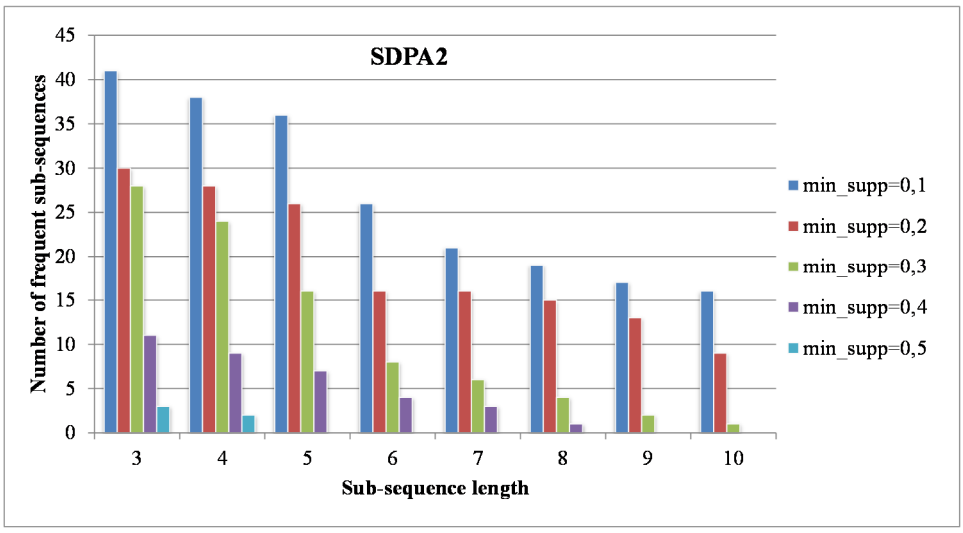

Fig. 10. The comparison of the number of frequent sub-sequences of the algorithm SDPA2 depending on the sub-sequence length.

Greater the length of the frequent sub-sequence, there are less different frequent sub-sequences. The results show that the number of 3-item, 4-item and 5-item different frequent sub-sequences slightly decrease, but the number of 5-item and 6-item different frequent sub-sequences obviously decrease.

To determine the maximum length of frequent sub-sequences, when the number of such sub-sequence is the highest, estimation of the values of probability characteristics *u*, *z* and probability *P*(0) can be used.

During the experiment the values of probability characteristics *u*, *z* and probability P(0) of the stochastic algorithm for mining frequent sub-sequences, SDPA1 and SDPA2 algorithms have been estimated. Evaluation of the values of these characteristics allows us to distinguish what the maximum length of frequent sub-sequences is when the number of such sub-sequences is the highest. Two independent samples of sub-sequences are analysed, their sizes are n_1 and n_2, and the most frequent sub-sequence occurred k_1 times in the first sample and k_2 times in the second sample. Test statistics *u* defines matching of the support of two adjacent length sub-sequences. Change of the test statistics *u*, depending on the length of the sub-sequence is presented in Figure 11. The length of the sub-sequence changes from 3 to 10 items.

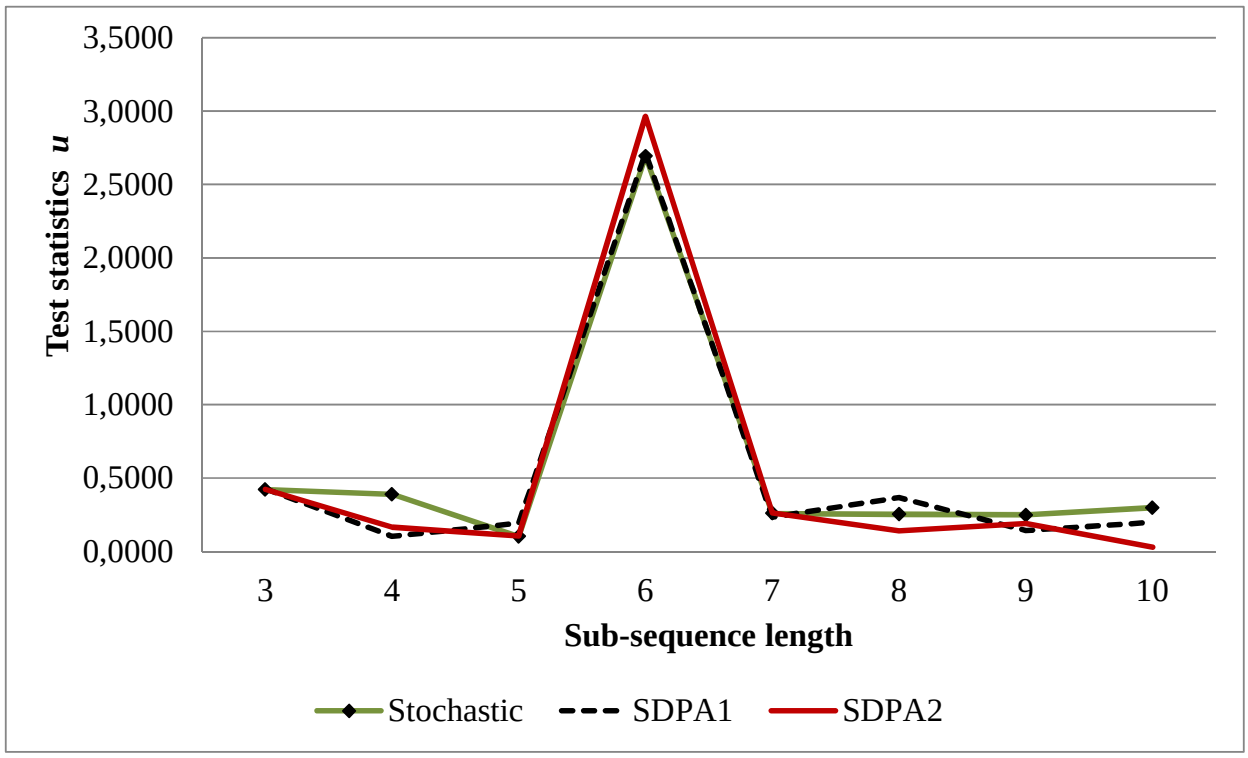

Fig. 11. Change of the test statistics *u*.

Evaluating test statistics *u* of matching of the support of two adjacent length sub-sequences using algorithm for mining frequent sub-sequences, algorithms SDPA1 and SDPA2 it has been estimated that the most frequent is 5-item sub-sequence, i.e. the most frequent market basket consists of 5 items, because when the length of the sub-sequence equals to 6, the statistical value *u* increases dramatically, which means that the support of 6-item sub-sequence is much lower than the support of 5-item sub-sequence.

Test statistics *z* also determines matching of the support of two adjacent length sub-sequences. Change of the test statistics *z*, depending on the length of the sub-sequence, when the length changes from 3 to 10 items is presented in Figure 12.

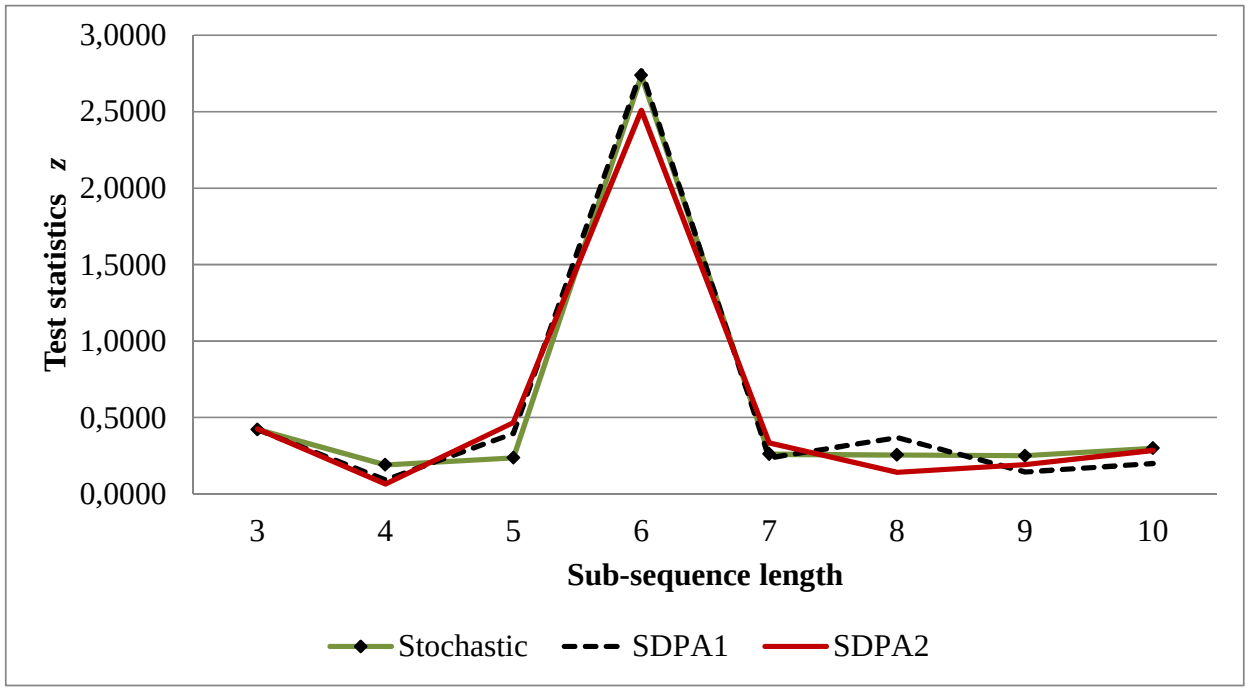

Fig. 12. Change of the test statistics *z*.

Evaluating test statistics *z* of matching of the support of two adjacent length sub-sequences using algorithm for mining frequent sub-sequences, algorithms SDPA1 and SDPA2 it has been estimated that the most frequent is 5-item sub-sequence, i.e. the most frequent market basket consists of 5 items, because when the length of the sub-sequence equals to 6, the statistical value *z* increases dramatically, which means that the support of 6-item sub-sequence is much lower than the support of 5-item sub-sequence.

The change of the values of probability function $P(0)$, when the length of sub-sequence changes from 1 to 10 items is presented in Figure 13.

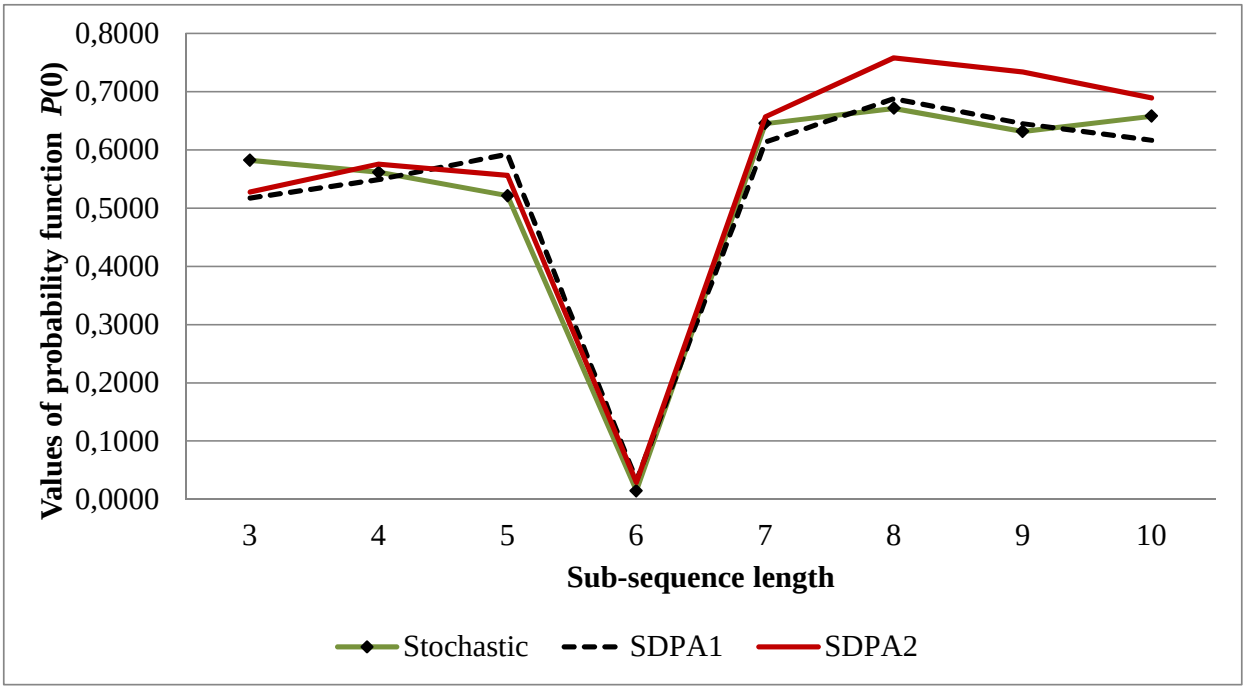

Fig. 13. The change of the values of probability function $P(0)$.

Evaluating the values of probability function $P(0)$.using algorithm for mining frequent sub-sequences, SDPA1 and SDPA2 algorithms, it has beens estimated that the most frequent is 5-item sub-sequence, i.e. the most frequent market basket consists of 5 items, because when the length of the sub-sequence equals to 6, the probability function $P(0)$ decreases dramatically, which means that the support of 6-item sub-sequence is much lower than the support of 5-item sub-sequence.

The database of transactions has been processed by the stochastic algorithm for discovery of association rules. Different values of support and confidence of association rules have been chosen to discover association rules and general association rules by the stochastic algorithm. General association rules have been created using two groups of items. The number of association rules estimated by the stochastic algorithm for discovery of association rules, when values of the support and confidence are different is given in Table 3.

In the analysed database of transactions, when the value of support is *min_supp*>40% association rules cannot be created. When the value of support is *min_supp*>25% association rules with 5 items cannot be created.

The stochastic algorithm for discovery of association rules approximately creates 19 association rules per second. The number of association rules and time are given in table 4.

Table 3. The number of association rules.

Support of the rule, %	Confidence of the rule, %									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
10	107236	107236	103780	93435	87363	82099	63853	44827	44663	44662
20	3056	3056	3056	2961	2739	2258	1905	1401	1237	1236
25	484	484	484	484	452	370	288	157	134	133
30	70	70	70	70	70	68	46	27	23	22
40	10	10	10	10	10	10	10	7	3	2
50	0	0	0	0	0	0	0	0	0	0

Table 4. The number of association rules and time.

Support and confidence of the rule, %	Number of rules	Time (seconds)	Number of generated rules per second
20; 100	1236	66.08	19
25; 10	484	27.67	17
25; 20	484	27.69	17
25; 30	484	27.69	17
25; 40	484	27.68	17
25; 50	452	25.68	18
25; 60	370	20.67	18
25; 70	288	20.66	14
25; 80	157	10.65	15
25; 90	134	9.65	14
25; 100	133	8.67	15
30; 10	70	3.27	21
30; 20	70	3.28	21
30; 30	70	3.27	21
30; 40	70	3.26	21
30; 50	70	3.19	22
30; 60	68	2.47	28
30; 70	46	2.17	21
30; 80	27	1.16	23
30; 90	23	1.16	20
30; 100	22	1.16	19
40; 10	10	0.54	19
40; 20	10	0.54	19
40; 30	10	0.54	19
40; 40	10	0.54	19
40; 50	10	0.54	19
40; 60	10	0.52	19
40; 70	10	0.52	19
40; 80	7	0.46	15
40; 90	3	0.22	14
40; 100	2	0.12	17

General association rules determined by the stochastic algorithm for discovery of association rules are given in Table 5.

In the analysed database of transactions, when the value of support is greater than 50%, general association rules, which satisfy the value, cannot be created.

Table 5. The number of general association rules.

Support of the rule, %	**Confidence of the rule, %**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
10	1794	1703	1537	1212	1023	863	646	393	382	382
20	170	170	159	137	122	98	74	38	27	27
25	86	86	86	73	64	55	43	24	17	17
30	20	20	20	16	16	12	10	9	8	8
40	6	6	6	6	6	4	3	3	2	2
50	2	2	2	2	2	2	2	2	1	1
60	0	0	0	0	0	0	0	0	0	0

The results of the research show, that it is important to properly choose the values of support and confidence of association rules. If these values are very low, then a large number of association rules is created. If great values of support and confidence are determined, then a small number of association rules will be created.

General Conclusions

To discover frequent sub-sequences and association rules in large databases approximate stochastic algorithm for mining frequent sub-sequences, its modifications SDPA1 and SDPA2, and stochastic algorithm for discovery of association rules have been designed. The main strategy of the new proposed stochastic algorithms is scanning the database once to select random length sub-sequences, when random amount of symbols is omitted. The lengths of selected sub-sequences and the number of omitted symbols are distributed according to the uniform distribution. The new proposed algorithms allow us to combine two important tests, i.e. accuracy and time. Real and simulated databases have been used for experimental research of these algorithms. The results of the experiments have been compared with other existing exact and approximate algorithms. The research carried in the dissertation draws the following conclusions:

1. The results of the comparison of exact SPADE, GSP, Apriori algorithms and recursive algorithm for mining frequent sub-sequences show that SPADE algorithm operates fastest and time consumption is the greatest of recursive algorithm for mining frequent sub-sequences.
2. The accuracy of new proposed stochastic algorithm for mining frequent sub-sequences, SDPA1 and SPDA2 algorithms have been researched by empirical experiments. The average first type error of the algorithms is about 2.4 %, the average second type error is about 5.6 %, confidence interval is [0.95; 0.99].
3. The rate of designed stochastic algorithm for mining frequent sub-sequences, SDPA1 and SPDA2 algorithms is not influenced by different number of database items, because when scanning the database random length sub-sequences are selected.
4. The new proposed stochastic algorithm for mining frequent sub-sequences, SDPA1 and SPDA2 algorithms determine same frequent sub-sequences as exact algorithms.
5. The fastest from all proposed algorithms is SDPA1 algorithm, when the value of the parameter of the algorithm $g \in [0; 0.6]$. When the value $g \in (0.6; 1]$, then processing time of both: algorithm SDPA1 and stochastic algorithm for mining frequent sub-sequences is approximately equal. Time consumption of algorithm SDPA2 is slightly greater than algorithm SDPA1 and stochastic algorithm for mining frequent sub-sequences, but the support of frequent sub-sequences is greater.
6. The stochastic algorithm for discovery of association rules approximately creates 19 association rules per second.
7. New proposed stochastic algorithm for mining frequent sub-sequences, SDPA1 and SPDA2 algorithms, and stochastic algorithm for discovery of association rules are fairly accurate and fast.

List of Publications on Topic of Dissertation

- Savulionienė, L. Dažnų posekių paieškos algoritmai ir jų rezultatai. *IV respublikinės mokslinės – praktinės konferencijos mokslinių straipsnių rinkinys*. ISSN 2029-2279, 2011, p. 107-113.

- Savulionienė, L., Sakalauskas, L. Statistinis dažnų posekių paieškos algoritmas. *Informacijos mokslai*, Vol. 58. ISSN 1392-0561, 2011, p.126-143.
- Savulionienė, L., Sakalauskas, L. Stochastinis dažnų posekių paieškos algoritmas. *Jaunųjų mokslininkų darbai*, Vol. 4(33). ISSN 1648-8776, 2011, p. 138-145.
- Savulionienė, L. Stochastinis modifikuotas dažnų posekių paieškos algoritmas. *VI respublikinės mokslinės – praktinės konferencijos mokslinių straipsnių rinkinys*. ISSN 2029-2279, 2013, p. 80-91.
- Savulionienė, L., Sakalauskas, L. Modifikuoto stochastinio dažnų posekių paieškos algoritmo tikimybinės charakteristikos. *XVI Kompiuterininkų konferencijos mokslo darbai*. ISBN 978-9986-34-293-9, 2013, p. 75-87.
- Savulionienė, L., Sakalauskas, L. Modified Stochastic Algorithm for Mining Frequent Subsequences. *Information and Software Technologies*, Vol. 403. ISBN: 978-3-642-41946-1 (Print), 978-3-642-41947-8 (Online), 2013, p. 222-235.
- Savulionienė, L., Sakalauskas, L. Stochastic frequent set search algorithm for association rules discovery. *Information technology and control*. ISSN: 1392-124X (Print); ISSN: 2335-884X (Online). *Accepted*.

About the Author

Loreta Savulionienė was born on the 30th of December in 1971, Vilkaviškis, Lithuania. In 1990, she graduated from Šakiai Zigmas Angarietis secondary school awarded silver medal. She got bachelor's degree in mathematics in Vilnius University in 1996, and master's degree in mathematics in Vilnius University in 1998. From 2008 till 2013 she was a PHD student of Vilnius University, Institute of Mathematics and Informatics. From 1998 till 2004 she worked as a lecturer in Vilniaus kolegija/ University of Applied Sciences. From 2004 till 2013 she worked as the head of Study Organization department and a lecturer in Vilniaus kolegija/ University of Applied Sciences. From 2013 she has been working as the vice dean of the Faculty of Electronics and Informatics and a lecturer in Vilniaus kolegija/ University of Applied Sciences.

In 2013 she was awarded UAB "Visma" grant for best Lithuanian programming lecturers.

SUSIETUMO TAISYKLIŲ PAIEŠKA DIDELĖSE DUOMENŲ BAZĖSE

Tyrimo sritis ir problemos aktualumas

Bet kokios įmonės veikla šiandien susijusi su dideliais informacijos ir duomenų kiekiais. Tarp didelių informacijos kiekių slepiasi ir svarbi, ir niekinė informacija. Efektyvus informacijos, slypinčios duomenyse, atskleidimas ir panaudojimas yra svarbiausias konkurencingumo didinimo veiksnys šiuolaikinėje dinamiškoje tyrimų ir verslo aplinkoje. Šioms problemoms spręsti taikoma duomenų tyryba.

Dažnų posekių paieškos ir susietumo taisyklių nustatymo uždavinių pagrindas yra šablonų, atvaizduojančių duomenų tarpusavio sąryšius, koncepcija. Šie šablonai atskleidžia vidinę duomenų struktūrą bei dėsningumus, būdingus duomenų poaibiams, kurie išreiškiami vartotojui suprantamu pavidalu – susietumo taisyklėmis.

Susietumo taisyklių paieška taikoma versle, finansinėse institucijose, medicinoje, nuotoliniame mokyme ir kitose srityse, kur tenka apdoroti didelius informacijos kiekius bei aptikti sąryšius tarp duomenų.

Nustatyti sąryšiai tarp duomenų padeda analitikams greičiau bei tiksliau priimti sprendimus, todėl susietumo taisyklių paieška yra svarbus uždavinys.

Disertacijoje yra pasiūlytas naujas apytikslis dažnų posekių ir susietumo taisyklių paieškos algoritmas, jo modifikacijos bei pateiktas algoritmo paklaidų įvertinimas. Algoritmų rezultatai buvo palyginti su kitais tiksliaisiais bei apytiksliais algoritmais.

Tyrimų objektas

Disertacijos tyrimo objektas yra duomenų tyrybos algoritmai ir metodai, skirti dažnų posekių ir susietumo taisyklių nustatymo uždaviniams spręsti. Disertacijoje aprašytuose tyrimuose naudojami imituotos ir realios duomenų bazės.

Darbo tikslas ir uždaviniai

Disertacijos tikslas yra sudaryti naują apytikslį dažnų posekių paieškos bei susietumo taisyklių nustatymo algoritmą ir jo modifikacijas bei pateikti algoritmo paklaidų įvertinimą.

Siekiant įgyvendinti užsibrėžtą tikslą sprendžiami šie uždaviniai:

- Išnagrinėti dažniausiai naudojamus duomenų tyrybos metodus ir algoritmus, skirtus dažnų posekių bei susietumo taisyklių nustatymui.
- Sukurti naują apytikslį dažnų posekių paieškos algoritmą.
- Įvertinti naujo algoritmo tikslumą, greitį bei statistines charakteristikas.
- Palyginti sudarytą apytikslį algoritmą su Apriori, GSP, SPADE, rekursiniu ir tikimybiniu dažnų sekų nustatymo ProMFS algoritmu.
- Realizuoti sukurto algoritmo modifikacijas algoritmo tikslumui bei greičiui padidinti.
- Realizuoti sukurto algoritmo modifikaciją susietumo taisyklėms nustatyti.
- Sukurti programinę įrangą eksperimentams atlikti.
- Atlikti eksperimentus su imituotais ir realiais duomenimis bei palyginti su kitais tiksliaisiais ir apytiksliais algoritmais.

Mokslinis darbo naujumas

Šioje disertacijoje yra nagrinėjamas aktualus susietumo taisyklių paieškos uždavinys, kurio sprendimui pasiūlytas naujas stochastinis dažnų posekių paieškos algoritmas bei jo modifikacijos susietumo taisyklių paieškai, įvertintos šių algoritmų tikimybinės charakteristikos. Šie dažnų posekių ir susietumo taisyklių paieškos algoritmai yra apytiksliai, kurie vieno duomenų bazės skenavimo metu nustato dažnus posekius bei susietumo taisykles. Naujai pasiūlytų algoritmų paklaidų įvertinimas atliktas naudojantis statistiniais metodais. Šių algoritmų veikimas greitesnis lyginant su tiksliaisiais ir su nagrinėtais apytiksliais dažnų posekių paieškos algoritmais.

Tyrimo metodika

Pagrindiniai tyrimo metodai taikomi disertacijoje – informacijos paieška, duomenų imitavimas, informacijos sisteminimas, analizė, lyginamoji analizė, apibendrinimas, statistinė analizė, žvalgomasis tyrimas, eksperimentinis tyrimas. Analizuojant kitų autorių mokslinius ir eksperimentinius pasiekimus dažnų posekių ir susietumo taisyklių paieškos srityje, buvo naudoti informacijos paieškos, duomenų imitavimo, sisteminimo, analizės, lyginamosios analizės, žvalgomojo tyrimo ir

apibendrinimo metodai. Sudarytiems algoritmams įvertinti buvo naudotas eksperimentinio tyrimo metodas ir statistinė analizė.

Darbo praktinė reikšmė

Darbe sukurtas stochastinis dažnų posekių paieškos algoritmas bei jo modifikacijos SDPA1, SDPA2, stochastinis susietumo taisyklių paieškos algoritmas. Sukurta programinė įranga, kurioje realizuoti Apriori, GSP, SPADE, rekursinis, ProMFS, stochastinis dažnų posekių paieškos, SDPA1, SPDA2 bei stochastinis susietumo taisyklių paieškos algoritmai. Programinė įranga buvo naudojama eksperimentams atlikti ir yra parengta realiam naudojimui.

Ginamieji teiginiai

- Darbe sukurti stochastinis dažnų posekių paieškos, SDPA1, SDPA2 ir stochastinio susietumo taisyklių paieškos algoritmai yra apytiksliai, tačiau yra pakankamai tikslūs ir greiti.
- SDPA1 yra stochastinio dažnų posekių paieškos algoritmo modifikacija, kuri padidina algoritmo tikslumą, kai pasirenkamas parametras $g \in [0{,}6;\ 1]$.
- SDPA2 algoritmas yra stochastinio dažnų posekių paieškos algoritmo modifikacija, kuri naudoja vieno elemento dažnus posekius, nustatytus pasirinktu tiksliuoju dažnų posekių paieškos algoritmu. SPDA2 tikslesnis už stochastinį dažnų posekių paieškos ir SPDA1 algoritmus.

Darbo rezultatų aprobavimas

Pagrindiniai tyrimo rezultatai atspausdinti 7 mokslinėse publikacijose, rezultatai pristatyti 2 tarptautinėse mokslininkų konferencijose ir 5 respublikinėse konferencijose.

Pranešimai skaityti šiose konferencijose:

- Respublikinė konferencija: LOTD 3-oji jaunųjų mokslininkų konferencija: Operacijų tyrimai verslui ir socialiniams procesams, Lietuva, Vilnius, 2010 m. spalio 1 d.

- Respublikinė konferencija: IV respublikinė mokslinė – praktinė konferencija: Mokslo taikomųjų tyrimų įtaka šiuolaikinių studijų kokybei, Lietuva, Vilnius, 2011 m. gegužės 5 d.
- Respublikinė konferencija: LOTD 4-oji jaunųjų mokslininkų konferencija: Operacijų tyrimai versle, inžinerijoje ir informacinėse technologijose, Lietuva, Kaunas, 2011 m. rugsėjo 30 d.
- Konferencija: Kompiuterininkų dienos – 2013, Lietuva, Šiauliai, 2013 m., rugsėjo 19-21 d.
- Respublikinė konferencija: VI respublikinė mokslinė – praktinė konferencija: Mokslo taikomųjų tyrimų įtaka šiuolaikinių studijų kokybei, Lietuva, Vilnius, 2013 m. gegužės 6 d.
- Tarptautinė konferencija: Distributed Systems and Big Data – Towards New Horizons, NORDUGRID 2013, Lietuva, Šiauliai, 2013 m. birželio 4-6 d.
- Tarptautinė konferencija: The 19th International Conference on Information and Software Technologies (ICIST 2013), Lietuva, Kaunas, spalio 10-11 d., 2013.

Disertacijos autorės publikacijų sąrašas:

- Savulionienė, L. Dažnų posekių paieškos algoritmai ir jų rezultatai. *IV respublikinės mokslinės – praktinės konferencijos mokslinių straipsnių rinkinys*. ISSN 2029-2279, 2011, p. 107-113.
- Savulionienė, L., Sakalauskas, L. Statistinis dažnų posekių paieškos algoritmas. *Informacijos mokslai*, Vol. 58. ISSN 1392-0561, 2011, p.126-143.
- Savulionienė, L., Sakalauskas, L. Stochastinis dažnų posekių paieškos algoritmas. *Jaunųjų mokslininkų darbai*, Vol. 4(33). ISSN 1648-8776, 2011, p. 138-145.
- Savulionienė, L. Stochastinis modifikuotas dažnų posekių paieškos algoritmas. *VI respublikinės mokslinės – praktinės konferencijos mokslinių straipsnių rinkinys*. ISSN 2029-2279, 2013, p. 80-91.
- Savulionienė, L., Sakalauskas, L. Modifikuoto stochastinio dažnų posekių paieškos algoritmo tikimybinės charakteristikos. *XVI Kompiuterininkų konferencijos mokslo darbai*. ISBN 978-9986-34-293-9, 2013, p. 75-87.

- Savulionienė, L., Sakalauskas, L. Modified Stochastic Algorithm for Mining Frequent Subsequences. *Information and Software Technologies*, Vol. 403. ISBN: 978-3-642-41946-1 (Print), 978-3-642-41947-8 (Online), 2013, p. 222-235.
- Savulionienė, L., Sakalauskas, L. Stochastic frequent set search algorithm for association rules discovery. *Information technology and control*. ISSN: 1392-124X (Print); ISSN: 2335-884X (Online). Priimtas (bus atspausdintas 2014 m. birželio mėn., T.43 Nr.2)

Disertacijos struktūra

Disertaciją sudaro 5 skyriai, literatūros sąrašas. Disertacijos skyriai: įvadas, susietumo taisyklių nustatymas, dažnų posekių paieškos algoritmai, stochastiniai dažnų posekių paieškos algoritmai, tyrimo rezultatai. Disertacijos apimtis 125 puslapiai, 10 lentelių, 21 paveikslas, 4 priedai. Disertacijoje remtasi 108 šaltiniais.

Bendrosios išvados

Dažnų posekių ir susietumo taisyklių paieškai didelėse duomenų bazėse buvo sukurtas apytikslis stochastinis dažnų posekių paieškos algoritmas bei šio algoritmo modifikacijos SDPA1, SDPA2 ir stochastinis susietumo taisyklių paieškos algoritmai. Naujai pasiūlytų stochastinių algoritmų pagrindinė strategija – vieno duomenų bazės skenavimo metu atrinkti atsitiktinio ilgio posekius, praleidžiant atsitiktinį simbolių kiekį. Paimamų posekių ilgis ir praleidžiamų simbolių skaičius yra pasiskirstę pagal tolygųjį skirstinį. Naujai pasiūlyti algoritmai leidžia suderinti du svarbius kriterijus, t.y. tikslumą ir laiką. Šių algoritmų eksperimentinis tyrimas buvo atliktas naudojant realias bei imitacines duomenų bazes. Eksperimentų rezultatai palyginti su kitas egzistuojančiais tiksliaisiais bei apytiksliais algoritmais. Šioje disertacijoje atlikti tyrimai leido padaryti tokias išvadas:

1. Palyginus tiksliuosius SPADE, GSP, Apriori, rekursinį dažnų posekių paieškos algoritmus rezultatai parodė, kad greičiausiai veikia SPADE algoritmas, o rekursinio algoritmo laiko sąnaudos yra pačios didžiausios.
2. Naujai pasiūlytų stochastinio dažnų posekių paieškos, SDPA1, SDPA2 algoritmų tikslumas ištirtas empyriniais eksperimentais. Algoritmų vidutinė pirmos rūšies

klaida yra apie 2,4 %, vidutinė antros rūšies klaida – apie 5,6 %, pasikliovimo tikimybės intervalas yra [0,95; 0,99].

3. Darbe sukurtų stochastinio dažnų posekių paieškos, SDPA1, SPDA2 algoritmų greičiui neturi įtakos skirtingų duomenų bazės elementų skaičius, nes duomenų bazės skenavimo metu pasirenkami atsitiktinio ilgio posekiai.
4. Naujai pasiūlyti stochastinis dažnų posekių paieškos, SDPA1, SDPA2 algoritmai nustato tuos pačius dažnus posekius kaip ir tikslieji algoritmai.
5. Greičiausias iš visų pasiūlytų stochastinių algoritmų yra SDPA1 algoritmas, kai algoritmo parametro reikšmė $g \in [0; 0,6]$. Kai reikšmė $g \in (0,6; 1]$, tai algoritmų SDPA1 ir stochastinio dažnų posekių paieškos algoritmo vykdymo laikas apytiksliai vienodas. SDPA2 algoritmo laiko sąnaudos nežymiai didesnės už SDPA1 ir stochastinio dažnų posekių paieškos algoritmų, tačiau didesnis dažnų posekių dažnumas.
6. Stochastinis susietumo taisyklių paieškos algoritmas per vieną sekundę sudaro vidutiniškai 19 susietumo taisyklių.
7. Naujai pasiūlyti stochastinis dažnų posekių paieškos, SDPA1, SDPA2 ir stochastinis susietumo taisyklių paieškos algoritmai yra greiti ir pakankamai tikslūs.

Trumpai apie autorę

Loreta Savulionienė gimė 1971 m. gruodžio 30 d. Vilkaviškyje.

1990 m. sidabro medaliu baigė Šakių Zigmo Angariečio vidurinę mokyklą. 1996 m. Vilniaus universiteto Matematikos fakultete įgijo matematikos bakalauro kvalifikacinį laipsnį. 1998 m. Vilniaus universiteto Matematikos fakultete įgijo matematikos magistro kvalifikacinį laipsnį. 2008–2013 m. doktorantė Vilniaus universiteto Matematikos ir informatikos institute. 1998–2004 m. dirbo Vilniaus kolegijoje lektore. 2004–2013 m. dirbo Vilniaus kolegijoje studijų organizavimo skyriaus vedėja ir lektore. Nuo 2013 dirba Vilniaus kolegijos Elektronikos ir informatikos fakulteto prodekane ir lektore.

2013 m. laimėjo UAB „Visma“ įsteigtą premiją geriausiems Lietuvos programavimo dėstytojams.