Computer Modelling and New Technologies, 2006, Vol.10, No.4, 46-52
Transport and Telecommunication Institute, Lomonosova 1, Riga, LV-1019, Latvia

NEW STATISTICAL CHARACTERISTICS FOR MINING FREQUENT SEQUENCES IN LARGE DATABASES

R. TUMASONIS[1], R. RASTENIS[2]

[1] Institute of Mathematics and Informatics
Akademijos 4, Vilnius, LT-08663, Lithuania
E-mail: romanas@viko.lt, phone: (+370)68656105

[2] Vilnius Business College
Filaretu 36 a, Vilnius, LT-01210, Lithuania
E-mail: r.rastenis@kolegija.lt, phone: (+370)52154884

The paper deals with the search and analysis of the subsequence in large volume sequences (texts, DNA sequences, etc.). A new algorithm ProMFS for mining frequent sequences is proposed and investigated. It is based on the estimated probabilistic-statistical characteristics of the appearance of elements of the sequence and their order. The algorithm builds a new much shorter sequence and makes decisions on the main sequence in accordance with the results of analysis of the shorter one.

Keywords: *Data Mining, frequent sequences*

1. Introduction

Generally, data mining (sometimes called data or knowledge discovery) is the process of analysing data from different perspectives and summarizing it into useful information – information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analysing data. It allows users to analyse data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

Although data mining is a relatively new term, the technology is not. Companies have used powerful computers to sift through volumes of supermarket scanner data and analyse market research reports for years. However, continuous innovations in computer processing power, disk storage, and statistical software are dramatically increasing the accuracy of analysis while driving down the cost.

Data are any facts, numbers, or text that can be processed by a computer. Today, organizations are accumulating vast and growing amounts of data in different formats and different databases. This includes:

1) operational or transactional data such as, sales, cost, inventory, payroll, and accounting;
2) non-operational data;
3) meta data.

The patterns, associations, or relationships among all this *data* can provide *information*. Information can be converted into *knowledge* about historical patterns and future trends.

Dramatic advances in data capture, processing power, data transmission and storage capabilities are enabling organizations to integrate their various databases into *data warehouses*. Data warehousing is defined as a process of centralized data management and retrieval. Data warehousing, like data mining, is a relatively new term although the concept itself has been around for years. Data warehousing represents an ideal vision of maintaining a central repository of all organizational data. Centralization of data is needed to maximize user access and analysis. Dramatic technological advances are making this vision a reality for many companies. And, equally dramatic advances in data analysis software are allowing users to access this data freely. The data analysis software is what supports data mining.

Companies primarily use data mining today with a strong consumer focus such as retail, financial, communication and marketing organizations. It enables these companies to determine relationships among "internal" factors such as price, product positioning or staff skills, and "external" factors such as economic indicators, competition, and customer demographics. Moreover, it enables them to determine the impact on sales, customer satisfaction and corporate profits. Finally, it enables them to "drill down" into summary information to view detail transactional data.

Data mining consists of five major stages:
1. Extract, transform and load transaction the data onto the data warehouse system.
2. Store and manage the data in a multidimensional database system.
3. Provide data access to business analysts and information technology professionals.
4. Analyse the data by application software.
5. Present the data in a useful format, such as a graph or table.

Different levels of analysis are available:
1. Artificial neural networks;
2. Genetic algorithms;
3. Decision trees;
4. Nearest neighbour method;
5. Rule induction;
6. Data visualization.

2. Sequential pattern mining

Sequential pattern mining [3-6], which finds the set of frequent subsequence 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 of 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 the patterns are extracted from natural language text rather than from structured databases of facts. Databases are designed for programs to process automatically; a text is written for people to read. Text mining methods can be used in bio-informatics for analysis of DNA sequences. A DNA sequence (sometimes 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 typically these are printed abutting one another without gaps, as in the sequence AAAGTCTGAC (see [2] for details).

The problem of mining sequential patterns is formulated, e.g., in [3, 4]. Assume we have a set $L=\{i_1, i_2, \ldots, i_m\}$ consisting of m distinct elements, also called *items*. An *item set* is a nonempty unordered collection of items. A *sequence* is an ordered list of item-sets. A *sequence* α is denoted as $(\alpha_1 \rightarrow \alpha_2 \rightarrow \ldots \rightarrow \alpha_q)$, where the sequence *element* α_j is an *item set*. An item can occur only once in an item set, but it can occur multiple times in different item sets of a sequence [4]. We solve a partial problem, where *item set* consists of one item only. A sequence $\alpha=(\alpha_1 \rightarrow \alpha_2 \rightarrow \ldots \rightarrow \alpha_n)$ is a *subsequence* of another sequence $\beta=(\beta_1 \rightarrow \beta_2 \rightarrow \ldots \rightarrow \beta_m)$, if there exist such numbers $t_1, t_2, \ldots, t_n$, where $t_{j+1} = t_j + 1,\ j = 1, \ldots, n$ and $\alpha_j = \beta_{t_j}$ for all a_j. Here β_{t_j} are elements of the set L. We analyse a sequence (the main sequence) S that is formed from single elements of L (not their sets, as is used in the classical formulation [3, 4] 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 subsequence in S. The problem is to find subsequence whose appearance frequency is more than some threshold called minimum support, i.e. the subsequence is frequent if it occurs in the main sequence not less than the minimum support times.

The most popular algorithm for mining frequent sequences is the GSP (Generated Sequence Pattern) algorithm. It has been examined in a lot of publications (see, e. g., [1, 3]). While searching frequent sequences in a long text, a multiple reviewing is required. The GSP algorithm minimizes the number of reviewing, but the searching time is not satisfactory for large sequence volumes. Other popular algorithms are. SPADE [3], PrefixSpan [8], FreeSpan [9] and SPAM [10].

In this paper, a new algorithm for mining frequent sequences (ProMFS) is proposed. It is based on estimated statistical characteristics of the appearance of elements of the main sequence and their order. It is an approximate method. The other method of this class is ApproxMAP [11]. The general idea of ApproxMAP is that, instead of finding exact patterns, it identifies patterns approximately shared by many sequences. The difference of our method is that we estimate the probabilistic-statistical characteristics of elements of the sequence database, generate a new much shorter sequence and make decisions on the main sequence in accordance with the results of analysis of the shorter one.

3. GSP (Generated Sequence Pattern) algorithm

Let us note that if the sequence is frequent, each its possible subsequence is also frequent. For example, if the sequence *AABA* is frequent, all its subsequence *A*, *B*, *AA*, *AB*, *BA*, *AAB*, and *ABA* are

frequent, too. Using this fact, we can draw a conclusion: if a sequence has at least one infrequent subsequence, the sequence is infrequent. Obviously, if a sequence is infrequent, all newly generated (on the second level) sequences will be infrequent, too.

At first we check the first level sequences. We have m sequences. After defining their frequencies, we start considering the second level sequences. There will be m^2 of such sequences (i_1i_1, i_1i_2, ..., i_1i_m, i_2i_1, ..., i_2i_m, ..., i_mi_1, ..., i_mi_m). However, we will not check the whole set of sequences. According to the previous level, we will only define which sequences should be checked and which not. If the second level sequence includes an infrequent sequence of the previous level (first level), then it is infrequent and we can eliminate it even without checking it up. Let us analyse an example. Suppose that some sequence is given:

$S = ABCCCBBCABCABCABCBABCCCABCAABABCABC.$ (1)

We will say that the sequence is frequent if it occurs in the text not less than 4 times, i.e. the minimum support is equal to 4. Frequent sequences will be A, B, C, AB, BC, CA, CC, ABC, BCA, CAB.

4. The probabilistic algorithm for mining frequent sequences (ProMFS)

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 following:

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

Let:

1) $P(i_j) = \dfrac{V(i_j)}{VS}$ be the probability of occurrence of element i_j in the main sequence, where $i_j \in L,\ j = 1,...,m$. Here $L=\{i_1, i_2, \ldots, i_m\}$ 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 seeking from i_j to the end of the main sequence, where $D(i_j \mid i_v)$ includes i_v. The distance between two neighbouring elements of the sequence is equal to one.

4) $\widehat{A}$ 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 $\widetilde{C}$ is generated. The length of this sequence is l. Denote its elements by c_r, $r = 1,...,l$. The model sequence $\widetilde{C}$ 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. A complementary function $\rho(c_r, a_{rj})$ is introduced. This function increases the value of characteristics $Q(i_j, c_r)$ by one. The first element c_1 of

the model sequence $\widetilde{C}$ is that from *L*, that corresponds to $\max(P(i_j))$, $i_j \in L$. According to c_1, it is activated the function $\rho(c_1, a_{1j}) \Rightarrow Q(i_j, 1+a_{1j}) = Q(i_j, 1+a_{1j}) + 1$, $j = 1,...,m$. Remaining elements c_r, $r = 2,...,l$, are chosen in the way below. Consider the *r*-th element c_r of the model sequence $\widetilde{C}$. 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 in dependence 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 $r = 2,...,l$. In such way we get the model sequence $\widetilde{C}$, which is much shorter than the main one and which may be analysed by the GSP algorithm with much less computational efforts.

Consider the previous example with the main sequence (1) given in Section 2. *L*={*A, B, C*}, i.e. *m*=3, $i_1 = A$, $i_2 = B$, $i_3 = C$. The sequence has *VS*=35 elements.

After one checking of this sequence such probabilistic characteristic are calculated:

$$P(A) = \frac{10}{35} \approx 0.2857,\ P(B) = \frac{12}{35} \approx 0.3429,\ P(C) = \frac{13}{35} \approx 0.3714,\ P(A \mid A) = 0.1,$$

$$P(A \mid B) = 0.9,\ P(A \mid C) = 0,\ P(B \mid A) \approx 0.1667,\ P(B \mid B) = 0.0833,$$

$$P(B \mid C) \approx 0.7500,\ P(C \mid A) \approx 0.4615,\ P(C \mid B) = 0.1538,\ P(C \mid C) \approx 0.3077.$$

TABLE 1. The matrix $\widehat{A}$ of average distances

	A	*B*	*C*
A	3.58	1.10	2.50
B	2.64	2.91	1.42
C	2.33	2.25	2.67

Let us compose a model sequence $\widetilde{C}$, whose length is *l*=8. At the beginning, the sequence $\widetilde{C}$ is empty, and $Q(i_j, c_r) = 0$, $r = 1,...,l$, $j = 1,...,m$:

r	1	2	3	4	5	6	7	8
A	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0
Model sequence $\widetilde{C}$	-	-	-	-	-	-	-	-

The first element of $\widetilde{C}$ is determined according to the largest probability $P(i_j)$. In our example, it is *C*, i.e. $c_1 = C$. Recalculate $Q(i_j, c_1)$, $j = 1,2,3$, according to the average distances. The situation becomes as follows:

r	1	2	3	4	5	6	7	8
A	0	0	**1**	0	0	0	0	0
B	0	0	**1**	0	0	0	0	0
C	0	0	0	**1**	0	0	0	0
Model sequence $\widetilde{C}$	***C***							

Let us choose c_2. All three values $Q(i_j, c_1)$, $j = 1,2,3$, are equal. Moreover, they are equal to zero; therefore, c_2 will be determined by maximal value of conditional probabilities. max(*P*(*C*|*A*), *P*(*C*|*B*), *P*(*C*|*C*)=*P*(*C*|*A*)= 0.4615. Therefore, $c_2 = A$. Recalculate $Q(i_j, c_2)$, $j = 1,2,3$, according to the average distances. The situation becomes as follows:

r	1	2	3	4	5	6	7	8
A	0	0	1	0	0	**1**	0	0
B	0	0	**2**	0	0	0	0	0
C	0	0	0	1	**1**	0	0	0
Model sequence $\widetilde{C}$	*C*	***A***						

Next three steps of forming the model sequence are given below:

r	1	2	3	4	5	6	7	8
A	0	0	1	0	0	**2**	0	0
B	0	0	2	0	0	1	0	0
C	0	0	0	**1**	**2**	0	0	0
Model sequence $\widetilde{C}$	*C*	*A*	***B***					

r	1	2	3	4	5	6	7	8
A	0	0	1	0	0	**3**	0	0
B	0	0	2	0	0	**2**	0	0
C	0	0	0	1	2	0	**1**	0
Model sequence $\widetilde{C}$	*C*	*A*	*B*	***C***				

r	1	2	3	4	5	6	**7**	8
A	0	0	1	0	0	3	**1**	0
B	0	0	2	0	0	2	**1**	0
C	0	0	0	1	2	0	1	**1**
Model sequence $\widetilde{C}$	*C*	*A*	*B*	*C*	***C***			

The resulting model sequence is $\widetilde{C} = CABCCABC$. The GSP algorithm has determined that the longest frequent subsequence of the model sequence is *ABC* when the minimum support is set to 2. Moreover, the GSP algorithm has determined the second frequent subsequence *CAB* of the model sequence for the same minimum support. In the main sequence (1), the frequency of *ABC* is 8 and that of *CAB* is 5. However, the subsequence *BCA*, which frequency is 5, has not been determined by the analysis of the model sequence. One of the reasons may be that the model sequence is too short.

5. Experimental results

The probabilistic mining of frequent sequences was compared with the GSP algorithm. We have generated the text file of 100000 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 (100000 symbols) was investigated with the GSP algorithm. The results are presented in Figures 1 and 2. They will be discussed more in detail together with the results of ProMFS. ProMFS generated the following model sequence $\widetilde{C}$ of length $l = 40$:

$\widetilde{C} = BBCABBCABBCABBCABBCABBCABBCABBCABBCABBCA.$

This model sequence was examined with the GSP algorithm using the following minimum support: 8, 9, 10, 11, 12, 13 and 14. The results are presented in Figures 1 and 2. Fig. 1 shows the number of frequent sequences found both by GSP and ProMFS. Fig. 2 illustrates the consumption of computing time used both by GSP and ProMFS to obtain the results of Figure 1 (the minimum support in ProMFS is *Ms*=8; the results are similar for larger *Ms*). The results in Figure 1 indicate that, if the minimum support in GSP analysing 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 till 6000, the number of frequent sequences by GSP decreases and by ProMFS increases. In the range of [2500, 6000], the number of frequent sequences found both by GSP and ProMFS is rather similar. When 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 Figure 2), 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 computing time as compared with GSP to obtain the similar result.

Conclusions

The new algorithm ProMFS for mining frequent sequences is 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 average distance between different elements of the sequence. The algorithm builds a new, much shorter model sequence, and makes decision on the main sequence in accordance with the results of analysis of the shorter one. The model sequence may be analysed by the GSP or other algorithm for mining frequent sequences: the subsequence frequency in the main sequence is estimated by the results of the model sequence analysis. The experimental investigation indicates that the new algorithm allows saving the computing time in a large extent. It is very important when analysing very large data sequences.

Moreover, the model sequence, that is much shorter than the main one, may be more understandable and easily perceived: in the experimental investigation, the sequence of 100000 elements has been modelled by a sequence of 40 elements. However, the sufficient relation between the length of the model sequence and the main sequence needs a more thorough investigation – both theoretical and experimental. In the paper, we present the experimental analysis of the proposed algorithm on the artificial data only. Further research should prove the efficiency on the real data. The research should disclose the optimal values of algorithm parameters (e.g. the length l of the model sequence $\widetilde{C}$, the minimum support for analysis of the model sequence). Another perspective research direction is the development of additional probabilistic-statistical characteristics of large sequences. This may produce the model sequence that is more adequate to the main sequence.

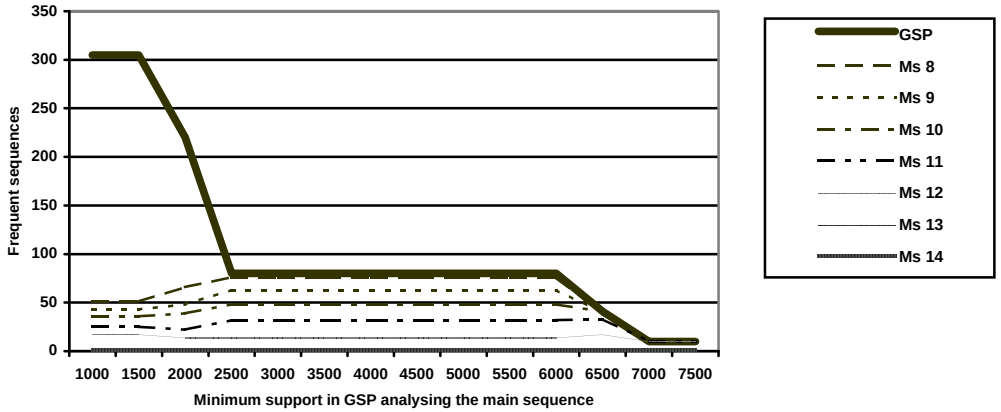

Figure 1. Number of frequent sequences found both by GSP and ProMFS (minimum support in ProMFS is *Ms* = 8,…,14)

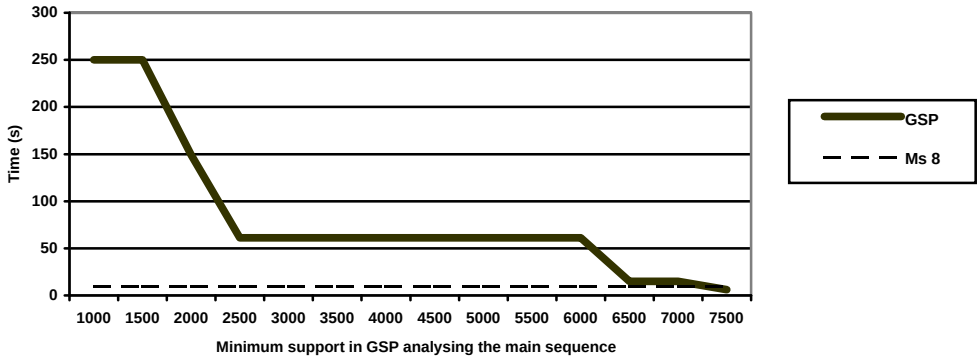

Figure 2. Computing time used both by GSP and ProMFS (minimum support in ProMFS is *Ms* = 8)

References

[1] Agrawal R. C., Agrawal C. C., Prasad V. V. (2000) Depth first generation of long patterns. In: *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Boston, Massachusetts, 108-118.

[2] http://en.wikipedia.org/wiki/DNA_sequence

[3] Zaki M. J. (2001) SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning Journal* **42** (1/2), 31-60. (D. Fisher, ed. Special issue on Unsupervised Learning)

[4] Zaki M.J. (2000) *Parallel sequence mining on shared-memory machines. In: Large-scale Parallel Data Mining. Lecture Notes in Artificial Intelligence* / M. J. Zaki, Ching-Tien Ho, eds. Vol. 1759. Berlin, Heidelberg, New York: Springer-Verlag, 161-189.

[5] Pei P. J., Han J., Wang W. (2002) Mining Sequential Patterns with Constraints in Large Databases. In: *Proceedings of the 11th ACM International Conference on Information and Knowledge Management (CIKM'02)*. McLean, VA, 18-25.

[6] Pinto P., Han J., Pei J., Wang K., Chen Q., Dayal U. (2001) Multi-Dimensional Sequential Pattern Mining. In: *Proceedings of the 10th ACM International Conference on Information and Knowledge Management (CIKM'01)*. Atlanta, Georgia, 81-88.

[7] Pei J., Han J., Mortazavi-Asl B., Pinto H., Chen Q., Dayal U., Hsu M.-C. (2001) PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: *Proc. 17th International Conference on Data Engineering ICDE.* Heidelberg, 215-224.

[8] Han J., Pei J. (2000) FreeSpan: Frequent pattern-projected sequential pattern mining. In*: Proc. Knowledge Discovery and Data Mining.* 355-359.

[9] Ayres J., Flannick. J., Gehrke J., Yiu T. (2002) Sequential pattern mining using a bitmap representation. In*: Proc. Knowledge Discovery and Data Mining.* 429-435.

[10] Kum H. C., Pei J., Wang W. (2003) ApproxMAP: Approximate Mining of Consensus Sequential Patterns. In: *Proceedings of the 2003 SIAM International Conference on Data Mining (SIAM DM '03)*. San Francisco, CA, 311-315.

Received on the 22nd of September 2006