

A probabilistic algorithm for mining frequent sequences

Romanas Tumasonis, Gintautas Dzemyda

Institute of Mathematics and Informatics, Akademijos str. 4, 08663 Vilnius, Lithuania
romanas@viko.lt, dzemyda@ktl.mii.lt

Abstract. The subject of the paper is to analyze the problem of the frequency of the subsequences in large volume sequences (texts, databases, etc.). A 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 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.

1. Introduction

Many traditional companies see the enormous opportunities in using e-commerce sites, especially e-stores, as way to reach customers outside the traditional business channels. Simply running an e-commerce site will not improve customers satisfaction and retention however. While a user-friendly e-commerce site may attract new customers and strengthen relationships with current customers [1]. Various data mining tools are implemented in such sites.

Data mining research during the last years has led to the development of a variety of algorithms for finding frequent sequential patterns in very large databases. These patterns can be used to find sequential association rules or extract prevalent patterns that exist in the sequences, and have been effectively used in many different domains and applications. A sequential pattern is a subsequence that appears frequently in a sequence database. Sequential pattern mining [2-6], which finds the set of frequent subsequences in sequence databases, is an important datamining task and has broad applications, such as business analysis, web mining, security, and bio-sequences analysis.

The problem of mining sequential patterns is formulated, e.g., in [2-4]. Assume we have a set $L=\{i_1, i_2, \dots, i_m\}$ consisting of m distinct elements. We consider the sequence (the main sequence) S that is formed from elements of the set L . In general, the number of elements in S is much larger than that in L . We have to find the most frequent subsequences in S . The problem is to find subsequences whose appearance frequency is more than some threshold called minimum support, i.e. the subsequence is frequent iff it occurs in the main sequence 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 publication (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 reviewings, but the searching time is not satisfactory for large sequence volumes. Other popular algorithms are e.g. SPADE [7], 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.

In Section 2, we present some extended details on the GSP algorithm because this algorithm is used as a constitutive part of the new algorithm. In Section 3, the new algorithm is presented. The experimental investigation results are given in Section 4.

2. 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 subsequences *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. For example, if the sequence *AA* is infrequent, all new upper level sequences *AAB* and *AAA* 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 ($i_1i_1, i_1i_2, \dots, i_1i_m, i_2i_1, \dots, i_2i_m, \dots, i_mi_1, \dots, i_mi_m$). However, we will not check whole the sequences set. 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. When we go over to the next (third) level, which will be created from the previous (second) level, we will have m^3 candidates, but again, we will check not all sequences, just sequences, which have not infrequent

subsequences. We can conclude that we will check not all the $\sum_{j=1}^m m^j$ combinations,

but only $\sum_{j=1}^m (m^j - p_{j-1})$ combinations, where p_{j-1} is the number of infrequent

subsequences from the previous level. Let us analyze an example. Suppose that some sequence is given:

$$S = ABCCCBBCABCABCABCABCCBCCABCAABABCABC \quad (1)$$

We will say that the sequence is frequent iff it occurs in the text not less than 4 times, i.e. the minimum support is equal to 4. We can see that all the sequences in the first level (see Table 1) are frequent. Now we will generate the next level according to these sequences. We have checked all the sequences in the second level (see Table 2), because all the previous the level sequences were frequent. Now we will generate next level.

Table 1. The first level

Level	Sequence	Frequency	Is it frequent?
1	<i>A</i>	10	+
1	<i>B</i>	13	+
1	<i>C</i>	13	+

Table 2. The second level

Level	Sequence	Shall we check it?	Frequency	Is it frequent?
2	<i>AA</i>	+	1	-
2	<i>AB</i>	+	9	+
3	<i>AC</i>	+	0	-
2	<i>BA</i>	+	2	-
2	<i>BB</i>	+	1	-
2	<i>BC</i>	+	9	+
2	<i>CA</i>	+	6	+
2	<i>CB</i>	+	2	-
2	<i>CC</i>	+	4	+

Table 3. The third level

Level	Sequence	Shall we check it?	Frequency	Is it frequent?
3	<i>ABA</i>	-	-	-
3	<i>ABC</i>	+	8	+
3	<i>ABB</i>	-	-	-
3	<i>BCA</i>	+	5	+
3	<i>BCB</i>	-	-	-
3	<i>BCC</i>	+	2	-
3	<i>CAB</i>	+	5	+
3	<i>CAA</i>	-	-	-
3	<i>CAC</i>	-	-	-
3	<i>CCA</i>	+	1	-
3	<i>CCB</i>	-	-	-
3	<i>CCC</i>	+	2	-

We will not check six newly got sequences: ABA , ABB , BCB , CAA , CCB of the third level (see Table 3) since they include infrequent subsequences of the previous level. The reduced amount of checking increases the algorithm efficiency and reduces time input. As the third level does not contain frequent sequences, we will not form the forth level and say that the performance of the algorithm is completed.

3. 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 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) the new much shorter model sequence \tilde{C} is generated according to these characteristics;
- 3) the new sequence is analyzed with the GSP algorithm (or any similar one);
- 4) the subsequences frequency in the main sequence is estimated by the results of the GSP algorithm applied on the new sequence.

Let:

1) $P(i_j) = \frac{V(i_j)}{VS}$ be the probability of occurrence of element i_j in the main sequence, where $i_j \in L$, $j = 1, \dots, m$. Here $L = \{i_1, i_2, \dots, i_m\}$ 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) $P(i_j | i_v)$ be the probability of appearance of element i_v after element i_j , where $i_j, i_v \in L$, $j, v = 1, \dots, m$. Note that $\sum_{v=1}^m P(i_j | i_v) = 1$ for all $j = 1, \dots, m$.

3) $D(i_j | i_v)$ be the distance between elements i_j and i_v , where $i_j, i_v \in L$, $j, v = 1, \dots, m$. In other words, the distance $D(i_j | i_v)$ 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 $D(i_j | i_v)$ includes i_v . The distance between two neighboring elements of the sequence is equal to one.

4) \widehat{A} be the matrix of average distances. Elements of the matrix are as follows: $a_{jv} = \text{Average}(D(i_j | i_v), i_j, i_v \in L), j, v = 1, \dots, m$. 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 $c_r, r = 1, \dots, l$. The model sequence \widetilde{C} will contain elements from $L: i_j \in L, j = 1, \dots, m$. For the elements $c_r, r = 1, \dots, l$, a numeric characteristic $Q(i_j, c_r), j = 1, \dots, m$, 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 $\rho(c_1, a_{1j}) \Rightarrow Q(i_j, 1 + a_{1j}) = Q(i_j, 1 + a_{1j}) + 1, j = 1, \dots, m$. Remaining elements $c_r, r = 2, \dots, l$, 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 $\max(P(c_{(r-1)} | i_p), P(c_{(r-1)} | i_t))$: $c_r = i_p$ if $P(c_{(r-1)} | i_p) > P(c_{(r-1)} | i_t)$, and $c_r = i_t$ if $P(c_{(r-1)} | i_p) < P(c_{(r-1)} | i_t)$. If these values are equal, i.e. $P(c_{(r-1)} | i_p) = P(c_{(r-1)} | i_t)$, then c_r is chosen in dependence on $\max(P(i_p), P(i_t))$. 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 $r = 2, \dots, l$. In such way we get the model sequence \widetilde{C} which is much shorter than the main one and which may be analyzed 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:

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

$$P(A|A) = 0.1, P(A|B) = 0.9, P(A|C) = 0,$$

$$P(B|A) \approx 0.1667, P(B|B) = 0.0833, P(B|C) \approx 0.7500,$$

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

Table 4. The matrix \hat{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 \tilde{C} , whose length is $l=8$. At the beginning, the sequence \tilde{C} is empty, and $Q(i_j, c_r) = 0$, $r = 1, \dots, l$, $j = 1, \dots, m$:

	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 \tilde{C}		-	-	-	-	-	-	-	-

The first element of \tilde{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 \tilde{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 \tilde{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 \tilde{C}	C	A	B					

r	1	2	2	3	4	4	6	7
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 \tilde{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 \tilde{C}	C	A	B	C	C			

The resulting model sequence is $\tilde{C} = CABCCABC$. The GSP algorithm determined that the longest frequent subsequence of the model sequence is ABC when the minimum support is set to two. 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 , whose 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.

4. 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 \tilde{C} of length $l=40$:

$$\tilde{C} = BBCABBCABBCABBCABBCABBCABBCABBCABBCABBCA$$

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 Fig. 1 (the minimum support in ProMFS is $M_s=8$; the results are similar for larger M_s).

The results in Fig. 1 indicate that, if the minimum support in GSP analyzing the main sequence is comparatively small (less than 1500 with the examined data set), GSP finds much more frequent sequences than ProMFS. When the minimum support in GSP grows from 2500 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 Fig. 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.

5. 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 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 on the results of analysis of the shorter one. The model sequence may be analyzed by the GSP or other algorithm for mining frequent sequences: the subsequences frequency in the main sequence is estimated by the results of the model sequence analysis.

The experimental investigation indicates that the new algorithm allows to save the computing time in a large extent. It is very important when analyzing very large data sequences.

Moreover, the model sequence, that is much shorter than the main one, may be easier understandable and perceived: in the experimental investigation, the sequence of 100000 elements has been modeled by a sequence of 40 elements. However, the sufficient relation between the length of the model sequence and the main sequence needs for a more deep 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 (the length l of

the model sequence \tilde{C} , the minimum support for analysis of the model sequence, etc.).

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.

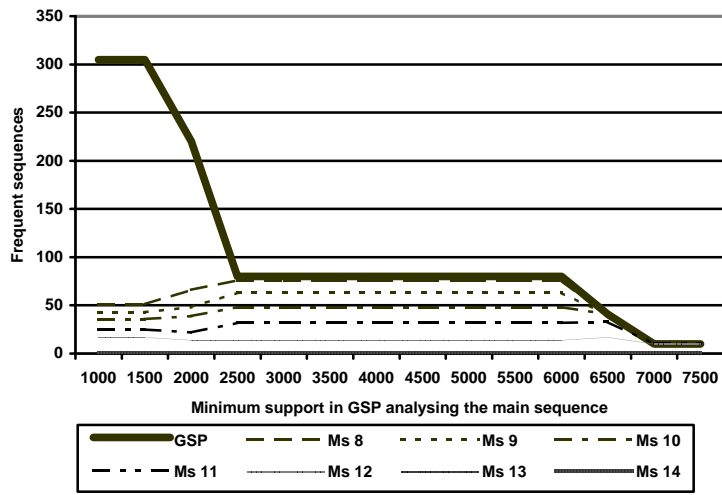


Fig. 1. Number of frequent sequences found both by GSP and ProMFS (minimum support in ProMFS is $M_s=8, \dots, 14$)

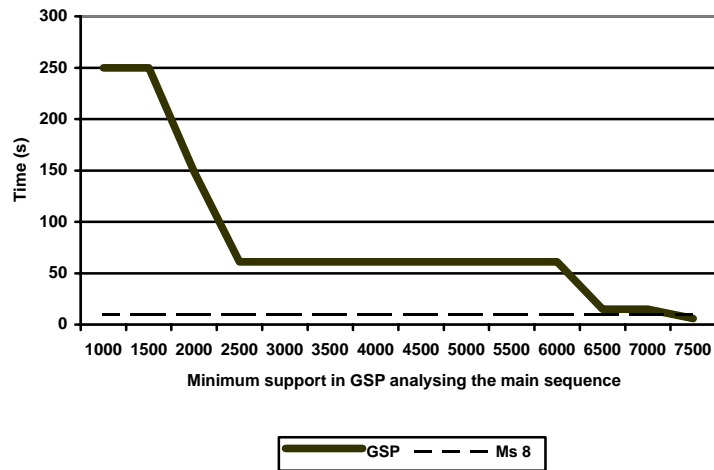


Fig. 2. Computing time used both by GSP and ProMFS (minimum support in ProMFS is $M_s=8$)

References

1. Agrawal, R.C., Agrawal C.C., Prasad V.V.: Depth first generation of long patterns. In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Boston, Massachusetts (2000) 108-118
2. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. Proc. 2000 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'00), Dallas TX (2000) 1-12
3. Zaki, M.J.: SPADE: An efficient algorithm for mining frequent sequences. Machine Learning Journal, **42** (1/2) (2001) 31-60 (Fisher, D. (ed.): Special issue on Unsupervised Learning)
4. Zaki, M.J.: Parallel sequence mining on shared-memory machines. In: Zaki, M.J., Ching-Tien Ho (eds): Large-scale Parallel Data Mining. Lecture Notes in Artificial Intelligence, Vol. 1759. Springer-Verlag Berlin Heidelberg New York (2000) 161-189.
5. Pei, P.J., Han, J., Wang, W.: 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 (2002) 18-25
6. Pinto, P., Han, J., Pei, J., Wang, K., Chen, Q., Dayal, U.: Multi-Dimensional Sequential Pattern Mining. In Proceedings of the 10th ACM International Conference on Information and Knowledge Management (CIKM'01), Atlanta, Georgia (2001) 81-88
7. Zaki, M.J., Parthasarathy, S.: Parallel algorithm for discovery of association rules. Data Mining and Knowledge Discovery **1** (1997) 343-374
8. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.-C.: PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In Proc. 17th International Conference on Data Engineering ICDE2001. Heidelberg (2001) 215-224
9. Han, J., Pei, J.: FreeSpan: Frequent pattern-projected sequential pattern mining. In Proc. Knowledge Discovery and Data Mining 2000 355-359
10. Ayres, J., Flannick, J., Gehrke, J., Yiu, T.: Sequential pattern mining using a bitmap representation. In Proc. Knowledge Discovery and Data Mining 2002 429-435
11. Kum, H.C., Pei, J., Wang, W.: ApproxMAP: Approximate Mining of Consensus Sequential Patterns. In Proceedings of the 2003 SIAM International Conference on Data Mining (SIAM DM '03), San Francisco, CA (2003) 311-315