Design of a prefixspan algorithm based on prefix position form

Youcheng Su[*], *Xin* He, *Zilong* Wang, and *Jianyu* Wang

Nanjing University of Science and Technology, School of Automation, Nanjing,China

Abstract. PrefixSpan algorithm based on sequence pattern is an association algorithm without generating candidate sequences and only needs to scan the original sequence database twice in the whole operation process. Compared with other sequence algorithms, the PrefixSpan algorithm is more efficient and fast. In this paper, PrefixSpan Base on Prefix Position Form, PPFPrefixSpan is proposed by studying the characteristics of the PrefixSpan algorithm and optimizing it based on it. , which solves the problem that the PrefixSpan algorithm produces too many duplicate projection databases in the process of operation, and optimates the PrefixSpan algorithm in space/time.

Keywords: PrefixSpan, Prefix position form, Repetitive projection database.

1 Introduction

In the algorithm of sequential pattern mining, experts and scholars spent a lot of time and energy to put forward many typical methods, such as SPADE, GSP, PrefixSpan algorithm, etc. Among them, the PrefixSpan algorithm based on prefix projection is lower than the other two algorithms in the required running time or the resource consumption of running space. PrefixSpan algorithm is not only more efficient in terms of computation speed, but also more stable in terms of memory usage compared to SPADE and GSP algorithms. Therefore, the PrefixSpan algorithm is taken as the research object in this paper, and the algorithm is improved and optimized.

2 Basic concepts

2.1 PrefixSpan algorithm

PrefixSpan algorithm[1] is based on the prefix projection sequence pattern mining algorithms, the algorithm principle of operation is the first to scan sequence database to generate the initial prefix length of 1[2], and once again scanning sequence database to projection of initial prefix, generate the corresponding initial projection database, then the projection database

[*] Corresponding author: 120110023044@njust.edu.cn

for recursive scanning generated frequent sequence length is 2, And so on, until you can't recursively mine longer frequent sequences.

The PrefixSpan algorithm flows as follows:

1) Scan the sequence database D, count each sequence of length 1, and get the sequence support degree of each single item.

2) The single item whose sequence support is lower than the minimum support min_sup[3] is deleted from the data set, and the frequent sequence of length 1 is obtained as the initial prefix, I =1. Each initial prefix is projected to generate the corresponding initial projection database.

3) Recursive mining is performed for each prefix of length I and its corresponding projection database.

a) Generate the projection database for the prefix. If the projected database is empty, the recursion is returned.

b) Scan the projection database and count the individual items. If all items have sequence support below min_sup, the recursion is returned.

4) Combine each single item that satisfies the minimum support min_sup with the current prefix to get several new prefixes.

5) Make I = I +1, and the prefix is each new prefix after merging the single item, and recursively perform Step 3 respectively.

2.2 Repetitive projection database

PrefixSpan algorithm is mainly based on the idea of using divide-and-conquer. It only needs to scan the sequence database twice, and recursively generates many smaller projection databases[4] and frequent pattern prefixes, until it cannot recursively mine longer frequent sequences. However, in the process of generating the projection database, the same projection database may be projected to different prefixes, thus causing repeated recursive calculation, which is the problem of the repeated projection database.

2.3 Prefix position form

In general, a single item is mapped, usually to the place where the single item first appears in the sequence. To facilitate the representation of sequence-based patterns, the definition of item position is proposed. The position of the term refers to the position of the term in the sequence, given the sequence $\alpha =< \alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_i >$, for any item b, if $b \in \alpha_j (1 \leq j \leq i)$, says item b in the sequence of a position of the item for j, remember to $d_{(\alpha,j)}$.

The position of the prefix refers to the position of the prefix in the sequence, given the sequence $\alpha =< \alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_i >$, for any prefix $\beta =< \alpha_1, \alpha_2, \ldots, \alpha_j >$ in the sequence α, the position of the prefix β in the sequence a is determined by the position of the last term α_j in the prefix, i.e.,j, and is denoized as $q_{(\alpha,j)}$.

When there are two different prefixes α and β in the database, the set of prefix positions generated for each sequence is the same, then the projection database generated for their mapping is also the same. And the number of prefix positions in the database that can be found for any prefix is equal to the sequence support of that prefix in the database.

An itemset is a frequent itemset when the sequence support of an itemset meets given minimum support. The nonempty subset of any frequent pattern is necessarily frequent. All supersets of infrequent patterns must be infrequent patterns.

There are prefixes α and β, and when their set of prefix positions is equal and the number of their prefix positions meets the minimum support, recursively generating the same set of frequent sequence patterns on the projection database of prefix α and β.

Based on the PrefixSpan algorithm, the PPFPrefixSpan algorithm (PrefixSpan Base on Prefix Position Form) is proposed. PPFPrefixSpan algorithm needs to construct a prefix position information table as auxiliary information in the operation process. Prefix position information table, which only saves the frequent subsequences in the operation process as the prefix, the position of each sequence in the database.

3 PPFPrefixSpan algorithm

PPFPrefixSpan algorithm is based on the PrefixSpan algorithm, based on the prefix projection sequence pattern association algorithm. Differences from the PrefixSpan algorithm, PPFPrefixSpan algorithm introduces a table of prefix location information. Because the set of frequent sequence patterns generated for the same projection database is the same. If the position information of any prefix a is the same as that of prefix β in the prefix position information table, then we can directly return the set of frequent sequence patterns generated by the projection database of prefix a and generate a new set of frequent sequence patterns generated by prefix β. Therefore, depth-first traversal is adopted. A single term of the initial prefix is first taken and recursed from length 1 to length L, which is used as the initial reference set. Because the probability of prefix positions repeating between new prefixes recursively generated from the same prefix is much lower than the probability of prefix positions repeating between prefixes of the same length through breadth-first traversal. Therefore, the PPFPrefixSpan algorithm adopts depth-first traversal.

In the actual operation of the algorithm, the prefix position information table is saved through a Hash table. Hash table is a kind of data structure that stores access through key-value, which can be directly queried by the Key Value to speed up the query. There are three important pieces of information in the algorithm, prefix, prefix position, and the frequency sequence of the corresponding projection database. Therefore, this information is stored in a secondary Hash table. The first level stores prefixes -- the prefix location, which is the Key-Value, and the prefix location, which is the Value. The second level stores the prefix position -- the frequent sequence. The prefix position is the Key-Value, and the frequent sequence corresponding to the prefix in the projection database is the Value.

First, the algorithm scans the entire sequence database, counts all single items, sorts the support degree of each single item, and takes the first item with a high support degree as the initial prefix. Depth-first traversal is adopted, the projection database is constructed for the first initial prefix first, and if the prefix position information table is empty, the prefix position information is saved directly. In the projection database, the single item that meets the minimum support is counted to form a new prefix with the original prefix, the prefix position information of the new prefix is saved, and a new projection database is formed. In the iteration, the projection database is empty. Save all sequence pattern results. Get the prefix position information of the second initial prefix, scan the prefix position information table, if there is a prefix with the same position as the initial prefix, stop recursion, directly return the sequence pattern set generated by the prefix, otherwise, save the initial prefix position information, and continue recursion. Returns a collection of all sequence patterns until all initial prefixes have been computed.

PPFPrefixSpan algorithm flow:

1) Scan the sequence database D, count each single item of length 1, get the sequence support degree of each single item and arrange it in descending order. The single term of the first μl item that satisfies the minimum support min_sup is selected as the initial prefix.

2) Calculate the prefix position of the first initial prefix and save it in the prefix position information table to generate its projection database. Iterate over it until no new projection database can be generated. Save the collection of frequent sequence patterns generated by each projection database.

3) Start from the second initial prefix until all the initial prefixes have been calculated.

a) Generate the projection database for the prefix. If the projected database is empty, the recursion is returned.

b) Scan the projection database and count the individual items. If all items have sequence support below min_sup, the recursion is returned.

c) Combine each single item that satisfies the minimum support min_sup with the current prefix to get several new prefixes. And calculate the prefix position of the new prefix. If there is a prefix equal to the prefix position in the prefix position information table, the set of frequent sequence patterns generated by the prefix in the prefix position information table is directly returned, and Step3 is returned. Instead, the prefix position information table holds the position information of the new prefix and generates new projection data, returning a).

4 Experimental design and verification

The experimental environment of this chapter is carried out on a PC, and the relevant programs are written in Python, and the software is run in Windows. The experiment adopts the real data set selected in the previous chapter. The data set is from all sales records of a retail supermarket since January 1, 2015 solstice and April 30, 2015. The dataset included 42,817 purchases made by 2,000 users over a four-month period.

The execution efficiency and execution space of EiPrefixSpan algorithm and PPFPrefixSpan algorithm with support of 2%, 4%, 6%, 8%, 10%, 12%, 16%,20% on the data set are compared respectively. The experimental results of execution efficiency are shown in Table I., and the experimental results of execution space are shown in Table II.

Table 1. Execution efficiency comparison results.

min_sup value	2	4	6	8	10	12	16	20
PrefixSpan	386	147	99	82	68	59	53	45
PPFPrefixSpa (μ=30)	125	89	61	55	48	37	31	28

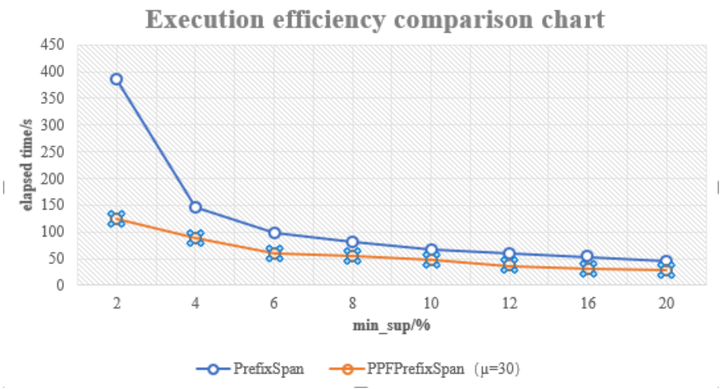

Fig. 1. Comparison diagram of execution efficiency.

Table 2. Perform spatial comparison results.

min_sup value	2	4	6	8	10	12	16	20
PrefixSpan	12.27	11.58	11.79	11.21	10.64	10.93	10.32	10.18
PPFPrefixSpan (μ=30)	7.45	7.13	6.78	6.94	6.69	6.73	6.63	6.55

As shown in Fig.1., the execution efficiency of the PPFPrefixSpan algorithm is significantly better than that of the PrefixSpan algorithm. In the process of sequence pattern mining, the PPFPrefixSpan algorithm avoids the repeated recursion of repeated projection database to generate frequent sequences by means of prefix position information table and depth-first traversal and reduces the operation time of the algorithm. Therefore, the more data sets with repeated projection databases, the more effective the algorithm will be. Therefore, the mining efficiency of the sequence pattern is improved.

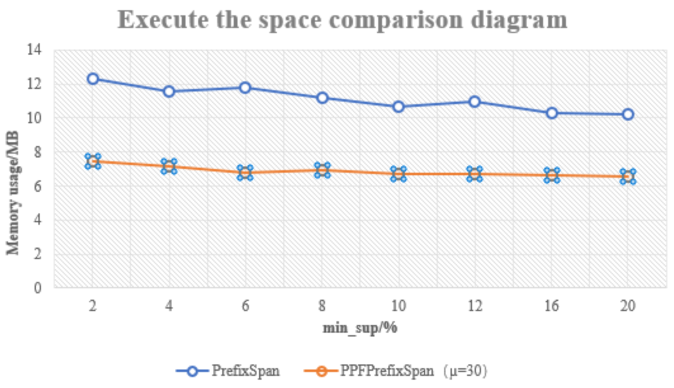

Fig. 2. Comparison diagram of execution space.

As shown in Fig.2., you can observe that the execution space of the PPFPrefixSpan algorithm is significantly better than that of the PrefixSpan algorithm. PPFPrefixSpan algorithm avoids generating duplicate projection databases in the process of sequential pattern mining, which reduces the memory usage required by the PPFPrefixSpan algorithm in the process of operation.

The experimental results show that the PPFPrefixSpan algorithm saves time and space resources compared with the PrefixSpan algorithm when there are more repeated projection databases generated on the data set in the process of algorithm operation. It proves that the algorithm optimization of the PPFPrefixSpan algorithm by introducing the prefix position information table is feasible.

5 Conclusion

In this paper, it is proposed that the PrefixSpan algorithm may have the problem of repeated projection database in the operation, which leads to the mining and division of the repeated projection database in the operation, resulting in a certain amount of repeated calculation, thus increasing the time/space consumption. Then the possibility of the existence of the repeated projection database is proved by an example and the prefix position information table is proposed. Through an example, it is deduced that when any two prefix position sets are equal and the number of their prefix positions meets the minimum support, then the recursively generated frequent sequence pattern sets of these two projection databases are the same. Based on this principle, the PPFPrefixSpan algorithm is proposed based on the PrefixSpan algorithm, which uses the assistance of the prefix position information table and the depth-first traversal to avoid the repeated calculation of the projection database. Finally, the experiment proves that the PPFPrefixSpan algorithm is better than the PrefixSpan algorithm in execution efficiency and execution space.

Reference

1. KUN Z, Yangyong Zhu. Sequence Pattern Mining Without Duplicate Project DataBase Scan[J].Computer Research and Development,2007, 44(1): 126-132.

2. WANG L L, Fan J. Improved Algorithm for Sequential Pattern Mining Based on PrefixSpan[J]. Computer Engineering,2009,35(23):56-58.
3. Han J, Kamber M.Data Ming: Concepts and Techniques[J]. Data Mining Concepts Models Methods & Algorithms Second Edition,2012,5(4):1-18.
4. KUN Z, Yangyong Zhu. Sequence Pattern Mining Without Duplicate Project DataBase Scan[J]. Computer Research and Development, 2007, 44(1): 126-1