Research on Massive News Events Evolution Prediction Based on Improved PrefixSpan Algorithm

Biao Wang¹,a, Xiao Dai²,b, Jing Yang²,c, Shun Li²,*
¹Network Center University of International Relations Beijing, China
²School of Information Science & Technology University of International Relations Beijing, China
³wangbiao@uir.edu.cn, ⁴Devin1212@163.com, ⁵20147107@uir.edu.cn, ⁶lishunmail@foxmail.com

Abstract. The Internet reports massive news event every day. Sequential pattern mining is adopted to study the timing relationship between news event, which can provide reference for the development or prediction of news events. With its advantages in performance and efficiency, PrefixSpan often becomes preferred algorithm in the field of sequence pattern mining. But unfortunately, because of the large amount of news and the long time span between some news events, the event sequences is long and dense, which results many subsequences in the frequent pattern and reduces the algorithm performance. In this paper, we propose an improved PrefixSpan algorithm by integrate subsequences of the supersequences in the frequent pattern and introduce news event weight in the pruning step of the PrefixSpan algorithm. Experiment results show that without affecting expression ability of frequent patterns, the improved PrefixSpan algorithm increases the efficiency by about 20 times as before and eliminates about 10% of the redundant subsequences to get a more concise frequent patterns set.

1. Introduction

Powered by the rapid development of Internet technology, the network media have derived the function of reporting real time news while creating mass of texts. History does not replicate itself, however, astonishing parallels can often be found in history. The type of news event is the action or activity initiated by the subject to the object at a specific time. The event related news was clustered by word bag method to form the news set of related events. Under this news set, news event is correlated to a greater or lesser extent. For example, when America makes trade threats to China, China imposes administrative restrictions on America. Through mining the sequence patterns of related news sets to obtain the news events correlation or knowledge that users are interested in, which can provide a reference for the development or prediction of the event.

Sequence pattern mining (SPM) finds sequence patterns that exceed the minimum support threshold in the sequence database. SPM is similar to association pattern mining in many ways, but it is more concerned with the sequential relevance of data. SPM is an important data mining problem with widespread application, including web access pattern, stock trend, natural disaster, event prediction, user behavior pattern, and so on. For example, AprioriAll algorithm[1] based on width search priority, GSP algorithm[2] with constraints such as time and class hierarchy, FreeSpan algorithm[3] and PrefixSpan algorithm[4] based on sequence growth pattern. Among the above SPM algorithms, AprioriAll and GSP belong to class Apriori algorithm will generate a number of candidate sequences in calculation process,
which is not suitable for mining dense data sets and long sequence patterns. In comparison, FreeSpan and PrefixSpan do not generate many candidate sequences and they are more efficient and feasible. Moreover, compared with FreeSpan, PrefixSpan is only based on frequent prefix subsequence for projection, and its performance in time and space is better than Freespan.

With its advantages in performance and efficiency, PrefixSpan often becomes preferred algorithm in the field of sequence pattern mining. However, because of the large amount of news and the long time span between some events, the event sequences is long and dense, which results many subsequences in the frequent pattern and reduces the algorithm performance. In this paper, we propose an improved PrefixSpan algorithm by integrate subsequences of the supersequences in the frequent pattern and introduce news event weight in the pruning step of the PrefixSpan algorithm. Experiment results shows that without affecting expression ability of frequent patterns, the improved PrefixSpan algorithm increase the time efficiency by about 20 times as before and eliminates about 10% of the redundant subsequences to get a more concise frequent patterns set.

2. Related Work

Li[5] proposed an Improved PrefixSpan algorithm for Mining Sequential Patterns(IPMSP) algorithm by avoid produce duplicated project databases with the same prefix pattern through checking the prefix with regard to prefix of the sequence database and abnegating the non-frequent items and project databases which sequential number is lower than minimum support in the recursive mining process. Yang[6] proposed an IPISP algorithm solve the problem that the PrefixSpan algorithm needs to rerun the algorithm in the process of incremental updating of the sequence pattern, needs to scan the original database repeatedly in the mining process. The IPISP algorithm uses the sequence pattern base to store all the mining sequences and their supports. Sun[7] proposed a tourist route recommendation algorithm based on PrefixSpan algorithm which combines the unique attributes of the tourist route, such as the number of days traveled and the type of person. Kim[8] developed PrefixSpan algorithm using Apache spark to improve sequence pattern search efficiency and scalability. Zhang and others[9-10] proposed the user behavior pattern mining algorithm based on PrefixSpan algorithm to provide a basis for personalized services. Wang and others[11-12] proposed the prediction algorithms of application usage based on PrefixSpan algorithm with Bayesian network or Support Vector Machine(SVM). Li and others[13-14] used the PrefixSpan algorithm to extract knowledge patterns from massive security logs and massive codes to provide support for corresponding research.

In recent years, academic research and exploration mainly focused on projection databases and application innovations, which has better solved the problem of duplication when the algorithm divides the database according to frequent prefix. However, regarding to the problem of generating enormous subsequences when PrefixSpan is mining long and dense sequences of news events, there is still a long way to go. The emphasis of this study lies in how to use PrefixSpan algorithm to mine a more concise frequent patterns set, save storage space, and improve algorithm efficiency.

3. Background

3.1. Introduction of PrefixSpan algorithm

Han proposed the PrefixSpan algorithm based on sequence growth pattern. With its advantages in performance and efficiency, PrefixSpan algorithm is usually preferred algorithm in the field of sequence pattern mining.

3.1.1. Related Definition and Theorem

**Definition1** (Event sequence) Given a sequence \( s = \langle n_1, n_2, ..., n_k \rangle \), \( k \) is the length of the event sequence, \( n_i (1 \leq i \leq k) \) is event item.

**Definition2** (Prefix and postfix) Given sequences \( a = \langle a_1, a_2, ..., a_m \rangle \) and \( b = \langle b_1, b_2, ..., b_n \rangle \) \( (1 \leq m \leq n) \), if \( i \leq n - 1, a_i = b_i \), \( b \) is the prefix of \( a \). For example, the prefix of sequence ABBDCAE includes A, AB, ABB, ABBD. The part of sequence data without prefix is postfix. For example, the postfix of
sequence ABBDCAE includes CAE and DCAE.

**Definition 3** (Projected database) Suppose sequence $a$ is frequent pattern in sequence database $S$, the projection database of $a$ is its set of postfix in $S$ about prefix $a$.

**Definition 4** (Support) The support of sequence $s_t$ is the number of $s_t$ contained in sequence database $S$.

$$\text{Support}(s_t) = \# \{\text{tid}, s > | (< \text{tid}, s >) \in S \land (s_t \in s)\}$$ \hspace{1cm} (1)

**Definition 5** (Frequent sequence pattern) When the support of a sequence is not less than the minimum support threshold (minS), the sequence is called the frequent sequence pattern.

**Definition 6** (Subsequence and supersequence) Given sequences $a = \langle a_1, a_2, ..., a_m \rangle$ and $b = \langle b_1, b_2, ..., b_n \rangle$, if there exist integers $i_1 < i_2 < ... < i_m < n$, such that $a_i = b_i$, $a_{i+1} = b_{i+1}$, ..., $a_m = b_m$, then $a$ is contained by $b$, $a$ is called a subsequence of $b$ and $b$ a supersequence of $a$.

3.1.2. Algorithm Steps

PrefixSpan algorithm adopts the idea of divide and rule. It divides the large database into small projection database continuously, and then mines the sequence pattern on the projection database. The main steps are as follows:

**Step 1:** Find length-1 sequential patterns from the sequence database, mark $S_1$.

**Step 2:** Divide sequence database by $S_1$, and length-1 frequent pattern is used as a prefix to construct a projection database and a postfix set corresponding to the prefix.

**Step 3:** Recursive mining frequent sequence pattern in the projection database constructed in **Step 2** until the result is empty.

**Step 4:** Record all mined frequent sequence patterns.

3.2. News event sequence mining and analysis

The main idea of news event sequence mining analysis is to mine the sequence pattern of relevant news event sets, to obtain the temporal relation between the news event, and to provide a reference for the development prediction of the news event. According to the characteristics of news, news event sequence mining analysis can be divided into three stages: (1) news event data preprocessing, (2) event sequence pattern mining, and (3) frequent event sequence analysis.

![Fig.1 News event sequence mining analysis process](image)

(1) News event data preprocessing

Before mining sequence patterns of news events, it is necessary to preprocess the news data. Firstly, cluster the news text data to form a related news data set, perform data cleaning and statute and encode the event into a triple consisting of subject, event type, and object: Event=(Actor1EventClassActor2). For example, A1E02A2 indicates that the E02 event type occurred between the subject A1 and the object A2. Based on this, a news event database is formed as shown in Table 1, which records the events that occur every day in the relevant context.

| Date | Event |
|------|-------|
| D1   | [A1E02A2, A3E07A2, A8E01A3] |
| D2   | [A8E02A1, A2E05A1, A3E04A6, A4E02A2] |
| D3   | [A8E07A3, A1E06A2] |
| D4   | [A6E07A3] |
| D5   | [A2E07A2, A1E02A2] |
| D6   | [A2E04A4] |
The news event database is composed of a series of related events with time continuity. However, events contain both big and small time intervals. For example, A1E02A2 occurs on D1 day, and its response event A2E05A1 occurs on D2 day. But D2 day A3E04A6 response event A6E07A3 occurs on D4 day. In order to ensure the continuity of events in time, we use the sliding time window method to construct a sequence database and generate event sequences according to the specified time window length.

\[ S_i = SW < D_i, D_{i+1}, ..., D_{i+\text{window\_num}-1} > \ (i = 1, 2, 3, \ldots) \] (2)

As shown in Table 2, we make time window length equal to 3 to generate a sequence database.

Table 2. News Event Sequence Database

| SID | Event sequence |
|-----|----------------|
| S1  | \{ [A1E02A2, A3E07A2, A8E01A3], [A8E02A1, A2E05A1, A3E04A6, A4E02A2], [A8E07A3, A1E06A2] \} |
| S2  | \{ [A8E02A1, A2E05A1, A3E04A6, A4E02A2], [A8E07A3, A1E06A2], [A6E07A3] \} |
| S3  | \{ [A8E07A3, A1E06A2], [A6E07A3], [A2E07A2, A1E02A2] \} |
| S4  | \{ [A6E07A3], [A2E07A2, A1E02A2], [A2E04A4] \} |

(2) Event sequence pattern mining

This stage is the core of news event sequence mining analysis, which mined the sequences that exceed minimal support threshold (minS) from the event sequence database. For example, when the minS =50%, \{[A8E02A1], [A2E04A4]\} is the frequent sequence pattern. The frequent sequence pattern reflects the temporal relationship between events, indicating that A2E04A4 may occur after the occurrence of A8E02A1. Therefore, we can provide a reference for the development and prediction of certain events based on the frequent sequence pattern.

(3) Frequent event sequence analysis

Frequent event sequences are analyzed and interpreted using appropriate tools and techniques to select sequences of interest to relevant researchers.

4. Problem definition

We use PrefixSpan to mine the temporal relationship between news event, using sentence-based events, that is, events exist in a sentence. A news has multiple events, as shown in Table 3.

Table 3. News and event

| News | Event |
|------|-------|
| China has decided to purchase American agricultural products. US Trade Representative Lighthizer will also hold talks with Chinese Vice Premier Liu He in Washington. The two sides tried to alleviate trade tensions by exchanging goodwill. | **Event1:** China has decided to purchase American agricultural products; **Event2:** US Trade Representative Lighthizer will also hold talks with Chinese Vice Premier Liu He in Washington. |

The Internet reports massive news every day. A news contains multiple events, so there are massive events in a period of time. After the sliding time window operation, the corresponding event sequence is long and dense, resulting in a large number of subsequences in frequent patterns, and the algorithm running time is long when mining long sequences.

5. Improved PrefixSpan Algorithm

On the basis of apriori principle, if a sequence is frequent, all its subsequences must also be frequent. Moreover, news sequence pattern mining is based on the clustering of news text, and sentence-based event mining. There are some unrelated events that increasing traversal complexity. Aiming at the above problems, we propose an improved PrefixSpan algorithm by integrate subsequences of the supersequences in the frequent pattern and introduce news event weight in the pruning step of the PrefixSpan algorithm.
5.1. Related Definition and Theorem

**Definition 8** (Event weight) the ratio of the number of days the event occurred in the news database to the total number of days.

\[
\text{Weight} = \frac{\sum D(\text{event})}{\sum D}
\]  

(3)

5.2. Improved PrefixSpan Algorithm Main Steps

**Step 1:** Scan the news database to calculate the weight of each event.

**Step 2:** Scan the news sequence database, delete events whose event weight is less than the minimum event weight, find the 1-length frequent pattern, and divide the projection database by 1-length frequent pattern.

**Step 3:** Based on Step 2, recursively mining frequent sequence pattern in the projection database was constructed, record the frequent sequence patterns, and sort by the sequence length;

**Step 4:** Give a function allin \((s_1, s_2)\) judging whether frequent sequences have supersequence, when \(\text{len}(s_2) > \text{len}(s_1)\), Is \(s_1\) a subsequence of \(s_2\)? if yes, \(s_2\) absorbs \(s_1\).

**Step 5:** Record all frequent sequence patterns.

The improved Prefixspan algorithm is described as follows:

**Input:** Original database(DB), Sequence database(SDB), minS, minimal event Weight(minW)

**Output:** Frequent sequence pattern

1. CalculateWeight(DB) //calculate the weight of each event
2. Prefixspan() 
3. frequentS1=find_1frequent(SDB,minS,minW)//scan SDB use minS and minW double filtering
4. for n in frequentS1:
5. S n = build_projected_database(n)
6. frequentS = prefixspan(Sn,minS) //recursively mining frequent sequences
7. end for
8. frequentS.sortby(len(frequentS))//record frequent sequence and sort by the sequence length
9. for fs1 in frequentS:
10. for fs2 in frequentS(len(fs) > len(fs1)):
11. if allin(fs1,fs2) == true: //determine whether the frequent sequence is contained by the sequence longer than it
12. Delete (fs1) //if true, delete fs1
13. End if
14. End for
15. End for
16. frequentS//get the final frequent pattern

The improved PrefixSpan algorithm reduces the number of frequent sequence patterns by integrate subsequences of the supersequences. And introduce news event weight, use minS and minW double filtering, decrease the size of the sequence database and shorten the algorithm traversal time.

6. Experiment and Result Analysis

6.1. Experimental environment and data set

The operating environment is CPU 2.3 GHz, memory 8G 2133 MHz LPDDR3, MacOS. All algorithms are written in python2.7 and implemented using pycharm.

The experiment uses the real data set TradeWar for algorithm comparison. TradeWar data set derives from GDELT of Google on May 1, 2019 to July 1, 2020, which is about the trade disputes between China and the United States.
6.2. Analysis of experimental results
Considering the event time span, time window length is set to 5 and a comparison between the following two aspects is given:

1. Contrast original PrefixSpan (prefixspan1) with PrefixSpan that introduces event weights (prefixspan2) in algorithm running time and expression ability of frequent patterns.

2. Contrast prefixspan2 with PrefixSpan that introduces event weights and drops subsequence (prefixspan3) in the number of frequent pattern.

6.2.1. prefixspan1 and prefixspan2
Contrast prefixspan1 to the prefixspan2 with minW 0.08, 0.16, 0.24 in algorithm running time. Respectively set minS 0.6, 0.65, 0.7, 0.75, 0.8.

As shown in Fig.2, under the same minS, the running time of prefixspan2 is far lower than prefixspan1. And with the increase of minW, the running time of prefixspan2 is also reduced.

Compare the differences between prefixspan1 and prefixspan2 in the number of frequent sequence patterns under different minW. Analyze the impact of weights on the expression ability of sequence patterns, as shown in Fig.3:

As it can be seen from the data in Fig.3. With the increase of minW, the expression ability of frequent patterns will be affected. For example, the frequent pattern of prefixspan2 with minW=0.24 is reduced by 1% compared to prefixspan1. However, when minW is 0.08,0.16, there is no difference between the
two algorithms in the same minS. It can be concluded that under the reasonable setting of minW, the frequent patterns of prefixspan1 and prefixspan2 have the same expression ability.

6.2.2. prefixspan2 and prefixspan3

The analysis of time efficiency and expression ability is given above. When the minw is 0.16, it does not affect the expression ability of frequent sequences and has a good time efficiency. In summary, given minW=0.16, a contrast between prefixspan2 and prefixspan3 is shown and the analysis of the removal effect of redundant subsequence is given.

As can be seen from Fig.4,5:

(1) With the same minS, the frequent pattern of prefixspan3 is reduced by 10% ~15% compared to prefixspan2. It shows that prefixspan3 gets a more concise frequent patterns without affecting its expression ability, eliminating 10% ~ 15% redundant subsequences.

(2) From Fig.5, The running time of Prefixspan3 is higher than prefixspan2 when the minS is low. And the difference between them is not significant when the minS is higher. In general, prefixspan3 and prefixspan2 is far lower than prefixspan1, about 1 / 20 of the running time of prefixspan1.

Finally, the results of the experiment indicate that the improved PrefixSpan algorithm is superior to the PrefixSpan algorithm.
7. Conclusion

In this paper, an improved algorithm is proposed to solve the problem that there is a large number of subsequences in the PrefixSpan algorithm, which reduce the efficiency of the algorithm. The experimental results suggest that the improved algorithm mines more concise sequence patterns, improves the efficiency of the algorithm, and effectively obtains the temporal relationship between the news events. Through analyzing and studying the frequent event sequences, the event correlation and knowledge that users are interested in are acquired. It is possible to provide reference for the development or prediction of news events. In the further study, more effort will be put on exploring how to use frequent event sequences and prediction models to predict event types.

Acknowledgments

Thanks are owed to my tutor, who spent much time in and effort on guiding me finish this paper. Grateful acknowledgement is also made to me laboratory members, which provided the experiment data and gave my many valuable suggestions. Finally, I am indebted to my parents for their continuous support and encouragement.

References

[1] Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In Eleventh International Conference on Data Engineering. IEEE Xplore.

[2] Srikant, R. and Agrawal, R. (1996). Mining Sequential Patterns: Generalizations and Performance Improvements. In International Conference on Extending Database Technology: Advances in Database Technology. Springer-Verlag.

[3] Han, J., Pei, J., Mortazavi-Asl, B., et al. (2000). Freespan: frequent pattern-projected sequential pattern mining. In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining.

[4] Han, J., Pei, J., Mortazavi-Asl, B., et al. (2001). Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In Proceedings of the 17th International Conference on Data Engineering.

[5] Li, P. and Shan, S. (2016). An Improved Algorithm of PrefixSpan. In 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS). Fukuoka.

[6] Yang, Y. and Zhou, X. (2017). An Incremental Updating Algorithm for Sequence Patterns Based on Improved PrefixSpan. In 2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC). Hangzhou.

[7] Sun, W., Chang, L., et al. (2019). Travel Route Recommendation Based on Knowledge Graph and Frequent Sequence Mining. In Computer Science, 46(02):56-61.

[8] Kim, B. and Yi, G. (2019). Location-Based Parallel Sequential Pattern Mining Algorithm. In IEEE Access, vol. 7, pp. 128651-128658.

[9] Zhang, W., Zhou, W. and Luo, J. (2019). Mining and Application of User Behavior Pattern Based on Operation and Maintenance Data. In 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). Arlington, VA, USA.

[10] Zhu, Y., Bao, P. and Genling, J. (2018). A parallel algorithm for mining user frequent moving patterns. In Journal of University of Science and Technology of China, 48(1): 57-64.

[11] Wang, K., Bao, Q., Yushuai, Z. and Jichao, W. (2019). Prediction Algorithm of App Usage Based on Smartphone User Behavior Habits. In Computer Applications and Software, 36(08):82-86+195.

[12] Wang, A. (2019). Research on App Usage Behavior Analysis and Prediction Algorithms for Mobile Users. In 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA). Qiqihar, China.

[13] Li, K., Li, Y., Liu, J., Zhang, R. and Duan, X. (2019). Attack pattern mining algorithm based on security.
log," 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), Beijing, 2017, pp. 205-205.

[14] B. Liu, J. Liu, J. Zhao and C. Jiao, "Research on Digital Serialization Method of Program Code," 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 2019, pp. 616-621.