Analysis of Web Access Sequence Based on the Improved PrefixSpan Algorithm

Yang Xu[1, a],Yu Wang[2,b]

[1]Department of Computer and Information,HoHai University,NanJing,China

[2]Department of Computer and Information,HoHai University,NanJing,China

[a]xuyang091@sina.com, [b]won9805@gmail.com

Keywords: Data mining; Sequential pattern; PrefixSpan; IPS; Web access sequence

Abstract. PrefixSpan is an important algorithm for sequential pattern mining algorithm, but it's projected database cost more redundant memory and scan-time, so this paper present an improved PrefixSpan algorithm(IPS) which is based on PrefixSpan. IPS decreases the redundant memory and scan-time by abnegating the non-frequent items and projection database which sequential number is lower than minimum support. This paper applied IPS to web access sequence mining, by mining the web access records database to find frequent access sequence to provide reasonable suggestions for Web building.

Introduction

With the rapid development of the Internet, the network data grows at a geometric rate, thus users face magnanimity information. How to make good use of the data, improve information utilization, easier transfer, exchange, obtain useful information and mining information hidden behind data has became the focus of experts and scholars in relevant field. The rapid development of Web makes it become the largest public data source in the world, and Web data mining goal is to search useful information from Web hyperlinks, web content and usage logs. Based on the primary kind of data used in the mining process, Web mining[1]tasks are categorized into three main types: Web structure mining, Web content mining and Web usage mining. In this paper, the main research is Web usage mining.

The sequential pattern mining[2] problem was first introduced by Agrawal and Srikant in 1995, it's initial motivation is to find the frequent sequences in transaction database with transaction time attributes. Agrawal and Srikant also proposed the AprioriSome, AprioriAll and DynamicSome which are all based on Apriori algorithm[2]; then Srikant further proposed generalized sequential pattern mining algorithms(GSP)[3]; Zaki proposed Sequential Pattern Discovery using Equivalent classes algorithm(SPADE)[4], it is a sequential pattern mining algorithm based on Aporiori using a vertical data format and the extension of the vertical format frequent item mining method; Both FreeSpan[5] and PrefixSpan[6] algorithm are two pattern growth method for sequential pattern mining, PrefixSpan algorithm is better than FreeSpan algorithm in performance for fewer projection library and subsequence connection. In mining process, the major costs of PrefixSpan algorithm is the massive construction of projected database, this became the bottleneck of the PrefixSpan algorithm.

PrefixSpan algorithm. A sequence database S is a set of tuples $< sid, s >$, where sid is a sequence identifier and s is an sequence. If sequential α is the subsequence of s, then tuples $< sid, s >$ contains α. The support count of α means the number of tuples which contains α in database, denotes as support(α). Give a support threshold, min_sup, a sequence S_a is a frequent sequence in the sequence database if the support of the sequence S_a is no less than a minimum support threshold. A sequence with length l is called an l-sequence.

Definition 1: Prefix. Suppose all the items in an element are listed alphabetically. Given a sequence $\alpha = (e_1, e_2, \cdots e_n)$, $e_i (1 \le i \le n)$ is a frequent item in S. A sequence $\beta = (e_1', e_2', \cdots e_m')$ is called

a prefix of α if and only:1) $e_i^{'} = e_i (i \le m-1)$; 2) $e_m^{'} \subseteq e_m$; 3)All the items in $(e_m - e_m^{'})$ are alphabetically after $e_m^{'}$.

Definition 2: Projection. Give sequence α and β, such that β is a subsequence of $\beta\ (\beta \subseteq \alpha)$. $\alpha^{'}$ is the subsequence of $\alpha\ (\alpha^{'} \subseteq \alpha)$. $\alpha^{'}$ is called a projection of α w.r.t prefix β if and only if: $\alpha^{'}$ has prefix β and there exits no proper super-sequence $\alpha^{''}$ of $\alpha^{'}$ such that $\alpha^{''}$ is a subsequence of α and also has prefix β.

Definition 3: Suffix. Given a sequence $\alpha = (e_1, e_2, \cdots, e_n)$, $e_i (1 \le i \le n)$ is a frequent item in S. Suppose $\beta =< e_1, e_2, \cdots e_{m-1}, e_m^{'} > (m \le n)$, β is the prefix of α, sequence $\gamma =< e_m^{''}, e_{m+1}, \cdots e_n >$ is called the postfix of α w.r.t prefix β, denoted as $\gamma = \alpha / \beta$, where $e_m^{''} = (e_m - e_m^{'})^2$, we also denote as $\alpha = \beta \bullet \gamma$.

Definition 4: Projected Database and the support of Projected Database. Suppose α is a sequential pattern in sequential database S, α is the prefix of β, then the α-projected database denoted as $S|_\alpha$, is the collection of suffix of sequence in S w.r.t prefix α. The support count of β in α-projected database $S|_\alpha$ is the number of γ in $S|_\alpha$ such that $\beta \subseteq \alpha \bullet \gamma$.

Pattern growth is a method of frequent pattern mining without candidate generation, its basic ideas are as follows: First, find all frequent items, then generate the set of projection database, each projected database associated with a frequent item and mined separately. The algorithm construct the prefix pattern, it connect with suffix pattern to get frequent patterns, thus avoiding generate candidate sequences. The steps of PrefixSpan describes as: 1)Scan the sequential database to get all frequent item, namely the set of frequent sequence with 1 length; 2)Divide the set into subset for n, each subset has different prefix; 3)Construct corresponding projection database and mining the subset of frequent sequence recursively.

Improved PrefixSpan Algorithm

Improved method. The proposed IPS algorithm make improvement mainly in two aspects:

1)When constructing projected database, IPS adds a pruning step, removes the non-frequent, and no longer scans projected database that the number of sequence is less than the minimum support;

2)For certain specific sequential patterns don't generate and scan projected database, directly generate from a sequence database, thereby reducing the size of projected database and the time of scanning.

Theorem1: Suppose sequential database S, $< \alpha \cdots \alpha >$ is a sequential pattern with the length of L-1 $(2 \le L \le n)$, for the projected database w.r.t $< \alpha \cdots \alpha >$, when the support count of $< \alpha >$ is no less than min_sup, the L-sequence pattern $< \alpha \cdots \alpha >$ which is directly generated from projected database S is equivalent with the one which is directly generated from sequential database.

Proof:Use mathematical induction to prove.

When L=2, the support count of $< \alpha >$ is no less than min_sup, mining the projected database with prefix $< \alpha >$ to get the $< \alpha >$ is a sequential pattern, then get the $< \alpha\alpha >$ is a sequential pattern, which is equivalent with $< \alpha\alpha >$ which is directly generated from sequential database. Assume when L=n-1, propositional establishment, we can get sequential pattern $< \alpha \cdots \alpha >$ with n-1, as a prefix, mining the projected database further. The support count of $< \alpha >$ is no less than min_sup, so $< \alpha >$ is a sequential pattern of projected database, then get the $< \alpha \cdots \alpha >$ is a sequential pattern with L=n, which is equivalent with the sequential pattern with L=n which is directly generated from sequential database. Theorem is proved.

Improved algorithm. The Improved PrefixSpan algorithm is described in detail as follows:

Input: A sequential database S and minimum support threshold min_sup

Output: The complete set of sequential patterns

Method: Call $IPS(<>,0,S)$

Subroutine: $IPS(\alpha, L, S|_\alpha)$

Parameters: α: a sequential pattern; L: the length of $<\alpha>$; $S|_\alpha$: the α-projected database, if $\alpha \neq <>$; otherwise, the sequence database S;

1）Scan $S|_\alpha$ once, find the set of frequent items b such that b can be assembled to the last element of α to form a sequential pattern

2）For each sequential pattern b, append it to α to form a sequential pattern α', and output α';

3）For each α', construct α'-projected database, get the set of suffixes about α', if the number of the suffixes in projected database is more than min_sup, then save the projected database, else abandon it. In the set of suffixes, when the support count of α' is no less than min_sup, directly get sequence pattern $\alpha'\alpha'$ by Theorem 1, then construct projected database $S|_\alpha$, for the other sequences except α', call $IPS(\alpha', L+1, S|_\alpha)$.

Analysis of the web access Sequence based on the Improved PrefixSpan

Experimental preparation.

1)Experimental data: The data comes from Internet Information Server (IIS) logs for msnbc.com for the entire day of September, 28, 2013 (Pacific Standard Time).

2)Experimental environment: Operating systems: Windows XP; Database: SQL server 2008, Running tools: VC++6.0; Language: C++; Hardware configuration:Intel(R)Core(TM)i5-3230M; RAM: 8.00G;

Data Preparation. Each sequence in the database corresponds to page views of a user during that twenty-four hour period. Then requests are not recorded at the finest level of detail---that is, at the level of URL, but rather, they are recorded at the level of page category.The categories are "frontpage", "news", "tech", "local", "opinion", "on-air", "misc", "weather", "health", "living", "business", "sports", "summary", "bbs" , "travel", "msn-news", and "msn-sports".Each event in the sequence corresponds to a user's request for a page. Table1 shows the data of user's request.

Then, we turn the user's request data into boolean values, discrete the user IP into 1,2,3,4,5,6,7... each category is associated--in order--with a letter starting with "A". For example, "frontpage" is associated with A, "news" with B, and "tech" with C. Then, we generate Boolean value table, as shown in Table 2.

Table 1.The data of web access

IP	Access sequence
195.112.164.137	on-air,travel,sports,travel,opinion,misc,local,misc,local,misc,local,sports,local
195.100.066.131	frontpage,news,news
195.090.053.041	frontpage,frontpage,frontpage,frontpage,frontpage
.....	
195.090.049.196	local,misc,local,local,health,news,travel,on-air,travel,opinion,travel,opinion,local,local,bbs
195.112.173.024	on-air,on-air
195.090.048.016	news
.....	

Table 2.web access Boolean value table

IP	Access sequence
1	F O L O E G D G D G D L D F
2	A B B
3	A A A A A
.....	
1678	D G A A I B O F O E O E D D N
1679	F F
1680	B
.....	

Analysis of experimental results

Then, Set the minimum support degree: min_sup=6%. According to the improved PrefixSpan algorithm, the frequent sequences is: "A", "AA", "ABC", "B", "BB", "C", "D", "F", "H", "HIJ", "I", "M", "N", "NN". This leads to the conclusion that: the most visited web page in mscn.com is "frontpage", "news", "tech", "local", "on-air", "weather", "health", "summary" and "bbs". Besides, the visitors tend to visit "news" and "tech" after"frontpage", visit "health" and "summary" after "weather".

In order to compare the efficiency of the improved algorithm, using both classical PrefixSpan

algorithm and IPS algorithm to mine frequent sequence. Performance comparison of the two algorithms is shown by Figure 1.

After the time performance analysis and testing, we can get that the Improved PrefixSpan algorithm is more efficient than the classical one. This is because the IPS algorithm avoid produce duplicated projected database with the same prefix pattern through checking the prefix with regard to prefix of the sequence database and abnegating the non-frequent items and projected databases which sequential number is lower than minimum support in the recursive mining process. So it improves the efficiency of frequent item sets and reduces the time complexity.

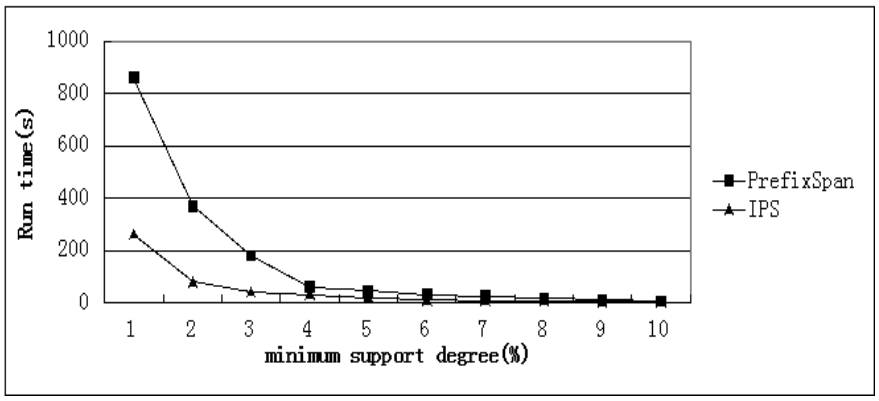

Figure 1.Performance comparison

Acknowledgments

This work was support by the Natural Science Foundation of Jiangsu Provience (BK20130852); by Information metadata application of volunteer computing(61103017);by Lianyungang Technology Plan(cg1215)

References

[1]Soumen Chakrabarti. Mining the Web: Discovering Knowledge from Hypertext Data. Morgan Kaufmann Publishers, 2002:10-16

[2]Agrawal R, Srikant R. Mining sequential patterns[C]. Proceeding of the 11th International Conference on Data Engineering. CA: Los Alamitos, IEEE Computer Society, 1995:3-14

[3]Srikant R, Agrawal R. Mining sequential patterns: generalizations and performances improvements [C]//EDBT 96: Proceedings of the 5th International Conference on Extending Database Technology: Advance in Database Technology. Berlin: Springer-Verlag, 1996:3-17

[4]Zaki M. SPADE: an efficient algorithm for mining frequent sequence[J]. Machine Learning, 2001,42(1):31-60.

[5] Han J, Pei J, Mortazavi-Asl B, et al. FreeSpan: frequent pattern-projected sequential pattern mining [C]//Proceed Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Ming. New York: ACM, 2000: 355-359

[6]Pei J, Han J, Mortazavi-Asl B, et al. Mining sequential patterns by pattern-growth: the prefixspan approach[J]. IEEE Transactions On Knowledge and Date Engineering, 2004, 16(11):1424-1440