Sequential Pattern Mining on Library Transaction Data

Imas Sukaesih Sitanggang
Computer Science Department
Bogor Agricultural University
Bogor, Indonesia
e-mail: imas.sitanggang@ipb.ac.id

Nor Azura Husin
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
43400 Serdang Selangor, Malaysia
email: nazura1112@gmail.com

Anita Agustina
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
43400 Serdang Selangor, Malaysia
email: anita.aja@gmail.com

Naghmeh Mahmoodian
Islamic Azad University
Mashhad Branch, Iran
email: naghmeh.ma@gmail.com

Abstract— **Application of data mining techniques in library data results interesting and useful patterns that can be used to improve services in university libraries. This paper presents results of the work in applying the sequential pattern mining algorithm namely AprioriAll on a library transaction dataset. Frequent sequential patterns containing book sequences borrowed by students are generated for minimum supports 0.3, 0.2, 0.15 and 0.1. These patterns can help library in providing book recommendation to students, conducting book procurement based on readers need, as well as managing books layout.**

Keywords- Sequential Pattern Mining; AprioriAll; Library Transaction Data .

I. INTRODUCTION

Nowadays university libraries have utilized computer-based systems to provide better services to users. Huge numbers of library data are recorded in the systems including book dataset, book transaction, and user profiles. Extracting useful and interesting patterns from large library data is an important task in order to know student's behavior in their book's loan records. Analyzing the transaction of book data can be used to determine the frequently borrowed books so that librarians can decide which books are needed to consider in the book procurement.

Data mining techniques have been widely applied to extract interesting patterns from transaction datasets in many areas such as customer shopping behavior analysis. In this task, we find association rules representing items that are frequently associated or purchased together. Association rule mining has been used to analyze library data. Li and Chen (2008) extracted strong association rules from book reader's loan records using the Apriori algorithm [6]. Zhu and Wang (2007) discovered useful knowledge from library circulation data using the improved association rule algorithm [14].

In addition to association rule mining, sequential pattern mining can be applied to extract new useful and interesting patterns from library data in which time parameter is involved. In sequential pattern mining, we take time stamp into account then find the proper rules, for example for those customers whose purchase computers, they also tend to buy antivirus software within a week. Sequential pattern mining deals with a sequence database consisting of sequences of ordered elements or events. In the case of customer purchase analysis in a supermarket, a record in a sequence database consists of transaction date and items bought in the transaction. Sequential pattern mining algorithms have been applied in many areas such as customer shopping, finding telephone calling patterns, weblog click streams, and DNA sequences and gene structures. Sequential pattern mining from alarm data also can be performed in finding problems in networks and possibly in predicting severe faults [10].

Sequential pattern mining is the mining of frequently occurring ordered events or subsequences as patterns [4]. The sequential pattern mining problem was first introduced by Agrawal and Srikant in 1995 [2]. They proposed two algorithms AprioriSome and AprioriAll to extract sequential patterns from a database of customer transactions. Each transaction in the database consists of the following fields: customer-id, transaction-time, and items purchased in the transaction. All transaction of a customer can together be viewed as a sequence, where each transaction correspond to a set of items, and the list of transactions ordered by increasing transaction-time, corresponds to a sequence. Sequential pattern mining algorithms find all frequent subsequences i.e. subsequences whose occurrence frequency in the dataset with no less than user-specified minimum support threshold.

This work applies the sequential pattern mining algorithm namely AprioriAll proposed by Agrawal and Srikant (1995) [2] on the library transaction dataset to discover the most common book's loan paths by students and from this information a library can recommend one or more books to students after they borrow another book. In addition, a library can arrange books position in which books in a sequence are located on adjacent position such that students can find these books easily.

The paper are organized as follows. Section 2 explains about literature review related to sequential pattern mining focus on the AprioriAll algorithm. Section 3 will discuss

978-1-4244-6716-7/10/$26.00 ©2010 IEEE

about the preprocessing phase for the library dataset. Result and discussion are summarized in Section 4. Conclusion is given in Section 5.

II. Sequential Pattern Mining

In Agrawal et al. (1993) the problem of discovering "what items are bought together in a transaction" over basket data was introduced [1]. The problem of finding what items are bought together from an unordered set of items is concerned with finding intra-transaction patterns. Sequential pattern mining deals with data represented as sequences (a sequence contains ordered sets of items). Sequential patterns indicate the correlation between transactions [13]. The problem of finding sequential patterns is concerned with extracting inter-transaction patterns [2]. In sequential pattern mining, the input data is a set of sequences in which a sequence represents a list of transactions. Each transaction is a set of items.

Many works have been performed in developing sequential pattern mining algorithms. Some algorithms for sequential mining problem are AprioriSome, AprioriAll, DynamicSome [2], Generalized Sequential Pattern (GSP) [9], SPIRIT: Sequential Pattern Mining with Regular Expression Constraints [3], SPADE: An Efficient Algorithm for Mining Frequent Sequences [12], PrefixSpan [7], SLPMiner: An Algorithm for Finding Frequent Sequential Patterns using Length-Decreasing Support Constraint [8], and Closed Sequential Pattern Mining (CloSpan) [11].

There are some important terminologies related to the task as described as follows [5]. Let I = {i_1, i_2, …, i_n} be a set of all **items**. An **itemset** is a subset of items. A **sequence** is an ordered list of itemsets. A sequence s is denoted by $<s_1s_2..s_l>$, where s_j is an itemset. S_j is also called an **element** of the sequence, and denoted as $(x_1x_2…x_m)$, where x_k is an item. Number of instances of items in a sequence is called the **length** of the sequence. A sequence with length l is called an **l-sequence**. A sequence $\alpha = <a_1a_2…a_n>$ is called a **subsequence** of another sequence $\beta = <b_1b_2…b_n>$ and β a **super-sequence** of α, denoted as $\alpha \subseteq \beta$, if there exist integers $1 \leq j_1 < j_2 < … < j_n \leq m$ such that $a_1 \subseteq b_{j1}$, $a_2 \subseteq b_{j2}$, …, $a_n \subseteq b_{jn}$ [5]. An itemset with minimum support is called a **large itemset** or **litemset** [2].

A. AprioriAll

The AprioriAll algorithm was widely used for identifying patterns in customer transactions in the retail industry. Given a database D of customer transactions, the problem of mining sequential patterns is to find the maximal sequences among all sequences that have a certain user specified minimum support. Each maximal sequence represents a sequential pattern. A sequence satisfying the minimum support constraint is called a large sequence. There are five steps in the AprioriAll algorithm: sort phase, litemset phase, transformation phase, sequence phase, and maximal phase [2]. In the sort phase, we sort the database D, with customer-id as the major key and transaction time as the minor key and then the original transaction database is converted into a database of customer sequences. In the litemset phase, the set of all litemsets L is found and the set of all large 1-sequences $\{<l>|l \in L\}$ is simultaneously obtained. The set of litemset is then mapped to a set of contiguous integers. The purpose of the mapping is to reduce the time required to check if a sequence is contained in a customer sequence [2]. In transformation phase, each customer sequence is transformed into an alternative representation. Each transaction is replaced by the set of all litemsets contained in the transaction based on the following conditions [2]:

- If a transaction does not contain any litemset, it is not retained in the transformed sequence.
- If a customer sequence does not contain any litemsets, this sequence is dropped from the transformed database.

In the sequence phase, multiple passes over the data are performed. In each pass, we start with a seed set of large sequences. New potentially large sequences called candidate sequences are generated based on the seed. At the end of the pass, the large candidate sequences are determined and a large candidate becomes the seed for the next pass. All Support for sequences are counted while the algorithm passes the data. The sequence phase involves the Count-all method based on the Apriori algorithm. Figure 1 shows the AprioriAll algorithm [2].

```
L1 = {large 1-sequences};
for (k=2; L_{k-1} ≠ ∅; k++) do
        begin
        C_k = New candidates generated from L_{k-1}
        foreach customer-sequence c in the database do
                Increment the count of all candidates in C_k
                        that are contained in c.
        L_k = Candidate in C_k with minimum support.
        end
Answer = Maximal Sequences in $\bigcup_k L_k$;
```

Fig. 1. AprioriAll Algorithm [2].

Below as the Apriori Candidate Generation applied in the AprioriAll algorithm [2]:

```
insert into C_k
select p.litemset_1, …, p.litemset_{k-1}, q.litemset_{k-1}
from L_{k-1} p, L_{k-1} q,
where p.litemset_1 = q.litemset_1, …,
        p.litemset_{k-2} = q.litemset_{k-2};
```

Fig. 2. Apriori Candidate Generation [2].

In the maximal phase, the maximal sequences among the set of large sequences are generated. In a set of sequences, a sequence s is *maximal* if s is not contained in any other sequence [2].

III. Data Praprocessing

A. Dataset

Each record in the dataset represents book's code that a student borrows in a particular date. The dataset consists of 7799 records and three attributes: transaction ID (Tid), book's code (Item), and transaction date (Date) when a student borrow a book. Tid and book's code are categorical, and Date is represented in date format (dd/mm/yyyy). An example of transaction ID (Tid) is A14103005. A Tid contains 9 characters. The first digit is a character denoting major of a student. This digit is followed by some number representing code of year when a student enrolls to the university and sequence number to identify a student in a particular major. Book's code (Item) is represented in 3 digits, for example 631, 311, 512 and so on. The first digit represents general topic, the second one is more specific than the first, and the last one is more specific than the second one.

B. Data Transformation

A book's code consisting three digits represents a category of book's subject. For examples: 512 for Algebra, 632 for Plant disease, and 664 for Food technology. Each digit in book's code is related to a book's subject. The first two digits represent more general book's subjects compared to code containing three digits. For example, the code 510 (denoted as 51 in the following discussion) is used for Mathematics book, meanwhile the code 512 is used for Algebra books. As we know Algebra is one of area in mathematics. Figure 3 shows the hierarchy of book's code.

It is difficult to find interesting book transaction patterns from the dataset at such raw or lowest level data (level 3 in Figure 3). For example, a transaction in which a student borrows a book Algebra (code: 512) and then she/he borrows two books: macro economic (339) and finance economic science (code: 332) may occur in a small fraction in the dataset. Therefore it may be difficult to be a frequent sequence involving these specific items. For that, we use the second level as a generalization of the third level of book's code in the sequential pattern mining.

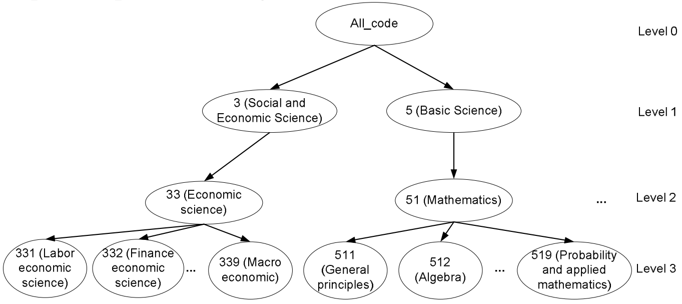

Fig. 3. Hierarchy of book's code.

In sequential pattern mining, all book transaction of a student can together be viewed as a sequence, where each transaction corresponds to a set of book items, and list of book transactions corresponds to a sequence, ordered by increasing transaction-time. To prepare the dataset containing Tid, Date, and lists of book items, we develop a computer program using SAP. The program will concatenate a book item (code) to others if the transactions containing items have the same Tid and date of transaction (Date). Table I represents list of book item sequences. A list of items in brackets, for example (65, 63, 33) represents a sequence of book's code borrowed by a student in the same date. The dataset contains 1037 list of sequences.

TABLE I. LIST OF BOOK'S ITEM SEQUENCES

Tid	List of Book's item
A14103005	63, 63, 63
A14103010	63, 33, (65, 63, 33), 63
A14103013	63
A14103019	33, 65, (33, 65)
A14103023	65, (68, 51)
…	…

We developed a computer program using SAP to find frequent (large) sequences from the library dataset using the AprioriAll algorithm. We performed experiment using some values of minimum support threshold i.e. 0.3, 0.2, 0.15 and 0.1. There are no frequent sequences generated for minimum support above 0.3.

IV. RESULT AND DISCUSSION

Mining sequential patterns using AprioriAll is conducted in five phases: Sort phase, litemset phase, transformation phase, sequence phase, and maximal phase [2]. In the sort phase the dataset is sorted, with Student-id (Tid) as the major key and transaction-time as the minor key in order to create a database of customer sequences. This step has been done in the preprocessing phase. In the litemset phase we find the set of all litemsets L. We also simultaneously determine the set of all large 1-sequences, since this set is just $\{< l >| l \in L\}$. In the transformation phase each transaction is replaced by the set of all litemsets contained in that transaction. If a transaction does not contain any litemset, it is not retained in the transformed sequence. If a sequence does not contain any litemset, this sequence is dropped from the transformed databases. But it still contributes to the count of total number of transactions. The set of litemsets is mapped to a set of contiguous integers in order to reduce the time required in executing the algorithm. Some records in the dataset before and after the transformation phase and after mapping as well are given in Table II. In this table, the original dataset includes infrequent litemsets printed in bold. These infrequent itemsets will be removed to result transformed dataset.

TABLE II. TRANSFORMED DATASET AND DATASET AFTER MAPPING

Original dataset	Transformed dataset	After Mapping
(61,63),(61,63), (61,63), (61,63), 61,(61,63),63	(61,63,(61,63)), (61,63,(61,63)),(61,63, (61,63)),(61,63,(61,63), 61,(61,63,(61,63)), 63	(3,4,10),(3,4,10), (3,4,10), (3,4,10),3,(3,4,10),4
50,53,57,**53,53**	57	2
(61,63),61,61,61,63,63	(61,63,(61,63)),61,61,61,63,63	(3,4,10),3,3,3,4,4
(61,63),(61,63,67),61, 66, (63,66)	(61,63,(61,63)),(61,63), 61,66,(63,66,(63,66))	(3,4,10),(3,4),3,6, (4,6, (4,6))
53,66,66, **(57,65)**	66,66	6,6
(57,63),63,63,63,63,63	(57,63,(57,63)),63,63,63,63, 63	(2,4,9),4,4,4,4,4
66,**(54,61)**	66,61	6,3
67,67,**69**,63,**67**,**(63,67)**	63,63	4,4

The sequence phase applies the AprioriAll algorithm on the mapping dataset. We used some values of minimum support to find *l*-large sequences, *l* = 2, 3, 4. For minimum support

0.3, 0.2, and 0.15, 2-large sequence is 33 63, 33 63, and 33 57 respectively. Large l-sequences, l = 2, 3, and 4, for minimum support 0.1 are given in Table III.

TABLE III. LARGE L-SEQUENCE, FOR L = 2, 3, 4 AND MINIMUM SUPPORT 0.1.

2-large sequence	***3- large sequence***	***4- large sequence***
33 57, 33 65, 33 66, 57 63, 57 66 63 65, 63 66, 57 (57,63), 63 (57,63)	33 57 63 33 63 66 57 63 66 57 63 (57,63)	33 (61,63) 61 63

After we have 1-large sequences, 2-large sequences and so on, we find maximal sequences among the set of large sequences. Maximal sequences for minimum support 0.3, 0.2, and 0.15 are <33 63>, <33 63>, and <33 57> respectively. Maximal sequences for minimum support 0.1 are <33 65>, <63 65>, <33 57 63>, <33 63 66>, <57 63 66>, <57 63 (57,63)>, and <33 (61,63) 61 63>.

This maximal sequences represent sequential pattern of books that frequently borrowed by students. By observing/utilizing this maximal sequences, a university Library can improve its services. Based on above result, books that frequently borrowed by students are 33 (Economic science), 63 (Agriculture), 57 (Life science), 66 (Chemistry Technology), and 61 (Medical Science). Therefore library should increase number of copies for these books to support students in learning subjects related to Economic science, Agriculture, Life science, Chemistry Technology and Medical Science.

The results also show student's behavior in borrowing books in the library. For example, students who borrow the book 33 (Economic science) will also borrow the book 63 (Agriculture) in the following day(s). This pattern is supported by 30% records in the transaction. Another result is that 10% transactions have the maximal sequence <57 63 66>. It means a student who borrow the book 57 (Life science), in the following day(s) she/he will borrow 63 (Agriculture), and followed by 66 (Chemistry Technology) in the next time period. Based on this sequence, library may provide readers who borrow Life science book recommendation to read Agriculture and Chemistry Technology next days. In addition, library may locate Life science, Agriculture and Chemistry Technology books in the same cluster or put close together to provide easy access and convenient for readers in searching these books.

V. CONCLUSION

This paper discusses the application of classical sequential pattern mining algorithm AprioriAll on a library dataset to extract frequent book sequences borrowed by students. One of the results show that students who borrow the book 33 (Economic science) will also borrow the book 63 (Agriculture) in the following day(s) for minimum support 0.3. The patterns can be used by library in order to improve its services to students effectively. Number of copies for books occurred in the frequent sequences can be increased to support students in learning related subjects. Library may also give readers recommendations to read other books after they finish reading a certain book. Based on the book occurrences in frequent sequences, layout of books can be arranged such that readers can find easily the books.

REFERENCES

[1] R. Agrawal, T. Imielinski and A. Swami, "Mining association rules between sets of items in large databases," Proceeding of the ACM SIGMOD Conference on Management of Data, Washington, D.C., May 1993, pp. 207-216.

[2] R. Agrawal and R. Srikant, "Mining sequential patterns," Proceeding of the 11th Int'l conference on Data Enggineering, Taipei, Taiwan, March 1995.

[3] M. N. Garofalakis, R. Rastogi and K. Shim, " SPIRIT: Sequential pattern mining with regular expression constraints," Proceedings of the 25th VLDB Conference Edinburgh, Scotland, 1999.

[4] J. Han and M. Kamber, "Data mining concepts and tchniques," San Diego, USA: Morgan-Kaufmann, 2006.

[5] J. Han, J. Pei, and X. Yan, "Sequential pattern mining by pattern-growth: principles and extensions," 2005.

[6] J. Li and P. Chen, "The application of Association rule in Library system," IEEE, 2008, pp. 248-251.

[7] J. Pei, J. Han, B. Mortazavi-Asl and H. Pinto, "PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth," 2001.

[8] M. Seno and G. Karypis, "SLPMiner: an algorithm for finding frequent sequential patterns using length-decreasing support constraint," 2002.

[9] R. Srikant, and R. Agrawal, "Mining sequential patterns: generalizations and performance improvements," Springer-Verlag Proceeding 5th International Conference, 1996.

[10] P. Wu, W. Peng and M. Chen, "Mining sequential alarm patterns in a telecommunication database," Proceeding of the International Workshop on Databases in Telecommunications (VLDB 2001), Roma, Italy, 10 September 2001.

[11] X. Yan, J. Han, and R. Afshar, "CloSpan: mining closed sequential patterns in large datasets," 2003.

[12] M. J. Zaki, "SPADE: an efficient algorithm for mining frequent sequences," Machine Learning, 0, 1-31 (2000) Kluwer Academic Publishers, Boston, 2000.

[13] Q. Zhao and S. S. Bhowmick, "Sequential pattern mining: a survey," Technical Report, CAIS, Nanyang Technological University, Singapore, No.2003118, 2003.

[14] Z. Zhu and J. Wang, "Book recommendation service by improved association rule mining algorithm," IEEE The Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August 2007, pp. 3864-3869.