Prediction of Yahoo! Music Sequences on user's musical taste

Dr. Sunita Mahajan[1], Alpa Reshamwala[2], Nisha Sharma[3], Divya Vineet[4], Akshay Sharma[5], Parshwa Shah[6]
[1]Principal, Institute of Computer Science, M.E.T, Bandra, Mumbai, India
[2]Assistant Professor, [3,4,5,6]Student, Computer Engineering Department
MPSTME, SVKM's NMIMS University, Mumbai, India

***Abstract*—Sequential pattern mining is an important data mining problem with broad applications. In this paper, we have implemented Apriori a candidate generation algorithm and SPAM (Sequential Pattern Mining) algorithm on Yahoo! Music KDD Cup 2011, which is the annual Data Mining and Knowledge Discovery competition organized by ACM Special Interest Group on Knowledge Discovery and Data Mining, the leading professional organization of data miners.Yahoo! Music has amassed billions of user ratings for musical pieces. When properly analyzed, the raw ratings encode information on how the popularity of songs, albums and artists vary over time and above all, which songs users would like to listen to. Such an analysis introduces new scientific challenges. From these discovered patterns, we can know what patterns or music sequences which are frequently heard and in what order they are recommended. Experimental results have shown that SPAM performs well for large datasets like Yahoo! Music datasetis due to the bitmap representation of the data for efficient counting.**

***Keywords*—Data mining, sequence data, time interval, Yahoo! Music, User music tastes.**

I. INTRODUCTION

Data mining extracts implicit, previously unknown and potentially useful information from databases. The discovered information and knowledge are useful for various applications, including market analysis, decision support, music recommendation, fraud detection, intrusion detection and business management. Many approaches have been proposed to extract information, and mining sequential patterns is one of the most important ones [1][2][3].

People have been fascinated by music since the dawn of humanity. A wide variety of music genres and styles has evolved, reflecting diversity in personalities, cultures and age groups. Yahoo! Music has amassed billions of user ratings for musical pieces. When properly analysed, the raw ratings encode information on how songs are grouped, which hidden patterns link various albums, which artists complement each other, how the popularity of songs, albums and artists vary over time and above all, which songs users would like to listen to. Users could rate songs, artists, albums and even genres on a 5 star system, or using a slider interface. These ratings were used by Yahoo! Music to generate recommendations that match the user's taste, based on either the taxonomy of items or on recommendations of other users with similar musical tastes. Such an analysis introduces new scientific challenges. An example of such a pattern can be listening to Classical music followed by Jazz music and then Rock / Hip Hop. From these discovered sequential patterns, we can know what patterns or music sequences are frequently heard and in what order they are recommended.

It has been a great challenge to improve the efficiency of Apriori algorithm. Since all the frequent sequential patterns are included in the maximum frequent sequential patterns, the task of mining frequent sequential patterns can be converted as mining maximum frequent sequential patterns. SPAM [4] algorithm is based on the lexicographic sequence tree and can make either depth or width first traversal.

In this paper, both these algorithmsare implemented to mine frequent sequential patterns on KDD Cup 2011 track 1 dataset of Yahoo! Music sequences to predict sequence of users rating of musical items with similar musical tastes.Items can be tracks, albums, artists and genres. Items form a hierarchy, such that each track belongs to an album, albums belong to artists, and together they are tagged by genres.

I. RELATED WORK

The problem of mining sequential patterns was first introduced by Agarwal and Srikant [1] which discovers patterns that occur frequently in a sequence database. A sequence database is formed by a set of data sequences. Each data sequence includes a series of transactions, ordered by transaction times. After mid 1990's, following Agrawal and Srikant [1], many scholar provided more efficient algorithms[5][6][7][8]. Besides these, works have been done to extend the mining of sequential patterns to other time-related patterns.

Existing approaches to find appropriate sequential patterns in time related data are mainly classified into two approaches. In the first approach developed by Agarwal and Srikant [9], the algorithm extends the well-known Apriori algorithm. This type of algorithms is based on the characteristic of Apriori—that any subpattern of a frequent pattern is also frequent [1]. The later, uses a pattern growth approach [5], employs the same idea used by the Prefix-Span algorithm.

This algorithm divides the original database into smaller subdatabases and solve them recursively. Previous research addresses time intervals in two typical ways, first by the time-window approach, and second by completely ignoring the time interval. First, the time window approach requires the length of the time window to be specified in advance. A sequential pattern mined from the database is thus a sequence of windows, each of which includes a set of patterns. Patterns in the same time window are bought in the same time period. In the algorithm [7], Shrikant and Agrawal, specified the maximum interval (max-interval), the minimum interval (min-interval) and the sliding time window size (windowsize).

Moreover, they cannot find a pattern whose interval between any two sequences is not in the range of the window-size. Agrawal and Srikant[1], introduced mining traditional sequential mining, by ignoring the time interval and including only the temporal order of the patterns.

To address the intervals between successive patterns in sequence database, Chen *et al.* have proposed a generalization of sequential patterns, called time-interval sequential patterns, which reveals not only the order of patterns, but also the time intervals between successive patterns [10]. Chen *et al.* developed algorithms tofind sequential patterns using both the approaches [10]. Their work, by assuming the partition of time interval as fixed, developed two efficient algorithms -I-Apriori and I- PrefixSpan. The first algorithm is based on the conventional Apriori algorithm, while the second one is based on the PrefixSpan algorithm. An extension of the algorithm developed by Chen *et al* [10], to solve the problem of sharp boundaries to provide a smooth transition between members and non-members of a set, is addressed in Chen *et al* [11]. The sharp boundary problems can be solved by the concept of fuzzy sets. Two efficient algorithms, the FTI-Apriori algorithm and the FTI-PrefixSpan algorithm, were developed for mining FTI sequential patterns. There are several other reasons that support the use of FTI in place of crisp interval. First, the human knowledge can be easily represented by fuzzy logic. Second, it is widely recognized that many real world situations are intrinsically fuzzy, and the partition of time interval is one of them. Third, FTI is simple and easy for users.

In contribution to the ongoing research on FTI sequential pattern mining, Mahajan and Reshamwala proposes an algorithm to detect and classify audit sequential patterns in network traffic data [13]. The paper also defines the confidence of the FTI audit sequences, which is not yet defined in the previous researches. The authors have also proposed an algorithm which uses a fuzzy genetic approach to discover optimized sequences in thenetwork traffic data to classify and detect intrusion [14].

This paper focuses on mining KDDCup 2011 track 1 dataset of Yahoo! Music sequences, using FTI sequential pattern to detect music recommendation with similar music tastes.

II. KDD 2011 DATASET

Yahoo! Music offers a wealth of information and services related to many aspects of music. We have compiled a dataset of user ratings of music items collected during a decade of using the Yahoo! Music website. The dataset was released within the first track of the KDD Cup 2011 contest [12]. It comprises of 262,810,175 ratings of 624,961 items by 1,000,990 users. The ratings include date and one-minute resolution timestamps, allowing refined temporal analysis. Each item and each user has at least 20 ratings in the whole dataset. The available ratings were split into train, validation and test sets such that the last 6 ratings of each user were placed in the test set and the preceding 4 ratings were used in the validation set.

A distinctive feature of this dataset is that user ratings are given to entities of four different types: *tracks, albums, artists, and genres.* The majority of items (81.15%) are tracks, followed by albums (14.23%), artists (4.46%) and genres (0.16%). The ratings however, are not uniformly distributed: Only 46.85% of the ratings belong to tracks, followed by 28.84% to artists, 19.01% to albums and 5.3% to genres. Moreover, these proportions are strongly dependent on the number of ratings a user has entered. All rated items are tied together within a taxonomy. That is, for a track we know the identity of its album, performing artist and associated genres. Similarly we have artist and genre annotation for the albums. There is no genre information for artists, as artists may switch between many genres in their career.

A. *Track 1 dataset*

For each user, user rating data is grouped by user.

<UsedId>|<#UserRatings>\n

Each of the next <#UserRatings> lines describes a single rating by <UsedId>, sorted in chronological order. Rating line format is:

<ItemId>\t<Score>\t<Date>\t<Time>\n

UsedId & ItemId: Consecutive integer, both starting at zero

Score: Integer between 0 and 100

Date: Integer describing number of days elapsed since an undisclosed date

Example:

0|2

507696 90 5103 07:34:00

137915 90 5103 07:39:00

B. *Track 2 dataset*

For each user, user rating data is grouped by user.

<UsedId>|<#UserRatings>\n

Each of the next <#UserRatings> lines describes a single rating by <UsedId>. Rating line format is:

<ItemId>\t<Score>\n

UsedId & ItemId: Consecutive integer, both starting at zero

Score: Integer between 0 and 100

Example:

0|2

28341 90

51210 90

C. *Item Taxonomy*

A unique feature of the datasets is a taxonomy annotating known relations between the items. Such taxonomy is expected to be particularly useful here, due to the large number of items

and the sparseness of data per item (mostly attributed to "tracks" rather than to "artists").

Recall that item id's can represent tracks, albums, artists or genres. The type of each item, including a hierarchical structure linking tracks, albums, artists and genres, is stored (separately for each the two datasets) in the following four files:

Track information formatted as:

<TrackId>|<AlbumId>|<ArtistId>|<Optional GenreId_1>|...|<Optional GenreId_k>\n

Album information formatted as:

<AlbumId>|<ArtistId>|<Optional GenreId_1>|...|<Optional GenreId_k>\n

Artist listing formatted as:

<ArtistId>\n

Genre listing formatted as:

<GenreId>\n

D. Sequence dataset

A sequence is an ordered list of items [1]. All the ratings of a user can together be viewed as a sequence, where each musical rating at a given time t corresponds to a set of musical items ordered by rating date and time. We call such a sequence as a user musical items rating sequence.

For example, consider the dataset shown in table I, Table II shows the dataset expressed as a set of user's item sequential ratings.

Given a database D of user's ratings on musical items, the problem of mining sequential patterns is to find the maximum sequences among all sequences that satisfy the user specified minimum support. Each such maximum sequence represents a sequential pattern.

TABLE I
TRACK 1DATASET

UserId	ItemId	Score	Date	Time
3	5980	90	3811	13:24:00
3	11059	90	3811	13:24:00
4	239721	70	4502	14:18:00
4	279399	70	4503	3:47:00
4	304395	70	4503	3:54:00
4	434567	70	4503	4:19:00

TABLE II
TRACK 1SEQUENCE DATASET

UserId	User Sequence
3	(5980 11059)
4	((239721) (279399304395434567))

III. ALGORITHMS

KDDCup 2011 track 1 dataset of Yahoo! Music, user's ratings sequenceswith similar music tastesare predicted using,the algorithms: AprioriAll and SPAM, which find sequential patterns without time intervals using traditional approach.

AprioriAll[1] is based on Apriori algorithm, in each pass we use the large sequences from the previous pass to generate the candidate sequences and then measure their support by making a pass over the database.

SPAM [4] algorithm is based on the lexicographic sequence tree and can make either depth or width first traversal. Taking current frequent subsequent node $n=(s1\rightarrow\ldots\rightarrow sk)$, s-step extension candidate items list $S_n$and i-step extension candidate items list $I_n$as input parameters, SPAM generates a new sub-sequence by either s-step or i-step. And this process recursively goes on until no more extension can be performed. Hence, this algorithm also implements a pruning method using a bitmap representation to store each sequence.

IV. RESULTS AND DISCUSSION

In this section we perform a simulation study to compare the performances of the algorithms: the Apriori [1] and SPAM[4], which find sequential patterns without time intervals.
These algorithms were implemented by Sun Java language and tested on a Intel Core Duo Processor, 2.10 GHz with 2GB main memory under Windows XP operating system.

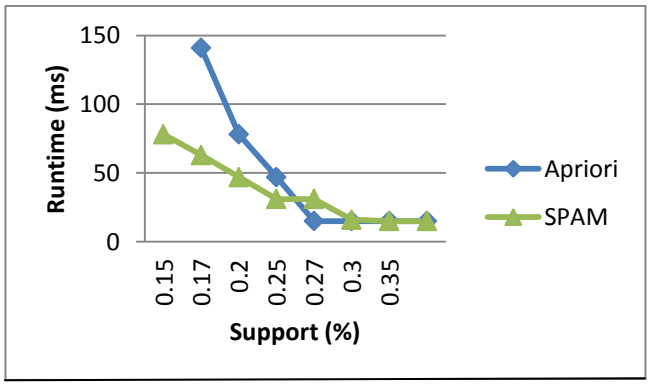

Fig. 1 Execution Times

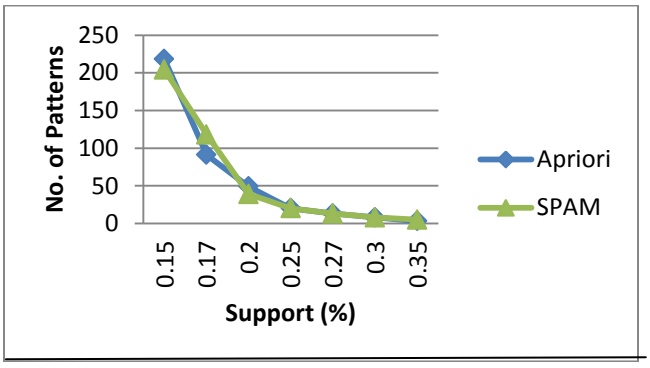

Fig. 2 No. of patterns verses pattern type

The comparison is carried out on the sequence dataset of KDD cup 2011track 1 training data. The dataset comprises 262,810,175 ratings of 624,961 music items by 1,000,990 users collected during 1999-2010.

The first comparison executed based on the dataset where the minimum support threshold is varied 15 % to 35%. Figure 1 summarizes those results. The above results show that SPAM is faster than Apriori.SPAM performs well for large datasets like Yahoo! Music is due to the bitmap representation of the data for efficient counting [4].

The second comparison is done on the number of frequent sequence patterns found executing these algorithms with the varying minimum support threshold. From the results in figure 2, it is shown that Apriori gives the maximum number of all sequential patterns due to its property of counting non-maximal sequences.

V. Conclusions and Future Work

In this paper we have implemented Apriori a candidate generation algorithm and SPAMalgorithm on Yahoo! Music to predict user's ratings on music recommendation with similar music tastes. Experimental results have shown that SPAM performs well for large datasets like Yahoo! Musicdatasetis due to the bitmap representation of the data for efficient counting and implementing a pruning method using a bitmap representation to store each sequence [4].

The second comparison is done on the number of frequent sequence patterns found executing these algorithms with the varying minimum support threshold and from the results, it is shown that Apriori gives the maximum number of all sequential patterns due to its property of counting non-maximal sequences.

In future work, as in these experiments we have found sequence patterns, by ignoring the time interval and including only the temporal order of the patterns.To address the intervals between successive patterns in sequence database, Chen *et al.* have proposed a generalization of sequential patterns, called time-interval sequential patterns, which reveals not only the order of patterns, but also the time intervals between successive patterns [10]. An extension of the algorithm developed by Chen *et al* [10], to solve the problem of sharp boundaries to provide a smooth transition between members and non-members of a set is addressed in Chen *et al* [11] can also be implemented.

In contribution to the ongoing research on FTI sequential pattern mining, Mahajan and Reshamwala proposes an algorithm to detect and classify audit sequential patterns in network traffic data [13]. The paper also defines the confidence of the FTI audit sequences, which is not yet defined in the previous researches. The authors have also proposed an algorithm which uses a fuzzy genetic approach to discover optimized sequences in thenetwork traffic data to classify and detect intrusion [14].

References

[1] R. Agrawal and R. Srikant: Mining sequential patterns. In Proc. Int. Conf. Data Engineering, pp. 3–14(1995)

[2] Y. L. Chen, S. S. Chen, and P. Y. Hsu: Mining hybrid sequential patterns and sequential rules. Inf. Syst., vol. 27, no. 5, pp. 345–362 (2002)

[3] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. New York: Academic, (2001)

[4] J. Ayres, J. Gehrke, T. Yiu, and J. Flannick,Sequential PAttern Mining using A Bitmap Representation, In Proceedings of ACM SIGKDD on Knowledge discovery and data mining, pp. 429-435, 2002.

[5] Pei, J., Han, J., Pinto, H., Chen, Q., Dayal, U., & Hsu, M.-C. : PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. Proceedings of 2001 International Conference on Data Engineering, pp. 215–224 (2001)

[6] Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., & Hsu, M.-C. : FreeSpan: Frequent pattern-projected sequential pattern mining. Proceedings of 2000 International Conference on Knowledge Discovery and Data Mining, pp. 355–359 (2000)

[7] Srikant, R., & Agrawal, R.: Mining sequential patterns: Generalizations and performance improvements. Proceedings of the 5th International Conference on Extending Database Technology, pp. 3–17 (1996)

[8] Zaki, M. J.: SPADE: An efficient algorithm for mining frequent sequences. Machine Learning Journal, 42(1/2), 31–60 (2001)

[9] R. Agrawal and R. Srikant: Fast algorithms for mining association rules, in *Proc. Int. Conf. Very Large Data Bases*, pp. 487–499(1994)

[10] Y. L. Chen, M. C. Chiang, and M. T. Ko: Discovering time-interval sequential patterns in sequence databases, *Expert Syst. Applicat.*, vol.25, no. 3, pp. 343–354(2003)

[11] Yen-Liang, Tony Cheng-Kui Huang: Discovering Fuzzy Time-Interval Sequential Patterns in Sequence Databases, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, vol.35, pp.959-972(2005)

[12] G. Dror, N. Koenigstein, and Y. Koren. Yahoo! Music recommendations: Modeling music ratings with temporal dynamics and item taxonomy. In Proc. 5thACM Conference on Recommender Systems (RecSys'11), 2011.

[13] Sunita Mahajan and Alpa Reshamwala, "Amalgamation of IDS Classification with Fuzzy techniques for Sequential pattern mining ",*IJCA Proceedings on International Conference on Technology Systems and Management (ICTSM)* (3):9–14, 2011.

[14] Sunita Mahajan and Alpa Reshamwala, "An Approach to Optimize Fuzzy Time-Interval Sequential Patterns Using Multi-objective Genetic Algorithm", ICTSM 2011, CCIS 145, pp. 115–120, 2011, Springer-Verlag Berlin Heidelberg 2011.