IJCSET – Volume 3, Issue 2 – February 2017. ISSN: 2455-9091 Pages: 9-15

SEQUENTIAL DATA MINING FOR BUSINESS STATISTIC ANALYSIS

Asst.Prof. [1] S. Siamaladevi, UG Students: [2] M. Kavinkumar, [3] K. Harirajan, [4] I. Chanthirahari,
[1, 2, 3, 4] Department of Computer Science and Engineering,
[1, 2, 3, 4] Sri Krishna College of Technology,
[1, 2, 3, 4] Coimbatore, Tamil Nadu, India.

ABSTRACT: Data mining involves the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data sets. A sequence database is a set of ordered elements or events, stored with or without a concrete notion of time. Each itemset contains a set of items which include the same transaction-time value. While association rules indicate intra-transaction relationships, sequential patterns represent the correlation between transactions. Sequential pattern mining (SPM) is the process that extracts certain sequential patterns whose support exceeds a predefined minimal support threshold. In this paper we propose to identifying the right pattern granularity for both sequential pattern mining and modeling, we have successful applications in B2B (Business-to-Business) marketing analytics, healthcare operation and management, and modeling of the product adoption in digit markets, as three case studies in dynamic business environments. The advantage of proposed algorithm is that it dosen't need to generate conditional pattern bases and sub- conditional pattern tree recursively. And the results of the experiments show that it works faster than previous algorithms.

Keywords: [Sequential Pattern Mining, KNN, Aprior algorithm.]

1. INTRODUCTION

Sequential pattern analysis targets on finding statistically relevant temporal structures where the values are delivered in sequences. This is a fundamental problem in data mining with diversified applications in many science and business fields, such as multimedia analysis (motion gesture/video sequence recognition), marketing analytics (buying path identification), and financial modelling (trend of stock prices). Given the overwhelming scale and the dynamic nature of the sequential data, new techniques for sequential pattern analysis are required to derive competitive advantages and unlock the power of the big data. In this dissertation, we develop novel approaches for sequential pattern analysis with applications in dynamic business environments. In particular, sequential pattern modelling infers a statistical model with a set of parameters, with which the model is able to simulate the modelled processes without breaking statistically significant characteristics. Hence, sequential pattern modeling provides parsimonious descriptions for the sequential data and the underlying complex dynamics hidden in the data. With the sequential pattern modeling techniques, the dynamics can be proactively monitored, quantitatively audited, and intuitively inspected.

Frequent pattern mining is a rather broad area of research, and it relates to a wide variety of topics at least from an application specific-perspective. Broadly speaking, the research in the area falls in one of four different categories:

• **Technique-centered:** This area relates to the determination of more efficient algorithms for frequent pattern mining. A wide variety of algorithms have been proposed in this context that use different enumeration tree exploration strategies, and different data representation methods. In addition, numerous variations such as the determination of compressed patterns of great interest to researchers in data mining.

• **Scalability issues:** The scalability issues in frequent pattern mining are very significant. When the data arrives in the form of a stream, multi-pass methods can no longer be used. When the data is distributed or very large, then parallel or big-data frameworks must be used. These scenarios necessitate different types of algorithms.

• **Advanced data types:** Numerous variations of frequent pattern mining have been proposed for advanced data types. These variations have been utilized in a wide variety of tasks. In addition, different data domains such as graph data, tree structured data, and streaming data often require specialized algorithms for frequent pattern mining. Issues of interestingness of the patterns are also quite relevant in this context.

• **Applications:** Frequent pattern mining have numerous applications to other major data mining problems, Web applications, software bug analysis, and chemical and biological applications. A significant amount of research has been devoted to applications because these are particularly important in the context of frequent pattern mining.

Our approach to identifying the meaningful granularity level for sequential pattern mining, the key idea is to summarize the temporal correlations in an undirected graph. Then, the "skeleton" of the graph serves as a higher granularity level on which hidden temporal patterns are more likely to be identified. In the meantime, the manifold embedding of the graph topology allows us to translate the rich temporal content into a metric space. This opens up new possibilities to explore, quantify, and visualize sequential data. Furthermore, by extending the robust temporal correlations, our approach can be utilized to model the dynamic systems which are generally measured by multivariate time series. Evaluation on a Business-to-Business (B2B) marketing application demonstrates that our approach can effectively discover critical purchase patterns from noisy customer behavior records. Indeed, our work will not only provide new opportunities to improve the marketing practice but also further the research of marketing science. For example, since we can identify dynamic buying stages of customers, we can improve the traditional static customer segmentation practice with dynamic extensions, which allows us to target each customer with the marketing campaigns most relevant to his/her current buying stage.

2. LITERATURE REVIEW

Computational complexity of finding frequent sequential patterns is huge for large symbol sets. Specific sequential pattern decreases significantly with the growing cardinality. Semantically meaningful patterns can exist at a higher granularity level, therefore pattern mining on the original. The grouping in these methods is performed irrespective of the temporal content and may fail to find statistically relevant temporal structures in sequential data.

Mining Sequential Patterns [1] Present three algorithms to solve this problem, and empirically evaluate their performance using synthetic data. Two of the proposed algorithms, AprioriSome and Apriori All, have comparable performance, albeit AprioriSome performs a little better when the minimum number of customers that must support a sequential pattern is low. Sequential Pattern Mining using A Bitmap Representation [2] An efficient algorithm

called SPAM (Sequential PAttern Mining) that integrates a variety of old and new algorithmic contributions into a practical algorithm. SPAM assumes that the entire database (and all data structures used for the algorithm) completely fit into main memory. Mean Shift, Mode Seeking, and Clustering [3] Cluster analysis is treated as a deterministic problem of finding a fixed point of mean shift that characterizes the data. Mean shift is also considered as an evolutionary strategy that performs multistart global optimization. Trajectory Pattern Mining [4] A novel form of spatio-temporal pattern, which formalizes the mentioned idea of aggregate movement behaviour.PrefixSpan: Mining Sequential Patterns by Prefix-Projected Pattern [5]The high promise of the pattern-growth approach may lead to its further extension toward efficient mining of other kinds of frequent patterns, such as frequent substructures. Much work is needed to explore new applications of frequent pattern mining. Frequent pattern mining: current status and future directions [6] The FP-growth algorithm transforms the problem of finding long frequent patterns to searching for shorter ones recursively and then concatenating the suffix. On Spectral Clustering: Analysis and an algorithm [7] A simple spectral clustering algorithm that can be implemented using a few lines. Using tools from matrix perturbation theory, we analyze the algorithm, and give conditions under which it can be expected. Adaptive Signal Processing Laboratory (ASPL) Electrical and Computer Engineering Department University of Florida [8] Identification of the clustering algorithm that can best discover inherent structure in real-world Extremely low frequency (ELF) data that is subjected to interference. A consensus about which algorithms are best suited to the present application is arrived at by comparing the performance of the different methods using these indices.

3. PROPOSED WORK

Improve the temporal skeletonization framework. To Predicting the behaviour of customers is challenging, but important for service oriented businesses. Data mining techniques are used to make such predictions, typically using only recent static data.We propose a frequent pattern mining algorithm based on procedure in the embedding space of the temporal graph. More importantly, we provide principled guidance on selecting the important parameters, such as the temporal order parameter, the Aprior algorithm and KNNare used to find out the frequent path in sequential dataset.

Study validates the effectiveness of the temporal skeletonization and the comparison demonstrates that our approach can effectively identify critical buying paths from noisy marketing data.

Proposed method for finding interesting buying paths from real-world B2B marketing data

Identify the right granularity for sequential pattern analysis.

Variety of difficulties in mining sequential patterns from massive data represented by a huge set of symbolic features.

Reduces the representation of the sequential data by uncovering significant, hidden temporal structures

3.1 Dataset Collection and preprocessing

We have collected huge amount of purchase event data for the customers of a big company. We normalize the original location traces for work- flow modeling. Specifically, we project each raw coordinate to a semantic location of the building, such as a room in the hospital, based on the floor maps of the building. This data preprocessing drastically reduces the computational cost, since we significantly reduce the number of records in the data after the projection. Also, this preprocessing step greatly smooths out the noise and alleviates the impact of errors on the workflow modeling tasks.

3.2 Prediction of buying stages

We plot the embedding of selected 503 events, and mark it with the clustering results. Each detected cluster, we extract dominant semantic keywords for the events in that cluster Semantic information is only used to summarize each temporal cluster for better understanding of our results. Temporal clusters could be partially consistent with attribute-based clusters, while meanwhile revealing more fine-grained structure by exploiting the temporal correlations. This is where the extra value comes from.

3.3 Findout Critical Buying Paths

Detected temporal clusters, we can transform the original event sequences to sequences of temporal clusters, and apply the KNN algorithms on the skeletonized sequences. Nearest Neighbor (KNN from now on) is one of those algorithms that are very simple to understand but works incredibly well in practice. Also it is surprisingly versatile and its applications range from vision to proteins to computational geometry to graphs and so on. Most people learn the algorithm and do not use it much which is a pity as a clever use of KNN can make things very simple.

3.4 Frequent Items mining

We apply the Apriori algorithms on the raw sequences and report for the patterns identification. With a very small support threshold, the maximum pattern length is only three, among which the top 10 with the largest support are listed. However, the resultant patterns mainly represent simple action sequences well expected by common sense.

Note that, there are thousands of patterns returned, which are difficult to be investigated individually.

4. METHODOLOGY

The first algorithm we shall investigate is the k-nearest neighbor algorithm, which is most often used for classification, although it can also be used for estimation and prediction. k-Nearest neighbor is an example of instance-based learning, in which the training data set is stored, so that a classification for a new unclassified record may be found simply by comparing it to the most similar records in the training set.

We have seen above how, for a new record, the k-nearest neighbor algorithm assigns the classification of the most similar record or records. A distance metric or distance function is a real-valued function d, such that for any coordinates x, y, and z:

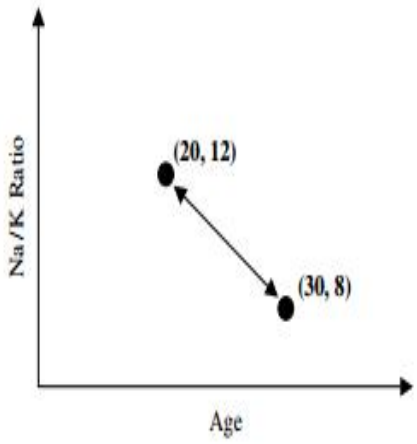
1. $d(x,y) \geq 0$, and $d(x,y) = 0$ if and only if $x = y$
2. $d(x,y) = d(y,x)$
3. $d(x,z) \leq d(x,y) + d(y,z)$

The most common distance function is Euclidean distance, which represents the usual manner in which humans think of distance in the real world:

$$d_{\text{Euclidean}}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

where x = x1, x2, . . . , xm, and y = y1, y2, . . . , ym represent the m attribute values of two records. For example, suppose that patient A is x1 = 20 years old and has a Na/K ratio of x2 = 12, while patient B is y1 = 30 years old and has a Na/K ratio of y2 = 8. Then the Euclidean distance between these points.

$$d_{\text{Euclidean}}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2} = \sqrt{(20-30)^2 + (12-8)^2}$$
$$= \sqrt{100 + 16} = 10.77$$



When measuring distance, however, certain attributes that have large values, such as income, can overwhelm the influence of other attributes which are measured on a smaller scale, such as years of service. To avoid this, the data analyst should make sure to normalize the attribute values. For continuous variables, the min–max normalization or Z-score standardization,

Min–max normalization

$$X^* = \frac{X - \min(X)}{\text{range(X)}} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Z-score standardization:

$$X^* = \frac{X - \text{mean}(X)}{\text{SD}(X)}$$

For categorical variables, the Euclidean distance metric is not appropriate. Instead, we may define a function, "different from," used to compare the ith attribute values of a pair of records, as follows:

$$\text{different}(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{otherwise} \end{cases}$$

where xi and yi are categorical values. We may then substitute different (x, i yi) for the ith term in the Euclidean distance metric.

For instance-based learning methods such as the k-nearest neighbor algorithm, it is vitally important to have access to a rich database full of as many different combinations of attribute values as possible. It is especially important that rare classifications be represented sufficiently, so that the algorithm does not only predict common classifications. Therefore, the data set would need to be balanced, with a sufficiently large percentage of the less common classifications. One method to perform balancing is to reduce the proportion of records with more common classifications.

AprioriAlgorithm(Frequent Mining)

The Apriori algorithm is the most well known association rule algorithm and is used in most commercial products. The use of support for pruning candidate itemsets is guided by the following principles.

Property 1: If an itemset is sequential, then all of its subsets must also be sequential.

Property 2: If an itemset is insequential, then all of its supersets must also be insequential.

The algorithm initially scans the database to count the support of each item. Upon completion of this step, the set of all sequential 1-itemsets, F1, will be known. Next, the algorithm will iteratively generate new candidate kitemsets using the sequential (k-1)-itemsets found in the previous iteration. Candidate generation is implemented using a function called Apriori-gen.

5. EXPERIMENTAL RESULT AND DISCUSSION

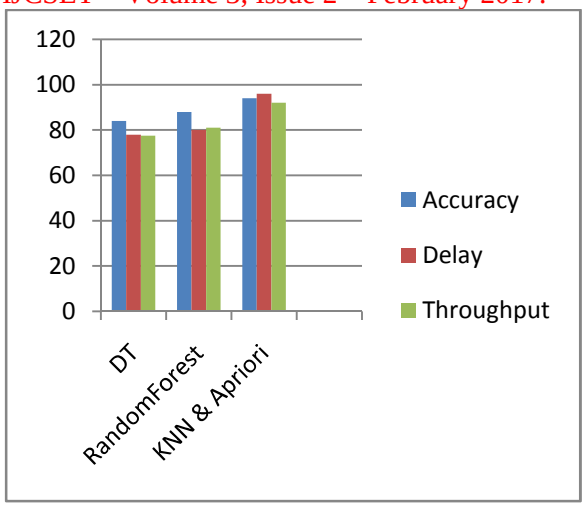
In order to verify the performance of the proposed algorithm, we compare it with Apriori algorithm. These algorithms are performed on a computer with a 2.00GHz processor and 512MB memory, running windows vista. The program is developed by Java with Mysql. We present experimental results using the database. The experimental result is showed in Figures. As shown in the Figure, proposed algorithm is more super than apriori ,because it dosen't need to generate 2-candidate itemsets and reduce the search space, and proposed algorithm dosen't need to much extra spaces on the mining process, so proposed algorithm has a better space scalability

Compared to existing algorithms our performance is increased. The below tables represent the accurate values of current process and existing values.

Technique	Accuracy	Delay	Throughput
Decision Tree	84%	78%	77.5%
Random Forest	88%	80%	81%
KNN &Apriori	94%	96%	92%

Table 1 : Performance Table

The accuracy rate obtained by applying the classification algorithms on the data sets.



The individual accuracy rates obtained from different feature selection methods on the classifier. Different feature selection metrics are applied on the classifier.

CONCLUSION

In this paper, a algorithm is proposed which combined Apriori algorithm and the KNN. The experimental results shows that this new algorithm works much faster than Apriori. The future work is to optimize the technique for counting the support of the candidates and expand it for mining more larger database. In the future, to cope with different applications, it is interesting to generalize these works with an unified probabilistic framework which simultaneously identifies the pattern granularity levels and estimates the model parameters

REFERENCE

[1] R. Agrawal and R. Srikant, "Mining sequential patterns," in Proc. Int. Conf. Data Eng., 1995, pp. 3–14.

[2] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu, "Sequential pattern mining using a bitmap representation," in Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2002, pp. 429–435.

[3] M. Belkin and P. Niyogi, "Laplacianeigenmaps and spectral techniques for embedding and clustering," in Proc. Adv. Neural Inf. Process. Syst., 2001, vol. 14, pp. 585–591.

[4] Y. Cheng, "Mean shift, mode seeking, and clustering," IEEE Trans. Pattern Anal. Mach. Intell., vol. 17, no. 8, pp. 790–799, Aug. 1995.

[5] P. Demartines and J. H_erault, "Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets," IEEE Trans. Neural Netw., vol. 8, no. 1, pp. 148–154, Jan. 1997.

[6] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2007, pp. 330–339.

[7] J. Han and Y. Fu, "Mining multiple-level association rules in large databases," IEEE Trans. Knowl. Data Eng., vol. 11, no. 5, pp. 798– 805, Sep./Oct. 1999.

[8] J. Han, J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth," in Proc. 17th Int. conf. Data Eng., 2001, pp. 215–224.

[9] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: Current status and future directions," Data Mining Knowl. Discovery, vol. 15, no. 1, pp. 55–86, 2007.

[10] B. K. Joseph, "Nonmetric multidimensional scaling: A numerical method," Psychometrika, vol. 29, no. 2, pp. 115–129, 1964.

[11] J.-G. Lee, J. Han, X. Li, and H. Cheng, "Mining discriminative patterns for classifying trajectories on road networks," IEEE Trans. Knowl. Data Eng., vol. 23, no. 5, pp. 713–726, May 2011.

[12] C. Liu. Demo: Temporal Skeletonization [Online].Available:http://liuchuanren.xminer.org/publications/S2Net.htm, 2015.

[13] C. Liu, K. Zhang, H. Xiong, G. Jiang, and Q. Yang, "Temporal skeletonization on sequential data: Patterns, categorization, and visualization," in Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2014, pp. 1336–1345.

[14] Y. N. Andrew, I. J. Michael, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in Proc. Adv. Neural Inf. Process. Syst., 2002, vol. 2, pp. 849–856.

[15] M. O. Lane, E. A. Les, and D. B. Gary, “Automatic clustering of vector time-series for manufacturing machine monitoring,” in Proc. Int. Conf. Acoust., Speech Signal Process., 1997, vol. 4, pp. 3393–3396.
[16] J. Pei, G. Dong, W. Zou, and J. Han, “On computing condensed frequent pattern bases,” in Proc. Int. Conf. Data Mining, 2002, pp. 378–385.
[17] K. Zhang, Q. Wang, Z. Chen, I. Marsic, V. Kumar, G. Jiang, and J. Zhang, “From categorical to numerical: Multiple transitive distance learning and embedding,” in Proc. SIAM Int. Conf. Data Mining, 2015, pp. 46–54.