Mining Quantitative Maximal Hyperclique Patterns: A Summary of Results

Yaochun Huang[1], Hui Xiong[2], Weili Wu[1], and Sam Y. Sung[3]

[1] Computer Science Department, University of Texas - Dallas, USA,
{yxh038100,wxw020100}@utdallas.edu
[2] MSIS Department, Rutgers University, USA
hui@rbs.rutgers.edu
[3] Dept. of Computer Science, South Texas College, USA
sysung@southtexascollege.edu

Abstract. Hyperclique patterns are groups of objects which are strongly related to each other. Indeed, the objects in a hyperclique pattern have a guaranteed level of global pairwise similarity to one another as measured by uncentered Pearson's correlation coefficient. Recent literature has provided the approach to discovering hyperclique patterns over data sets with binary attributes. In this paper, we introduce algorithms for mining maximal hyperclique patterns in large data sets containing quantitative attributes. An intuitive and simple solution is to partition quantitative attributes into binary attributes. However, there is potential information loss due to partitioning. Instead, our approach is based on a normalization scheme and can directly work on quantitative attributes. In addition, we adopt the algorithm structures of three popular association pattern mining algorithms and add a critical clique pruning technique. Finally, we compare the performance of these algorithms for finding quantitative maximal hyperclique patterns using some real-world data sets.

1 Introduction

A hyperclique pattern [11, 5] is a new type of association pattern that contains items which are highly affiliated with each other. More specifically, the presence of an item in one transaction strongly implies the presence of every other item that belongs to the same hyperclique pattern. Conceptually, the problem of mining hyperclique pattern in transaction data sets can be viewed as finding approximately all-one sub-matrix in a 0-1 matrix where each column may correspond to an item and each row may correspond to a transaction. For the rest of this paper, we refer to this problem as the binary hyperclique mining problem.

However, in many business and scientific domains, there are data sets which contain quantitative attributes (e.g. income, gene expression level). How to define and efficiently identify hyperclique patterns in data sets with quantitative attributes remains a big challenge in the literature. To this end, the focus of this paper is to address the quantitative hyperclique pattern mining problem.

To the best of our knowledge, there is no previous work on developing algorithms for finding quantitative maximal hyperclique patterns. Our approach for mining quantitative hyperclique patterns is built on top of the normalization scheme [9]. A side effect of the normalization scheme is that there is no support

pruning for single items. To meet with this computational challenge, we design a **clique pruning** method to dramatically remove a large number of items which are weakly related to each other, and thus effectively improving the overall computational performance for finding quantitative hyperclique patterns. We adopt structures of three popular association pattern mining algorithms including FP-tree [6], diffEclat [12], and Mafia [3] as the bases of our algorithms. The purpose of these algorithms is to find quantitative maximal hyperclique pattern, which is a more compact representation of quantitative hyperclique patterns and is desirable for many applications, such as pattern preserving clustering [10]. A hyperclique pattern is a maximal hyperclique pattern if no superset of this pattern is a hyperclique pattern. Finally, we briefly introduce the results of using our approach on some real-world data sets.

2 Normalization and Quantitative Hyperclique Patterns

Normalization. In this paper, we adopt the normalization method proposed in [9]. For a vector $x =< x_1, x_2, \ldots, x_n >$, our normalization will turn the vector as $x' =< x'_1, x'_2, \ldots, x'_3 > =< \frac{x_1}{|x|}, \frac{x_2}{|x|}, \ldots, \frac{x_n}{|x|} >$, where $|x| = \sqrt{\sum_{k=1}^{n} x_k^2}$. After data normalization, we define the support of every individual item i, $\sigma_{L_2^2}(i) = {x'_1}^2 + {x'_2}^2 + \ldots + {x'_n}^2 = 1$ and the support of an itemset X is defined as $\sigma_{\min,\ \mathrm{L}_2^2}(X) = \sum_{i \in T}(min\{T(i,j)|j \in I\})^2$, where T(i, j) means the normalized value of item j in the transaction i.

One advantage of this normalization is that the resulting support is a number between 0 and 1. Such normalization is natural in many domains, e.g., text documents. However, a side-effect of this is that individual items can no longer be pruned using a support threshold since all single items have a support of 1.

Quantitative Hyperclique Patterns. A traditional binary hyperclique pattern [11] is a frequent itemset with the additional constraint that every item in the itemset implies the presence of the remaining items with a minimum level of confidence known as the h-confidence. Specifically, we have the following:

Definition 1. *A set of attributes, X, forms a hyperclique pattern with a particular level of h-confidence, where h-confidence is defined as*

$$\mathrm{hconf}(X) = \min_{i \in X}\{\mathrm{conf}(\{i\} \rightarrow \{X - \{i\}\})\} = \sigma(X) / \max_{i \in X}\{\sigma(i)\} \qquad (1)$$

Where σ is the standard support function [1].

H-confidence, just like standard support, is in the interval $[0, 1]$ and it has the anti-monotone property; that is, the h-confidence of an itemset is greater than or equal to that of its any superset. Also, hyperclique patterns have the high affinity property, i.e., items in a pattern with a high h-confidence are guaranteed to have a high pairwise similarity as measured by the cosine metric. Additionally, there is an important relationship between h-confidence of binary hyperclique patterns and the support function $\sigma_{\min,\ \mathrm{L}_2^2}(X)$. In particular, since $\sigma_{\min,\ \mathrm{L}_2^2}(X)$ is equivalent to standard support for binary data, we can substitute $\sigma_{\min,\ \mathrm{L}_2^2}(X)$ for the standard support function $\sigma(X)$ in Equation 1. It is then interesting to note

that if we normalize all attributes to have an L_2 norm of 1, i.e., $\sigma_{\min,\ L_2^2}(i) = 1$ for all items i, then, by Equation 1, $\text{hconf}(X) = \sigma_{\min,\ L_2^2}(X)$, since the normalization sets the support of all the item to 1, we get $\max_{i \in X}\{\sigma(i)\} = \text{support(i)} = 1$.

In a nutshell, finding continuous hyperclique patterns first proceeds by normalizing the attributes to have an L2 norm of 1. Then, for each row, we take the minimum of the specified attributes. Finally, we square each of these values and add them up. The resulting value is the h-confidence and is a lower bound on the pairwise cosine similarity.

3 Algorithm Descriptions

Here, we present the algorithms for mining quantitative maximal hyperclique patterns. Our algorithms are built on top of three state-of-the-art association pattern mining algorithms including FPTree [6], diffEclat [12], and Mafia [3].

Clique Pruning. We design a clique pruning method for eliminating weakly related single items. Specifically, we first compute h-confidence of all item pairs on the normalized data. For each item, we then identify the maximum h-confidence value among all pairs including this item. Finally, for a user-specified threshold, we prune all items whose maximum h-confidence is less than this threshold.

Algorithm based on FP-Tree: FP-Tree [6] is a compact tree structure which allows to identify frequent patterns without generating the candidate patterns. Here, we adopt the FP-tree algorithm for finding quantitative maximal hyperclique patterns. First, we store float values instead of integer values, since the support of the normalized data are continuous. Second, the support values should be squared before added to the FP-Tree since they have an L_2 Norm. Finally, we need to split squared transactions and make the support of preceding item not less than the successor item, before adding them into the FP-Tree.

Algorithm based on MAFIA. MAFIA [3] is a depth-first searching algorithm for mining maximal frequent patterns. For the data set with continuous attributes, we change the algorithm to store not only the tidset, but also the support (normalized data) for each transaction. For this purpose, the algorithm needs a float vector to store the support information. Each element in the vector presents the support for each transaction in order.

Algorithm based on DiffEclat. DiffEclat uses a vertical data representation, called **diffset**, for efficiently mining maximal frequent patterns[12]. The diffset only store the different set of transaction ID between the pattern and its parant pattern. The key modification that we made is to store both transaction IDs and the support information. However, for diffset, we store the support different between a pattern and its parent pattern instead of the support itself.

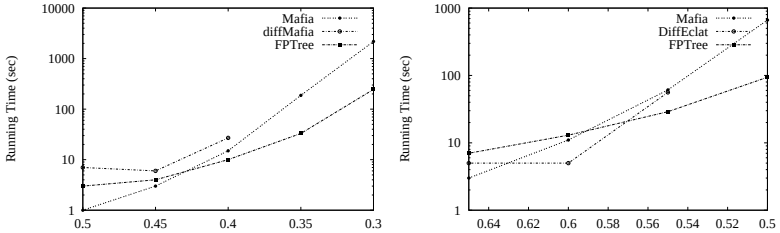
4 Experimental Evaluation

Experimental Setup Our experiments were performed on two real-life gene expression data sets, Colon Cancer and NCI [2, 7]. Table 1 shows some characteristics of these gene expression data sets.

DATASET	Colon Cancer		NCI
# Gene	2000		9905
# Sample	62		68
# Class	2		9
CLASS	NAME	# SAMPLE	
C1	Tumor	40	
C2	Normal	22	

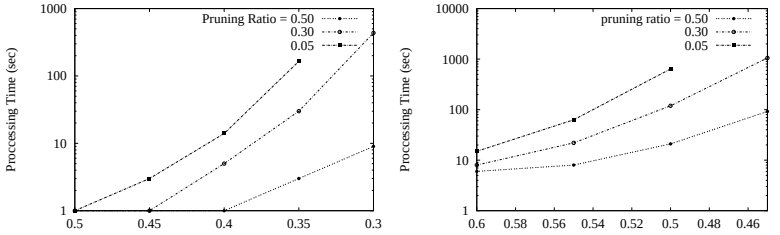
Table 1. The Characteristics of Gene Expression Data Sets

A Performance Comparison. Figure 1 (a) shows the running time of three algorithms on the Colon Cancer data set. As can be seen, when the h-confidence threshold is less than 0.35, the FP-Tree can be an order of magnitude faster than Mafia and DiffEclat is not very efficient and become unscalable when the h-confidence threshold is low. Also, Figure 1 (b) shows the performance of the proposed algorithms for mining sample patterns on the NCI data set. Similar to the observation from the Colon Cancer data set, we can also observe that when the h-confidence threshold is less than 0.5, the FPTree can be an order of magnitude faster than Mafia. However, MAFIA has a better performance when the h-confidence threshold is high. Another observation is that the performance DiffEclat is not scalable when the h-confidence threshold is low.



(a) *Colon* (b) *NCI*

Fig. 1. The Performance Comparison on Colon and NCI data sets.



(a) *Colon* (b) *NCI*

Fig. 2. The Effect of Clique Pruning on Colon Cancer and NCI Data Sets.

The Effect of Clique Pruning. Figure 2 demonstrate the effect of clique pruning on Colon and NCI data sets using the algorithm based on Mafia. As

can be seen from both figures, with the increase of the clique pruning ratio, the running time is reduced significantly. The running time can be orders of magnitude faster if we target on hyperclique patterns with high affinity. Another benefit is that, the proposed algorithm can even identify patterns at a very low level support when the clique pruning ratio is at a certain level.

5 Conclusions

In this paper, we addressed the problem of mining quantitative maximal hyperclique patterns in the data sets with continuous attributes. Instead of mapping continuous attributes into binary attributes, we applied a data normalization method. Also, we provided algorithms for finding quantitative maximal hyperclique patterns. These algorithms are built on top of three state-of-the-art association pattern mining algorithms and have included a clique pruning method to perform pruning for individual items. Finally, the performance of the algorithms have been demonstrated using real-world data sets.

Acknowledgement

This paper was partially supported by NSF grant #ACI-0305567 and NSF grant #CCF-0514796.

References

1. Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In *SIGMOD 93*, May 1993.
2. U. Alon, N. Barkai, D.A. Notterman, and et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissuese probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci*, 96:6745–6750, June 1999.
3. D Burdick, M Calimlim, and J Gehrke. Mafia: A maximal frequent itemset algorithm for transactional databases. In *ICDE*, 2001.
4. Eui-Hong Han, George Karypis, and Vipin Kumar. Tr# 97-068: Min-apriori: An algorithm for finding association rules in data with continuous attributes. Technical report, Department of Computer Science, University of Minnesota, 1997.
5. Y. Huang, H. Xiong, W. Wu, and Z. Zhang. A hybrid approach for mining maximal hyperclique patterns. In *ICTAI*, 2004.
6. J.Han, J.Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *ACM SIGMOD*, 2000.
7. D.T. Ross, U. Scherf, and et al. Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics*, 24(3):227–234, 2000.
8. Ramakrishnan Srikant and Rakesh Agrawal. Mining quantitative association rules in large relational tables. In *SIGMOD 96*, 1996.
9. Michael Steinbach, Pang-Ning Tan, Hui Xiong, and Vipin Kumar. Extending the Notion of Support. In *ACM SIGKDD*, 2004.
10. H. Xiong, M. Steinbach, P. Tan, and V. Kumpar. HICAP: Hierarchial Clustering with Pattern Preservation. In *Proc. of SIAM Int'l Conf. on Data Mining*, 2004.
11. H. Xiong, P. Tan, and V. Kumar. Mining strong affinity association patterns in data sets with skewed support distribution. In *ICDM 2003, USA*, 2003.
12. Mahommed Zaki and Karam Gouda. Fast vertical mining using diffsets. In *ACM SIGKDD*, 2003.