Approximate Frequent Pattern Mining

Philip S. Yu, Xifeng Yan
IBM TJ Watson Research Center
{psyu, xifengyan}@us.ibm.com

Jiawei Han, Hong Cheng, Feida Zhu
University of Illinois at Urbana-Champaign
{hanj, hcheng3, feidazhu}cs.uiuc.edu

Summary

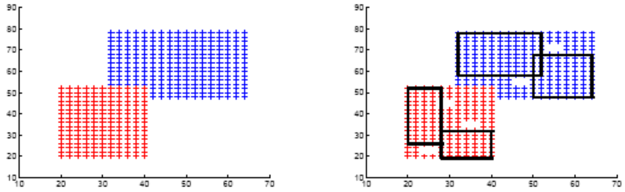
Frequent pattern mining has been a focused theme in data mining research and an important first step in the analysis of data arising in a broad range of applications. The traditional exact model for frequent pattern requires that every item occurs in each supporting transaction. However, real application data is usually subject to random noise or measurement error, which poses new challenges for the efficient discovery of frequent pattern from the noisy data. Mining approximate frequent pattern in the presence of noise involves two key issues: the definition of a noise-tolerant mining model and the design of an efficient mining algorithm. In this paper, we will give an overview of the approximate itemset and sequential pattern mining.

Extended Abstract

Frequent pattern mining has been a focused theme in data mining research with a large number of scalable mining methods proposed [1, 5] and various extensions including closed patterns, maximal patterns and so on [3, 12]. Frequent patterns have found broad applications in areas like association rule mining [1], indexing [11], classification [8] and clustering [10]. In these applications, the ultimate goal is to discover interesting associations between objects and attribute subsets, rather than association among attributes alone.

Lets first consider the frequent itemset mining. One important experimental application of frequent itemset mining is the exploration of gene expression data, where the joint discovery of both the set of conditions that significantly affect gene regulation and the set of co-regulated genes is of great interest. Another important application of frequent pattern mining is frequent pattern-based classification, where the associations between attributes and their relation to the class labels or functions are explored. Despite the exciting progress in the field of frequent pattern mining and its extensions, an intrinsic problem with the exact frequent pattern mining is the rigid definition of support.

An itemset *x* is supported by a transaction *t*, if each item of *x* exactly appears in *t*. An itemset *x* is frequent if the number of transactions supporting it is no less than a user-specified minimum support threshold (denoted as *min sup*). However, in real applications, a database is typically subject to random noise or measurement error, which poses new challenges for the discovery of frequent itemsets. For example, in a customer transaction database, random noise could be caused by an out-of-stock item, promotions or some special event like the world cup, holidays, *etc.*. Measurement error could be caused by noise from experiments, uncertainty involved in discretizing continuous values, stochastic nature of the study field, *etc.*. In privacy-preserving data mining [2], random noise is added to perturb the true values of the original database. Such random noise can distort the true underlying patterns. Theoretical analysis by [9] shows that in the presence of even low levels of noise, large frequent itemsets are broken into fragments of logarithmic size, thus the itemsets cannot be recovered by the exact frequent itemset mining algorithms. Figure 1 shows two transaction databases. The *x*-axis represents items and the *y*-axis represents the transactions. Figure 1 (a) shows the embedded true patterns in the database without random noise while (b) shows the observed distorted patterns in the presence of random noise (several “holes" appear in the embedded patterns due to random noise). If the exact frequent itemset mining algorithms are applied to a database subject to random noise, as in Figure 1 (b), the original embedded true patterns will be fragmented into several smaller ones which are highlighted by the bounding boxes.



(a) Embedded True Patterns

(b) Observed Distorted Patterns

Figure 1. Patterns with and without random noise

To tackle such a problem, we propose to recover the approximate frequent itemsets from “core patterns”. Intuitively, an itemset *x* is a core pattern if its exact support in the noisy database *D* is no less than αs, where $\alpha \in$ [0, 1] is a *core pattern factor*, and *s* is the *min sup* threshold. To support our approach, we use a random noise model to evaluate the probability of a true frequent pattern remaining as a core pattern in *D* in the presence of noise. With some realistic parameter settings, we can show that such true patterns remain as core patterns in the noisy database with high probability. Therefore, although our approach could miss certain true patterns if they fail the core pattern requirement, we have some confidence about the core pattern recovery approach. With the core pattern constraint, we only retain the set of core patterns as initial seeds for approximate pattern generation. For each core pattern, an extension set of patterns with a small amount of mismatch (determined by the noise level) from it is identified. The approximate support under noise is the sum of the exact supports for the core pattern and its extension set. To further reduce the output size, we define the concept of *approximate closed itemset* and only mine the closed ones. Based on the problem formulation, we design an efficient algorithm AC-Close [4] to mine the approximate closed itemsets. A *top-down* mining strategy is exploited where the large-size approximate itemsets are discovered before the small size ones, which makes full use of the pruning power of *min sup* and closeness and thus, narrows down the search space dramatically.

Next we relax the core pattern requirement and consider the case of sequential pattern mining. We again look at bioinformatics for example. The identification of repeats serves as a critical step in many biological applications on a higher level such as a preprocessing step for genome alignment, whole genome assembly and a postprocessing step for BLAST queries. For repeat families that are relatively new in the evolution, the set of repeats found under the Hamming distance model captures almost the complete set. Furthermore, the limited knowledge that biologists currently have of these repeats makes it often hard for them to evaluate the relative significance among different repeats. It is therefore worth the effort to mine the complete set. Existing tools like RepeatMasker [6] only solve the problem of pattern matching, rather than pattern discovery without prior knowledge. Many research works for the repeating patterns have been on an important subtype: the tandem repeats [7], where repeating copies occur together in the sequence. However, these methods would miss those patterns whose supporting occurrences appear globally in the entire data sequence, which account for the majority of the complete set of frequent patterns.

To uncover more interesting approximate patterns in DNA sequences, we establish a more general model for approximate sequential pattern mining problem. Our general philosophy is a “break-down-and-build-up” one based on the following observation. Although for an approximate pattern, the sequences in its support set may have different patterns of substitutions, they can in fact be classified into groups, which we call *strands*. Each strand is a set of sequences sharing a unified pattern representation together with its support. The idea is that by “breaking down” the support sets of the approximate patterns into strands, we are able to design efficient algorithms to compute them. Here the core pattern constraint is not required. Using a suffix-tree-based algorithm, we can in linear time mine out the initial strands, which are all the exactly matching repeats. These initial strands will then be iteratively assembled into longer strands in a local search fashion, until no longer ones can be found. In the second “build-up” stage, different strands are then grouped based on their constituting sequences to form a support set so that the frequent approximate patterns would be identified. By avoiding incremental growth and global search, we are able to achieve great efficiency without losing the completeness of the mining result. Instead of mining only the patterns repeating within a sliding window of fixed sizes, our algorithm is able to mine all globally repeating approximate patterns.

REFERENCES

[1]. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. VLDB'94*, pages 487-499, Sept. 1994.

[2]. R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proc. of SIGMOD*, pages 439-450, 2000.

[3]. D. Burdick, M. Calimlim, and J. Gehrke. MAFIA: A maximal frequent itemset algorithm for transactional databases. In Proc. ICDE'01, pages 443-452, April 2001.

[4]. H. Cheng, P.S. Yu, and J. Han, AC-Close: efficiently mining approximate closed itemsets by core pattern recovery. *In Proc. ICDM’06*, Dec. 2006.

[5]. J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proc. SIGMOD'00*, pages 1-12, May 2000.

[6]. Institute for Systems Biology. Repeatmasker. In http://www.repeatmasker.org/webrepeatmaskerhelp, 2003.

[7]. G. M. Landau, and J. P. Schmidt. An algorithm for approximate tandem repeats. In Proceedings of the 4th Annual Symposium on Combinatorial Pattern Matching, number 684, pages 120–133, 1993.

[8]. B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proc. of KDD*, pages 80-86, 1998.

[9]. J. Liu, S. Paulsen, X. Sun, W. Wang, A. Nobel, and J. Prins. Mining approximate frequent itemsets in the presence of noise: Algorithm and analysis. In *Proc. SDM'06*, pages 405-416, April 2006.

[10]. K. Wang, C. Xu, and B. Liu. Clustering transactions using large items. In *Proc. of CIKM*, pages 483-490, 1999.

[11]. X. Yan, P.S. Yu, and J. Han. Graph Indexing: A frequent structure-based approach. In *Proc. of SIGMOD*, pages 335-346, 2004.

[12]. M.J. Zaki, and C. J. Hsiao. CHARM: An efficient algorithm for closed itemset mining. In *Proc. SDM'02*, pages 457-473, April 2002.