行政院國家科學委員會專題研究計畫　成果報告

找尋序列間關聯法則之研究

計畫類別：個別型計畫
計畫編號：NSC92-2213-E-002-066-
執行期間：92 年 08 月 01 日至 93 年 07 月 31 日
執行單位：國立臺灣大學資訊管理學系暨研究所

計畫主持人：李瑞庭

報告類型：精簡報告

處理方式：本計畫可公開查詢

中　華　民　國 93 年 11 月 18 日

找尋序列間關聯法則之研究
A study on mining inter-sequence association rules

計畫編號：92-2213-E-002-066
執行期間：92/08/01-93/07/31

主持人：李瑞庭　臺灣大學資訊管理學系　副教授

一、中文摘要(關鍵詞: 資料探勘、關聯法則、序列中的關聯法則、序列間的關聯法則)

在循序樣式的資料庫中，一個交易(transaction)只包括一個序列(sequence)且每一個序列皆互為獨立的，彼此不相關。我們稱此類的資料探勘為序列中的關聯法則(intra-sequence association rules)。我們進一步探討序列間樣式的關聯，我們稱之為序列間的關聯法則(inter-sequence association rules)。就我們所知，目前，並沒有任何的資料探勘的技術，特別設計來找尋序列間的關聯法則。而序列間的關聯法則可應用於分析許多應用層面的資料：如，WWW 的路徑追蹤樣式、通信資料、疾病症狀、氣候、股票波動與 DNA 序列等等。

因此，在本計劃中，我們提出一個找尋序列間的關聯法則的演算法。首先，我們使用 PrefixSpan 的演算法找尋所有的循序樣式，然後使用 level-wise 的演算法檢查序列集合是否為大序列集合(large sequence-set)。同時，我們將每一個循序樣式發生的時間紀錄在一個時間串列中(time point list)，然後，我們將這些時間串列分成好幾個群組，並將它們儲存於 L-buckets 中。因為我們使用時間串列及 L-buckets 加速支持度的計算(support counting)，使得我們所提出的演算法比 Apriori-like 演算法更有效率。

英文摘要(Keywords: Data mining, Association rules, intra-sequence association rules, Inter-sequence association rules)

There are many algorithms proposed to find sequential patterns in sequence databases where a transaction contains a sequence. Previously proposed algorithms treat each sequence as an independent one. This kind of mining belongs to intra-sequence patterns mining, because all the patterns found just describe characteristics within a sequence. We would like to go further to investigate relationships between sequential patterns in different sequences, called inter-sequence association rules mining. To the best of our knowledge, there are no data mining techniques specially designed to analyze the inter-sequence association rules. Mining inter-sequence association rules is used in many application areas. We can use inter-sequence association rules to analyze web page traversal, telecommunication, disease symptoms, weather changes, stock movements, DNA sequences, and etc.

Therefore, in this project, we proposed an algorithm to mine inter-sequence association rules. First, we use the PrefixSpan algorithm to find all sequential patterns, and then we use a level-wise method to check if an sequence-set is large. We use a time point list to collect all the time points at which sequential patterns occur. Then, we divide time point lists into several groups, and store them in buckets, called L-buckets. Since our proposed algorithm uses L-buckets and time point lists to accelerate the process of support counting, our proposed algorithm outperforms the Apriori-like algorithm.

二、計畫的緣由與目的

With the increasing of tremendous amount of data in various applications, mining implicit knowledge from large databases has attracted much attention recently. There is a large amount of valuable information embedded in databases or data warehouses which is useful for analyzing customer’s buying

behavior and thus improving the business decisions.

Data mining is an application-specific issue and various mining techniques have been developed to solve different application problems, such as mining association rules [2, 3, 6, 13, 15, 17, 18, 19, 21, 22, 25, 27], classification [1, 5, 11, 16], clustering [8, 9, 10, 20, 26, 28], sequential patterns [4, 14, 23, 24], partial periodic patterns [12], and path traversal patterns in World Wide Web (WWW) [7].

There are many algorithms [4, 14, 23, 24] proposed to find sequential patterns in sequence databases where each transaction contains one sequence. Previously proposed algorithms treat each sequence as an independent one. This kind of mining belongs to intra-sequence patterns mining, because all the patterns found just describe characteristics within a sequence. We would like to go further to investigate relationships between sequential patterns in different sequences, called inter-sequence association rules mining.

To the best of our knowledge, there are no data mining techniques specially designed to analyze the inter-sequence association rules. We can use inter-sequence association rules to analyze web page traversal, telecommunication, disease symptoms, weather changes, stock movements, DNA sequences, and etc.

Let's consider an example. A telecommunication company may have many mobile phone base stations. The order of busy base stations within a call can be viewed as a sequence. When we mine intra-sequence association rules, we may find two association rules: 1) station B often becomes busy after station A becomes busy; 2) station D often becomes busy after station C becomes busy. When we mine inter-sequence association rules, we may find one more rule: if station B becomes busy after station A becomes busy some day, another sequential pattern (station D becomes busy after station C becomes busy) may happen in two hours. This rule can help the telecommunication company to make an arrangement on working schedules and to control the message flow.

Web page traversal also can be viewed as a sequence. When we mine intra-sequence association rules, we may find two association rules. One is that users like to browse page B after browsing page A, and the other is that users like to browse page D after browsing page C. When we mine inter-sequence association rules, we may find that if a user browses page B after browsing page A, he/she is very likely to browse page D after browsing page C in two days.

Therefore, in this project, we propose an algorithm to mine inter-sequence association rules. First, we use the PrefixSpan algorithm to find all sequential patterns, and then we use a level-wise method to check if a sequence-set is large. We use a time point list to collect all the time points at which sequential patterns occur. Then, we divide time point lists into several groups, and store them in buckets, called L-buckets. Candidates are stored in buckets, too, called C-buckets. Candidates in the same bucket are very likely to need information in the same L-buckets. When counting the supports of candidates, we treat a C-bucket as a basic unit of candidates to be read into the memory and thus we just need to read relevant L-buckets into the memory. Thus, it will improve the performance of our proposed algorithm.

三、研究方法與成果

The problem of mining association rules can be decomposed into two steps: (1) find all large sequence-sets in a database. (2) use the large sequence-sets found in step 1 to generate all association rules.

To find all large sequence-sets in a database, we first use the PrefixSpan algorithm to find all large sequences. For each large sequence found, all the time points at which the large sequence occurs are collected into a corresponding time point list. Then, we divide time point lists into several groups, and store them in buckets, called L-buckets. Then, we use a level-wise method to generate candidate patterns across different sequences and check if a candidate is

large. Once we generate a candidate, we count the support of the candidate. To count the support of a candidate, we just need to read relevant time point lists.

If memory is not large enough to contain all the L-buckets, we record all L-buckets in disk and the L-buckets are clustered in disk. We can read L-buckets into memory as many as possible. When a needed time point list is not in memory, we remove some L-buckets not currently needed from memory and read the L-bucket that contains the time point list into memory. Because of candidate generation method, candidates are sorted according to their alphabetical order. Many successive candidates very likely need several common L-buckets. With the property, the number of disk accesses can be reduced.

The experimental results show that the execution time of the Apriori-like and our proposed algorithms increases linearly when the number of transactions is increased from 10k to 100k. This is because there are more transactions for the Apriori-like algorithm to scan and check, and there are more time points in L-buckets for our proposed algorithm to check too. As the minimum support decreases, the execution time of the Apriori-like algorithm and our proposed algorithm increases. When the total number of items in a transaction increases, the execution time increases for both algorithms. Under the same support threshold, if the average transaction length gets longer, more number of candidates needs to be counted. When the number of distinct items in the database increases, the number of large 1-itemset increases slightly, but the number of large 2-itemsets decreases sharply. Because the count of each candidate decreases, the number of large sequence-sets is smaller and the execution time of both algorithms decreases. In summary, our proposed algorithm uses L-buckets and time point lists to accelerate the process of support counting. The construction of time point lists and L-buckets is quite straight-forward and efficient. Thus, our proposed algorithm outperforms the Apriori-like algorithm in all cases.

四、結論與討論

In this project, we proposed an algorithm to mine inter-sequence association rules in which a rule can be used to describe the relationships of patterns among different sequences. Its basic idea is to find patterns within a transaction first and collect the time point lists at the same time. Then, we find patterns between different sequences by a candidate generation-and-test method. We use L-buckets and time point lists to accelerate the process of counting support of candidates. The construction of time point lists and L-buckets is quite straight-forward and efficient. The experimental result shows that our proposed algorithm is efficient and runs much faster than the Apriori-like algorithm.

There are other issues worth further study. First, we are still working on using a more efficient data structure, called CHAIN, to mine the inter-sequence association rules. Second, we only discuss one-dimensional sequence relationships, but we may consider extending it to multi-dimensional relationships, so that we can mine a more general rule like “If a user browses page B after browsing page A at school, he/she is very likely to browse page D after browsing page C at home in two days.” Furthermore, inter-sequence association rule mining may suffer from generating a huge number of rules, and we need a way to impose a constraint on the mining process to quickly find our interested rules.

五、參考文獻

[1] R. Agrawal, S.Ghosh, T. Imielinski, B. Iyer, and A. Swanmi. An interval classifier for database mining applications. *Proceedings of the 18th International Conference on Very Large Data Bases*, pp. 560-573 (1992).

[2] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, ACM

Press, Washington, D.C., pp. 207-216 (1993).

[3] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *Proceedings of the 20th International Conference on Very Large Data Bases*, pp. 478-499 (1994).

[4] R. Agrawal and R. Srikant. Mining sequential patterns. *Proceedings of the 11th International Conference on Data Engineering*, pp. 3-14 (1995).

[5] T.M. Anwar, H.W. Beck, and S.B. Navathe. Knowledge mining by imprecise querying: a classification-based approach. *Proceedings of the 8th International Conference on Data Engineering*, pp. 622-630 (1992).

[6] S. Brin, R. Motwani, J.D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, ACM Press, Tucson, Arizona, pp. 255-264 (1997).

[7] M.-S. Chen, J.S Park, and P.S. Yu. Efficient data mining for path traversal patterns. *IEEE Transactions on Knowledge and Data Engineering*, 10(2), pp. 209-221 (1998).

[8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial database with noise. *International Conference on Knowledge Discovery in Databases and Data Mining (KDD-96)*, AAAI Press, Portland, Oregon, pp. 226-231 (1996).

[9] M. Ester, H.-P. Kriegel, and X. Xu. A database interface for clustering in large spatial databases. *International Conference on Knowledge Discovery in Databases and Data Mining (KDD-95)*, AAAI Press, Montreal, Canada, pp. 94-99 (1995).

[10] S. Guha, R. Rastogi, and K. Shim. CURE: A clustering algorithm for large databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, ACM Press, pp. 73-84 (1998).

[11] J. Han, Y. Cai, and N. Cercone. Knowledge discovery in databases: An attribute-oriented approach. *Proceedings of the 18th International Conference on Very Large Data Bases*, pp. 547-559 (1992).

[12] J. Han, G. Dong, and Y. Yin. Efficient mining of partial periodic patterns in time series database. *Proceedings of the 15th International Conference on Data Engineering*, pp. 106-115 (1999).

[13] J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. *Proceedings of the 21st International Conference on Very Large Data Bases*, pp. 420-431 (1995).

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

[15] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. *Proceedings of the ACM SIGMOD International Conference on Management of*

Data, ACM Press, Dallas, Texas, pp. 1-12 (2000).

[16] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms, *Machine Learning*, 40(3), pp. 203-228 (2000).

[17] H. Mannila. Methods and problems in data mining. *Proceedings of the 6th International Conference on Database Theory*, Springer, Delphi, Greece, pp. 478-499 (1997).

[18] H. Mannila, H. Toivonen, and A.I. Verkamo. Efficient algorithms for discovering association rules. *Proceedings of the AAAI Workshop on Knowledge Discovery in Databases*, AAAI Press, Seattle, Washington, pp. 181-192 (1994).

[19] A.M. Mueller. Fast sequential and parallel algorithms for association rules mining: a comparison. *Technical report*, Faculty of the Graduate School of The University of Maryland (1995).

[20] R.T. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. *Proceedings of the VLDB Conference*, Santiago, Chile, pp. 144-155 (1994).

[21] P. Nicolas, B. Yves, T. Rafik, and L. Lotfi. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24(1), pp. 25-46 (1999).

[22] J.S. Park, M.-S. Chen, and P.S. Yu. Using a hash-based method with transaction trimming for mining association rules. *IEEE Transactions on Knowledge and Data Engineering*, 9(5), pp. 813-825 (1997).

[23] J. Pei, J. Han, B. Mortazavi-Asl, and H. Pinto. PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. *Proceedings of the 17th International Conference on Data Engineering*, pp. 106-115 (2001).

[24] J. Pei, J. Han, B. Mortazavi-Asl, and H. Zhu. Mining access pattern efficiently from web logs. *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Kyoto, Japan, Springer, Berlin, pp. 396–407 (2000).

[25] A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining associate rules in large databases. *Proceedings of the 21st International Conference on Very Large Data Bases*, pp. 432-444 (1995).

[26] G. Sudipto, R. Rajeev, and S. Kyuseok. ROCK: A robust clustering algorithm for categorical attributes. *Information Systems*, 25(5), pp. 345-366 (2000).

[27] H. Toivonen. Sampling large databases for association rules. *Proceedings of the 22nd International Conference on Very Large Data Bases*, pp. 134-145 (1996).

[28] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: An efficient data clustering method for very large databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Montreal, Canada, pp. 103-114 (1996).