

Medical Data Mining Based on Association Rules

Ruijuan Hu

Dep of Foundation, PLA University of Foreign Languages, Luoyang 471003, China

E-mail: huruijuan01@126.com

Abstract

Detailed elaborations are presented for the idea on two-step frequent itemsets Apriori algorithm of Association Rules. An improved method called Improved Apriori algorithm is brought forward owing to the disadvantages of Apriori algorithm. Moreover, based on Improved Apriori algorithm, data mining for breast-cancers is carried out for the relationship between breast-cancer recurrences and other attributes by making use of SQL Server 2005 Analysis Services. Results show the availability of Association Rules in medical data mining.

Keywords: Data mining, Apriori algorithm, Improving Apriori algorithm, Breast-cancer

1. Introduction

The application of computer information technology in medical promotes the digitalization of medical information, so amount of hospital database information is perpetual expanding. The database within the medical treatment of most hospitals is low-end operations at present, lacking data integration and analysis, letting alone mine deeper, implicit and valuable knowledge in the large amount of data resources. In this context, medical data mining came into being (Daniel, 2008, pp. 77-102).

As is well known, many algorithms including Association Rules, Decision Tree and Clustering for data mining were presented over time. A trial of medical data mining was made on 285 cases of breast disease patients in HIS (Hospital Information System) using Association Rules algorithm.

2. Medical data mining based on Association Rules

In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. Many algorithms for generating association rules were presented over time. Some well known algorithms are Apriori, DHP and FP-Growth. Apriori is the best-known algorithm to mine association rules.

2.1 Apriori algorithm

Apriori is a fast mining algorithm first introduced by R.Agrawal et al for market basket data analysis (R. Agrawal, 1993, pp. 207-216). It uses a breadth-first search strategy to counting the support of itemsets and uses a candidate generation function which exploits the downward closure property of support. It is a set of ideas based on the frequency of two-stage approach which can be decomposed into two sub-problems (Sriphaew K, 2003, pp. 476-484). First, find all Frequent Itemsets that have transaction supports above minimum support called minsup. The support for an itemset is defined as the fraction of total transactions that contains this itemset. Itemsets with minimum support are called large itemsets, and all the others small itemsets. Second, use the large itemsets to generate the desired rules with the recursive method. There is a straightforward algorithm for this task.

Its core idea is as follows (Rakesh Agrawal, 1994, pp. 487-499):

```

C1 = {candidate 1-itemsets}
L1 = { c ∈ C1 | c.count ≥ minsup };
for (k=2; Lk-1 ≠ ∅; k++) do begin
    Ck = apriori_gen (Lk-1);           // New candidates
    for all transaction t ∈ D do begin
        Ct = subset (Ck, t);         // Candidates contained in t
        for all candidates c ∈ Ct do
            c.count++;
    end
    Lk = { c ∈ Ck | c.count ≥ minsup }
end
end

```

Answer = $\cup_k L_k$;

The first pass of the algorithm simply counts item occurrences to determine the large 1-itemsets. A subsequent pass, say pass k , consists of two phases. First, the large itemsets L_{k-1} found in the $(k-1)$ th pass are used to generate the candidate itemsets C_k using the apriori-gen function described below. Next, the database is scanned and the support of candidates in C_k is counted.

The apriori_gen function (Mohammed J., 2000, pp. 372-390) takes as argument L_{k-1} , the set of all large $(k-1)$ -itemsets. The function works as follows.

Procedure apriori_gen (L_{k-1} : frequent $(k-1)$ _itemsets; minsup)

```

for each itemset p ∈ Lk-1
  for each itemset q ∈ Lk-1
    if (p.item1 = q.item1) ∧ (p.item2 = q.item2) ∧ ... ∧ (p.itemk-2 = q.itemk-2) then
      {c = p ∪ q
       if has_infrequent_subset (c, Lk-1) then
         delete c
       else add c to Ck;
      }

```

return C_k ;

Procedure has_infrequent_subset (c: candidate k _itemsets; L_{k-1} _itemsets)

```

for each s ( $k-1$ ) subsets of c
  if s ∈ Lk-1 then
    return true;
return false;

```

In this algorithm, k _itemsets means itemsets which include k sets; L_k means all k _itemsets which are greater than minimum support, which is large k _itemsets; C_k means k _itemsets which meet the following conditions: each $(k-1)$ _itemsets subset of k _itemsets belongs to L_{k-1} , and it is generally called the candidate sets. The algorithm mainly consists of two steps: generate candidates C_k and count the candidate sets. Generating candidates also can be divided into two stages (M. J. Zaki., 2001, pp. 31-60):

- (1)Connection: Merge the two equal $(k-1)$ _itemsets of pre- $k-2$ sets in L_{k-1} , resulting in candidate k itemsets.
- (2)Delete: Delete the $(k-1)$ subsets of C_k not belonging to L_{k-1} .

The efficiency of the algorithm lies in its use of large itemsets' closure. It greatly reduces the number of itemsets needing to calculate the degree of support, namely, it avoids the calculation of those which can not become a large set of candidate sets.

2.2 Improved Apriori algorithm

In Apriori algorithm all the candidate itemsets with the same length must be stored in the memory, which results in a waste of space. To generate large itemsets, the database is passed as many times as the length of the longest large itemsets. Namely, the database is scanned and the support of each candidate itemset is counted after the new candidate itemsets are generated, which results in a waste of time for large database. This is the performance bottleneck of Apriori algorithm.

The basic idea of the improved algorithm (Zelic I, 2000, pp. 799-803) (J. Han, 1992, pp. 547-559) is proposed according to the above deficiencies. In the improved algorithm, which is fundamentally different from Apriori, we need not store all the candidate itemsets in the memory and pass over the database only once. Find out all the high frequency 1-dimensional data itemsets L_1 , and then L_1 is used to identify all the high frequency 2-dimensional data itemsets L_2 , what's more, use L_2 to find C_2 , the rest may be deduced by analogy until no new high frequency itemsets exist. The realization from L_{k-1} to L_k is connecting L_{k-1} and its own to generate a candidate set of k -dimensional set of data itemsets, denoted by C_k , and then counting the frequency of C_k 's data itemsets, discarding low-frequency data itemsets, forming L_k . The connection process is taking out p and q from L_{k-1} . If p and q are the same as the pre- $k-2$ items, make a connection (S. Muggleton, 1992). The improved function apriori-gen is as follows:

```

Procedure apriori_gen ( $L_{k-1}$ : frequent (k-1) _itemsets; minsup)
  for each itemset  $p \in L_{k-1}$ 
    for each itemset  $q \in L_{k-1}$ 
      if ( $p.item_1 = q.item_1$ )  $\wedge$  ( $p.item_2 = q.item_2$ )  $\wedge$  ...  $\wedge$  ( $p.item_{k-2} = q.item_{k-2}$ ) then
        {  $c = p \cup q$ 
          for each itemset  $p \in L_{k-1}$  //scan all elements of  $L_{k-1}$ 
          for each itemset  $c \in C_k$  // scan all elements of  $C_k$ 
            if  $p$  is the subset of  $c$  then //determine whether each element of  $L_{k-1}$  contains  $C_k$ 
               $c.count++$ ;
           $C_k = \{c \in C_k | c.count = k\}$ ;
        }
  return  $C_k$ ;

```

In order to reduce the size of candidate sets, the improvement is set proposed. Suppose $|L_{k-1}|$ indicate the number of data itemsets in L_{k-1} , $p = |C_k|$ indicate the number of data itemsets in C_k . From the knowledge of sets, we can see, the number of n elements set's subsets is 2^n . Therefore, the original algorithm needs a total of $2^p * |L_{k-1}|$ times operations. The new algorithm just need $p * (|L_{k-1}| + 1)$ times operations.

The improved algorithm has the excellent property that the database is not used repeatedly. Rather, the encoding of the database is employed for judging whether a candidate is a large itemset. In the later passes, the size of this encoding can become much smaller than the database, thus saving much reading effort (Carlos Ordonez, 2000, pp. 78-85). Obviously the improved algorithm is superior when the number of data itemsets continuously increases.

2.3 Realization of medical data mining

2.3.1 Data preparation

A trial of medical data mining is made on 285 cases of breast disease patients provided by Puyang City People's Hospital HIS (Hospital Information System), of which 201 cases of no-recurrence-events, 84 cases of recurrence-events. Through communicating with doctors and learning the knowledge of pathology, we extract 8 attributes of patients as the attributes of each case: Age, Tumor-size (unit: mm) , the number of lymph node invasion (Inv-nodes) , whether Node-caps (Node-caps) are, malignant degree (Deg-malig) , tumor location (Breast) , where is the quadrant tumor (Breast-quad) , radiotherapy or not (Irradiat) , whether recurrence-events or no-recurrence-events (Class). Establish the relationships of Class and other attributes through data mining. The concrete settings of attributes are as follows:

```

age {'10-19', '20-29', '30-39', '40-49', '50-59', '60-69', '70-79', '80-89', '90-99'}
tumor-size {'0-4', '5-9', '10-14', '15-19', '20-24', '25-29', '30-34', '35-39', '40-44', '45-49', '50-54', '55-59'}
inv-nodes {'0-2', '3-5', '6-8', '9-11', '12-14', '15-17', '18-20', '21-23', '24-26', '27-29', '30-32', '33-35', '36-39'}
node-caps {'yes', 'no'}
deg-malig {'1', '2', '3'}
breast {'left', 'right'}
breast-quad {'left-up', 'left-low', 'right-up', 'right-low', 'central'}
irradiat {'yes', 'no'}
Class {'no-recurrence-events', 'recurrence-events'}

```

According to the attributes, we build the breast-cancer database of the cases by making use of SQL Server 2005.

2.3.2 Concrete realization

The steps of data mining using SQL Server 2005 Analysis Services for the realization of Association Rules are as follows (Zhu Deli. 2007, Chapter 11):

Step 1: Select the breast-cancer database created previously as the data source, and set up a data source view.

Step 2: Create Association Rules mining structure, and select breast-cancer table as the instance table, that is, the data contained in the table is the historical data relied on the analysis of algorithm.

Step 3: Create the mining model structure. Establish the mining model between the attributes of Class and Age, Tumor-size, Inv-nodes, Deg-malig, Irradiat according to the defined question what factors tumor recurrence is related to.

Step 4: Generate the mining results.

3. Analysis of the mining results

Realize Association Rules algorithm by making use of SQL Server 2005 Analysis Services. Several illustrations of Microsoft Association Rules are brought forward.

(1) Probability is put to use instead of Confidence.

(2) How to calculate the importance of Association Rules?

$$\text{IMPORTANCE}_{A \rightarrow B} = \log \frac{p(B | A)}{P(B | \text{not}A)} \quad (1)$$

(3) Set the parameters of the algorithm.

The mining rules are shown in Figure 1, which sort on the basis of importance and probability of association.

The minimum support is 30 and the minimum itemset is 0 in the case. These two values can be set according to the actual conditions. The interface shown in Figure 1 is sorting rules mined by data mining algorithm according to the probability intensity and the importance degree. As shown in Figure 1, the blue line above represents the recurrent probability of the patient whose Tumor-size is 35 to 39 and Age is 30 to 39. Figure 2 shows the strength of different Association Rules. To change the intensity level you can see, tumor recurrences most happen on the patients who have a great degree of malignancy and no radiotherapy.

A part of Association Rules are as follows according to the above:

Rule 1: Tumor-size='30-34' \cap Deg-malig='3' \Rightarrow Class='recurrence-events' (36.1%, 61.5%);

Rule 2: Inv-nodes='6-8' \cap Tumor-size='40-44' \Rightarrow Class='recurrence-events' (41.1%, 100%);

Rule 3: Inv-nodes='3-5' \cap Deg-malig='3' \Rightarrow Class='recurrence-events' (44.3%, 100%);

Rule 4: Inv-nodes='9-11' \cap Irradiat='no' \Rightarrow Class='recurrence-events' (44.3%, 100%).

Rule 1 means that 36.1% (support) of the studied patients' Tumor-size is 30 to 34mm, Deg-malig is '3' (the highest), and the possibility of tumor recurrence is 61.5% (confidence). Rule 2 means that 41.1% (support) of the studied patients' Inv-nodes is 6 to 8, Tumor-size is 40 to 44mm, and the possibility of tumor recurrence is 100% (confidence). Rule 3 means that 41.3% (support) of the studied patients' Inv-nodes is 3 to 5, Deg-malig is the highest, and the possibility of tumor recurrence is 100% (confidence). Rule 4 means that 41.3% (support) of the studied patients' Inv-nodes is 9 to 11, and they have no radiotherapy, so the possibility of tumor recurrence is 100% (confidence).

Based on Association Rules, the data mining results show that the higher Tumor-size and Deg-malig, the more Inv-nodes, the more chances of recurrence are; patients who have a smaller Tumor-size and receiving radiotherapy have a fewer possibilities of recurrence.

4. Conclusion

An Improved Apriori algorithm is proposed to reduce the size of candidate sets by studying on Apriori algorithm of Association Rules and the deficiencies of Apriori algorithm. Conclusions are made on association rules between tumor recurrence and other attributes by doing data mining on breast cancer patients provided by HIS using SQL Server 2005 Analysis Services. The results corresponding with the background knowledge of diagnosis of breast disease can be used as important references in breast diseases.

References

- Carlos Ordonez, Cesar A. Santana, & Levien de Braal.(2000). Discovering interesting association rules in medical data. In *ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD 2000)*, pp.78-85
- Daniel Kunkle, Donghui Zhang, & Gen Cooperman. (2008). Mining Frequent Generalized Itemsets and

Generalized Association Rules Without Redundancy. *Journal of Computer Science & Technology*, 23, 77-102.

J. Han, Y. Cai, & N. Cercone. (1992). Knowledge discovery in databases: An attribute oriented approach. *In Proc. of the VLDB Conference*, Vancouver, British Columbia, Canada, pp. 547-559.

M. J. Zaki. (2001). SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning Journal*, 42, 31-60.

Mohammed J. Zaki. (2000). Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12(3), 372-390.

R. Agrawal, T. Imielinski, & A. Swami. (1993). Mining Association Rules between Sets of Items in Large Databases, *SIGMOD Conference* pp.207-216.

Rakesh Agrawal, & Ramakrishnan Srikant. (1994). Fast algorithms for mining association rules in large databases. *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB*, pp. 487-499.

S. Muggleton, & C. Feng. (1992). E-cient induction of logic programs. In S. Muggleton, editor, *Inductive Logic Programming*. Academic Press.

Sriphaew K, & Theeramunykong T. (2003). Mining generalized closed frequent itemset items of generalized association rules. In Proc. *International Conference on knowledge-Based Intel-ligent Information and Engineering Systems (KIS)*, pp. 476-484.

Zelic I, Bercic B, Pikec M. & Slavec S. (2000).Implementation and deployment of healthcare management information system. *Stud Health Techno Inform*, 2(77), 799-803

Zhu Deli. (2007). *SQL Server 2005 Data Mining complete solutions and business intelligence*. Beijing: Electronic Industry Press, (Chapter 11).

probability	importance	rules
0.900	0.482	Inv-nodes = '6-8', Deg-malig = '3' -> Class = 'recurrence-events'
1.000	0.443	Inv-nodes = '9-11', Irradiat = 'no' -> Class = 'recurrence-events'
1.000	0.443	Tumor-size = '35-39', Age = '30-39' -> Class = 'recurrence-events'
0.529	0.427	Deg-malig = '3' -> Class = 'recurrence-events'
0.833	0.421	Inv-nodes = '6-8', Age = '50-59' -> Class = 'recurrence-events'
0.778	0.414	Inv-nodes = '6-8', Irradiat = 'yes' -> Class = 'recurrence-events'
1.000	0.411	Inv-nodes = '9-11', Age = '30-39' -> Class = 'recurrence-events'
1.000	0.411	Inv-nodes = '6-8', Tumor-size = '40-44' -> Class = 'recurrence-events'
0.655	0.402	Irradiat = 'yes', Deg-malig = '3' -> Class = 'recurrence-events'
0.692	0.381	Inv-nodes = '3-5', Deg-malig = '3' -> Class = 'recurrence-events'
0.700	0.374	Inv-nodes = '3-5', Tumor-size = '30-34' -> Class = 'recurrence-events'
0.647	0.363	Tumor-size = '25-29', Deg-malig = '3' -> Class = 'recurrence-events'
0.615	0.361	Tumor-size = '30-34', Deg-malig = '3' -> Class = 'recurrence-events'

Figure 1. The result of data mining based on Association Rules

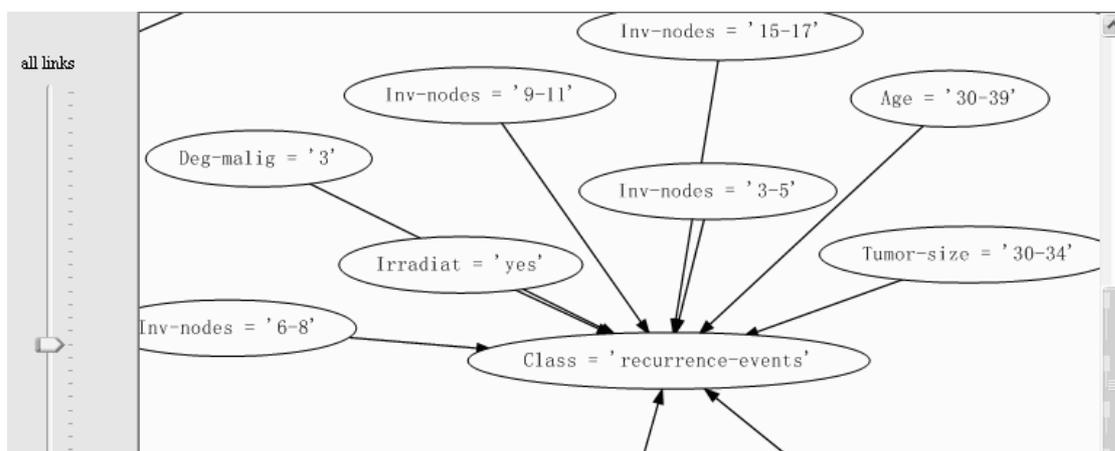


Figure 2. Relationship network of association