Mining Association Rules From Relational Data – Average Distance Based Method*

Vladimír Bartík¹, Jaroslav Zendulka¹

¹ Brno University of Technology, Faculty of Information Technology, Božetěchova 2,
61266 Brno, Czech Republic
{bartik, zendulka}@fit.vutbr.cz

Abstract. The paper describes a new method for association rule discovery in relational databases, which contain both quantitative and categorical attributes. Most of the methods developed in the past are based on initial equi-depth discretization of quantitative attributes. These approaches bring the loss of information. Distance-based methods are another kind of methods. They try to respect the semantics of data. The basic idea of the new method is to separate processing of categorical and quantitative attributes. The first step finds frequent itemsets containing only values of categorical attributes and then quantitative attributes are processed one by one. Discretization of values during quantitative attributes processing is distance-based. A new measure called average distance is introduced for these purposes. The paper describes the method and results of several experiments on real world data.

1 Introduction

At present, databases contain a huge amount of data and sometimes it is very difficult for people to deduce some conclusions from this data. Therefore, the use of knowledge discovery techniques is necessary. Knowledge discovery in databases is a process of derivation of useful information, which is not explicitly stored in a database but may be useful in practice.

Association rules are one kind of knowledge discovered in databases. They are typically used for market basket analysis, where transactional databases are used. A transactional database is defined as a set of transactions where a transaction is a set of items. Here, the goal of mining association rules is to find rules of a form $A \Rightarrow B$ where A and B are sets of items. If this association rule is found, it means: "If a transaction contains a set of items A , it is likely to contain a set of items B ". The sets A and B must be disjoint. We call them itemsets. An itemset that contains k items will be called k -itemset. Potential usefulness of a rule is usually expressed by means of two measures – support and confidence. Then, a frequent itemset is an itemset satisfying user-defined minimum support and a strong rule is a rule generated from frequent itemsets that satisfies also user-defined minimum confidence.

* This work has been supported by the Grant of FRVS MSMT, FR824/2003/G1, "Discovery of Association Rules In Relational Databases" and by the long-term grant project of Ministry of Education No. MSMT 262200012 "Research of information and control systems".

In the past, the research was mostly focused on methods for association rules in transactional databases. A lot of algorithms were proposed in this area. Probably the best known is Apriori algorithm [1] and its modifications [2]. The FP-tree method proposed in [3] and a graph-oriented method proposed in [4] are other examples.

The research of association rules in relational data is quite different. Here, the mining process is performed over a relational table containing only scalar values. There are two types of attributes in relational databases: categorical and quantitative. A categorical attribute (e.g. town) is an attribute the domain of which contains a finite set of values. On the other hand, the domain of a quantitative attribute (e.g. price) contains either an infinite set of values or the cardinality of the domain is “high”, and the ordering must be defined on the domain.

Mining on categorical attributes is the same as in transactional databases - one row in a relational table can be considered as a transaction in a transactional database. Therefore slightly modified algorithms for mining transactional data can be used.

However, this approach cannot be applied to process quantitative attributes. The reason is that, in general, only few association rules would be generated due to a low support of individual values of a quantitative attribute. As a result, the discretization of values of the attribute must be performed the objective of which is to decrease the number of “representative values” of the attribute. In this case, the “representative values” are intervals. Minimum support and confidence, that control mining, are not considered during discretization. Equi-depth bucketing method is used very often. The intervals satisfying minimum support may be too wide in this case. It can result in high loss of information by discretization. Therefore, an additional measure for discretization “quality” should be introduced.

The paper is organized as follows: Chapter 2 contains a brief overview of methods for mining association rules in relational databases containing quantitative attributes. A new measure called the average distance, which is used to control discretization and our new method based on this measure are introduced in Chapter 3. Results of experiments that show some properties of the method are presented in Chapter 4.

2 Related Work

The term *Quantitative association rules* was defined by Srikant and Agrawal in [5]. The quantitative association rules may contain equality predicates ($Attr=val$) and range predicates (intervals). To indicate the “quality” of an interval, a metrics called K-partial completeness was introduced. It was designed to ensure that intervals are neither too big, nor too small. The basic idea is as follows: if the interval is too big, it may hide rules between the sub-intervals. The method requires an initial discretization. It is the equi-depth one. This approach does not reflect the semantics of the data, for example the distance between the data points. The initial equi-depth discretization may cause that two very near values (data points) lapse into two different intervals.

A method for mining *Optimized association rules* was proposed by Fukuda in [6]. The method uses no additional measure for the interestingness of an association rule. Because the minimum support and confidence thresholds are antipodal, either so

called optimized support rules or optimized confidence rules are obtained with this method. The goal of the optimized support rules is to find rules with support as high as possible with respect to minimum confidence. Similarly, the optimized confidence rules maximize confidence of the rule while satisfying the minimum support requirement. The method proposed in [6] uses the equi-depth bucketing for the quantitative attribute discretization. The disadvantages of the method are the same as of the previous one.

The definition of association rules was extended by Zhang in [7]. The *Fuzzy Quantitative Association Rules* are introduced there. They may contain crisp values, intervals, and fuzzy terms in both antecedent and consequent. This approach combines interval partition with pre-defined fuzzy terms. Therefore, it is more general.

A *distance-based method*, in which the partitioning respects the semantics of numeric data, was proposed by Miller and Yang in [8]. The method uses a clustering method to find the intervals contained in an association rule. The algorithm works in two phases. In the first phase, the clusters over an attribute are identified using, for example, the algorithm Birch [9]. In the second phase, the clusters are combined to form rules using the classic Apriori algorithm. Frequent itemsets of clusters are the result of this phase.

A distance-based method that combines the approach of merging small intervals into larger ones with clustering was presented in [10]. This method is called *An Adaptive Method For Numerical Attribute Merging*. It takes into account both distance between data points and their density. First, each quantitative value is assigned to its own interval (containing only one value). Then, two hitherto most neighboring intervals are merged together. The criterion that controls the merging process respect both distance and density of quantitative values.

In this paper, another distance-based method that reflects the distances between data points will be proposed. The association rule generated by this method contains a value of a quantitative attribute (a “representative value”), which has a sufficient number of neighbors in a sufficiently short distance from it. A new measure called an average distance controls discretization. The method and the measure will be described in the next chapter.

Most of the distance-based methods were mainly developed to mine in values of quantitative attributes. Our method processes quantitative and categorical attributes separately. This fact allows us to use faster algorithms for categorical attributes processing.

3 Average Distance Based Method

3.1 Average Distance

Assume that we have a sorted array of quantitative values. We want to find a value, which has a number of neighbors in a short distance defined by a measure. The measure will be called the *average distance* (we will use the acronym AD for it).

The average distance of a quantitative value v will be defined as:

$$AD(v) = \sum_{i=1}^n \frac{(v - v_i)}{n} \quad (1)$$

where v is a quantitative value, v_i are the values that lie close to the value v , and n is the number of “close” values. The set of “close” values will be called a cluster. A cluster will be stored, for example, in the form of interval represented by its borders.

The number n , which is used to obtain AD, can be computed from the minimum support threshold. We can obtain it as:

$$n = \text{minsup} * \text{numrows} \quad (2)$$

where *minsup* is the user-defined minimum support threshold and *numrows* is the number of rows in the mined relational table. The AD of v is then the average distance from v of such number of values (data points) that satisfy the minimum support threshold.

We will introduce a maximum value of AD, which will be a parameter of a data mining task, that determines if v is an interesting value. If the average distance $AD(v)$ does not exceed the value of maximum AD, v will be called an interesting value.

Besides this parameter, we assume another parameter called *precision* (we will denote it P) of a value in an association rule. For example, if the quantitative attribute will be Age, the value ($\text{age}=19.7647$) is not interesting for the user. If the user will enter the precision $P=1$, the result will be an integer value, either ($\text{age}=19$) or ($\text{age}=20$). In other words, the value P determines the finest granularity of discretization.

Two parameters of an association rule mining task - maximum average distance (*MaxAD*) and precision P - defined in this chapter have to be entered by the user separately for each quantitative attribute in the table because each attribute in the table has different semantics. For example, the requirements of the user concerning an attribute *age* can differ from requirements concerning an attribute *price*.

3.2 Method Overview

Assume that there is a relational table containing both categorical and quantitative attributes. The result of the method is a set of association rules containing values of categorical and numerical attributes. An association rule may look like this:

$$(\text{car} = \text{"VW Golf"}) \Rightarrow (\text{age_of_customer} = 32 \text{ [MaxAD}=1, P=1])$$

But the term ($\text{age_of_customer}=32$) does not mean that this condition satisfies the minimum support threshold. It means that there is a satisfactory number of customers whose age is near the number ‘32’. That’s why the minimum support and confidence are not enough to say that the association rule is strong. Alternatively, the interval containing values close to the value ‘32’ may be used in the rule instead of this specific value.

A relational table that can contain both categorical and quantitative attributes, minimum support and confidence, maximum average distance, and precision are

inputs of the method. As mentioned above, the maximum average distance and precision must be entered for each quantitative attribute.

The task of mining association rules is performed in the following steps:

- **Categorical attributes processing:** Finding the frequent itemsets over the categorical attributes of the table.
- **Quantitative attributes processing** (getting frequent itemsets): Extending the frequent itemsets by adding quantitative items. The quantitative attributes are processed one by one. For each selected attribute and each frequent itemset from the previous step, its values are discretized first, based on the average distance measure, and the obtained interesting values are added to the itemset.
- **Association rules generating:** When all frequent itemsets are found, the set of strong association rules is generated by the well-known approach based on the minimum confidence entered by the user. This approach is straightforward and is described for example in [11].

3.3 Categorical Attributes Processing

The frequent itemsets over the categorical attributes can be obtained by very simple and well-known methods. In this case, we can view a row of the relational table as one transaction in a transactional database and use, for example, a slightly modified Apriori algorithm.

The result of this step is the set of all frequent itemsets over the part of the input table containing only categorical attributes. These itemsets will be extended with items of the quantitative attributes in the next step.

Notice that the categorical and quantitative attributes of the input table are processed separately. This fact significantly reduces the number of database scans because only fields in the table that are contained in categorical frequent itemsets are scanned in the following step of the process.

As a result of this step, we have a set of frequent itemsets containing only categorical values. We will call such itemsets the frequent categorical itemsets.

3.4 Quantitative Attributes Processing

Let us assume that the number of categorical items in frequent itemsets obtained in the previous step is k . We have frequent categorical itemsets, support of which is known. The quantitative attributes are processed one by one. For each frequent categorical itemset, values of a selected quantitative attribute from rows in which values of categorical attributes match the values in the actual categorical itemset are stored into an ordered number sequence. This sequence of numbers can be stored, for example, in a sorted array.

If we have this number sequence, we can find the interesting values for association rules using the average distance and precision measures. If any interesting value is found, it is added to the frequent categorical itemset, for which this sequence of numbers was constructed. The result will be a new frequent itemset.

Construction of a Number Sequence. For each frequent itemset, the algorithm scans the table. If the values in the row correspond to the values contained in the frequent itemset, the value from the quantitative column is added to the number sequence.

The algorithm looks as follows:

```

For each categorical frequent itemset FI do
    For each row R in the table do
        For each item i in FI do
            {
                if (i.catvalue <> R.catvalue) break;
                Insert_Into_Sequence (R.numvalue, Arr);
            }

```

The value of actual quantitative attribute is denoted as *numvalue*, the value of actual categorical attribute is denoted as *catvalue*. The procedure *Insert_Into_Sequence* will insert a new numeric value into the sorted array *Arr* of quantitative values.

Finding the interesting values. If we have the number sequence, we can find interesting values in it based on the average distance measure. The number of steps of this algorithm depends on the precision *P*.

The input of this algorithm is the sorted array *Arr*, maximum average distance *MaxAD* and the precision *P*. The output will be a set *S* of interesting quantitative values.

The algorithm can look as follows:

```

N=minsup*numrows;
Val=Round(Min_value_in_array, P);
While Val < Max_value_in_array
{
    RD=Count_AD (N, Arr, Val);
    If (AD<MaxAD) Add_To_Interesting(Val, S);
    Val=Val+P;
}
return (S);

```

At first, the algorithm sets the actual value *Val* to the lowest value in *Arr*. This value must be rounded with respect to the precision *P*. For each value, an average distance is counted by the procedure *Count_AD*. If the maximum AD threshold is satisfied, then the value *Val* is interesting and is added to *S* by the procedure *Add_To_Interesting*.

The procedure *Count_AD* can look as follows:

```

Procedure Count_AD (Arr, Val, n) {
    int p=Search_Hithermost(Val, Arr);
    int less=1; more=0; RD=0;
    for (i=0, i<=n, i++) {
        if ((Val-Arr[p-less])>=(Arr[p+more]-Val))
        { AD+=(Arr[p+more]-Val); more++; }
        else
        { AD+=(Val-Arr[p-less]); less--; }
    }
}

```

```

    }
    AD/=n;
}

```

The procedure *Search_Hithermost* searches the number sequence *Arr* for the value closest to the value *Val*. It returns the index of a field in the number sequence with a higher value closest to *Val* (or equal to *Val*).

If the interesting values are added as items to the competent frequent itemset, then we have a new set of frequent itemsets, which contain values of one quantitative attribute.

3.5 Frequent Itemsets With Several Quantitative Attributes

If the input table contains more than one quantitative attribute, we have to continue computing frequent itemsets in additional iterations. Each iteration adds one interesting value of another quantitative attribute to the existing frequent itemset.

Assume that we have a frequent itemset (we will use the 2-itemset $\{(car="VW Golf"), (age_of_customer=32)\}$ as an example). The process will be the following:

1. Let f_i be a frequent itemset ($\{(car="VW Golf"), (age_of_customer=32)\}$) containing a value v (32 in our example) of a quantitative attribute q_1 ($age_of_customer$ in our example). Let c be a cluster of values of the attribute represented by the value v (this cluster is stored together with the "representative value" v during the previous steps).
2. Select a quantitative attribute (for example *price*) from the attributes that have not been processed yet.
3. Construct a number sequence for the attribute q_2 containing the values of q_2 from those rows of the table the value of which in the attribute q_1 is present in the cluster c .
4. Find the interesting values in the number sequence constructed in step 3 for the attribute q_2 (this procedure of finding the interesting values is described in the previous section).
5. Create a new set of frequent itemsets, constructed from the itemset f_i by adding the interesting values (from step four) to it (in our example, the new 3-itemset may be, for example, $\{(car="VW Golf"), (age_of_customer=32), (price=10.000\$)\}$).
6. Repeat the steps 2, 3, 4 and 5 for each quantitative attribute of the table that has not been processed yet.

The processes of finding the interesting values are repeated for all frequent itemsets that contain values of quantitative attributes.

If there is no frequent $(k+1)$ -itemset generated from a set of frequent k -itemsets, the algorithm stops and the actual set of frequent itemsets is a result of the process.

Notice that the generated quantitative items in itemsets may depend on the order of quantitative attributes processing. This implies that we can obtain different results for different orders of processing.

4 Experimental Results

The first version of average distance based method was implemented as a part of a data mining system [12]. The implementation was made in Java. A database containing data from a medical study¹ was used for experiments.

The association rule mining task contained a relational table with four attributes. Two of them were categorical (frequency of drinking alcohol and physical activity in work) and the other two attributes were quantitative (weight and systolic blood pressure). The input relational table contained 1141 rows.

At first, we searched the dependences between time complexity and some user-defined parameters of a task (minimum support, maximum average distance and precision). These dependences are depicted in Figure 1.

Fig. 1. Dependency between time complexity of the method, minimum support and maximum average distance (value of average distance was the same for both quantitative attributes)

The experiments confirmed the assumption that processing of categorical attributes is much faster than processing of quantitative attributes. A classic Apriori algorithm was used to process the categorical attributes. Times of processing of both kinds of

¹ The study (STULONG) was realized at the 2nd Department of Medicine, 1st Faculty of Medicine of Charles University and Charles University Hospital, U nemocnice 2, Prague 2 (head. Prof. M. Aschermann, MD, SDr, FESC), under the supervision of Prof. F. Boudík, MD, ScD, with collaboration of M. Tomečková, MD, PhD and Ass. Prof. J. Bultas, MD, PhD. The data were transferred to the electronic form by the European Centre of Medical Informatics, Statistics and Epidemiology of Charles University and Academy of Sciences (head. Prof. RNDr. J. Zvárová, DrSc). The data resource is on the web pages <http://euromise.vse.cz/challenge2003>. At present time the data analysis is supported by the grant of the Ministry of Education CR Nr LN 00B 107.

attributes are shown in Figure 2. It indicates that separated processing may be advantageous.

As mentioned above, the result of mining can depend on the order of processing of quantitative attributes. Therefore we tried to find out a dependency between resultant association rules, the order of processing and values of other parameters (e.g. maximum average distance). The results of these experiments showed that it is better to process first the attributes with higher maximal average distance, lower precision and lower range of values. The solution of this problem may be a heuristics, which will determine the order of attributes to be processed. This heuristics may be based on three parameters mentioned above - maximum average distance, precision and range.

Fig. 2. Differences between time complexity of categorical (Phase 1) and quantitative attributes processing (Phase 2)

5 Conclusion

A new method for discovery of association rules in relational databases, where the association rule contains both categorical and quantitative items was proposed in this paper. A new measure called average distance for interestingness of a value was introduced. The measure is used for discretization of quantitative attributes.

The presented method has two main advantages. First, the results of this method may contain association rules with quantitative items in the form (*Attr=Value*). This expression associated with the average distance may give more useful information than an interval, which is used as a result of most of other methods. But it is possible to use intervals in the resultant association rules too.

The second advantage is the separation of categorical and quantitative attributes processing. Some advanced (and faster) methods may be used for processing of categorical attributes. This fact is of good promise to make this method more effective compared to processing both kinds of attributes together. The experiments confirmed that processing of categorical attributes takes much shorter time.

Nevertheless, there are some topics for further research. The first one concerning dependency on the order of quantitative attributes processing was briefly discussed in chapter 4. One of the possible solutions is to find heuristics that determines the best order. This heuristics may include the data mining task parameters entered by the user for the attributes and some properties of values of the attributes.

Another topic for the future research is to find more effective structure and algorithms for searching interesting values. The array presented in this paper was used only to demonstrate the principles of the method and for verification of the method functionality.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules Between Sets of Items In Large Databases. Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington, DC, 1993.
2. Kotásek, P., Zendulka, J.: Comparison of Three Mining Algorithms For Association Rules, Proceeding of MOSIS 2000 – Information Systems Modeling ISM 2000, Rožnov pod Radhoštěm, Czech Republic, 2000, pp. 85-90
3. Han, J., Pei, J., Yin, Y.: Mining Frequent Patterns without Candidate, Proc. 2000 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD'00), Dallas, TX, May 2000.
4. Lee, G., Lee, K., L., Chen, L., P.: Efficient Graph-Based Algorithms for Discovering and Maintaining Association Rules in Large Databases, Knowledge and Information Systems, Springer-Verlag London Ltd., 2001, pp. 338-355.
5. Srikant, R., Agrawal, R.: Mining Quantitative Association Rules In Large Relational Tables. Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, 1996.
6. Fukuda, T, et al.: Mining Optimized Association Rules For Numeric Attributes. Proceedings of ACM PODS'96, Montreal, Canada, 1996, pp. 182-191.
7. Zhang, W.: Mining Fuzzy Quantitative Association Rules. Proceedings of 11th IEEE International Conference on Tools with Artificial Intelligence, IEEE Computer Society, Chicago, Illinois, 1999, pp. 99-102.
8. Miller, R. J., Yang, Y.: Association Rules Over Interval Data. Proceedings of 1997 ACM SIGMOD, Tucson, Arizona, USA, 1997, pp. 452-461.
9. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: An Efficient Data Clustering Method For Very Large Databases. Proceedings of the ACM SIGMOD International Conference on Management of Data, Montreal, Canada, 1996.
10. Li, J., Shen, H., Topor, R.: An Adaptive Method of Numerical Attribute Merging for Quantitative Association Rule Mining, Proceedings of the 5th international computer science conference (ICSC), 1999, pp. 41 - 50, Springer.
11. Hua, Z.: On-Line Analytical Mining of Association Rules, PhD. thesis, Simon Fraser University, 1998.
12. Bartík, V.: Association Rule Discovery In Databases, Proceedings of 5th International Conference ISM'02, Rožnov pod Radhoštěm, Czech Republic, 2002, pp 21-27.