Quantitative and Ordinal Association Rules Mining (QAR Mining)

Filip Karel

Department of cybernetics, Czech Technical University in Prague,
Technická 2, Praha 6, 166 27
karelf1@fel.cvut.cz

Abstract. Association rules have exhibited an excellent ability to identify interesting association relationships among a set of binary variables describing huge amount of transactions. Although the rules can be relatively easily generalized to other variable types, the generalization can result in a computationally expensive algorithm generating a prohibitive number of redundant rules of little significance. This danger especially applies to quantitative and ordinal variables. This paper presents and verifies an alternative approach to the quantitative and ordinal association rule mining. In this approach, quantitative or ordinal variables are not immediately transformed into a set of binary variables. Instead, it applies simple arithmetic operations in order to construct the cedents and searches for areas of increased association which are finally decomposed into conjunctions of literals. This scenario outputs rules that do not syntactically differentiate from classical association rules.

Keywords: association rules, quantitative attributes, ordinal attributes.

1 Introduction

The discovery of quantitative and ordinal association rules in large databases is considered as an interesting and important research problem. Recently, different aspects of the problem have been studied, and several approaches have been presented in the literature. Authors mostly agree that standard algorithms are ineffective and often provide us with illogical or unusable results. Quantitative or ordinal attributes bring many specifics. One of them is the optimalization of discretization of quantitative attributes. Another question is how to measure the quality of generated rules. Some authors use classical measures like support and confidence [3],[4],[8], other authors point out that in the case of quantitative it is better to use different measures [5],[9],[10]. Principle of proposed approaches is almost always very similar. It is based on finding suitable intervals of attributes which are then used in some kind of binary tests which are the basic principle of classical association rules mining. Distinctions among authors are in the methodology of selecting intervals and in measures used to evaluate these intervals. In general the authors show better results of their algorithms and more intuitive concept, but they are not primarily focused on time consumption and also they

B. Gabrys, R.J. Howlett, and L.C. Jain (Eds.): KES 2006, Part I, LNAI 4251, pp. 195–202, 2006.

often do not provide a complex algorithm. Authors in [11] discus this problem, but do not provide any algorithm which saves time during QAR mining. Authors also do not address the problem of attributes combination in detail.

In this paper an innovative algorithm for QAR mining is proposed. The main advantages of this algorithm are significantly lower time consumption and a reduction of the number of redundant rules. On the other hand, as this algorithm is not based on a complete search through the state space it cannot be guaranteed that all rules are found.

Association rule is an implication $X \rightarrow Y$. The left hand side of this implication is called *antecedent*, the right hand side is *succedent*. Generally antecedent and succedent can be called *cedents*. Cedents are constructed of *attributes*, which are included in the database. Cedent can consist of one attribute - *trivial cedent*, or more attributes - *non-trivial cedent*.

The rest of paper is organized as follows. Section 2 provides an overview of different types of attributes and describes the preprocessing process. Section 3 is focused on combination of attributes into non-trivial cedents. The following section 4 examines areas of interest and describes how to decompose the non-trivial cedents and how to select valid rules. Section 5 gives examples of experimental results, summarizes the paper and discusses open problems.

2 Attributes' Preparation and Preprocessing

The data mining literature contains a variety of terms describing different types of data or attributes. In this paper the following division is used:

Qualitative attributes - they are only divided in categories, but not numerical measures

- nominal - attributes that are exhaustively divided into mutually exclusive categories with no rankings that can be applied to these categories (names, colors)
- ordinal - the categories into which they are classified can be ordered (evaluation of an action = {very good, good, bad, very bad})

Quantitative attributes - attributes that are measured on a numerical scale and to which arithmetic operations can be applied

- discrete - have a measurement of scale composed of distinct numbers with gap in between (number of cars)
- continuous - can ideally take any value (height, distance)

The proposed algorithm deals with qualitative ordinal, and quantitative discrete and continuous attributes.

Attributes can take different values and different number of values. These differences among attributes can very heavily influence the results of association rules mining, so it is necessary to preprocess them before combining. The common methods for attributes adaptation can be used - **normalization** and **discretization**.

Using normalization we reach all attributes take values in the same range. All attributes are normalized to values between 0 and max_value.

There are many advantages of using discrete values over continuous ones. Discrete features are closer to a knowledge-level representation than continuous ones. For both users and experts, discrete features are easier to understand, use, and explain. As reported in [12], discretization makes learning more accurate and faster. Discussing advantages and disadvantages of different types of discretization is out of the scope of this paper. Generally speaking, the higher degree of discretization (number of discrete values), the higher the computational costs. The lower degree of discretization, the higher information loss. One of the main advantages of algorithm proposed in this paper is its low time consumption. We can therefore afford higher degree of discretization.

3 Combining Attributes

To create non-trivial cedents, several attributes have to be combined together and they have to be represented by one number.

Combination can be realized by basic operations like adding and subtraction, which are simple enough and they are not increasing time consumption of the whole algorithm. Moreover, the following decomposition of the non-trivial cedent is simple when these operations are used. We can see on figure 1 there can be quite big differences among non-trivial cedents' histograms depending on the way how cedents are created. If there are only two operations (addition and subtraction) then we have 2^{n-1} of combinations how to create a non-trivial cedent, where n is the number of attributes in a non-trivial cedent. We have $\binom{m}{n}$ combinations of attributes which can create the non-trivial cedent, where m is total number of attributes in the database. Both antecedent and succedent have to be created so overall time consumption is $2^{n_a+n_s-2} * \binom{m_a}{n_a} * \binom{m_s}{n_s}$. Testing all possibilities could therefore be very time demanding.

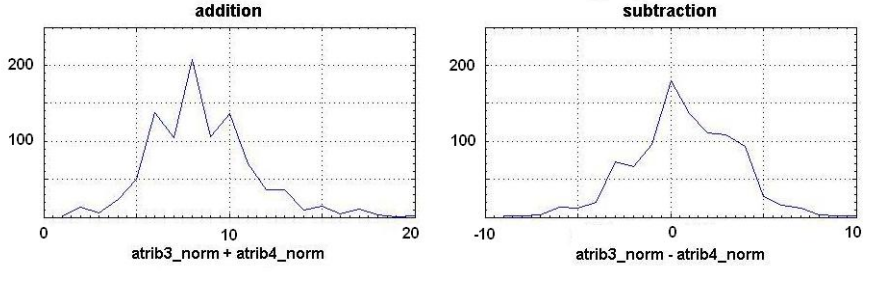

Fig. 1. Cedents histograms

Experiments have demonstrated it is more suitable to assess the sign to an attribute on the antecedent side in order to how it influences attribute(s) on

the succedent side (positively or negatively). For this decision a non-parametric *spearman rank correlation coefficient* is used as follows. The succedent is created by addition of all attributes. Then, spearman rank correlation coefficient is computed with each of the antecedent attributes. If the coefficient is positive (negative) then + (-) is assigned to the antecedent attribute.

If we combine attributes without normalization and discretization, the histogram of the resulting cedent is influenced more by attributes with high number of high values. This fact can heavily influence further rules generation. If we combine normalized attributes, all of them have equal weight and resulting cedent is influenced by all attributes equally. In following discretization, numbers of records in particular discretization bins are more balanced.

4 Areas of Interest

This section describes how to identify areas of interest and how to mine rules from these areas of interests. After constructing the non-trivial cedents, the contingency table of differences between real and expected values is used. The size of the table is $\sum_{i=1}^{n} D_A(i)$ x $\sum_{i=1}^{m} D_S(i)$, where n is the number of attributes creating antecedent, m is the number of attributes creating succedent, D_A is the degree of discretization of the i-th antecedent attribute, D_S is the degree of discretization of the i-th succedent attribute.

The interesting areas to identify are the areas with positive values. In these areas there are more records than expected in case of cedents independence. In most cases, all these areas cannot be examined because of huge time consumption. So the task is to identify the most interesting areas, this means the largest areas with the highest positive values.

For this task the standard clustering algorithm is followed. To increase the quality of generated rules and to reduce the number of redundant and similar rules, points 5 and 6 are added.

1. Place K points into the space represented by the objects (points with positive values) that are being clustered. These points represent initial groups' centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids (we use weightening according to the value of each point, the biggest value the more it influences the position of centroid).
4. Repeat Steps 2 and 3 until the centroids no longer move.
5. Move all centroids to the nearest point with the highest value if it is in some defined neighborhood.
6. If there are two (or more) centroids in each others' neighborhood, keep just one with the highest value.

After the clustering algorithm finishes, we have the position of final centroids in the matrix (contingency table). First point's coordinate represents the value of an antecedent, second represents the value of a succedent. We receive K points coordinates, where K is number of interesting areas we want to gain.

Decomposition points' coordinates into the concrete values of antecedent's and succedent's attributes has to follow now. I demonstrate it on the example of a non-trivial antecedent made of two attributes and non-trivial succedent made of 2 attributes. The degree of discretization is 10. Therefore the contingency table of differences has size of 20 x 20. Let us decide to gain just one interesting area and suppose that we received the centroid coordinates [13, 5], where 13 represents antecedent axis and 5 succedent axis. We now have to decompose these coordinate's values into the attributes' values.

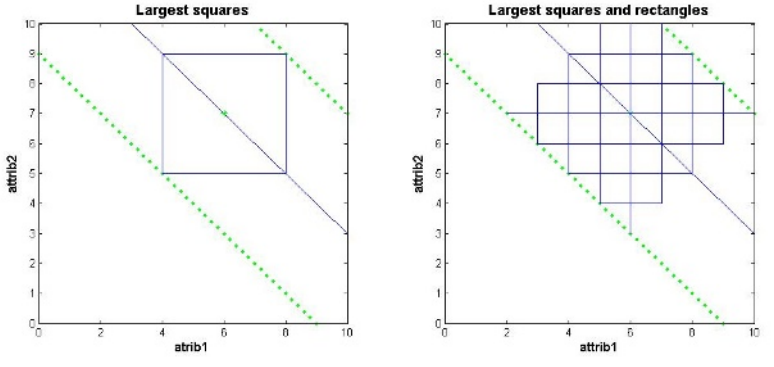

Fig. 2. Antecedent decomposition

We can imagine the situation in 2D space - Figure 2 (for more attributes, more dimensional space is used). Coordinates X and Y represents the two antecedent attributes. The full line represents points which satisfies the condition $attrib_1 + attrib_2 = 13$. Some tolerance area have to be set up to gain rules with sufficient support and to use the positive points in the centroid surrounding. The tolerance area is represented by dotted lines. All points in this area satisfies condition $10 \leq (attrib_1 + attrib_2) \leq 16$.

There are 3 options how to go through the space bordered by the lines

1. **Largest squares** - we take into account only the largest squares, we achieve lowest time consumption - $n_{ver} \approx (X_{max} - X_{min})^{D-1}$, where n_{ver} is number of verifications. We have of course limited resolution, The example square on the left side of figure 2 illustrate $attrib_1 = 4..8$ & $attrib_2 = 5..9$.
2. **Largest squares and rectangles** - we take into account all largest squares and rectangles, time consumption is higher, $n_{ver} \approx ((X_{max} - X_{min} * tol\_width))^{D-1}$. The example square and rectangles on the right side of figure 2 illustrate possibilities from $attrib_1 = 5$ & $attrib_2 = 3..10$ to $attrib_1 = 2..10$ & $attrib_2 = 7$.
3. **All squares and rectangles** - we consider all squares and rectangles in the area, time consumption of this possibility is the highest of all possibilities, $n_{ver} \approx (X_{max} - X_{min})^{tol\_width+(D-1)}$.

Obviously, the highest time consumption, the better resolution is obtained. But even the most time consuming option has lower time consumption in comparison with classical approaches.

All the combinations of antecedent and succedent squares and rectangles have to be verified for the valid rules. There are many measures for quality of association rules described in literature. I'm using the basic measures of *support* and *confidence* [1] ensembled by the *lift* measure [2]. One rule is described by one square (rectangle) of antecedent side and one square (rectangle) from succedent side of a particular centroid.

Maximum number of verifications between one antecedent and succedent is given by

$$n_{ver} = \sum_{i=1}^{K} n_{antrect}(i) * n_{succrect}(i),$$

where K is number of interesting areas we want to identify in the contingency table of differences, $n_{antrect}$ is number of squares (and rectangles) gained from i-th centroid antecedent coordinate, $n_{succrect}$ is number of squares (and rectangles) gained from i-th centroid succedent coordinate.

To prevent generation of too many redundant rules and reduce number of generated rules to minimum while keeping all dependencies

- minimum thresholds for support, confidence and lift (*minsupp, minconf* and *minlift*) are set up,
- from each area represented by one centroid only two best rules are selected - one with the highest confidence (lift), which also satisfies condition *minsupp* and one with highest support, which also satisfies conditions *minconf* and *minlift*.

The number of rules is limited by $2 * K$ for each atecedent-succedent combination. For the whole algorithm there are

$$n_{rules} \leq 2 * K * n_{comb},$$

where n_{rules} is number of rules, n_{comb} is number of possible antecedent-succedent combinations, $n_{comb} \leq \binom{m_a}{n_a} * \binom{m_s}{n_s}$.

So the number of generated rules can be influenced through parameter K. If we want to have just a few rules and gain the main dependencies in data, we set K lower, if we want to gain all valid rules we set up K higher.

5 Experimental Results and Conclusions

An innovative algorithm for QAR mining was introduced in this paper. This algorithm consists of four basic steps.

1. Preprocessing of attributes (normalization and discretization).
2. Non-trivial cedents creation (addition and subtraction).

3. Identification of interesting areas in contingency table.
4. Cedents decomposition into rules and best rules selection.

The comparison of classical mining and QAR mining was made over the STULONG data domain [13], [14]. It concerns a dataset describing the data collected during a longitudinal study of atherosclerosis prevention on around 1400 middle-aged men. I worked with data representing entry examination of men, ten basic attributes from the database were selected and I tried to find association among selected attributes.

Time consumption is represented by number of verifications, i.e. number of candidate rules tested for the *minsupp*, *minconf* and *minlift* thresholds. On figure 3 we can see how number of verifications depends on number of attributes in database and how number of valid rules depends on interval length. Interval length represents maximum range of one attribute in the rule. For example if we have interval length three, attribute in rule can take values 1..4, 2..5, ... and it can't take values 1..5, 2..6, ... Non-trivial antecedent made of two attributes and trivial succedent were used in this experiment.

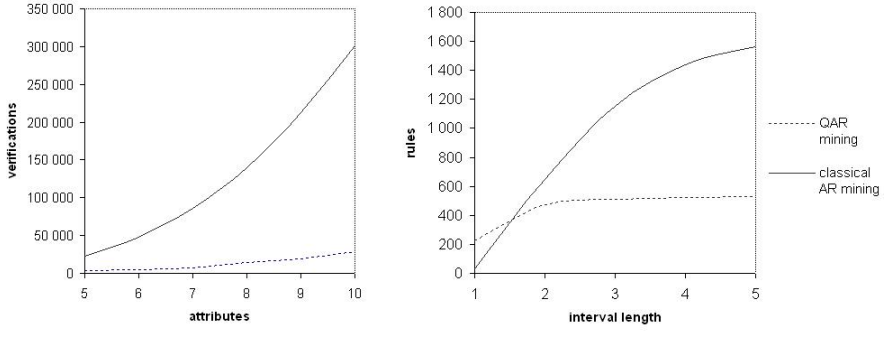

Fig. 3. Experimental results

The number of verifications grows very quickly with the number of attributes. For ten attributes we have over 300 000 verifications using classical approach, while the number of verifications using QAR mining approach is approximately ten times lower. The time needed for non-trivial cedent construction, interesting areas identifying and cedents decomposition have to be added to the QAR approach time consumption. Still the time consumption reaches approximately 15% of the classical approach time consumption.

The number of rules is higher using classical approach but number of dependencies described by these rules is almost the same. This means that QAR mining approach reduces the number of redundant rules and contributes to the transparency of generated rules.

The main disadvantage of the proposed algorithm is that it is not based on complete searching. It cannot be guaranteed that all rules which satisfy condi-

tions of *minsupp*, *minconf* and *minlift* are found. Practical experiments demonstrated that in most cases 90-95% of all dependencies are identified. The number of rules to describe the dependecies is significantly lower. Positive characteristics of proposed algorithm can be more valued in working with large databases with high number of attributes or when looking for more complicated dependencies with more attributes on antecedent or succedent side.

Acknowledgement. This work was supported by the Universities Development Fund CR, project nr. 877 /2006.

References

1. AGRAWAL R., IMELISKI T., SWAMI A., *"Mining Association Rules Between Sets of Items in Large Databases"*, In Proc. of ACM SIGMOD Conference on Management of Data, pages 207-216, Washington, D.C., 1993.
2. PIATETSKY-SHAPIRO G., *"Discovery, analysis, and presentation of strong rules"*, in Knowledge Discovery in Databases, Cambridge, MA: AAAI/MIT, 1991, pp. 229-248.
3. SRIKANT R., AGRAWAL R., *"Mining Quantitative Association Rules in Large Relational Databases"*, In Proc. of ACM SIGMOD Conference on Management of Data, Montreal, Canada, 1996.
4. FUKUDA T., MORIMOTO Y., MORISHITA S., TOKUYAMA T., *"Mining Optimized Association Rules for Numeric Attributes"*, In Proc. of ACM SIGMOD Conference on Management of Data, Montreal, Canada, 1996.
5. MILLER R.J., YANG Y., *"Association Rules over Interval Data"*, In Proc. of ACM SIGMOD Conference on Management of Data, Tuscon, AZ, 1997.
6. IMBERMAN S., DOMANSKI B., *"Finding Association Rules from Quantitative Data Using Data Booleanization"*, 1999.
7. WEBB G.I., *"Discovering Associations with Numeric Variables"*, ", In Proc. of ACM SIGMOD Conference on Management of Data, San Francisco, CA, 2001.
8. RASTOGI R. SHIM K., *"Mining Optimized Association Rules with Categorical and Numeric Attributes"*, IEEE Transactions on Knowledge and Data Engineering, vol. 14, no. 1, 2002.
9. GUILLAUME S., *"Discovery of Ordinal Association Rules"*, In Proceedings of the Sixth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'02), 322-327, Taipei, Taiwan, 6-8 May 2002.
10. AUMANN Y., LINDELL Y., *"A Statistical Theory for Quantitative Association Rules"*, Journal of Intelligent Information Systems, vol. 20, 255-283, 2003.
11. WIJSEN J., MEERSMAN R., *"On the Complexity of Mining Quantitative Association Rules"*, Data Mining and Knowledge Discovery, 2, 263–281, 1998.
12. DOUGHERTY J., KOHAVI R., SAHAMI M., *Supervised and unsupervised discretization of continuous features*, Proceedings of the Twelfth International Conference on Machine Learning (pp. 194–202), Tahoe City, CA, 1995.
13. STULONG project, WWW page, http://euromise.vse.cz/stulong.
14. KLEMA J., NOVAKOVA L., KAREL M., STEPANKOVA O., *Trend Analysis in Stulong Data*, In Proceedings of ECML/PKDD'04 Discovery Challenge, 2004.