Interpretable Associations over DataCubes: Application to Hospital Managerial Decision Making

Miguel PRADOS DE REYES a Carlos MOLINA b,1, Belén PRADOS c and Carmen PEÑA YAÑEZ a

c Department of Computer Sciences, San Cecilio Hospital, Granada, Spain

da Department of Computer Sciences, University of Jaen, Jaen, Spain

b Department of Software Engineering, University of Granada, Granada, Spain

Abstract. The world concern about the costs of the health care systems has raised the importance of counting on precise and interpretable tools, that help the health care institution’s managers to make decisions to optimize the use of health resources. In this paper we propose a new Classification based on Association Rules (CAR) algorithm that improves the interpretability of the results, making it specially useful for decision making. Changing the usual way to obtain the rules we follow four goals: first to improve the interpretability of the result by obtaining rules meaningful and interpretable by themselves, secondly to reduce the complexity of the result obtaining a lower number of rules; thirdly, to obtain simpler rules, with less size in number of antecedents; and finally to avoid the usual over-fitting problem of the classification methods by obtaining a generic final result set, where specific rules for specific cases are avoided unless they are necessary. To prove the utility of our proposal we have used it in an example of decision support regarding the planning of the surgery rooms.

Keywords. CAR, Complexity Reduction, Classification, DataCube, Multidimensional Model, Interpretability Improvement.

Introduction

Today it is undeniable the world concern about the costs of the health industry [1] and the need to count on appropriate mechanisms to plan, invest and control properly the available funding. However, the huge amounts of data produced in this sector makes difficult to process and understand the information. This is where data mining methods play a big role, as support in the decision making [2, 3, 4].

There is a wide variety of data mining methods. As Yoo et al. states in [5], they can be divided into three categories. Firstly there are classification methods where a set of attributes must be previously selected to perform the classification according to them. This is the case of the methods based on neural networks, support vector machine, bayesian algorithms and decision trees, among others. In the second place can be found algorithms for clustering, used mainly when there is no previous information
about the objects to be classified. In this category there are basic, hierarchical and partition based clustering techniques. Finally can be found techniques that, in addition to classify, are useful to discover relations between data items, like the classification based on association rules (CAR) proposals.

In the last few years important advances have been performed (e.g. [6]) in the application of data mining methods over health care information with research purposes or to support medical diagnosis (e.g. [7]). Nevertheless as Jensen et al. indicate in [8], this is still an under-used source of information, mainly due to the nature of the medical data itself, as well as to the difficulties for the user to understand results of the data mining techniques, and therefore make the most of them.

This fact is even worse when we focus on health care management decision support where, to our best knowledge, there are quite few academic proposals [9,10], but they are more related to the business process [9] and its framework [10] than to the decision making support. Most of the hospital information systems, in the best of the cases, use commercial software packages. However, the information available about them is the one in their web sites, usually related to their advantages and successes, and in most of the cases there is no mention about the algorithms used. Nevertheless it is well known [5] that amongst all the data mining techniques the most comprehensible are those based on trees [11], for the structure of the results; while the most precise ones are the CAR methods, since they are exhaustive and explore all the possibilities in the classification and prediction processes. It makes them the most appropriated to take advantage of all the potential that the health information provides.

The main inconvenience of these methods is, as pointed above, that their results are difficult to understand. This is due to the rule set obtained with CAR proposals, that is used as an ordered list of rules. This set is checked in order, discarding the rules unsatisfied and stopping the process at the first one satisfied. In real systems the antecedents of these rules are usually complex and not directly meaningful, so the user must take into account the previously discarded rules to understand the full meaning of the classification performed. Another problem that classification methods (not only CAR based) must face is the over-fitting of the data.

Here we propose and exemplify a technique to solve these problems and achieve the four goals mentioned above, based on three key points: the exploitation of the hierarchies defined over the dimensions of the data cubes, the change in the way the association rules are obtained, and also a change in the evaluation of the rules at the classification process.

In section 1, we show the adaptation of a previous algorithm to reduce the complexity of the resultant set of rules. An example about the performance of our proposal are presented in section 2, and finally we indicate our conclusions and final remarks in section 3.

1. Algorithm

Our proposal is based on an improvement of the Complexity Guided Association Rules Extraction method (COGARE) presented in [12]. Due to space limitation, in this section we just indicate the modifications required to extract association rules adapted for classification

*Itemset generation.* When generating the frequent itemsets we only change the candidates: the method only considers as candidates the itemsets containing an item of
the class variable. If the itemset is not frequent the method applies the generalization process but not on the class item (the granularity of this variable does not change in any moment).

Rule generation. At the rule generation stage, the method only takes into account the association rules of the form \( A \rightarrow C \), where \( C \) only has one item that belongs to the class variable. So for each frequent itemset found, the method only generates one rule.

Rule generalization. In the rule generalization phase the scheme is similar to the one used in the itemset generation, because the method tries to generalize the rule set considering all the variables but the class.

Iterative approach. One of the main changes of the algorithm is the use of an iterative approach. Most of the proposals first generate all the association rules and later select the classification rules from this set. In our approach we use the following iterative approach:

- First the method generates frequent itemsets of size 2 (one item from the class variable and another one from other variable).
- Then it generates the association rules of size two, pruning those with a certainty factor (CF) under the threshold given as parameter, and selects the rules from this set. Hence the method generates a classification model using simple rules (one antecedent and the class).
- Next the method calculates the frequent itemsets of size 3 using frequent itemsets of size 2. In this stage the method tries to reduce the number of rules deleting unnecessary frequent itemsets, as explained in the next subsection.
- After it the method generates the rules for the frequent itemsets of size 3, and selects the rules to classify.
- The method stops if the result is not better than using rules of size 2. If we obtain a better classification then we continue the process with itemsets of size 4, and so on.

Space reduction. The aim of this process is to reduce the candidate frequent itemsets to be generated. The main idea is to avoid building more specific rules/bigger itemsets in areas of the search space where the classification was made correctly with simpler rules.

The method takes the current classification rule set, and looks for areas correctly classified with the current rule set. Those areas are delimited by the rules with CF equal to 1 (all the elements which satisfy the antecedent of the rule have the same class). Next the process keeps out from the frequent itemsets those including the antecedent of any of these rules, and in the next iteration no itemset in those areas will be considered.

Avoiding the generation of larger and more complex rules in these areas, the use of unnecessary inserted items in the antecedent of the rules is also avoided, and hence the over-fitting they could introduce.

Rules selection. On each iteration the algorithm selects a set of rules to build the classifier. First the method sorts the rules according to the quality of each one in decreasing order (better rules first). Then it uses the first rule as initial classifier a computes its quality. The rest of the process checks iteratively whether the next rule improves the quality or not. If there is an improvement the technique selects the rule and proceeds to the next iteration. Otherwise, the rule is dismissed.
1: \( N \) = Number of variables

2: \( \text{FI}_2 \) = Generate frequent item sets of size 2

3: \( \text{R}_2 \) = Generate rules from \( \text{FI}_2 \)

4: \( \text{R}_1 \) = Sort \( \text{R}_2 \) according to rules quality (CF) descending

5: \( \text{R}_a \) = Selected rules from \( \text{R}_1 \)

6: \( \text{FI}_2 \) = reduceSpace(\( \text{FI}_2 \), \( \text{R}_a \))

7: for \( i = 3 \) to \( N \) do

8: \( \text{FI}_i \) = Generate frequent item sets of size \( i \) from \( \text{FI}_{i-1} \)

9: \( \text{R}_i \) = Generate rules from \( \text{FI}_i \)

10: \( \text{R} = \bigcup_{j=2}^{i} \text{R}_i \)

11: \( \text{R}' = \text{Sort} \text{R} \) according to rules quality descending

12: \( \text{R}_c = \text{Selected rule from} \text{R}' \)

13: if \( \text{Quality}(\text{R}_c) > \text{Quality}(\text{R}_a) \) then

14: \( \text{R}_a = \text{R}_c \)

15: \( \text{FI}_i = \text{reduceSpace}(\text{FI}_i, \text{R}_a) \)

16: else

17: Finish and return \( \text{R}_a \)

18: end if

19: end for

20: return \( \text{R}_a \)

---

**Rule evaluation.** As mentioned above in most of the CAR proposals the rule set obtained is used as an ordered list of rules. In our proposal, we do not establish an order between rules since we use a classic inference schema, considering that a rule is satisfied with a degree \( \oplus (\mu(A_1), \ldots, \mu(A_n)) \), where \( A_S \) are the items in the antecedent and \( \mu(A_i) \) the degree at which the record satisfies the item. As t-norm we use the minimum. We combine the result of each fired rule with the same consequent (class) with a t-conorm to obtain one unique value for each class (we use the maximum). The class with higher degree is the one selected to classify the new record.

2. **Examples**

Due to the lack of proposals designed to work over datacubes with which to perform a comparison, we test the validity of the proposed method in the well known Wisconsin Diagnostic Breast Cancer (WDBC) at UCI. Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

This schema is defined over data collected for non-postponed operations which were carried out in hospitals in Granada between 2002 and 2004. For the facts, we only consider the data when the patients are from Granada. There are 50,185 facts with one variable (amount) and 6 dimensions (Figure 1).

**Classification** With this DataCube we want to know the estimated duration of operations to be able to shuffle the surgery room in an efficient manner. So, we need to get the value for level range in dimension Duration considering the time in four classes: 0-1 hours, 2-4 hours, 5-10 and 10 or more hours. The main parameters for the process are the following:

- Support: 0.005
- CF: 0.1
To test the quality of the method we apply 10-fold cross-validation getting an average correct classification of 95.129%. As can be seen, the method gets a high accuracy.

3. Conclusions

In this paper we have presented a modification of the COGARE algorithm for the extraction of interpretable association rules. With it four goals are achieved: having simpler rules, getting a lower number of them, improve the interpretability and comprehension of the system, and increase the the quality of the classifier avoiding the data over-fitting.

Finally we have shown that it is possible to use this approach for decision support, since each rule is a result per se that can be interpreted independently. It means that each rule helps to understand not only the criteria under which the classification is being performed, but also the characteristics that determine each class. As an example we have tested it on a surgery planning scenario.

This research is still in progress and the algorithm is being tested under more data sets to be able to quantify the improvements in the above mentioned goals.

References


