Mining Association Rules for Ordinal Data Classification using an Unimodal Model

Cláudio Rebelo de Sá[1], Joaquim Costa[4], Carlos Soares[1,2], Paulo Azevedo[5], and Alípio Mário Jorge[1,3]

[1] INESC TEC Porto, Porto, Portugal
[2] Faculdade de Economia, Universidade do Porto
[3] DCC - Faculdade de Ciencias, Universidade do Porto
[4] DM - Faculdade de Ciencias, Universidade do Porto
[5] CCTC, Departamento de Informática, Universidade do Minho
claudio.r.sa@inescporto.pt, jpcosta@fc.up.pt, csoares@fep.up.pt,
pja@uminho.pt, amjorge@fc.up.pt

Abstract. Some real life problems require the classification of items into naturally ordered classes. Conventional methods, intended for the classification of nominal classes, are traditionally used to deal with these problems where the classes are ordered. This paper proposes an adaptation of association rules for classification intended for multi-class problems where the order relation is not ignored. The theoretical background assumes that the random variable class associated with a given query should follow a unimodal distribution. The adaptation, which uses class association rules (CAR's), is essentially in terms of the output handling, i.e the voting system for the predicted class. The experiments in real datasets are presented. Despite this very simple variant of association rules for classification, the results indicate that the method is making valid predictions and is competitive with state-of-the-art algorithms.

1 Introduction

Many classification problems require classifying examples into naturally ordered classes. These problems are commonly found in many study fields, such as econometric modeling and collaborative filtering. Other applications include biomedical classification problems, in which is very frequent that the classes are ordered, although that is almost never taken into account and the conventional methods, for nominal classes or regression, are used.

Conventional methods of supervised classification can be used, but first of all it is usually harder and slower to train with these methods and secondly the derived classifier might not be really appropriate. On the other hand, using regression methods introduces an arbitrary selection of numbers to represent the classes, which in turn influence both the prediction function and the usual measures of performance assessment that are used. However, the use of methods specifically designed for ordered classes results in simpler classifiers, making it easier to interpret the factors that are being used to discriminate among classes [9].

Association rules mining is a very important and successful task in data mining. Although its original purpose was only descriptive, several adaptations have been proposed for predictive problems like in [8]. This work proposes an adaptation of association rules for classification intended for multi-class problems where the order relation of the classes is not ignored. The method searches for Class Association Rules (CAR's) and handles them taking into account the order relation of the classes. This is done by forcing an unimodal distribution [5] of the class probabilities, even if their empirical distribution is not unimodal.

The paper is organized as follows: sections 2 and 3 introduce the unimodal paradigm and the task of association rule mining, respectively; section 4 gives a simple description of the method proposed here; section 6 presents the experimental setup and discusses the results; finally, section 7 concludes this paper.

2 The unimodal paradigm

Let us define a supervised classification problem with K ordered classes $\mathcal{C}_1 < \ldots < \mathcal{C}_K$ and denote the feature space as $\mathbb{X}$. In common supervised classification problems, the goal is to find a mapping:

$$f_T : \mathbb{X} \rightarrow \{\mathcal{C}_i\}_{i=1}^{K}$$

that minimizes certain cost functional relative to the ℓ examples in a given training set $T = \{(\mathbf{x}_i, \mathcal{C}_{\mathbf{x}_i})\}_{i=1}^{\ell} \subset \mathbb{X} \times \{\mathcal{C}_i\}_{i=1}^{K}$. Bayes decision theory aims to maximize the *a posteriori* probability $P(\mathcal{C}_k|\mathbf{x})$ in the classification of new examples $\mathbf{x}$ with the class $\mathcal{C}_k$. To that end, we must find a function which estimates the *a posteriori* probabilities:

$$f_T(\mathbf{x}) = \arg\max_{\mathcal{C}_k}\{P(\mathcal{C}_k|\mathbf{x})\}$$

However, should we consider this very same foreground in every classification problem? Let us assume we wave a temperature classification problem with $K = 5$ classes: {Very cold, Cold, Mild, Hot, Very hot}. It is intuitive to consider a natural ordering between these classes: Very cold < Cold < Mild < Hot < Very hot. Thus if the model obtains $P(C_4|\mathbf{x})$ as the highest *a posteriori* probability for a given query point $\mathbf{x}$, the second most probable class should be $P(C_3|\mathbf{x})$ or $P(C_5|\mathbf{x})$. This means that if the most likely is a Hot day, then the second most likely should either be a Mild day or a Very hot day. In other words, we can assume that the probabilities should decrease monotonically to the left and to the right of the class with maximum probability. By using classifiers which do not take into account the order relation presented, the second highest *a posteriori* probability can be, for instance, $P(\mathcal{C}_1|\mathbf{x})$, which makes no sense.

More formally, the unimodal paradigm introduced in [5] assumes that in a supervised classification problem with ordered classes, the random variable class $\mathcal{C}_\mathbf{x}$ associated with a given query point $\mathbf{x}$ should follow a unimodal distribution. We assume a particular unimodal discrete distribution for $\mathcal{C}_\mathbf{x}$, and a classifier f_T estimates the *a posteriori* probabilities by estimating the parameters of the assumed distribution.

3 Association Rules Mining

An association rule (AR) is an implication: $A \rightarrow C$ where $A \bigcap C = \emptyset$, $A, C \subseteq desc(\mathbb{X})$ where $desc(\mathbb{X})$ is the set of descriptors of instances in $\mathbb{X}$, typically pairs $\langle attribute, value \rangle$. We also denote $desc(x_i)$ as the set of descriptors of instance x_i.

Association rules are typically characterized by two measures, support and confidence. The support of rule $A \rightarrow C$ in T is sup if $sup\%$ of the cases in it contain A and C. Additionally, it has a confidence $conf$ in T if $conf\%$ of cases in T that contain A also contain C.

The original method for induction of AR is the APRIORI algorithm that was proposed in 1994 [1]. APRIORI identifies all AR that have a support and confidence higher than a given minimal support threshold ($minsup$) and a minimal confidence threshold ($minconf$), respectively. Thus, the model generated is a set of AR of the form $A \rightarrow C$, where $A, C \subseteq desc(\mathbb{X})$, and $sup(A \rightarrow C) \geq minsup$ and $conf(A \rightarrow C) \geq minconf$. For a more detailed description see [1].

Despite the usefulness and simplicity of APRIORI, it runs a time consuming candidate generation process and needs space and memory that is proportional to the number of possible combinations in the database. Additionally it needs multiple scans of the database and typically generates a very large number of rules. Because of this, many new pruning methods were proposed in order to avoid that. Such as the hashing [10], dynamic itemset counting [4], parallel and distributed mining [11], relational database systems integrated with mining [12].

Association rules were originally proposed for descriptive purposes. However, they have been adapted for predictive tasks such as classification (e.g., [8]) which is described in Section 3.2.

3.1 Pruning

AR algorithms typically generate a large number of rules (possibly tens of thousands), some of which represent only small variations from others. This is known as the rule explosion problem [3]. It is due to the fact that the algorithm might find rules for which the confidence can be marginally improved by adding further conditions to the antecedent.

Pruning methods are usually employed to reduce the amount of rules, without reducing the quality of the model. A common pruning method is based on the improvement that a refined rule yields in comparison to the original one [3]. The *improvement* of a rule is defined as the smallest difference between the confidence of a rule and the confidence of all sub-rules sharing the same consequent. More formally, for a rule $A \rightarrow C$

$$imp(A \rightarrow C) = min\left(\forall A' \subset A, conf\left(A \rightarrow C\right) - conf\left(A' \rightarrow C\right)\right)$$

As an example, if one defines $minImp = 0.1\%$, the rule $A_1 \rightarrow C$ will be kept, if, and only if $conf\left(A_1 \rightarrow C\right) - conf\left(A \rightarrow C\right) \geq 0.001$, where $A \subset A_1$.

3.2 Class Association Rules

Classification Association Rules (CAR), were proposed as part of the Classification Based on AR (CBA) algorithm [8]. A class association rule (CAR) is an implication of the form: $A \rightarrow \mathcal{C}$ where $A \subseteq desc\,(\mathbb{X})$, and $\mathcal{C} \in \mathcal{L}$, which is the class label. A rule $A \rightarrow \mathcal{C}$ holds in T with confidence $conf$ if $conf\%$ of cases in T that contain A are labeled with class $\mathcal{C}$, and with support sup in T if $sup\%$ of the cases in it contain A and are labeled with class $\mathcal{C}$.

CBA takes a tabular data set $T = \{\langle x_i, \mathcal{C}_i \rangle\}$, where x_i is a set of items and $\mathcal{C}_i$ the corresponding class, and look for all frequent *ruleitems* of the form $\langle A, \mathcal{C} \rangle$, where A is a set of items and $\mathcal{C} \in \mathcal{L}$. The algorithm aims to choose a set of high accuracy rules $\mathcal{R}_\mathcal{C}$ to match T. $R_\mathcal{C}$ matches an instance $< x_i, \mathcal{C}_i > \in T$ if there is at least one rule $A \rightarrow \mathcal{C} \in \mathcal{R}_\mathcal{C}$, with $A \subseteq desc(x_i), x_i \in \mathbb{X}$, and $\mathcal{C} \in \mathcal{L}$. If the available rules cannot classify a new example, a default class is given to it (e.g., the majority class in the training data).

4 Association Rules for unimodal class distribution

Although the original purpose of Association rules was only descriptive, several adaptations have been proposed for predictive problems like in [8]. The objective of this work is to propose a supervised classification method, built in association rules, more suitable for predicting ordinal classes. We believe that our contribution will improve the performance of the supervised classification in the presence of ordinal classes in the association rules domain.

After generating the CARs, given some user defined interest measures, these are grouped and counted per class for each unclassified example x. This is, considering the classes {Very cold, Cold, Mild, Hot, Very hot} the method can obtain a bunch of rules for an unclassified example, like, for instance:

(Very cold-5 rules, Cold-7 rules, Mild-15 rules, Hot-2 rules, Very hot-3 rules)

Which is an empirical distribution of the classes where each CAR contributes in equal terms for the counting. These numbers are then used to estimate the probabilities of each of the five classes $P(C_i|x)$. Now, the unimodal paradigm enters: instead of using the usual estimator for these probabilities, for instance $\hat{P}(\text{Verycold}|x) = 5/38$, we will use a different estimator that smooths these numbers and at the same time verify the unimodal paradigm described above; this paradigm is what makes sense, in our opinion, with ordered classes. To do so, we will use a specific unimodal parametric statistical model to be fitted to the numbers (5,13,15,2,3): the binomial distribution. Our purpose will then be to find a binomial distribution $B(4, p)$ whose 5 probabilities best approximate the empirical probabilities (5/38, 13/38, 15/38, 2/38, 3/38). In order to do so, the parameter p will be estimated by maximum likelihood [7].

This p is then used to calculate the probability for the set of classes. The class with higher probability will be the one chose for the recommendation. The method, as we propose, is independent of the CARs generator, once only the recommendation process is affected.

We believe that this method contributes for better performance because it will only force the unimodality in the presence of non unimodal distributions. In those cases where the empirical distribution is already unimodal the probabilities will be slightly smoothed only.

5 Error measures

In this supervised classification problem with ordered classes, we measure the performance of the classifier using two different measures. The Misclassification Error Rate (MER) and the Mean Absolute Error (MAE). The first one considers every misclassification equally costly. In the latter the performance of a classifier f_T is assessed in a dataset $\mathcal{O} \subset \mathcal{X}$ through

$$\mathrm{MAE} \ = \ \frac{1}{\mathrm{card}(\mathcal{O})} \sum_{\mathbf{x} \in \mathcal{O}} |g(\mathcal{C}_{\mathbf{x}}) - g(f_T(\mathbf{x}))|,$$

where $g(\cdot)$ corresponds to the number assigned to a class. Assuming this assignment is arbitrary the performance measurement given by MAE can be inadequate. However, as we are dealing with ordinal classes in this work, the assignment will take into account the classes natural order. This measure can be more informative than MER, because its values increase with the absolute differences between the "true" and "predicted" class numbers and so the misclassifications are measured as a distance factor.

6 Experimental Results

The datasets in this work (Table 1) were taken from:

- (A) www.gatsby.ucl.ac.uk/ chuwei/ordinalregression.html
- (B) www.cs.waikato.ac.nz/ ml/index.html

All the variables were discretized by the following order: (1) *recursive minimum entropy partitioning* criterion ([6]) with the *minimum description length* (MDL) as stopping rule (2) *equal width* bins. The division into *equal width* bins will only be used in the attributes left undiscretized by the previous method.

The evaluation measures used are the Misclassification Error Rate (MER) and Mean Absolute Error (MAE) and the performance of the method was estimated using ten-fold cross-validation. The performance of the Unimodal method was compared with a very simple CAR method, which classifies the examples with the most frequent class from the set of all the matching CARs. For the generation of frequent items we used CAREN [2] and *minsup* was fixed in 0.1% and *mimp* was set to 0.001%.

The number of each class per dataset is graphically presented in Fig. 1. Some datasets do not seem to have an unimodal distribution in the classes. This can be because their distribution is not binomial at all or due to the fact that the data has unbalanced information.

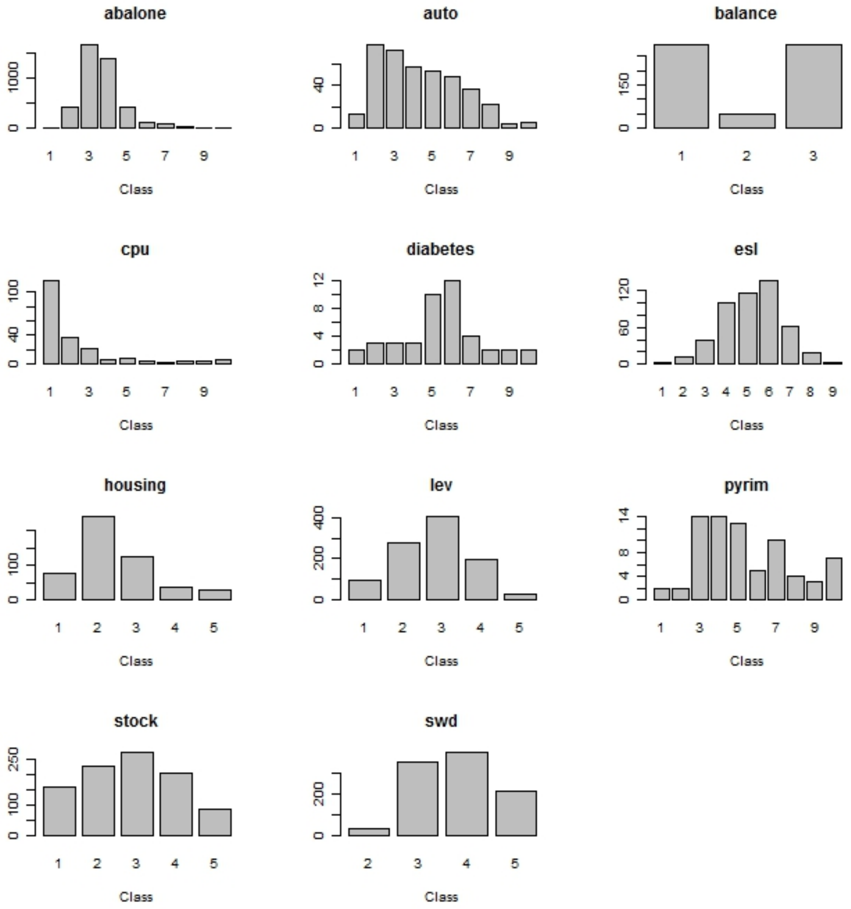

Fig. 1. Distribution of the classes by dataset

Table 1. Summary of the datasets

Datasets	source	#examples	#attributes	#classes
abalone	A	4177	10	10
auto	A	392	7	10
balance	A	625	4	3
cpu	A	209	6	10
diabetes	A	43	2	10
esl	B	488	4	9
housing	A	506	13	5
lev	B	1000	4	5
pyrim	A	74	27	10
stock	A	950	9	5
swd	B	1000	10	4

Table 2. Results obtained in terms of the Misclassification Error Rate

minconf	75%		50%		25%	
	CARs	*Unim*	*CARs*	*Unim*	*CARs*	*Unim*
abalone	.520	**.518**	.450	**.445**	**.466**	.485
auto	**.719**	.729	.566	**.559**	.528	**.492**
balance	**.248**	.333	**.248**	.333	**.248**	.333
cpu	.455	**.440**	.450	**.436**	.435	**.426**
diabetes	.780	.780	.905	**.880**	**.895**	.935
esl	.416	**.399**	.352	**.334**	.348	**.332**
housing	**.296**	.306	**.334**	.352	.391	**.389**
lev	.569	**.569**	.503	**.483**	.484	**.478**
pyrim	.755	**.689**	**.737**	.850	.834	**.771**
stock	**.149**	.183	**.161**	.202	**.143**	.225
swd	.521	**.519**	.446	**.441**	**.449**	.462

6.1 Results

The results presented in table 2 indicate that the classifier using the unimodal method obtains slightly better results than the ones without it. This error measure is only an indicator of the performance of a classifier because, as said before, when dealing with ordered classes there are more fair error measures that should be considered to measure the distance of the predicted class from the real class. For this reason Table 3 presents the results obtained with MAE, which penalizes the bigger deviations of the recommended class in comparison with the real ones. In this case, the Unimodal method clearly outperforms the other method in every confidence values considered.

Despite the absence of a statistical test, the improvement observed is a good motivation for a more throughout experimental study in the future. This indicates that this association rules method is in fact improving its performance by

Table 3. Results obtained in terms of the Mean Absolute Error

minconf	75%		50%		25%	
	CARs	*Unim*	*CARs*	*Unim*	*CARs*	*Unim*
abalone	.758	**.757**	**.599**	.601	**.605**	.626
auto	**1.941**	1.951	1.168	**1.160**	.694	**.645**
balance	.418	**.345**	.418	**.345**	.418	**.345**
cpu	1.231	**1.212**	1.177	**1.163**	.862	**.780**
diabetes	1.495	1.495	2.030	**1.875**	2.090	**1.940**
esl	.500	**.477**	.381	**.355**	.379	**.355**
housing	.332	**.330**	.405	**.397**	.462	**.435**
lev	.669	.669	.554	**.536**	.535	**.523**
pyrim	1.663	**1.264**	1.623	**1.479**	1.702	**1.454**
stock	**.151**	.184	**.164**	.203	**.145**	.225
swd	.638	**.634**	.481	**.479**	**.491**	.505

using the Unimodal method when the class distribution is ordered. This also means that, despite the simplicity of the adaptation, this can be considered a competitive method. We expect that the results can be significantly improved, for instance, by implementing more suitable pruning methods.

7 Conclusions

In this paper we present a simple adaptation of association rules for ordinal classification. This adaptation essentially consists of adapting a well known method taking into consideration the properties of the classes of the datasets. Which, in this particular case, is the natural orders of the classes.

These results are positive and clearly show that this method, in most cases, decreases the error rate when compared with normal association rules methods in this scenario. We think that it executes well its function to better classify classes with a binomial distribution.

This work uncovered several problems that could be better studied in order to study and improve the method's performance. They include: improving the prediction generation method; implementing better pruning methods; choice of parameters.

We also want to undertake statistical tests, in future work, to give a stronger analysis of the performance of the method.

Acknowledgments

This work was partially supported by the European Regional Development Fund through the programme COMPETE (PEst-C/MAT/UI0144/2011) from FCT and Palco AdI project Palco3.0 financed by QREN and Fundo Europeu de Desenvolvimento Regional (FEDER). We thank the anonymous referees for useful comments.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proc. 20th Int. Conf. Very Large Data Bases, VLDB. vol. 1215, p. 487499. Citeseer (1994)
2. Azevedo, P.J., Jorge, A.M.: Ensembles of jittered association rule classifiers. Data Min. Knowl. Discov. 21(1), 91–129 (2010)
3. Bayardo, R., Agrawal, R., Gunopulos, D.: Constraint-based rule mining in large, dense databases. Data Mining and Knowledge Discovery 4(2), 217–240 (2000)
4. Brin, S., Motwani, R., Ullman, J.D., Tsur, S.: Dynamic itemset counting and implication rules for market basket data. Proceedings of the 1997 ACM SIGMOD international conference on Management of data - SIGMOD '97 pp. 255–264 (1997), http://portal.acm.org/citation.cfm?doid=253260.253325
5. da Costa, J.F.P., Alonso, H., Cardoso, J.S.: The unimodal model for the classification of ordinal data. Neural Networks 21(1), 78–91 (2008)
6. Fayyad, Irani: Multi-interval discretization of continuous-valued attributes for classification learning. In: International Conference on Machine Learning. pp. 1022–1027 (1993)
7. Jae, I., Myung: Tutorial on maximum likelihood estimation. Journal of Mathematical Psychology 47(1), 90–100 (2003), http://www.sciencedirect.com/science/article/pii/S0022249602000287
8. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. Knowledge Discovery and Data Mining pp. 80–86 (1998)
9. Mathieson, M.: Ordinal models for neural networks (1996)
10. Park, J.S., Chen, M.S., Yu, P.S.: An effective hash-based algorithm for mining association rules. ACM SIGMOD Record 24(2), 175–186 (May 1995), http://portal.acm.org/citation.cfm?doid=568271.223813
11. Park, J., Chen, M., Yu, P.: Efficient parallel data mining for association rules. of the fourth international conference on (1995), http://portal.acm.org/citation.cfm?id=221270.221320
12. Thomas, S., Sarawagi, S.: Mining generalized association rules and sequential patterns using SQL queries. . Conf. on Knowledge Discovery and Data Mining (1998), http://www.aaai.org/Papers/KDD/1998/KDD98-062.pdf