Combining Classification and Model Trees for Handling Ordinal Problems

D. Anyfantis, M. Karagiannopoulos S. B. Kotsiantis, P. E. Pintelas

Educational Software Development Laboratory
Department of Mathematics
University of Patras, Hellas
{dany,mariosk,sotos,pintelas}@math.upatras.gr

Abstract

Given an ordered class, one is not only interested in minimizing the classification error, but also in minimizing the distances between the actual and the predicted class. This study offers an organized study on the various methodologies that have tried to handle this problem and presents an experimental study of these methodologies with a proposed voting technique, which combine the predictions of a classification tree and a model tree algorithm. The paper concludes that the proposed technique can be a more robust solution to the problem since it minimizes the distance between the actual and the predicted class and improves the classification accuracy.

Keywords: Data mining, Supervised Machine Learning, Ranking Learning.

1. Introduction

Ordinal classification (ranking learning or ordinal regression) can be viewed as a bridging problem between the two standard learning tasks of classification and regression. In ordinal classification, the target values are in a finite set (like in classification) but there is an ordering among the elements (like in regression, but unlike classification).

The ordering of the labels does not justify a metric loss function, thus casting the ranking learning problem as an ordinary regression (by treating the continuous variable with a coarse scale) may not be realistic. Settings in which it is natural to rank or rate instances arise in many fields such as information retrieval, visual recognition, collaborative filtering, econometric models and classical statistics.

Even though Machine Learning (ML) algorithms for ordinal classification are rare, there are a number of statistical approaches to this problem. However, they all rely on specific distributional assumptions for modeling the class variable and also assume a stochastic ordering of the input space [Potharst et. Al. (2000)]. The ML community has mainly addressed the issue of ordinal classification in two ways. One is to apply

classification algorithms by discarding the ordering information in the class attribute [Frank et. Al. (2001)]. The other is to apply regression algorithms by transforming class values to real numbers [Kramer et. Al. (2001)]. This paper proposes a voting technique, which combine the predictions of a classification tree and a model tree algorithm. Experimental results show that this technique minimizes the distances between the actual and the predicted class and improves the prediction accuracy.

This paper is organized as follows: the next section discusses the different techniques that have been presented for handling ordinal classification problems. In section 3, we describe the proposed technique. In Section 4, we present the experimental results of our methodology and compare these results with those of other approaches. In the final section of the paper we discuss further work and some conclusions.

2. Techniques for Dealing with Ordinal Problems

Problems of ordinal regression arise in many fields, e.g., in information retrieval and in classical statistics. They can be related to the standard machine learning paradigm as follows. In ordinal regression, we consider a problem which shares properties of both classification and metric regression. A variable of the above type exhibits an ordinal scale and can be thought of as the result of coarse measurement of a continuous variable. Classification algorithms can be applied to ordinal prediction problems by discarding the ordering information in the class at-tribute. However, some information that could improve the performance of a classifier is lost when this is done.

The use of regression algorithms to solve ordinal problems has been examined in [Pfahringer et. Al. (2000)]. In this case each class needs to be mapped to a numeric value. However, if the class attribute represents a truly ordinal quantity, which, by definition, cannot be represented as a number in a meaningful way, there is no upright way of devising an appropriate mapping and this procedure is ad hoc. Kramer [Kramer et. Al. (2001)] investigated the use of a regression tree learner in this way.

Another approach is to reduce the multi-class ordinal problem to a set of binary problems using the one-against-all approach [Allwein et. Al. (2000)]. In the one-against-all approach, a classifier is trained for each of the classes using as positive examples the training examples that belong to that class, and as negatives all the other training examples. The estimates given by each binary classifier are then coupled in order to obtain class probability membership estimates for the multi-class problem [Allwein et. Al. (2000)].

A more sophisticated approach that enables classification algorithms to make use of ordering information in ordinal class attributes is presented in [Frank et. Al. (2001)].

Similarly with previous method, this method converts the original ordinal class problem into a set of binary class problems that encode the ordering of the original classes. However, to predict the class value of an unseen instance this algorithm needs to estimate the probabilities of the m original ordinal classes using m − 1 models. For example, for a three class ordinal problem, estimation of the probability for the first ordinal class value depends on a single classifier: Pr(Target < first value) as well as for the last ordinal class: Pr(Target > second value). Whereas, for class value in the middle of the range, the probability depends on a pair of classifiers and is given by :

$$\Pr(\text{Target} > \text{first value}) * (1 - \Pr(\text{Target} > \text{second value}))$$

Other machine learning research that seems relevant to the problem of predicting ordinal classes is work on cost-sensitive learning. In the domain of propositional learning, some induction algorithms have been proposed that can take into account matrices of misclassification costs [Kotsiantis et. Al. (2004)]. Such cost matrices can be used to express relative distances between classes.

Har-Peled [Har-Peled et. Al. (2002)] proposed a constraint classification approach that provides a unified framework for solving ranking and multi-classification problems. Shashua [Shashua et. Al. (2003)] generalized the support vector formulation for ordinal regression by finding r-1 thresholds that divide the real line into r consecutive intervals for the r ordered categories.

3. Proposed Technique

A recent overview of existing work on decision trees and a taste of their usefulness to the newcomers in the field of machine learning are provided in [Murthy et. Al. (1998)]. Decision trees are trees that classify instances by sorting them based on feature values. Each node in a decision tree represents a feature in an instance to be classified, and each branch represents a value that the node can take. Instances are classified starting at the root node and sorting them based on their feature values. Most well-known decision tree algorithm is the C4.5 [Quinlan et. Al. (1993)].

Model trees are the counterpart of decision trees for regression tasks. The most well known model tree inducer is the M5´ [Wang et. Al. (1997)]. A model tree is generated in two stages. The first builds an ordinary decision tree, using as splitting criterion the maximization of the intra-subset variation of the target value [Witten et. Al. (2000)]. The second prunes this tree back by replacing sub-trees with linear regression functions wherever this seems appropriate. If this step is omitted and the target is taken to be the average target value of training examples that reach this leaf, then the tree is called a “regression tree” instead. Although the models trees are smaller and more accurate than the regression trees, the regression trees are more comprehensible [Frank et. Al. (1998)].

As we have already mentioned, the proposed technique combines the predictions of a classification tree and a model tree algorithm. When learners are combined using a voting methodology, we expect to obtain good results based on the belief that the majority of classifiers are more likely to be correct in their decision when they agree in their opinion. Voters can express the degree of their preference using a confidence score i.e. the probabilities of classifiers pre-diction.

In the proposed ensemble the sum rule is used -each voter gives the probability of its pre-diction for each candidate. Next all confidence values are added for each candidate and the candidate with the highest sum wins the election. It must be mentioned that the sum rule is one of the best voting methods for classifier combination according to [Van Erp et. Al. (2002)].

In detail, the proposed ensemble (Vote-C4.5-M5´) is schematically presented in Figure 1. Each classifier (C4.5, M5´) generate a hypothesis h1, h2 respectively. For M5´, class is binarized and one regression model is built for each class value [Frank et. Al. (1998)]. The a-posteriori probabilities generated by the individual classifiers are correspondingly denoted p1(i), p2(i) for each output class i. Next, the class represented by the maximum sum value of the a-posteriori probabilities is taken as the voting hypothesis (h*). The predictive class is computed by the rule:

$$predictive\_Class = \arg\max_{i} \sum_{i=1, j=1}^{i=number\_of\_classes, j=3} p_j(i)$$

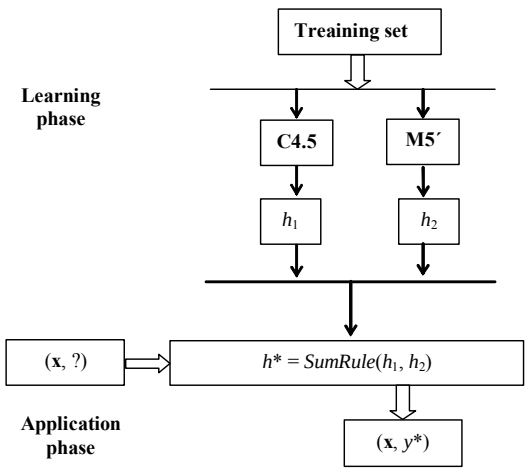

Figure 1. *The proposed ensemble*

It must also be mentioned that the proposed ensemble can easily be parallelized using a learning algorithm per machine. Parallel and distributed computing is of most importance for Machine Learning (ML) practitioners because taking advantage of a

parallel or a distributed execution a ML system may: i) increase its speed; ii) increase the range of applications where it can be used (because it can process more data, for example).

4. Experiments

To test the hypothesis that the above method improves the generalization performance on ordinal prediction problems, we performed experiments on real-world ordinal datasets donated by Dr. Arie Ben David (http://www.cs.waikato.ac.nz/ml/weka/). These training samples are labeled by ranks, which exhibits an ordering among the different categories. In contrast to metric regression problems, these ranks are of finite types and the metric distances between the ranks are not defined. These ranks are also different from the labels of multiple classes in classification problems due to the existence of the ordering information. We also used well-known datasets from many domains from the UCI repository [Blake et. Al. (1998)]. However, the used UCI datasets represented numeric prediction problems and for this reason we converted the numeric target values into ordinal quantities using equal-size binning. This unsupervised discretization method divides the range of observed values into three equal size intervals. The resulting class values are ordered, representing variable-size intervals of the original numeric quantity. This method was chosen because of the lack of numerous benchmark datasets involving ordinal class values.

All accuracy estimates were obtained by averaging the results from 10 separate runs of stratified 10-fold cross-validation. It must be mentioned that we used the free available source code for most algorithms by the book [Witten et. Al. (2000)].

In Table 1 and Table 2, for each data set the algorithms are compared according to classification accuracy (the rate of correct predictions) and to Root mean squared error: $\sqrt{(p_1 - a_1)^2 + \ldots + (p_1 - a_1)^2} \big/ n$, where pi: predicted values and ai: actual values. Moreover, in Table 1 and Table 2, we represent as “v” that the specific algorithm performed statistically better than the proposed method according to t-test with $p<0.05$. Throughout, we speak of two results for a dataset as being "significant different" if the difference is statistical significant at the 5% level according to the corrected resampled t-test [Nadeau et. Al. (2003)], with each pair of data points consisting of the estimates obtained in one of the 100 folds for the two learning methods being compared. On the other hand, “*” indicates that proposed method performed statistically better than the specific algorithm according to t-test with $p<0.05$.

Table 1 shows the results as far as the root mean square error is concerned for (a) the C4.5 algorithm, (b) the ordinal classification method presented in Section 2 in conjunction with C4.5 algorithm (C45-ORD) [Frank et. Al. (2001)], (c) using classification via regression (M5´) and (d) using the proposed voting technique (Vote-C4.5-M5).

As one can see from the aggregated results in Table 1, it manages to minimize the distances between the actual and the predicted classes. The reduction of the root mean square error is about 7% compared to the simple C4.5 and the C4.5-ORD, while it exceeds the 3% compared to M5´.

Table 1. *Comparing algorithms as far as the root mean square error is concerned.*

Datasets	Vote-C4.5-M5´	M5´		C4.5		C4.5-ORD	
auto93	0,28	0,32		0,31		0,33	
autoHorse	0,13	0,16	*	0,11		0,11	
autoMpg	0,29	0,29		0,31	*	0,31	*
autoPrice	0,21	0,22		0,23		0,22	
baskball	0,36	0,36		0,38		0,39	
bodyfat	0,08	0,11	*	0,07		0,07	
breastTumor	0,43	0,44		0,44		0,44	
cholesterol	0,37	0,36		0,4	*	0,4	*
cloud	0,2	0,22		0,21		0,2	
cpu	0,1	0,1		0,11		0,1	
echoMonths	0,4	0,4		0,44	*	0,41	
ERA	0,3	0,3		0,3		0,31	
ESL	0,23	0,23		0,24		0,25	*
fishcatch	0,09	0,13	*	0,06		0,06	
fruitfly	0,37	0,38		0,37		0,37	
housing	0,3	0,29		0,35	*	0,35	*
hungarian	0,29	0,28		0,33	*	0,33	*
LEV	0,32	0,33		0,33		0,34	*
lowbwt	0,4	0,39		0,43	*	0,43	*
pbc	0,42	0,41		0,47	*	0,46	*
pharynx	0,4	0,47	*	0,39		0,37	
pwLinear	0,32	0,33		0,36	*	0,34	
sensory	0,41	0,41		0,42		0,42	
strike	0,06	0,06		0,05		0,05	
SWD	0,37	0,37		0,38		0,38	
veteran	0,23	0,23		0,23		0,23	
Average Root Mean Square Error	0,28	0,29		0,30		0,30	
W-D-L		0/22/4		0/18/8		0/18/8	

In addition, the presented ensemble is significantly more accurate than M5´ in 4 out of the 26 datasets, whilst it has significantly higher root mean square error in none

dataset. The presented ensemble has also significantly lower root mean square error in 8 out of the 26 datasets than both C4.5 and C4.5-ORD, whereas it is significantly less accurate in none dataset.

Table 2. *Comparing algorithms as far as the classification accuracy is concerned.*

Datasets	Vote-C4.5-M5´	M5´		C4.5		C4.5-ORD	
auto93	80,69	81,24		80,83		78,31	*
autoHorse	96,88	94,74	*	96,78		96,93	
autoMpg	82,16	82,31		82,09		82,59	
autoPrice	90,19	89,5		89,25		89,68	
baskball	72,11	71,46		70,64	*	67,9	*
bodyfat	98,37	99,12		98,37		98,37	
breastTumor	58,25	55,25	*	58,71		58,6	
cholesterol	69,91	72,28		68,69		68,48	
cloud	90,63	88,58	*	90,54		91,12	
cpu	97,19	97,1		96,95		97,42	
echoMonths	63,38	66,08	v	62,46		65,54	
ERA	28,46	27,2		27,9		27,24	
ESL	65,77	65,81		65,03		65,53	
fishcatch	97,65	96,84		97,65		97,91	
fruitfly	73,67	73,67		73,67		73,67	
housing	79,41	82,33	v	78,22		79,07	
hungarian	80,56	82,44		80,22		80,22	
LEV	60,99	59,21		60,48		61,19	
lowbwt	60,3	61,3		60,09		60,04	
pbc	56,76	57,4		55,53		54,27	*
pharynx	69,85	43,34	*	69,4		73,59	v
pwLinear	77	78,3		76,55		78,5	
sensory	64,17	64,24		64,24		64,05	
strike	99,21	99,21		99,21		99,21	
SWD	57,49	59,13		57		58,05	
veteran	91,26	91,26		91,26		91,26	
Average Accuracy	75,47	74,59		75,06		75,33	
W-D-L		2/20/4		0/25/1		1/22/3	

Table 2 shows the results as far as the classification accuracy is concerned for (a) the C4.5 algorithm, (b) the ordinal classification method presented in Section 2 in conjunction with C4.5 algorithm (C45-ORD) [Frank et. Al. (2001)], (c) using

classification via regression (M5´) and (d) using the proposed voting technique (Vote-C4.5-M5).

As one can see from the aggregated results in Table 2, the proposed technique is slightly better in classification accuracy than the remaining approaches. The presented ensemble is significantly more accurate than M5´ in 4 out of the 26 datasets, whilst it has significantly higher error rate in two datasets. The presented ensemble has also significantly lower error rate in 3 out of the 26 datasets than C4.5-ORD, whereas it is significantly less accurate in one dataset. The proposed method is significantly more accurate than C4.5 in 1 out of the 26 data-sets, whilst it has significantly higher error rate in none dataset.

It must be mentioned that the ordinal technique [Frank et. Al. (2001)] outperforms the simple C4.5 only in classification accuracy. It does not manage to minimize the distance between the actual and the predicted class. Moreover, the M5´ seems to give the worst average accuracy results ac-cording to our experiments even though in several data sets its performance is much better than the performance of the remaining algorithms.

It must also be declared that a decision tree learning algorithm for monotone learning problems has been presented in [Potharst et. Al. (2000)]. In a monotone learning problem both the input attributes and the class attribute are assumed to be ordered. This is different from the setting considered in this paper because we do not assume that the input is ordered.

5. Conclusion

Many pattern recognition problems involve classifying examples into classes which have a natural ordering. Settings in which it is natural to rank instances arise in many fields, such as information retrieval, collaborative filtering and econometric modeling.

Most of the research in the machine learning community has gone into developing algorithms which are good for either classification or regression. Given that, some of the researchers have attempted to solve the ranking and sorting problems by posing them as either multi-class classification or regression problems. In this paper, we argue that this is not the right approach. If the ranking problem is posed as a classification problem then the inherent structure present in ranked data is not made use of and hence generalization ability of such classifiers is severely limited. On the other hand, posing the task of sorting as a regression problem leads to a highly constrained problem.

In this paper, we studied various ways of transforming a simple algorithm for ordinal classification tasks and we proposed a voting technique, which combine the predictions of a classification tree and a model tree algorithm. According to our

experiments in synthetic and real ordinal data sets, the proposed method manages to minimize the distances between the actual and the predicted classes, without harming but actually slightly improving the classification accuracy. More extensive experiments will be needed to establish the precise capabilities and relative advantages of this methodology.

References

Allwein, E. L., Schapire, R. E., and Singer, Y. (2000), *Reducing multiclass to binary: A unifying approach for margin classifiers.* Journal of Machine Learning Research 1, pp. 113–141.

Blake, C.L. & Merz, C.J. (1998), *UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science*. [http://www.ics.uci.edu/~mlearn/MLRepository.html].

Frank, E. and Hall M. (2001), *A simple approach to ordinal prediction*, L. De Raedt and P. Flach (Eds.): ECML 2001, LNAI 2167, Springer-Verlag Berlin pp. 145-156.

Frank, E., Wang, Y., Inglis, S., Holmes, G., and Witten, I.H. (1998), *"Using model trees for classification"*, Machine Learning, vol.32, No.1, pp. 63-76.

Har-Peled, S., D. Roth, and D. Zimak. (2002), *Constraint classification: A new approach to multiclass classification and ranking.* In Advances in Neural Information Processing Systems 15.

Kotsiantis, S., Pintelas, P. (2004), *A Cost Sensitive Technique for Ordinal Classification Problems,* Lecture Notes in Artificial Intelligence, Springer-Verlag vol 3025, pp. 220-229.

Kramer, S., Widmer, G., Pfahringer, B. and DeGroeve M. (2001), *Prediction of ordinal classes using regression trees.* Fundamenta Informaticae.

Murthy (1998), *Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey*, Data Mining and Knowledge Discovery 2 pp. 345–389, Kluwer Academic Publishers.

Nadeau, C. Bengio, Y. (2003), *Inference for the Generalization Error.* Machine Learning 52(3), pp.239-281.

Pfahringer, B., Kramer, S. Widmer, G. and M. de Groeve (2000), *Prediction of ordinal classes using regression trees.* In International Syposium on Methodologies for Intelligent Systems, pp. 426–434.

Potharst R. and Bioch J.C (2000), *Decision trees for ordinal classification.* Intelligent Data Analysis 4, pp. 97-112.

Quinlan J.R.: C4.5 (1993), *Programs for machine learning.* Morgan Kaufmann, San Francisco

Shashua, A. and A. Levin. (2003), *Ranking with large margin principle: two approaches.* In S. Becker, S. T. and K. Obermayer, editors, Advances in Neural Information Processing Systems 15, pp. 937–944

Van Erp, M., Vuurpijl, L.G., and Schomaker, L.R.B. (2002), *An overview and comparison of voting methods for pattern recognition.* In Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition pp. 195-200. Niagara-on-the-Lake, Canada.

Wang, Y. & Witten, I. H. (1997), *Induction of model trees for predicting continuous classes,* In Proc. of the Poster Papers of the European Conference on ML, Prague pp. 128–137. Prague: University of Economics, Faculty of Informatics and Statistics.

Witten I. and Frank E. (2000), *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations,* Morgan Kaufmann, San Mateo.