Random Ordinality Ensembles: A Novel Ensemble Method for Multi-valued Categorical Data

Amir Ahmad and Gavin Brown

School of Computer Science, University of Manchester,
Manchester, M13 9PL, UK
{ahmada,gbrown}@cs.man.ac.uk

Abstract. Data with multi-valued categorical attributes can cause major problems for decision trees. The high branching factor can lead to data fragmentation, where decisions have little or no statistical support. In this paper, we propose a new ensemble method, *Random Ordinality Ensembles* (ROE), that circumvents this problem, and provides significantly improved accuracies over other popular ensemble methods. We perform a random projection of the *categorical* data into a *continuous* space by imposing *random ordinality* on categorical attribute values. A decision tree that learns on this new continuous space is able to use binary splits, hence avoiding the data fragmentation problem. A majority-vote ensemble is then constructed with several trees, each learnt from a different continuous space. An empirical evaluation on 13 datasets shows this simple method to significantly outperform standard techniques such as Boosting and Random Forests. Theoretical study using an information gain framework is carried out to explain RO performance. Study shows that ROE is quite robust to data fragmentation problem and Random Ordinality (RO) trees are significantly smaller than trees generated using multi-way split.

Keywords: Decision trees, Data fragmentation, Random Ordinality, Binary splits, Multi-way splits.

1 Introduction

Ensembles are a combination of multiple base models for which the final classification depends on the combined outputs of individual models. Classifier ensembles have shown to produce better results than single models, if the classifiers are *accurate* and *diverse* [7,12].

Several different methods based on the principle of data randomization have been proposed to build diverse decision tree ensembles. Some methods manipulate the data, whereas some other methods manipulate the splitting criteria. *Bagging* [3] and *Boosting* [10] introduce randomization by manipulating the distribution of training patterns supplied to each classifier. Random Trees [8] and Random Forests [4] manipulate the splitting criteria to build ensembles of decision trees.

J.A. Benediktsson, J. Kittler, and F. Roli (Eds.): MCS 2009, LNCS 5519, pp. 222–231, 2009.

The majority of existing methods [5,13] for decision trees build a tree in a top-down approach and use various *impurity functions* to estimate the quality of the attributes in order to select the best one to *split on*. Whether there should be a *binary split* or *multi-way split* has been a question of extensive research [5,13,9,2]. While multi-way splits produce a more comprehensible tree, they may lead to the *data fragmentation* problem [14], where fine-grained partitioning of the training set at every tree node reduces the number of examples at lower-level nodes. As decisions in the lower levels nodes are based on increasingly smaller fragments of the data, some of them may not have much statistical significance.

Motivated by the advantages of binary decision trees (low data fragmentation) for *multi-valued categorical data* [5,13,9], in the proposed work, we build classifier ensembles of *binary decision trees for datasets consisting of multi-valued categorical attributes*.

The rest of the paper is organized as follows: in the next section, we discuss different binary-split and multi-way split criteria for decision trees. In section 3, we present the *Random Ordinality ensemble* technique. Theoretical study of RO attributes using information gain ratio framework is presented in section 4. The experiments are presented in section 5. The effect of data fragmentation on ROE and sizes of RO trees are studied in section 6. The paper ends with conclusions and future work.

2 Related Work with Split Criteria

In this section, we analyze various split criteria used in decision trees for *multi-valued categorical attributes*.

The CART [5] procedure proposed by Brieman uses the *Gini index* as its splitting criterion. As a multi-way split (for multi-valued categorical attributes) with the Gini index favours those with more values, CART enforces binary splits to overcome this problem. As CART procedure builds binary trees, the values of the categorical attribute at the node have to be divided into two groups. If the number of attribute values is $|A|$ then the number of nontrivial binary splits is given by $2^{(|A|-1)} - 1$. Selecting the best split is computationally expensive. Breiman [5] shows that for two class problems the best split can be found by examining only ($|A|$-1) possibilities.

C4.5 as proposed by Quinlan [13] uses the *information gain ratio* as the splitting criterion. C4.5 builds a binary tree for continuous data. There are two methods in C4.5 to handle multi-valued categorical attributes. In the first, it allows the multi-way split of nodes (one branch for each attribute value). In the second method, it uses a greedy approach to iteratively merge the attribute values into two groups. Another way to obtain a binary split for a multi-valued categorical attribute is to partition the data points using an attribute value [5,9]. In this method, all the data points with that attribute value form one group, whereas the other group is formed with the other examples. Geurts et al. [11] suggest a randomized method to create binary attributes from the multi-valued attributes; they divide the attribute values randomly into the two categories. As in this method the node split

decision is taken without considering the output, the classification accuracy of the tree may be poor.

Our method is between the methods proposed by Breiman (searching for the best split) [5] and completely random splits [11]. In RO trees, the best split at each node can be found by examining (|A|-1) possibilities. In this next section, we present RO ensembles.

3 Random Ordinality Ensembles

In this section we discuss our proposed method, for producing ensembles of binary decision trees on datasets with *multi-valued categorical attributes*. The handling of categorical attributes is difficult as the category values have no *intrinsic ordering*. For example (*dog, cat, cow*), have no natural order. This is distinct from discrete data, such as (*low, medium, high*), where there is a natural order to the attribute values. We can exploit this property to build an ensemble of binary decision trees. *We solve the node splitting problem under some random constraints.* Our method is between the methods proposed by Breiman (searching for the best split) [5] and completely random splits [11]. To find the best split at each node (|A|-1) possibilities are examined. Random constraints used in the proposed method are helpful in building classifier ensembles as the randomization helps in creating diversity. **This technique is based on data manipulation by imposing a random ordinality onto the categorical attribute values.** This implies a random projection of the categorical attributes into a continuous space. Our method is based on data manipulation, so it is not specific to any split criterion—Random Ordinality creates diverse *training datasets*.

3.1 Data Generation Using RO

As there is no natural order given for the categorical attribute values, we can *enforce a random ordinality* on these values. In other words, we create a random projection of categories to a continuous space. We explain our method by using the example data given in column one of Table 1. This data has four attribute values (Cow, Dog, Cat, Rat) for one of its attributes (attribute 1). We assign some integer number (1 to *number of attribute values*) to them randomly such that no two attribute values are assigned the same integer value. For example, we assign Dog = 1, Cow = 2, Rat = 3, Cat = 4 to the attribute values of the first attribute. The enforced ordinality is therefore Dog<Cow<Rat<Cat. We follow the same process for all the multi-valued categorical attributes independently. Our final dataset will be integer-valued, therefore having a natural ordering. Following this method we can generate diverse continuous datasets from the original training dataset.

3.2 Learning

Each decision tree in the ensemble learns on one dataset from the pool of different datasets created by RO. During learning, integer-valued attributes are treated

Table 1. Example of Random Ordinality for a single attribute A_1. The possible values of A_1 have no natural ordering—but can be randomly assigned with an ordinality, as shown by new attributes A_1', and A_1''. In A_1', we have Dog<Cow<Rat<Cat, while in A_1'', we have Rat<Cat<Cow<Dog.

A_1	$\rightarrow$	A_1'	A_1''
Cow	$\rightarrow$	2	3
Dog	$\rightarrow$	1	4
Cow	$\rightarrow$	2	3
Dog	$\rightarrow$	1	4
Rat	$\rightarrow$	3	1
Rat	$\rightarrow$	3	1
Cat	$\rightarrow$	4	2
Cat	$\rightarrow$	4	2

as *continuous attributes*. We have binary splits in the tree as for continuous data attributes the node is split at a threshold value. For our example, we have three possible splits, {(1), (2,3,4)}, {(1,2), (3,4)} and {(1,2,3), (4)}. The best split is decided by the desired split criterion. *We avoid the data fragmentation problem as there is a binary split.* Using this method, it is not necessarily true that we get the best split as shown by Breiman [5]. However, since we want to create an ensemble, different node splits are necessary to create diverse decision trees. Furthermore there is no change in the tree building process so no extra computational cost for the tree building phase. Results of different decision trees in the ensemble are combined using a majority voting scheme to get the final prediction. ROE algorithm is presented in Fig. 1. In the next section, we present theoretical study of RO attributes.

4 Study of RO Attributes in an Information Gain Framework

In RO, new attributes are created by randomly assigning order to different attribute values and treating these new attributes as continuous. The selected splitting criterion is used to decide the best binary split. In this section, we will use the information theoretic framework to discuss whether these attributes are good for classification.

Let D be a 2 class (Y = +1 and Y = -1) dataset with the same number of positive and negative examples. Let A be a multi-valued attribute with cardinality $|A|$ again with uniform prior probability. Half of these values correctly identify the positive class, whereas rest of the values correctly identify the negative class. For example, if attribute values are (a,b,c,d,e,f),

$$p(Y = +1|A = a) = 1, p(Y = +1|A = b) = 1, p(Y = +1|A = c) = 1. \quad (1)$$

$$p(Y = -1|A = d) = 1, p(Y = -1|A = e) = 1, p(Y = -1|A = f) = 1. \quad (2)$$

Input- Dataset T with m multi-valued categorical attributes and L size of the ensemble.

Training Phase
for i=1...L **do**
  Data Generation
  Apply Random Ordinality to generate integer valued dataset T_i.
  Learning Phase
  Treat dataset T_i as continuous, and learn decision tree D_i.
end for

Testing Phase
For a given data point **x**
for i=1...L **do**
  Convert **x** to $\mathbf{x}'$ using the ordinality of tree D_i.
  Get the prediction for $\mathbf{x}'$ from tree D_i.
end for
Combine the results of L decision trees by the chosen combination rule to get the final classification result (we use majority voting method).

Fig. 1. Algorithm for Random Ordinality Ensembles (ROE)

We calculate the information gain ratio of different attributes created by RO. We randomly assign order to (a,b,c,d,e,f) and calculate a binary split at each point, the maximum information gain ratio is taken as the information gain ratio associated with this random order. For example, if we assign

$$a < c < f < e < b < d. \tag{3}$$

The maximum information gain ratio is based on the split ((a,c) (f,e,b,d)) and this is taken as the information gain ratio associated with the random order presented in Eq. 3. We calculate the average information gain ratio of different possible random orders of attribute values. We carry out this exercise for attributes with different cardinality, whereas dataset and attribute values have the same properties as discussed above.

We also calculate the information gain ratio of binary splits created by random splitting of attribute values into two groups. Results are presented in table 2. Results indicate that the average gain ratio of attribute created using RO and the gain ratio of multi-valued attributes are quite similar, whereas random splits do not create good splits. As the cardinality of the attribute increases, the average information gain ratio of RO attributes decreases. The same is true for the multiway split as the value of normalizing factor ($\log_2 |A|$) increases. This suggests that on average we are creating binary splits from multi-valued categorical attributes that have similar information gain ratio. The theoretical study suggests that for multi-valued categorical attributes with certain properties, **the information gain**

Table 2. Gain ratio of attributes with different numbers of attribute values

Cardinality $\|A\|$ of the attribute A	Number of random attributes created	Average gain ratio for RO attributes (s.d.)	Average gain ratio for attributes with random split (s.d.)	Gain ratio for multi-way split
4	10^4	**0.59(0.29)**	0.37(0.26)	0.50
6	10^4	**0.47(0.20)**	0.20(0.24)	0.39
8	10^6	**0.40(0.16)**	0.13(0.18)	0.33
10	10^7	**0.35(0.12)**	0.10(0.14)	0.30
12	10^7	**0.32(0.10)**	0.08(0.11)	0.28
14	10^7	**0.29(0.09)**	0.06(0.09)	0.26

ratio of a binary split with some random constraints may be equal to or greater than a multi-way split.

In the next section, we present the comparative study of ROE against the other ensemble methods.

5 Empirical Evaluation

A study was carried out to compare the performance of ROE with Bagging [3], AdaBoostM1 [10] and Random Forest [4]. We created two types of RO ensembles. In the first, ROE with J48, we used the J48 (the WEKA [15] implementation of C4.5 as the base classifier (with the unpruned option)), which uses multi-way splits for multi-valued categorical attributes as per default. In the second, ROE with RS, we used *Random Trees*[15] as the base classifier. *Random Trees* [15] constructs a tree that considers K random features at each node. In other words, we combine the benefits of attribute randomization of *Random Subspaces* (RS) with *Random Ordinality*. We carried out experiments with Bagging and AdaBoost.M1 [10] using J48 (unpruned) as the base model, and Random Forests (WEKA implementations of these ensemble method were used). The sizes of the ensembles were set at 50 for these experiments. K (number of attributes to randomly investigate) is taken as the half of the attributes for Random Tree. Default settings were used for the rest of the parameters. The experiments were conducted following the 5×2 cross-validation [6]. The original test proposed by Dietterich [6] to compare the performance of classifiers suffers from low replicability. Alpaydin [1] propose a modification to the 5×2 cross-validation F test. We used this test for our experiments. We considered a confidence level of 95% for this test. Table 3 presents classification errors of different ensemble methods on different datasets.

Results suggest that, with the exception of Monks1 data, the performance of ROE with J48 is either statistically similar or better than that of other popular ensemble methods. *The performance of ROE with RS is either statistically similar or better than that of other popular ensemble methods for all datasets.* For Monks1 data the performance of ROE with J48 was poor. We discuss this dataset in detail to understand the limitations of RO ensembles.

Table 3. Classification error in % for different ensembles; bold numbers indicate best performance. Comparative results are presented *ROE with J48/ROE with RS* in bracket (if performance of these ensembles are different). '+/-' shows that performance of ROE is statistically better/worse than that algorithm for that dataset, 'Δ' shows that there is no statistically significant difference in performance for this dataset between ROE and that algorithm.

Dataset	RO with J48 ensemble	RO with RS ensemble	Bagging	AdaBoostM1	Random Forest	Single Tree (J48)
Promoter	13.1	**12.8**	15.5	19.6	13.4	28.5(+/+)
Hayes-Roth	16.9	**15.9**	22.8(+/+)	23.1(+/+)	22.2(+/+)	25.3(+/+)
Breast Cancer	30.3	30.1	**29.9**	35.6	32.4	35.9
Monks1	18.3	**1.5**	5.8(-/Δ)	5.9(-/Δ)	3.3(-/Δ)	15.9(-/+)
Monks2	33.9	**30.9**	46.9(+/+)	47.5(+/+)	50.4(+/+)	49.6(+/+)
Monks3	0	0	0	0	0	0
Balance	**19.6**	20.0	29.6(+/+)	30.3(+/+)	26.9(+/+)	31.4(+/+)
Soyalarge	8.8	**7.3**	8.2	**7.3**	7.9	9.7
Tic-tac-toe	6.6	**3.4**	10.0(+/+)	3.5	8.6(Δ/+)	18.4(+/+)
Car	**4.1**	4.2	8.3(+/+)	5.9(+/+)	8.3(+/+)	9.2(+/+)
DNA	4.5	4.4	6.2(+/+)	5.1	5.8	8.9(+/+)
Mushroom	0.1	0.1	**0**	**0**	**0**	**0**
Nursery	1.0	**0.9**	2.8(+/+)	1.3(+/+)	2.6(+/+)	3.6(+/+)
RO with J48 win/draw/lose			7/5/1	5/7/1	5/7/1	9/3/1
RO with RS win/draw/lose			7/6/0	5/8/0	6/7/0	9/4/0

Monks1 dataset has six attributes and two classes. The classification is Y = 1, if $(x_1 = x_2) \vee (x_5 = 1)$. All the other data points belong to class 2. When we treat data as continuous, the first concept $(x_1 = x_2)$ is a diagonal concept. J48 trees are restricted to *orthogonal* decision boundaries. In other words, decision trees divide the input attribute space into rectangular regions whose sides are perpendicular to the attribute axis. Decision trees have a representational problem because of this orthogonal property; they cannot learn diagonal concepts properly. Ensembles of decision trees solve this problem, as combined results of decision trees produce a good approximation of a diagonal concept [7]. The quality of the approximation depends on the diversity of decision trees in the ensemble. RO with RS trees are more diverse as compared to RO with J48 trees. Hence, ROE with RS can learn this diagonal concept in Monk1 data better than ROE with J48.

Building a good ensemble depends on the creation of diverse decision trees. We create diverse decision trees by imposing random ordinality to categorical attributes values that in turn create different node splits. The diversity in node splits is the key for diverse decision trees. If we have $|A|$ attribute values, these attributes will be present in different trees in different order, the possible number of different splits from these attribute values is $2^{(|A|-1)} - 1$. If $|A|$ is small, there is a large possibility that different trees have same node splits, and we may not get very diverse trees. Tic-Tac-Toe data has only 3 attribute values for each attribute.

Hence, there is only three possible node splits for each attribute (we are taking the case when all the attribute values are present in the node). Different trees can have one of the three possible node splits for a attribute. It means that there is a large possibility that different trees have same node splits. In this condition, the trees in the ensemble will not be very diverse. When we combine the attribute randomization of RS with RO, we observe a large improvement in the classification error as compared to ROE with J48 (the average error reduced from 6.6% to 3.4%). Better diversity of ROE with RS is the reason for this improvement. In the next section, we present the various studies to analyze RO trees and RO ensembles.

6 Study of RO Ensembles and RO Trees

One of the motivation of ROE is that it avoids data fragmentation problem. In this section we study the effect of data fragmentation on ROE and RO tree sizes.

6.1 Study of Data Fragmentation for ROE

Data fragmentation may affect the performance of decision trees. We have carried out a controlled experiment to see how different ensemble methods perform with respect to the number of attribute values. For this purpose, we selected two pure continuous datasets; Segment and Vehicle. We converted these datasets into categorical datasets using equal width discretization. We studied various ensemble methods on these discretized datasets; varying the numbers of bins to see its effect on different ensemble methods. We performed five replications of a two-fold cross-validation. The results (Fig. 2) suggest that classification errors of RO ensembles are relatively unaffected. When we increase the number of bins we have a small number of points in every bin; that leads to badly estimated probabilities and poor generalization. Whereas, RO ensembles have binary decision trees so they are more robust to the data fragmentation problem.

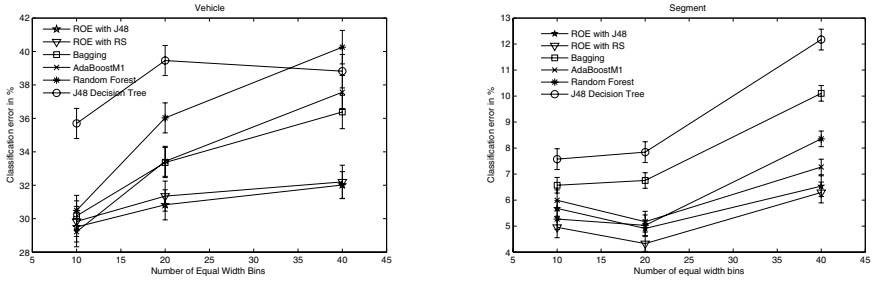

Fig. 2. Effect of equal width discretization on various ensemble methods for Vehicle and Segment datasets. RO resists fragmentation as space grows.

Table 4. The average sizes of RO trees and multi-split J48 trees for different datasets

Name of dataset	Size of the training data	The average number of leaves/size of RO trees (J48)	The average number of leaves/size of multi-way split J48 trees
Car	864	54/107	127/174
DNA	1587	76/151	211/281
Tic-Tac-Toe	479	49/97	92/142
Promoter	53	6/11	13/17

6.2 RO Tree Sizes

Smaller trees have greater statistical evidence at the leaves. Motivated by Occams Razor, small trees are preferable. As RO trees have binary splits RO trees are more likely to have smaller sizes than that of multi-split decision trees. We studied RO tree sizes for various datasets; Car, DNA, Tic-Tac-Toe and Promoter. The experiments were conducted following the 5×2 cross-validation and 50 RO trees are created in each run.

In the table 4, we present the average sizes of RO trees (J48 decision trees created using datasets generated by RO method) and normal multi-split J48 decision trees for different datasets. For all the datasets, RO trees are smaller that normal multi-split J48 decision trees. For example, for DNA dataset, the average size of RO trees is 151 whereas the average size of normal multi-split J48 decision trees is 281. **These results indicate that RO helps in creating smaller decision trees**.

7 Conclusion

In this paper, we have presented a new ensemble method to build *diverse binary decision trees for datasets consisting of multi-valued categorical attributes*. We convert categorical attributes into continuous attributes by randomly assigning integer values to categorical attribute values. As the transformation to continuous data is random, diverse datasets are created. When a decision tree is constructed by treating these new attributes as continuous ones, we have binary splits at the nodes giving binary decision trees. The theoretical study suggests that for multi-valued categorical attributes with certain properties, the information gain ratio of a binary split of RO attributes may be equal to or greater than a multi-way split. We create two types of ensembles using RO. In the first, we use J48 (the WEKA [15] implementation of C4.5) as the base model for the ensemble. In the second, we combine the attribute randomization of Random Subspaces with Random Ordinality. The comparative study on 13 different datasets from the UCI repository suggest that ROE significantly outperform other popular ensemble methods in terms of test error. The study shows that ROE avoids the data fragmentation problem and RO trees are significately smaller than multi-way split trees. ROE is easy to implement and parallel implementation of ROE is also possible.

In this present work, we imposed random ordinality to each attribute independently. In future we will also take interdependencies of attributes into consideration while imposing random ordinality. The "take-home" message of this paper is that, when categorical attribute values have no *intrinsic order*, this property can be exploited to build a successfully performing ensemble of diverse binary decision trees.

References

1. Alpaydin, E.: Combined 5 x 2 cv f Test Comparing Supervised Classification Learning Algorithms. Neural Computation 11(8), 1885–1892 (1999)
2. Bratko, I., Kononenko, I.: Learning Diagnostic Rules from Incomplete and Noisy Data, Seminar on AI Methods in Statistics, London (1986)
3. Breiman, L.: Bagging Predictors. Machine Learning 24(2), 123–140 (1996)
4. Breiman, L.: Random Forests. Machine Learning 45(1), 5–32 (2001)
5. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth International Group, CA (1985)
6. Dietterich, T.G.: Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. Neural Computation 10, 1895–1923 (1998)
7. Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
8. Dietterich, T.G.: An Experimental Comparison of Three Methods for Constructing Ensembles of Decision trees: Bagging, Boosting, and randomization. Machine Learning 40(2), 1–22 (2000)
9. Fayyad, U.M., Irani, K.B.: The Attribute Selection Problem in Decision Tree Generation. In: Proc. AAAI 1992. MIT Press, Cambridge (1992)
10. Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)
11. Geurts, P., Ernst, D., Wehenkel, L.: Extremely Randomized Trees. Machine Learning 63(1), 3–42 (2006)
12. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience, Hoboken (2004)
13. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
14. Vilalta, R., Blix, G., Rendell, L.: Global Data Analysis and the Fragmentation Problem in Decision Tree Induction. In: Proceedings of the 9th European Conference on Machine Learning, pp. 312–328 (1997)
15. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)