Nested Dichotomies Based on Clustering

Miriam Mónica Duarte-Villaseñor, Jesús Ariel Carrasco-Ochoa, José Francisco Martínez-Trinidad, and Marisol Flores-Garrido

Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro #1, 72840, Santa María Tonantzintla, Puebla, Mexico
{mduarte,ariel,fmartine,mflores}@inaoep.mx

Abstract. Multiclass problems, i.e., classification problems involving more than two classes, are a common scenario in supervised classification. An important approach to solve this type of problems consists in using binary classifiers repeated times; within this category we find nested dichotomies. However, most of the methods for building nested dichotomies use a random strategy, which does not guarantee finding a good one. In this work, we propose new non-random methods for building nested dichotomies, using the idea of reducing misclassification errors by separating in the higher levels those classes that are easier to separate; and, in the lower levels those classes that are more difficult to separate. In order to evaluate the performance of the proposed methods, we compare them against methods that randomly build nested dichotomies, using some datasets (with mixed data) taken from the UCI repository.

Keywords: Nested Dichotomies, Binarization, Multiclass Problems, Supervised Classification.

1 Introduction

Supervised classification is one of the main issues in pattern recognition, which is applied in different fields such as medicine, astronomy, and economy, among others. In the most well-known scenario there are only two different classes to which each object can be assigned (binary classification), however, it is common to find problems in which more than two classes are involved (multiclass classification). Although multiclass classifiers have been developed to deal with multiclass problems, these problems become harder when the number of classes grows and, therefore, it is more likely to make classification mistakes.

An alternative approach to solve a multiclass problem consists in decomposing the problem into several binary classification problems; in this way the original problem is simplified and it is expected to achieve better classification accuracy. This later alternative is called binarization and among its most important approaches we find *One-vs-One* (OVO) [1], *One-vs-All* (OVA) [2] and nested dichotomies [3].

In a nested dichotomy, a multiclass problem is divided into several binary classificacion problems by using a binary tree whose root contains the set of all the problem classes. Then, the classes are split into two subsets of classes,

L. Alvarez et al. (Eds.): CIARP 2012, LNCS 7441, pp. 162–169, 2012.

called superclasses, and a model is created to differentiate between them. This process is repeated, splitting superclasses until they contain a single class from the original set, i.e., each leaf of the tree contains only one class. In order to classify a new object, the constructed tree is traversed using a binary model to choose the branch that must be followed at each level; when a leaf is reached, its associated class is assigned to the object.

For each multiclass problem, it is possible to construct different nested dichotomies; finding a good one could help to reduce classification mistakes. There are several ways to build a nested dichotomy proposed in the literature [3–5], but most of them separate classes in a random way, which does not guarantee finding a good nested dichotomy. In this work, we propose three deterministic methods to build a nested dichotomy that separates first, at the upper levels of the tree, the more easily separable classes and leaves to lower levels the separation among classes that are hard to distinguish from each other. In a nested dichotomy, errors that appear in certain level cannot be corrected in lower levels and, therefore, they become classification errors in the final result. For this reason, it makes sense to try to reduce classification errors in upper levels and we believe that this can be achieved by following the proposed strategy, obtaining, as a consequence, a better classification accuracy.

The rest of this paper is organized as follows: in Section 2, we present some previous work on nested dichotomies. Section 3 contains a description of the proposed methods: Nested Dichotomy based on Clustering (NDC), Nested Dichotomy based on Clustering using Radius (NDCR) and Nested Dichotomy based on Clustering using Average Radius (NDCA). In Section 4, we show a series of experiments in which the proposed methods are compared against other state-of-the-art methods that use a random strategy. Finally, in Section 5, we present some conclusions and future research lines.

2 Previous Work

Choosing a nested dichotomy, given a multiclass problem, is not a trivial issue. Each multiclass problem can be decomposed in many different nested dichotomies. In [4] it is shown the recurrence relation that gives the number of nested dichotomies for a problem with n classes ($t(n) = (2n-3)t(n-1)$); for a problem with 12 classes, for instance, there are 13749310575 possible nested dichotomies. The classification results obtained by different nested dichotomies can vary, since each nested dichotomy contains different binary problems to be solved. Most of the works on nested dichotomies randomly choose the order in which the classes are separated. We can mention at least three of them.

In 2004, Frank and Kramer [3] propose the use of a binary tree that recursively splits the class set into dichotomies. At each internal node of the tree, including the root, the class set is partitioned into two subsets. The authors state that there is no reason to prefer a dichotomy over another one, and, therefore, they randomly choose the classes that go into every set partition. They also propose the use of an ensamble of nested dichotomies (END) in order to get better classification results.

Later, in 2005, Dong *et al.* [4] use an ensamble of nested dichotomies in which they consider only balanced dichotomies, i.e., when building a nested dichotomy, they randomly split the classes at each internal node but taking care of keeping an equilibrium between the number of classes at each child node; they call this method ECBND. They also propose a variant of the method (EDBND) in which the data (instead of the number of classes) is kept balanced at each internal node.

In 2010, Rodriguez *et al.* [5] consider that ensamble methods frequently generate better results than individual classifiers and propose the use of nested dichotomies of decision trees as base classifiers in an ensemble; they call this approach *Forest of Nested Dichotomies* (FND). In order to form the ensambles they consider three strategies: bagging [6], AdaBoost [7] and MultiBoost [8].

Finally, it is important to mention the work of Aoki and Kudo [9], who, in 2010, propose a top down method for building class decision trees, which are similar to nested dichotomies. However, class decision trees allow using multi-class classifiers to separate two groups of classes. The method proposed by Aoki and Kudo decides which classes must be separated, in each node, by testing different classifiers, and selecting those groups of classes producing lower error rates. Additionally, this method applies feature selection before evaluating the error rate of a classifier. Since the selection of the best classifier and the best subset of features for separating the best separable groups of classes, the method proposed by Aoki and Kudo is very expensive in time, and not always produces good results.

As stated before, when constructing a nested dichotomy, following certain criteria can help to obtain better classification quality. For this reason, in this paper we propose three methods that allow to build a nested dichotomy in a non-random and inexpensive way.

3 Proposed Methods

When a binary classifier in a nested dichotomy makes a mistake, the error is spread to lower levels of the tree, where it cannot be corrected. For this reason, it is important to reduce the number of errors in the upper levels of the tree. Following this idea, we propose to build nested dichotomies in which classes that are more easily separable are considered first, at the upper levels of the tree, and those classes that are harder to differentiate are postponed until lower levels.

3.1 Nested Dichotomy Based on Clustering (NDC)

The basic idea of the Nested Dichotomy based on Clustering (NDC) method is that classes with greater distance between each other are easier to separate and, therefore, they should be separated at the upper levels of a nested dichotomy. To determine the distance among classes, we compute the centroid of each class and measure the distance among them. Once these distances are computed, we find the two classes with centroids furthest away from each other, say m_1 and

m_2, and the rest of the classes are clustered into two groups using m_1 and m_2 as group centers.

We next describe the proposed method NDC, distinguishing between two main phases: the construction of the nested dichotomy and the classification process itself.

1. **Construction of the nested dichotomy.** Given a dataset of class-labeled objects, where each object is described by a set of attributes:
   (a) Choose the mean of each class as centroid of the class, taking into account all dataset instances. If there are non-numeric attributes, we choose, for each class, the object with the greatest similarity, on average, to all other instances of its class.
   (b) Create the tree root with all the classes and the centroids chosen in the previous step.
   (c) Create a dichotomy. This step is performed recursively, over each node containing two or more classes, until there is only one a class at each tree leaf.
      i. Choose the two classes whose centroids have the greatest distance between each other. Use the identified centroids as group centers.
      ii. Each class at the current node is grouped with the closest center, considering the distance between its class centroid and the defined group centers. If the distance toward the two centers is equal, the class is put into the first group.
      iii. A child node is created for each of the groups and the process is repeated.
   (d) Once the tree is built, a binary classifier is trained at each internal node of the tree in order to separate the groups of classes of its child nodes. For this purpose, all the instances from the training set corresponding to the classes grouped at each child node are used.
2. **Classification process.** Given an instance:
   (a) The tree is traversed, starting from the root and following the branches indicated by each binary classifier, until a leaf node is reached.
   (b) The class associated with the final tree leaf is assigned to the instance.

3.2 Nested Dichotomy Based on Clustering Using Radius (NDCR)

A drawback of the NDC method is that there can be classes that, despite having centroids that are far from each other, are difficult to separate due to considerable overlapping. The opposite case is also possible, i.e., classes with close centroids that, however, do not have overlap among them and, as a consequence, these classes are easy to separate. For this reason, we propose a variant of the NDC method that involves each class radius to compute the distance among classes. The class radius is computed as the distance between the class centroid and the element, within the class, furthest away from the centroid. Thus, in order to measure the distance between two classes, C_1 and C_2, given the centroid of each

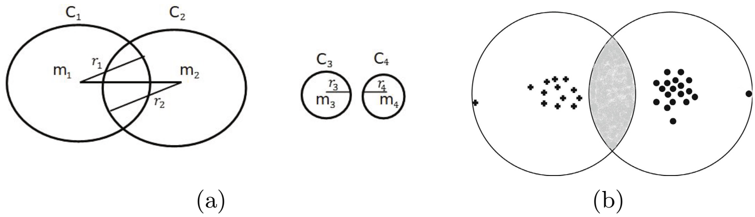

Fig. 1. (a) Representation of the distance between classes using their radius. (b) Example that shows the drawback of obtaining the class radius by computing the distance to the furthest away instance.

class, m_1 and m_2 respectively, the distance between them, $d(m_1, m_2)$, and the radius of each class, r_1 and r_2, we propose the function

$$D(C_1, C_2) = \frac{d(m_1, m_2)}{r_1 + r_2} \tag{1}$$

as a measure of the distance between classes. Note that $D = 1$ indicates that the classes are next to each other but they do not overlap, $D > 1$ that the classes are separated, and $D < 1$ that there is overlapping between the classes. The Fig. 1a shows, in general, the distance between two classes using their radius. In this figure, the centroids of classes C_1 and C_2, m_1 and m_2 respectively, are far from each other, but since both radius, r_1 and r_2, are big, there is overlapping between classes C_1 and C_2. Therefore, according to our distance, these classes are close to each other. On the other hand, the centroids of C_3 and C_4, m_3 and m_4 respectively, are close to each other, but since both radius, r_3 and r_4 are small, there is not overlapping between classes C_3 and C_4. Therefore, according to our distance, these classes are far from each other.

We call this method Nested Dichotomy based on Clustering using Radius (NDCR). The steps of the method are similar to the ones described for NDC, except that the Phase 1 requires an aditional step in which the radius of each class is computed, and that distance D, given in (1), is used in the step 1c instead of the distance between centroids.

3.3 Nested Dichotomy Based on Clustering Using Average Radius (NDCA)

A shortcoming of the method described in the previous section is its sensitivity to outliers. If an element of a class is far away from the rest of the class elements, the radius of the class will be big and, when it is used to compute the distance D, will mislead to think that there is overlapping between classes when, in fact, this might not be true; see Fig. 1b for an example. In order to deal with this scenario, we propose to compute each class radius as the average of the distance between the class centroid and all the elements of the class. Thus, this method is similar to NDCR, but the function used to measure the distance between classes is given by

$$D'(C_1, C_2) = \frac{d(m_1, m_2)}{r'_1 + r'_2} \quad (2)$$

where r'_i represents the average of the distance between the centroid m_i and the elements within the class C_i.

4 Experimental Results

We conducted experiments on 20 datasets taken from the UCI Machine Learning repository [11] that have been commonly used to evaluate methods for constructing nested dichotomies; the Table 1 shows details of these datasets.

In all the experiments, we used 10-fold cross validation, using the same folds for all the methods. The binary classifiers that we use are C4.5, Random Forest, 1-NN, 3-NN and Naive Bayes taken from Weka 3.7.1. In our methods, as well as for 1NN and 3NN, we used, as distance function, the Heterogeneous Euclidean-Overlap Metric (HEOM) [10], which allows comparing object descriptions that include numerical and non-numerical attributes (mixed data). We compared our methods against: ND [3], ND-CB (Class Balanced) and ND-DB (Data Balanced) [4], as well as ensembles based on these methods, all of them also taken from Weka 3.7.1. The proposed methods are implemented in Java. All the experiments were conducted on a PC with a Pentium Dual-Core processor at 2.9 Ghz and 3Gb of RAM, running Linux-Ubuntu 11.04.

The Table 2 shows the results obtained in our experiments. The columns show the average accuracy, over the 20 datasets, of the different methods to build nested dichotomies using different base classifiers, as well as the results obtained with nested-dichotomies ensembles built using bagging [6], AdaBoost [7] and MultiBoost [8] approaches. The best result for each classifier is highlighted in bold.

In the Table 2, for each method, it is also shown the average accuracy over all the used base classifiers. In all the cases, the methods NDC and NDCR got the highest average accuracy (this is even more clear in the general average shown at the bottom of the table), suggesting that a better classification quality is achieved through the nested dichotomies built in a non-random way.

Table 1. Datasets used in the experiments

		Attributes					Attributes		
Dataset	Instances	Num.	Nom.	Classes	Dataset	Instances	Num.	Nom.	Classes
Anneal	898	6	32	6	Optdigits	5620	64	0	10
Audiology	226	0	69	24	Page-blocks	5473	10	0	5
Balance-scale	625	4	0	3	Pendigits	10992	16	0	10
Car	1728	0	6	4	Primary-tumor	339	0	17	22
Dermatology	366	1	33	6	Segment	2310	19	0	7
Mfeat-factors	2000	216	0	10	Soybean	683	0	35	19
Mfeat-Karhunen	2000	64	0	10	Vehicle	846	18	0	4
Mfeat-morphological	2000	6	0	10	Vowel-context	990	10	2	11
Mfeat-pixel	2000	0	240	10	Waveform	5000	40	0	3
Nursey	12960	0	8	5	Zoo	101	1	15	7

Table 2. Results of the performed experiments using individual nested dichotomies as well as ensembles of them. The columns show the average of the accuracy obtained by each method, using different base classifiers; the first three columns correspond to the proposed methods.

		Method					
	Base Classifier	NDC	NDCR	NDCA	ND	ND-DB	ND-CB
	C4.5	**88.81**	88.58	88.48	87.22	87.63	88.02
	RandomForest	93.07	92.89	92.73	93.51	**93.63**	93.52
	1-NN	93.65	**93.74**	93.50	93.64	93.63	93.65
	3-NN	88.97	88.76	88.78	88.92	88.87	**88.99**
	Naive Bayes	77.32	**78.35**	78.01	73.10	72.59	72.60
	Average	88.37	**88.46**	88.30	87.28	87.27	87.36
Bagging	C4.5	91.60	91.72	91.70	92.33	92.22	**92.68**
Ensemble:	RandomForest	94.32	94.17	93.94	94.47	94.38	**94.50**
	1-NN	93.33	93.36	93.32	93.40	**93.43**	93.41
	3-NN	**90.24**	89.29	89.32	90.01	90.01	89.99
	Naive Bayes	**82.61**	80.56	79.49	80.32	80.49	80.65
	Average	**90.42**	89.82	89.55	90.11	90.11	90.25
AdaBoost	C4.5	93.83	93.69	93.47	93.79	93.64	**93.97**
Ensemble:	RandomForest	94.36	94.23	94.36	94.55	94.37	**94.58**
	1-NN	**93.31**	93.28	93.28	93.19	93.28	93.19
	3-NN	92.51	92.49	92.46	92.55	92.48	**92.65**
	Naive Bayes	83.55	**83.62**	82.70	80.22	81.15	81.31
	Average	**91.51**	91.46	91.25	90.86	90.98	91.14
MultiBoost	C4.5	93.37	93.29	93.21	93.40	**93.59**	93.47
Ensemble:	RandomForest	94.30	94.12	94.17	94.34	**94.36**	94.32
	1-NN	93.35	**93.47**	93.33	93.30	93.31	93.27
	3-NN	91.11	**91.32**	91.34	91.22	91.20	91.02
	Naive Bayes	82.39	**83.44**	82.39	80.45	80.70	80.16
	Average	90.90	**91.13**	90.89	90.54	90.63	90.45
	Gral. Average	**90.30**	90.22	90.00	89.70	89.75	89.80

5 Conclusions

We proposed three methods, NDC, NDCR and NDCA, to build nested dichotomies in a non-random way. The main idea in these methods is to separate in the upper levels of the tree the classes that are easier to separate and separate in the lower levels the classes that are harder to separate. The first method determines which classes are easier to separate by clustering the classes using the distance among the centroids of the classes. The second method takes into account, besides the distances among centroids, the radius of each class, trying to determine if there is overlapping among the classes. Finally, the third method replaces the radius of each class by the average of the distance between the class centroid and all other elements within the class.

Experiments were performed on individual nested dichotomies and ensambles of nested dichotomies. The three proposed methods showed, in average, better classification accuracy than ND, ND-CB (Class-Balanced) and ND-DB (Data-Balanced), which build nested dichotomies in a random way. It must be also highlighted that, whereas the proposed methods are deterministic, the methods that build nested dichotomies in a random way could show variations in their classification accuracy in different runs (sometimes for the worst). Although in the proposed methods there is an additional cost (respect to random methods) for choosing the separable classes at each level of the nested dichotomy, the proposed separability criteria are inexpensive to evaluate, compared to other approaches previously proposed in the literature.

In the future, it would be important to explore different ways to assess the overlapping among classes, for instance, measuring how many objects are located in regions where most of the objects belong to other classes. In addition, methods of attribute and/or instance selection could be used in the binary classifiers, in order to improve the classification accuracy.

References

1. Hastie, T., Tibshirani, R.: Classification by pairwise coupling. Ann. Statist. 26(2), 451–471 (1998)
2. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. J. Mach. Learn. Res. 5, 101–141 (2004)
3. Frank, E., Kramer, S.: Ensembles of Balanced Nested Dichotomies for Multi-class Problems. In: 21st International Conference on Machine Learning, ICML 2004, pp. 305–312. ACM, New York (2004)
4. Dong, L., Frank, E., Kramer, S.: Ensembles of Balanced Nested Dichotomies for Multi-class Problems. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 84–95. Springer, Heidelberg (2005)
5. Rodrguez, J.J., Garca-Osorio, C., Maudes, J.: Forests of Nested Dichotomies. Pat. Rec. Lett. 31(2), 125–132 (2010)
6. Breiman, L.: Bagging predictors. Mach. Learn. 24(2), 123–140 (1996)
7. Freund, Y., Schapire, R.E.: Game theory, on-line prediction and boosting. In: 9th Annual Conference on Computational Learning Theory, COLT 1996, pp. 325–332 (1996)
8. Webb, G.I.: MultiBoosting: A Technique for Combining Boosting and Wagging. Mach. Learn. 40(2), 159–196 (2000)
9. Kazuaki, A., Mineichi, K.: A top-down construction of class decision trees with selected features and classifiers. In: International Conference on High Performance Computing and Simulation, HPCS, pp. 390–398 (2010)
10. Wilson, D.R., Martinez, T.R.: Improved heterogeneous distance functions. J. Art. Intell. Res. 6, 1–34 (1997)
11. Frank, A., Asuncion, A.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA (2010), `http://archive.ics.uci.edu/ml`