IMPROVING CLASSIFICATION WITH AUTOMATED SELECTION OF A COMBINED CLASSIFIER

MA. A. CORONA-NAKAMURA
Depto. de Ciencias Computacionales, Universidad de Guadalajara
Av. Revolución No. 1500, S. R. Guadalajara, Jalisco 44840 México.
acorona@megared.net.mx

R. RUELAS
DIP - CUCEI Universidad de Guadalajara
Apdo. Postal 307. Zapopan, Jalisco 45101 México.
rruelas@newton.dip.udg.mx

ABSTRACT

In this paper we present how the classification results can be improved using a set of classifiers working together in a combined classifier. This one must be compound of different kind of classifiers in order to get better results. The combined classifier is defined from seven classifiers, each one working alone, and then selecting the best combination of them. The results show that the combined classifier acts like a more efficient classifier. This can be seen from the individual databases used, but it is more significant when different databases are considered.

KEYWORDS: Classification, Combined classifiers, Combination of multiple classifiers, Classifier fusion, Supervised learning, Pattern recognition.

INTRODUCTION

In many real-life problems there are objects that need to be recognized. Many times this task is done by humans but each time, more and more, this function is automated. This requires defining the features that make the objects similar or different among themselves, and using the system that can see these similarities/differences.

In this work we only consider objects of known class, or supervised classification. So, the problem of classification is to train a classifier in order to recognize the elements of each class; those used for the training and new ones.

The classification of objects is difficult because there is a few data to train and to test the classifier, because even if we have a lot of data, the number of features available are not sufficient to describe correctly the elements of each class, or because, according to the features of the objects, some of them could be included in more than one class. Then, some elements are very hard to recognize correctly. In those circumstances, the results do not always satisfy the requirements of the application; for example, in a supervised learning it would be desirable to have zero errors.

In order to try to get better results than those obtained with individual classifiers, a cooperation of classifiers was used [1, 2], considering that this would help to obtain better results and to approach the concept of optimal classifier. This last one is a classifier that could be applied to all databases and gives, for a criterion and all databases, zero errors.

The classifiers are evaluated according to the quality of their training and the accuracy in the classification of new examples. Usually, when some classifiers are combined we can get better results than using them in an individual way. However, this is not systematic. Nevertheless, this must be avoided if we really want to improve the classification with the combined classifier [3]. It is then necessary to look for the characteristics that help us to improve the results, or in the worst case to get results as good as those given by individual classifiers.

In this work we propose to use different types of classifiers for a combination. The results show that this last one works as well or better than its components for one database, and better results if several databases are considered. Also, as there are similar steps in the training and selection of classifiers, we propose an automated selection of the combined classifier. See figure 1.

The next sections are organized as follows: in the second section we present different individual classifiers and their combination. In the third section we present the results, and a discussion in the fourth section. In the last section we give some conclusions.

THE COMBINED CLASSIFIER

The combined classifiers are groups of classifiers working together in different ways. Generally, they are considered in three groups. These are: the parallel, the serial and the hierarchical (tree-like) groups [4]. Here we consider those of the first group.

The classifiers selected for this study are: TREE (T), PARZEN (P), FISHER (F), QD (Q), LD (L), KNN (K), and NM (N), as shows figure 1. And the selected databases are: Iris (IR), Iris23 (IR23), Glass (GL), Wine (WI), and Thyroid (TH). These databases were selected because they are very common for training and testing classifiers.

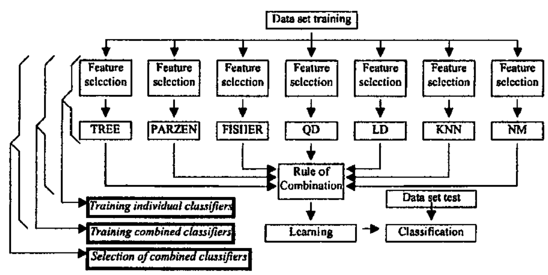

Figure 1. Automated selection of the best combined classifier.

As we are interested, not only in the combination of classifiers but also in the results obtained combining different numbers of them, we considered to compare the best combined classifier using 3, 5, and 7 of the total classifiers studied.

Features selection

In the applications of the classifiers, features selection is very important because it could help to have good classification results, to reduce the number of features, and to improve the reliability of the estimation of the performance [5, 6].

For feature selection we need two elements [7], the criterion to evaluate each feature and the search algorithm to find the best features. The criterion is based on the sum or the minimum of the distances, but it could be any untrained classifier [8]. For the feature selection, the algorithm takes the best feature each time, until a given number k of them is reached, or we can begin eliminating the worst feature each time until k [7]. For this work we apply the first approach.

The number of features for each classifier was determined according to the minimal error obtained. The features selected are as follows: 3 for the QD applying the NN (nearest neighbor) criterion, 5 in the LD applying the NN criterion, 3 in the KNN applying the Mahalanobis distance criterion, 3 in NM with the LD classifier for the feature selection, and for the TREE, PARZEN and FISHER classifiers the best results were obtained using all the features.

Classifiers

The TREE classifier. The TREE classifier partitions the feature space of all possible objects into subregions described in leaves. Different subsets of the original feature space are used at different levels of the tree. Each sample is then classified by the label of the leaf it reaches [9]. There are different criterion functions, stopping rules or pruning techniques [8]. The criterion functions were chosen for this study. We consider here the binary decision trees. So, a distance is calculated between the object to be classified and its neighbors, and a threshold is then necessary to decide the final class of the object [9].

The PARZEN classifier. This is a classifier based on the class-conditional probabilities. It models the conditional probabilities of class $P(x|c_i)$, for each class c_i, by the evaluation of the densities of the kernels. It uses the multi-normal density function, with mean consisting of all training samples and the diagonal covariance matrix with the overall variance h^2. The parameter h is determined by maximum-likelihood estimation. The approach used is feature-based which computes the Euclidean distances from the considered dissimilarities [9]. Its smoothing parameter can be estimated using an optimization of the density [8]. In this work this parameter is equal to one.

The FISHER classifier. The FISHER classifier finds the linear discriminant function between the classes in the training database by minimizing the errors in the least square sense. It produces a single linear classifier between each class and the combined set of other classes [8].

The linear discriminant classifier. For the linear discriminant analysis method, we need to look for linear combinations of selected variables, called discriminant functions, which provide the best separation between the classes. The discriminant functions are calculated from the mean values of the classes, the covariance matrix and a given data. In fact, the discriminant functions are hyperplanes that separate one class from the rest of them [10].

The quadratic discriminant classifier. A classifier with a non linear discriminant function in the features, is a non linear classifier. As such function offers a greater number of degrees of freedom, a larger number of training objects is needed also. Here we use the normal densities based on the quadratic classifier (QD). This classifier assumes normally distributed classes, and finds the quadratic surface which optimizes classification performance [11].

The KNN classifier. In the KNN (K-Nearest Neighbors) it is necessary to compute all the distances, the Euclidean distance, between a new object and all the objects used to train the classifier. The object is then considered as an element of the class to which appertain most of its k-nearest objects. There are many algorithms based in the principle of the KNN [9]. Here we use the KNN based on the prototypes, with k equal to 1.

The NEAREST MEAN classifier. The simplest linear classifier is the nearest mean classifier (NM), which assigns an unknown object to the class of the nearest mean. The decision rule is based on the (pseudo)Euclidean distance [12].

The combining rules

Once the classifiers are applied to a new object, the results are combined in order to decide the class of the object. There are different rules for the combination of classifiers. In this work we use the next: the min, the max, the mean, the product and the majority vote [13]. The table IV shows the results.

RESULTS

For the selection of the combined classifier we proceeded as follows: each classifier was trained alone, looking for a good criterion and the best number of features. Then, they were combined using different rules and considering all the possible combinations. Finally, the combined classifier with the minimum percentage of error was selected.

The next tables show the results obtained applying different classifiers, individual and combined, and different rules of combination. Each table presents the results in two parts. The first one where we use all the database to train and to test the classifiers, and the second one, where the database was divided in two parts, to train and to test the classifiers.

Table I. Selected combined classifiers 3 of 7 for each database.

ALL DATA					ODD - EVEN DATA				
IR	IR23	GL	WI	TH	IR	IR23	GL	WI	TH
T, P, K	T, P, K	P, K, N	T, P, F	T, P, Q	T, Q, L	T, Q, N	P, K, N	F, K, N	T, L, N

Table II. Individual results of the classifiers.

	ALL DATA					ODD - EVEN DATA				
	IR	IR23	GL	WI	TH	IR	IR23	GL	WI	TH
TREE	0	0	0	0	0	2	2	2	12	11
PARZEN	0	0	0	0	0	3	3	1	31	7
FISHER	16	3	49	0	29	8	3	24	0	15
QD	3	3	1	1	7	3	3	29	4	4
LD	3	3	9	0	17	3	3	11	2	9
KNN	0	0	138	50	65	3	3	69	30	32
NM	11	11	24	49	28	5	5	12	22	16
Combined	0	0	0	0	0	1	1	1	0	3

Table III. Results combining 3, 5, and 7 classifiers.

Classifiers	ALL DATA					ODD - EVEN DATA				
	IR	IR23	GL	WI	TH	IR	IR23	GL	WI	TH
3 of 7	0	0	0	0	0	1	1	1	0	3
5 of 7	0	0	0	0	0	1	2	1	0	3
7 of 7	0	0	0	0	4	3	2	1	4	5

Table IV. Results with different rules of combination and the classifiers of table I.

	ALL DATA					ODD - EVEN DATA				
	IR	IR23	GL	WI	TH	IR	IR23	GL	WI	TH
Prod	0	0	0	0	10	6	3	1	2	12
Mean	0	0	0	0	0	1	2	1	0	3
Max	0	0	0	0	0	1	1	1	0	3
Min	0	0	0	0	0	1	1	1	0	3
Major	0	0	8	0	0	2	3	5	9	6

DISCUSSION

In table II, for all data, the combined classifier is as good as the TREE and PARZEN classifiers, and it is better than the others. But, this does not imply that the classifier with the minimum of errors will be the best with new data. Now, using the odd-even data, the combined classifier works better than the others. This is important because we get good results even if some classifiers used in the combination produce a lot of errors.

From the results obtained combining the seven classifiers we can see that the combination with three of them gives the best results (see table III). The table I shows the classifiers selected for each database. As this table shows, we get a different combined classifier for each database. So, the characteristics of the classifiers, in some way, help to take into account some characteristics of each particular application. So, the selected classifiers do not depend only on their individual results because some of them produce the worst results from the set of classifiers applied to a particular database. Nevertheless in the combined classifier they help to improve the results.

For example, with the database IRIS23, the NM classifier was the worst but it helps to get better results in the combined classifier. The same happened with the KNN classifier and the GLASS database, the KNN and NM classifiers and the database WINE, and the TREE and NM classifiers are not among the best classifiers with the THYROID database.

CONCLUSIONS

When a simple classifier does not satisfy an application, it is possible to use a combination of some of them. Nevertheless, there are many parameters to take into account, so the possibilities to combine multiple classifiers. In order to simplify the selection process, in this work was considered the automatization of this task.

As the results show, very good results are obtained combining different kinds of classifiers. Besides, the combined classifiers are not necessarily compound of the best individual classifiers. Then the classifiers complement their advantages and they allow to build a more efficient classifier. This can be seen more clearly when different databases are used. So, using different kinds of classifiers helps to improve the learning and generalization capacities, situation that seems difficult to reach when we combine the same type of classifiers.

The number of classifiers used in a combination is a parameter that needs to be studied more deeply. Because increasing the number of them does not necessarily improve the results. In fact, combining a lot of classifiers degrades the results. Another point that needs to be studied is the fact that the combination of the best individual classifiers does not give the best combined classifier.

REFERENCES

1. Gama J. M. "*Combining classification algoritms.* " PhD thesis, Universidade do Porto, (1999).
2. Kuncheva L.I., Bezdek J.C., Duin R.P.W., "Decision templates for multiple classifier fusion: an experimental comparison.", *Pattern Recognition* 34 (2001), 299-314.
3. Tax D. M.J., Breukelen M., Duin R.P.W. , Kittler J. "Combining multiple classifiers by averaging or by multiplying? *Pattern Recognition.*" 33 (2000), 1475-1485.
4. Jain A.K., Duin R.P.W., Mao J. "Statistical Pattern Recognition: A Review." *IEEE Trans. on Pattern Analysis and Machine Intelligence* 22 (1) (2000), 4-37.
5. Kudo M., Sklansky J. "Classifier-Independent Feature Selection for Two-Stage Feature Selection." Advances in Pattern Recognition, Lecture Notes in Computer Science, vol. 1451, A. Amin, D. Dori, P. Pudil and H. Freeman(eds.), Springer (1998), 548-554.
6. Boz, O. "Feature Subset Selection by Feature Relevance." Submitted to: *The Nineteenth Int. Conf. on Machine Learning* (ICML-2002), (8-12 July 2002).
7. Tax D.M.J. "Exercises for Computer Service Practicum (CSP)." Course TN3571-p: Exploratory Data Learning to use basic data analysis tools. http://csp.tn.tudelft.nl/tn3571.html
8. Duin R. P. W. "*PRTools Version 3.0. A Matlab Toolbox for Pattern Recognition.*" Delft University of Technology, January (2000).
9. Pekalska E., Duin R.P.W. "Classification on dissimilarity data: a first look." *Proc. ASCI 2000, 6th Annual Conf. of the Advanced School for Computing and Imaging* (Lommel, Belgium, June 14-16), Delft (2000), 221-228.
10. Mjahed M. "Classification of Multi-jet Topologies in e+ e - collisions using Multivariate Analysis Methods and Morphological Variables." LATTICE 2001 The XIX International Symposium on Lattice Field Theory (August 2001).
11. Withagen P. J., Schutte K. Vossepoel A., Breuers M. "Automatic classification of ships from infrared (FLIR) images." *Signal Processing, Sensor Fusion, and Target Recognition VIII, SPIE Proceedings* Vol. 3270, Orlando, USA, (1999), 180-187.
12. Pekalska E., Paclik P., Duin R.P.W. "A generalized kernel approach to dissimilarity based classification." *Journal of Machine Learning Research* 2 (2) (2001), 175-211.
13. Duin R.P.W, Tax D.M.J. "Experiments with Classifier Combining Rules." J. Kittler, F. Roli (eds.), *Multiple Classifier Systems*, Springer Verlag, Berlin (2000), 16-29.