Bagging classifiers based on Kernel density estimators

Edgar Acuna*
University of Puerto Rico at Mayaguez, Department of Mathematics
Mayaguez, PR 00680
edgar@math.uprm.edu

Alex Rojas
University of Puerto Rico at Mayaguez, Department of Mathematics
Mayaguez, PR 00680
alexr@math.uprm.edu

Abstract

A lot of research is being conducted on combining classification rules (classifiers) to produce a single one, known as an *ensemble*, which in general is more accurate than the individual classifiers making up the *ensemble*. Two popular methods for creating *ensembles* are *Bagging* introduced by Breiman, (1996) and, Ada*Boosting* by Freund and Schapire (1996). These methods rely on resampling techniques to obtain different training sets for each of the classifiers. Previous work has demonstrated that combining techniques are very effective for unstable classifiers, such as decision trees, neural networks and naive Bayes. In this paper we present some results in application of *Bagging* to classifiers where the class conditional density is estimated using Kernel density estimators.

Key words: Bagging, adaboosting, kernel density estimation, ensembles, classification.

1. Introduction

Many researches have investigated the technique of combining the predictions of multiple classifiers to produce a single classifier: Wolpert (1992), Breiman (1996, 1998), Quinlan (1996), Freund and Schapire (1996), Maclin and Optiz (1997), Bauer and Kohavi (1999), etc. The resulting classifier, also known as an *ensemble*, is generally more accurate than the individual classifiers making up the *ensemble*. Three popular methods for creating ensembles are: Bagging (Bootstrap aggregating) introduced by Breiman, (1996), AdaBoosting by Freund and Schapire (1996) and Arcing (Adaptively resampling and combining) by Breiman (1998). These methods rely on resampling techniques to obtain different training sets for each of the classifiers. Stacking (Wolpert, 1992) is another method to combine classifiers but it does not use resampling, in this case the learning data set is divided in v parts similar to v-fold cross validation and then a classifier is estimated in each of the part. The stacked classifier is a linear combination of the single classifiers.

Breiman (1996) heuristically defines a classifier as unstable if a small change in the learning data L can make large changes in the classification. Unstable classifiers have low bias but high variance, meanwhile the opposite occurs for stable classifiers. CART and neural networks are not stable classifiers, linear discriminant analysis and K-nearest neighbor classifiers are stable. It is expected a reduction of the bias and variance after the classifiers are combined.

Table 1. Results of previous experiments in Bagging

Reference	Classifier	Relative Error Reduction (%)	# of datasets (w-l-t)
Breiman (1996)	CART	29.0	7 (7-0)
Freund & Schapire (1996)	C4.5	20.0	27(22-2-3)
Quinlan (1996)	C4.5	10.0	27(24-3)
Maclin & Opitz (1997)	C4.5	18.5	23(21-1-1)
Maclin & Opitz (1997)	Neural Nets	13.3	23(22-0-2)
Breiman (1998)	CART	36.0	11(11-0)
Bauer & Kohavi (1999)	MC4	14.5	14(14-0)

Table 2. Results of previous experiments with Adaboosting

Reference	Classifier	Relative Error Reduction (%)	# of datasets (w-l-t)
Freund & Schapire (1996)	C4.5	24.8	27(22-4-1)
Quinlan (1996)	C4.5	15.0	27(21-6)
Maclin & Opitz (1997)	C4.5	22.0	23(21-2)
Maclin & Opitz (1997)	Neural Nets	17.1	23(19-4)
Breiman (1998)	CART	48.4	11(10-1)
Bauer & Kohavi (1999)	MC4	27.0	14(10-4)

Bagging and Adaboosting are very effective for unstable classifiers such as decision trees: CART, C4.5 and MC4 (see Breiman (B96, B98), Quinlan (Q96), Freund and Schapire (FS96), Bauer and Kohavi (BK99)) and neural networks (see Maclin and Optiz (MacO97)). A summary of these results is shown in table 1. Adaboosting applying to decision trees and Naïve-Bayes performs generally better that Bagging, but not uniformly better, sometimes they degraded compared to the single classifier. The same conclusions were obtained for neural networks classifiers (MacO97). In table 2 appears a summary of previous results in adaboosting. When tree classifiers are used, Bagging mainly reduces the variance, whereas boosting reduces, both the bias and the variance (B98 and BK99).Bagging can be easily be implemented in parallel, but Adaboosting are essentially sequential and parallelized versions have not been implemented so far (BK99).

In the table 3, we show the misclassification errors of applying combining techniques to the 10 datasets used in this paper, they are described later in section 3. Note that in 2 of these datasets: *Vehicle* and *Iris* the best single classifier beats the combining techniques. In the breastw dataset the improvement is minimum and the remaining 7 the combining techniques do a good job.

Table 3. Comparison of results of experiments on Bagging, Boosting and Arcing with the best single classifier for the datasets used in this paper

Dataset	Q96 C4.5	FS96 C4.5	Mac097 C4.5	B96 CART	B98 CART	Mac0Nnet	Best Single Classifier
Iris	5.13b	5.0ab	4.6b	NA	NA	2.9x	2.0 LDA
Sonar	19.62a	19.0a	19.7a	NA	NA	12.5a	15.5 1-NN
Glass	23.55a	22.7a	28.4b	24.9b	21.6x	31.5x	23.8 1-NN
Heart-c	21.39a	20.9b	17.4a	NA	NA	16.7b	18.2 NN
Ionosphere	NA	5.8a	6.0ab	8.6b	6.3x	7.0x	8.1 C4.5
Breast-w	4.09a	3.2b	3.1a	4.2b	3.2a	3.2x	3.3 NN
Diabetes[1]	23.63b	24.4b	21.9b	18.8b	23.9b	22.8a	22.3 Log
Vehicle[1]	22.72a	22.6a	24.8a	NA	NA	19.5a	15.0 Qdisc
German[1]	25.81b	24.6b	22.8b	NA	NA	24.3b	28.3 NN
Segment[1]	1.87a	1.4a	2.3a	NA	NA	3.7x	3.0 Ker

1: Statlog Project
b: bagging
a: Adaboosting
x: Arcing

LDA: Linear discriminant analysis
1-NN 1-Nearest neighbour
Log Logistic regression
Qdisc: Quadratic discrimination

NN: Neural Network
Ker: Kernel classifier

2. Classifiers based on Kernel Density estimators

From a Bayesian point of view, supervised classification is equivalent to compare estimates of the probabilities of belonging to each class with each other, assigning an object with measurement vector **x** to the class with the largest $\hat{f}$ (j/**x**), j=1,2….J. In order to obtain such estimates, one can estimate them indirectly via the class conditional density f(**x**/j) using the Bayes' theorem. Kernel density estimators can be used to carry out that task. For a given class j and a random sample $\mathbf{X}_1,\mathbf{X}_2,\ldots..\mathbf{X}_n$ of the p-dimensional random vector **X** with continuos components, the product kernel of the class conditional density at the point **x** is given by

$$\hat{f}(\mathbf{x}) = \frac{1}{nh_1h_2....h_p}\sum_{i=1}^{n}\prod_{v=1}^{p} K\left(\frac{\mathrm{x}_v - \mathbf{X}_{iv}}{h_v}\right)$$

where the kernel K will be usually a radially symmetric unimodal density function, for instance the multivariate normal density, and h_v represents the bandwidth of the v-th predictor. There are several approaches to select the optimal bandwidth.

In the Statlog Project (Michie, et. al. 1994), where 23 classifiers are compared in 22 datasets, classifiers based on kernel density estimators (ALLOC80) performed better than CART (a 13-8 victory) and tied with C4.5 (11-11). However, ALLOC80 appeared as the top 5 classifier for 11 datasets whereas C4.5 and CART appeared only for 5 and 3 datasets respectively. Classifiers based on kernel density estimation are unstable due to singularities presented in the log-likelihood function, and to the effect of outliers in the selection of the bandwidth.

In this paper we have used kernel density estimation with both fixed and adaptive bandwidth. The fixed bandwidth is originated by assuming that the class conditional density is a multivariate normal (see Silverman, p86). This is, $h_{opt} = (4/n(p+2))^{1/(p+4)}$, where p is the number of predictors and n is the number of instances. In the adaptive kernel, the bandwidth varies from one point to another, and its value will depend on the concentration level of data in the neighborhood of the point . The basic idea of this approach is to combine the standard kernel density estimation with the nearest neighbor approach (for more details see Silverman, p101).

We have also considered kernel density estimators for categorical predictors: binary, nominal and ordinal predictors In the first case we have followed the proposal presented in Aitchison and Aitken's paper. That is, given n p-dimensional sample data points $\mathbf{x}_j$ the kernel density estimate of f at the point **x** is given by

$$\hat{f}(\mathbf{x}) = \frac{1}{n}\sum_{j=1}^{n} h^{p-d_j^2}(1-h)^{d_j^2}$$

where $\frac{1}{2} \leq h \leq 1$, and $d_j^2 = (\mathbf{x}-\mathbf{x}_j)'(\mathbf{x}-\mathbf{x}_j)$ is equivalent to the number of disagreements in corresponding components of **x** and $\mathbf{x}_j$. The smoothing parameter h has to be estimated, usually by crossvalidation. In 1980, Titterington gave a explicit formula for h and in addition he considered also kernel density estimates for nominal and ordinal variables. A kernel density estimator for a categorical variable X with k values and vector of sample proportions **r** can be written as $\hat{f}(x) = C'\mathbf{r}$, where C=I+(1-h)G is a square matrix of order k, and G is a matrix such that G_{ii}=-1, G**1**'=**0** and G_{ij}>0 for i≠j. El smoothing parameter h is estimated by minimizing the mean squared error and, is given by

$$\hat{h} = 1 + \frac{tr(VG)}{tr(G'\mathbf{rr}'G + VGG')}$$

where $V=n^{-1}(\Lambda-\mathbf{rr}')$ and $\Lambda=\mathrm{diag}(r_1,\ldots.r_k)$.

In the multivariate case if our vector of predictors **x** can be decomposed as $\mathbf{x}=(\mathbf{x}^{(1)},\mathbf{x}^{(2)})$, where $\mathbf{x}^{(1)}$ contains the p_1 categorical predictors and $\mathbf{x}^{(2)}$ includes the p_2 continuous predictors, then a mixed product kernel density estimator will be given by

$$\hat{f}(\mathbf{x}) = \frac{1}{nh_{21}\ldots h_{2p_2}}\sum K_{p_1}^{(cat)}(\mathbf{x}^{(1)};\mathbf{x}_j^{(1)},\mathbf{h}_1)K_{p_2}^{(cont)}\left(\frac{\mathbf{x}^{(2)}-\mathbf{x}_j^{(2)}}{\mathbf{h}_2}\right)$$

where $K_{p_1}^{(cat)}$ is the kernel estimator for the vector $\mathbf{x}^{(1)}$ of categorical predictors and $K_{p_2}^{(cont)}$ is the kernel density estimator of the vector $\mathbf{x}^{(2)}$ of continuous predictors.

3. Experimental Methodology

We chose 10 datasets coming from the Machine Learning Database at University of California Irvine (UCI) to evaluate the effect of combining KDE classifiers considering mixed type of predictors: continuous and categorical and fixed as well as adaptive bandwidths. Next we describe briefly each of these datasets.

Iris (iris): It contains 150 instances, 4 continuous features and 3 classes of Iris: Setosa, Versicolor and Virginica There exists multivariate normality of the predictors in the three classes. There is not high correlation among the predictors in each of the classes. The predictors do not present outliers.

Sonar (sonar): It includes 208 instances, 60 continuous features and two classes: metal and rock. There is not multivariate normality in both classes. Few predictors are highly correlated.

Forensic Glass (glass): 214 instances, 9 features and 6 classes. The feature 9 assume the same value on the class 5 and it has been omitted, the feature 8 also assume one value in class 5 and few different values in other class and also has been deleted, the same happens with feature 6 that assume the same value in class 5. There exists multivariate normality in three classes and several predictors have normality on some classes.

Cleveland Heart disease (heartc): There are 303 instances, 13 features: continuous (1,4,5,8,10), binary (2, 6, and 9), nominal (3,11,13) and ordinal (7,12) and two classes. There are 6 instances containing missing values (feature 12 has 5 missing and feature 13 has 2 missing) hence only 297 instances were considered. There is not multivariate normality, but there are some predictors where normality holds. The distribution of the predictors is quite similar in both classes and they have outliers. The predictors in general are low correlated.

Ionosphere (Ionosphere): 351 instances, 34 continuous features and two classes: "Good" and "Bad " radar returns. The second feature was eliminated because assume the same value for all instances. The first feature assumes the same value in one of the classes and was also eliminated. There is not multivariate normality in any of the classes. The predictors have plenty of outliers in the second class. The predictors in general are low correlated.

Wisconsin breast cancer (breastw): There are 699 instances, 9 continuous features and two classes. There are 16 instances containing missing values (all of them in the feature 6) therefore only 683 instances were considered. There is not multivariate normality in any of the classes. There is not high correlation among the predictors in each of the classes. The distribution of the predictors is quite different between the classes.

Pima Indian diabetes (diabetes) It includes 768 instances, 8 continuous features and two classes. There is not multivariate normality in any of the classes. The correlation coefficient among all the pair of features is low. The predictors have plenty of outliers and in general are low correlated. The distribution of the predictors is quite similar in both classes.

Vehicle Silhouettes (vehicle): 846 instances, 18 continuous features and 4 classes of vehicles. There is not multivariate normality in any of the four classes. Some predictors are highly correlated in all the classes. The distribution of the predictors is quite similar in the four classes and there are outliers.

Credit German (german): It contains 1000 instances, 20 features: continuous (2,5,8,11,13,16,18), binary (19 and 20), nominal (1,3,4,6,9,10,12,14,15,17) and ordinal (7) and 2 classes of customers. There is not multivariate normality in both classes. The predictors in general are low correlated. The distribution of the predictors looks somehow similar in both classes and there are outliers.

Segmentation (segment): It includes 2310 instances, 19 continuous features and 7 classes. The third feature was eliminated because assumes the same value for all instances. Also features 4 and 5 present only two or three different values in all the classes and they were omitted. There is not multivariate normality in any of the 7 classes. Some predictors are highly correlated in all the classes. The distribution of the predictors looks different in the 7 classes and there are outliers.

For each dataset we have performed the following procedure: The dataset is randomly divided in 10 parts, the first one is taken as the test sample and the remaining is considered as the learning sample. Next, 50 bootstrapped samples are taking from the learning sample and a KDE classifier is constructed with each of them. Finally, each instance of the test sample is assigned to a class by voting using the 50 classifiers previously constructed. The proportion of instances incorrectly assigned will be the bagged misclassification error. We repeat the steps considering now the second part as the test set and in this way we continue until the tenth part is considered as the test set. The procedure is repeated 10 times. The misclassification error of a single classifier is estimated by a 10-fold crossvalidation and averaged over 50 runs. The bagged misclassification error is averaged on 100 repetitions. We also computed the ratio of the misclassification errors of the bagged classifier versus the single one. The misclassification errors and the ratios are shown in the table 4. We have written a S-Plus program to carry out all our tasks. The program was tested in a DELL Workstation with a dual Processor Pentium Xeon running at 933 Mhz.

Table 4. Comparison of misclassification error rates for single and bagged KDE classifiers

Dataset	Standard Kernel			Adaptive Kernel		
	Single	Bagged	Ratio	Single	Bagged	Ratio
Iris	3.59	3.66	1.01950	4.47	4.26	0.95302
Sonar	17.40	17.59	1.0109	16.37	15.70	0.95907
Glass	45.39	45.18	0.99537	35.46	35.51	0.99859
Heart	22.44	21.97	0.97906	21.43	20.97	0.97853
Heart*	23.30	21.88	0.93906	22.27	20.80	0.93399
Heart**	22.30	20.80	0.93274	21.70	20.10	0.92626
Ionosfera	10.93	10.48	0.95883	10.33	10.08	0.97579
Breastw	3.66	3.81	1.04098	3.94	3.80	0.96446
Diabetes	26.37	26.27	0.99621	26.29	25.95	0.98706
Vehicle	35.33	34.85	0.98641	36.88	34.16	0.92624
German	34.61	33.81	0.97688	35.06	33.74	0.96235
German*	30.0	27.28	0.90933	30.0	25.9	0.86330
German**	28.50	27.90	0.97438	28.24	25.58	0.90580
Segment	15.86	15.32	0.96595	13.34	13.31	0.99770

* Here only continuous and binary variables are considered
**Here all the predictors are considered as continuous.

The average of the error reduction for the 10 datasets after Bagging using the standard Kernel was 2.24% whereas for the adaptive Kernel was 4.77%. When C4.5 classifier was bagged (Quinlan. 1997) the average error reduction for the same datasets was 8.83%. Notice that we get improvement with the adaptive kernel although is very slight in some datasets.

3.1 Bagging after Feature selection

To deal with the curse of dimensionality problem we perform feature selection. A forward selection procedure was used and repeated 10 times. First we select the single feature that produces the highest classification rate using the classifier based on kernel density estimator. Once that this is done we search for the second feature that together with the first one yields the highest classification rate. The procedure continues until the classification rate decreases. The table 5 shows the original number of features and the number of features selected in each dataset for both the standard kernel and the adaptive kernel classifier. The table also includes which variables were selected.

Table 5: Features selected for both types of kernel density estimators

Dataset	Number of Features (0/SK/AK)	Standard Kernel	Adaptive Kernel
Iris	4/3/2	2,4,3	3,4
Sonar	60/10/10	10,11,12,15,16, 17,21,22,37,46	11,12,16,18,20, 21,22,36,46,49
Glass	9/3/3	2,4,7	4,5,7
Heartc	13/5/6	2,6,8,12,13	2,3,8,10,12,13
Ionosphere	34/7/6	1,2,3,4,12,13,14	1,3,4,5,6,30
Breastw	9/4/4	1,2,4,6	1,2,3,6,7
Diabetes	8/4/3	2,6,7,8	2,6,8
Vehicle	18/8/7	1,3,4,5,6,7,10,18	1,4,5,6,9,10,18
German	20/5/4	1,2,3,5,10	1,2,3,4,6
Segment	19/7/7	1,2,11,13,14,16,19	1,2,8,14,16,18,19

Once that the predictors are selected we create two subsets of each of the datasets and then we perform bagging using the corresponding kernel classifier. In table 6, we show the misclassification rates of the single and bagged kernel classifiers after feature selection

4. Conclusions

Our experiments have lead us to the following conclusions

a) Increasing the number of bootstrapped samples seems to improves the misclassification error for both types of kernels. We have tried 10 and 50 and on average we gained 2%. But the computing time increases more than three times.
b) Without feature selection the adaptive kernel performs better than the standard kernel, but it requires between 3 and 6 times more computing time.
c) After feature selection the performance of bagging deteriorates for both time of kernels.
d) Features selection does a good job, because after that kernel density classifiers gives lower misclassification errors than CART classifiers.
e) Bagging classifiers gets better results when it is applied to datasets where the single classifier performs poorly. In most of the datasets of this paper kernel density classifiers have performed quite well.

Table 6. Comparison of bagging and boosting KDE classifiers after feature selection

Dataset	Standard Kernel			Adaptive Kernel		
	Single	Bagged	Ratio	Single	Bagged	Ratio
Iris	3.58	3.60	1.00559	3.59	3.66	1.01950
Sonar	15.58	15.76	1.01155	14.98	15.23	1.01669
Glass	35.66	36.73	1.03001	32.47	32.05	0.98706
Diabetes	24.75	22.81	0.92162	23.93	23.58	0.98537
Heart*	17.39	18.00	1.0358	18.76	19.10	1.01812
Heart**	20.34	20.26	0.99607	21.76	17.74	0.81526
Breastw	3.45	3.54	1.02609	3.24	3.42	1.05556
Ionosphere	7.09	7.12	1.00423	9.15	8.68	0.94863
Vehicle	29.53	29.20	0.98882	32.99	32.01	0.97029
German**	25.38	24.32	0.95823	25.17	23.74	0.94318
Segment	3.29	3.36	1.02128	4.36	4.48	1.02752

Acknowledgment

This work was supported by the Office of Naval Research grant N00014-00-1-0360 and by the PRECISE group of the Engineering School at the UPR-Mayaguez funded by NSF grant EIA 99-77071. Dr. Acuña wants to thank his research assistant Frida Coaquira for reviewing the computer programs and checking carefully the results.

REFERENCES

Bauer, E. and Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, Bosting and variants. *Machine Learning,* 36, 105-139.

Blake, C and Merz, C. (1998). UCI repository of machine learning databases. Department of Computer Science and Information. University of California, Irvine, USA.

Breiman, L. (1996). Bagging Predictors. *Machine Learning,* 26,123-140.

Freund, Y and Schapire, R. (1996). Experiments with a new boosting algorithm. *In Machine Learning, Proceedings of the Thirteenth International Conference,* 148-156., San Francisco, Morgan Kaufman..

Maclin, R. and Optiz, D. (1997). An empirical evaluation of Bagging and Bosting. Proceedings of the Fourteenth National Conference on Artificial Intelligence, AAAI/MIT Press.

Michie, D., Spigelhalter, D.J. and Taylor, C.C. (1994). *Machine Learning, Neural and Statistical Classification.* London: Ellis Horwood.

Quinlan, J.R. (1996). Bagging, Boosting and C4.5. *Proceedings of the Thirteenth National Conference on Artificial Intelligence,* 725-730. AAAI/MIT Press.

Silverman, B.W. (1986). Density Estimation for Statistics and Data Analysis. Chapman and Hall. London.

Titterington, D.M. (1980). A comparative study of kernel-based density estimates for categorical data. *Technometrics*, 22, 259-268.