Naïve Bayes Ensembles with a Random Oracle

Juan J. Rodríguez and Ludmila I. Kuncheva

[1] Departamento de Ingeniería Civil, Universidad de Burgos, Burgos, Spain
jjrodriguez@ubu.es
[2] School of Electronics and Computer Science, University of Wales, Bangor, UK
l.i.kuncheva@bangor.ac.uk

Abstract. Ensemble methods with Random Oracles have been proposed recently (Kuncheva and Rodríguez, 2007). A random-oracle classifier consists of a pair of classifiers and a fixed, randomly created oracle that selects between them. Ensembles of random-oracle decision trees were shown to fare better than standard ensembles. In that study, the oracle for a given tree was a random hyperplane at the root of the tree. The present work considers two random oracles types (linear and spherical) in ensembles of Naive Bayes Classifiers (NB). Our experiments show that ensembles based solely upon the spherical oracle (and no other ensemble heuristic) outrank Bagging, Wagging, Random Subspaces, AdaBoost.M1, MultiBoost and Decorate. Moreover, *all* these ensemble methods are better with any of the two random oracles than their standard versions without the oracles.

1 Introduction

Given its name and simplicity, the performance of the Naïve Bayes Classifier is often described as surprising [8,10,13]. A simple and accurate method is ideally suited as a base classifier for classifier ensembles. Nevertheless, NB is very stable and does not work well with some ensemble methods, such as Bagging [1]. The random oracle makes it possible to destabilize NB, introducing diversity in the classifiers of an ensemble.

Methods for constructing ensembles are often designed so as to inject randomness in the learning algorithm [6]. For instance, a Random Forest [4] is Bagging using random trees as base classifiers instead of standard decision trees. A random oracle makes it possible to introduce randomness for any base classifier model. Thus it can be considered that the presented approach consists of using an ensemble method with a different base classifier.

The paper is organised as follows. Section 2 details the random oracle approach to ensemble construction and the two random oracles considered. The experimental validation and results are given in Section 3. Finally, Section 4 concludes the study.

M. Haindl, J. Kittler, and F. Roli (Eds.): MCS 2007, LNCS 4472, pp. 450–458, 2007.

2 Ensembles with a Random Oracle

A random oracle classifier is a mini-ensemble formed by a pair of classifiers and a random oracle that chooses between them. It can be thought of as a random discriminant function which splits the data into two subsets with no regard of any class labels or cluster structure. A random oracle classifier can be used as the base classifier of any ensemble method. Given a classification method, the training of a random oracle classifier consists of:

- Select the random oracle (sample its parameters from a uniform distribution).
- Split the training data in two subsets using the random oracle.
- For each subset of the training data, train a classifier.

The random oracle classifier is formed by the pair of classifiers and the oracle itself. The classification of a test instance is done in the following way:

- Use the random oracle to select one of the two classifiers.
- Return the classification given by the selected classifier.

If the computational complexity of the oracle is low, both in training and classification, the computational complexity of a random oracle classifier is very similar to the complexity of the base classifier. In the classification phase, only one of the two classifiers is used. In the training phase, two classifiers are built. Nevertheless, they are trained with a disjoint partition of the training examples and the training time of any classification method depends, at least linearly, on the number of training examples.

In this work, two random oracles are considered: the linear and the spherical oracles.

2.1 The Linear Oracle

This oracle divides the space into two subspaces using a hyperplane. To build the oracle, two different training objects are selected at random (these can be from the same class). The oracle is the hyperplane delineating the Voronoi regions of the two objects, i.e., the hyperplane passing through the middle of the segment joining the objects and orthogonal to that segment. Using objects from the data set for constructing the oracle, we ensured that there will be training instances in both subspaces.

Since the data sets used in the experiment contain both numeric and nominal attributes, we used distances to the two selected objects rather than the computationally cheaper calculation of the hyperplane. We consider Euclidean space; all numerical attributes are scaled within [0,1]. The distance between two values of a nominal attribute is 0 if the values are equal and 1 otherwise.

2.2 The Spherical Oracle

The space is divided into two regions: inside and outside a hypersphere in a random subspace. The procedure for selecting the sphere is:

- Draw a random feature subset containing at least 50% of the features.
- Select a random training instance as the center of the sphere.
- Find the radius of the sphere as the median of the distances from the center to K randomly selected training instances. (For no specific reason, here we use $K = 7$.)

The objective of this procedure is to have training instances inside and outside of the sphere. The selection of a feature subset seeks to increase the diversity of the oracles (and therefore, of the random oracle classifiers). The effect of using such subset is that the distance between two objects can be different for different oracles. If the distances are always the same for a pair of objects, two close objects would be in the same subspace for the majority of random oracles.

2.3 Why Does Random Oracle Work?

Figure 1 shows an artificial data set and the classification regions for NB, NB with linear and spherical random oracles and two NB ensembles with random oracle. Clearly, NB on its own is not adequate for this kind of data. Classical ensemble methods of NB classifiers do not help on this data. The training error of the NB classifier on this data is 57.2%. AdaBoost needs weak classifiers with errors smaller than 50%. The base classifiers from Bagging are trained from samples of the data, they will be similar to the classifier obtained from all the data.

A random oracle classifier with two NB classifiers is better for this data, but the accuracy depends substantially on the randomly selected oracle. An ensemble or 25 Random Oracle classifiers approximates rather well the optimal classification boundary.

This example illustrates two possible reasons for the success of random oracles. First, the oracle splits the training data into two subsets and the classification task can be easier in the subsets than in the original data. This may lead to a better classifier (mini-ensemble) than the original NB.

The second reason for the success of random oracle is that the base classifiers can be much more diverse than the classifiers obtained with other ensemble methods. Classical ensemble methods are not able to introduce diversity in NB classifiers. The example shows that it is possible to obtain accurate ensembles from random oracle classifiers.

3 Experiments

3.1 Settings

The data sets used in the experiments, from the UCI Repository [7], are shown in table 1. The experiments were carried out using Weka [16] and our own code.

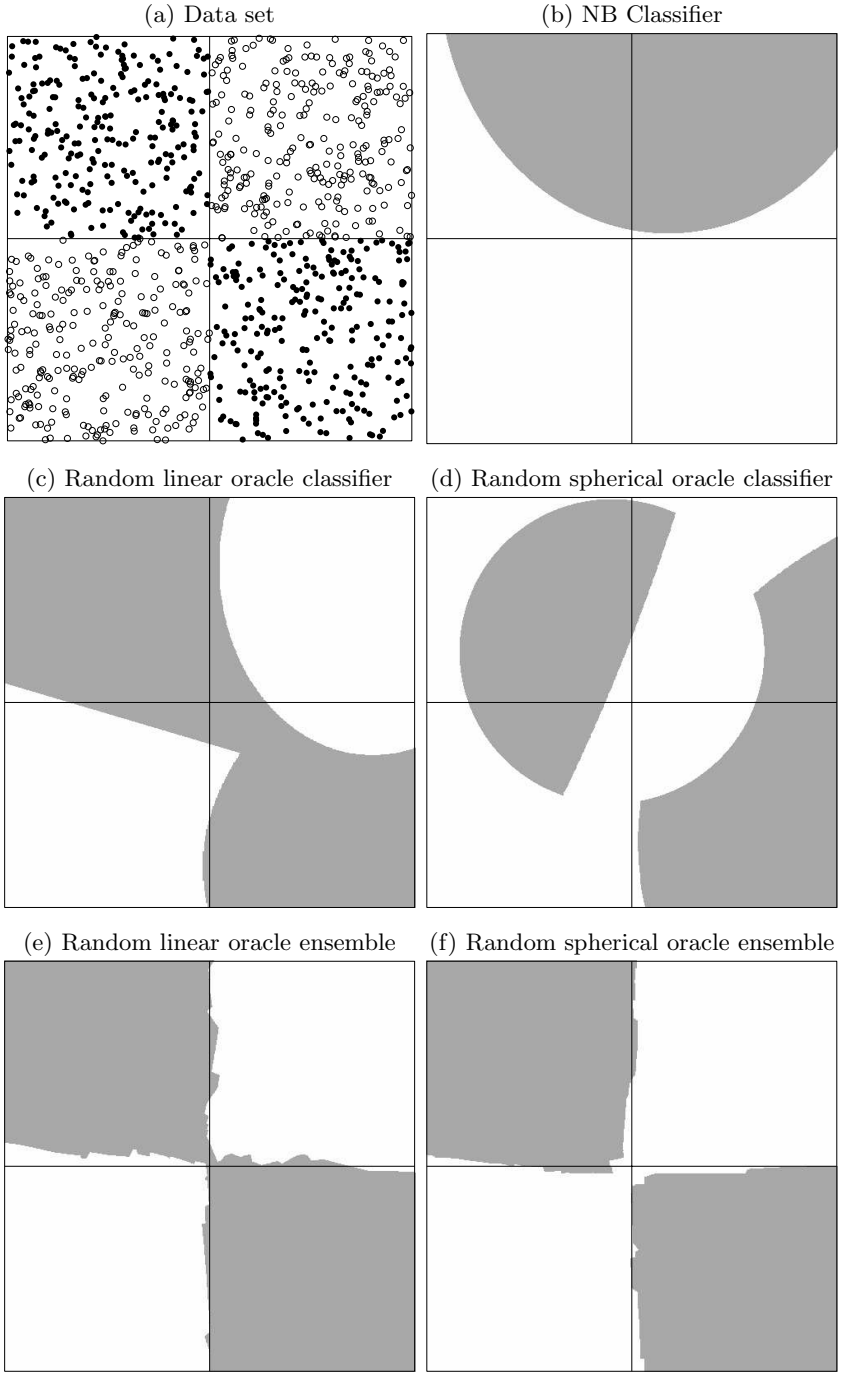

Fig. 1. Data set and classification regions for NB, NB with random oracle and NB ensembles with random oracle. Each ensemble consists of 25 classifiers.

Table 1. Summary of the 35 UCI Datasets used in the experiment

Data set	Classes	Objects	D	C	Data set	Classes	Objects	D	C
anneal	6	898	32	6	letter	26	20000	0	16
audiology	24	226	69	0	lymphography	4	148	15	3
autos	7	205	10	16	mushroom	2	8124	22	0
balance-scale	3	625	0	4	pima-diabetes	2	768	0	8
breast-cancer	2	286	10	0	primary-tumor	22	339	17	0
cleveland-14-heart	2	303	7	6	segment	7	2310	0	19
credit-rating	2	690	9	6	sick	2	3772	22	7
german-credit	2	1000	13	7	sonar	2	208	0	60
glass	7	214	0	9	soybean	19	683	35	0
heart-statlog	2	270	0	13	splice	3	3190	60	0
hepatitis	2	155	13	6	vehicle	4	846	0	18
horse-colic	2	368	16	7	vote	2	435	16	0
hungarian-14-heart	2	294	7	6	vowel-context	11	990	2	10
hypothyroid	4	3772	22	7	vowel-nocontext	11	990	0	10
ionosphere	2	351	0	34	waveform	3	5000	0	40
iris	3	150	0	4	wisconsin-bc	2	699	0	9
kr-vs-kp	2	3196	36	0	zoo	7	101	16	2
labor	2	57	8	8					

Note: 'D' stands for the number of discrete features and 'C' for the number of continuous-valued features.

There are several methods for handling continuous attributes in NB classifiers [2]; in this work the "Normal" method was used. The class-conditional pdf for attribute x_i, $p(x_i|\omega_j)$ is approximated as a normal distribution, and the discriminant function for class ω_j is $g_j(\mathbf{x}) = P(\omega_j) \prod_i p(x_i|\omega_j)$. Each ensemble was formed by 25 classifiers. The results were obtained using a 10-fold stratified cross validation, repeated 10 times.

3.2 Ensemble Methods

As the random oracle approach produces, in effect, a base classifier, it can be used with any ensemble heuristic or on its own. The ensemble methods considered in this work are:

- Bagging [3]. Each base classifier is trained on a bootstrap sample of the training data.
- Wagging [15,1]. For each base classifier, the training examples are weighted randomly using the Poisson distribution.
- Random Subspaces [11]. Each base classifier is trained with all the training examples, but using only a random subset of the features. Two values are considered for the number of randomly selected features here: 50% and 75%.
- AdaBoost.M1 [9]. This is the most well-known variant of Boosting. The training samples are also weighted. It is an incremental method; the weight on an object depends on the correctness of the classifications given by the

previous base classifiers. Both the re-sampling and the re-weighting version are considered here, denoted (S) and (W), respectively.
- MultiBoost [15]. This is a combination of Boosting and Wagging. It follows the AdaBoost method, but after a number of iterations the training examples are reweighted using the Wagging approach. The size of the sub-committees for this method was set to 5. It has the same two variants as AdaBoost.M1, both of them used in the experiment.
- Decorate [14]. This is an incremental method based on the boosting method. Each base classifier is trained using all training examples plus artificially generated examples. The method seeks diversity among the base classifiers by constructing the artificial examples in a specific way.

The total number of different methods together with their variants is 9 (6 methods, 3 of them with 2 options). Each method will be used with three base classifiers: NB, random linear oracle with NB and random spherical oracle with NB. Hence, the number of different configurations is 27.

Included in the experiments were also the following three methods: a single NB (denoted further as 'Single') and ensembles obtained using *only* the random oracle heuristic, linear (denoted L-Ensemble) or spherical (denoted S-Ensemble).

3.3 Results

Table 2 shows a summary of the experimental results. The methods are sorted according to their average rank, following the method described in [5]. For each data set, all the methods are sorted. The best method has rank 1, the second best has rank 2, and so on. If there are ties, the methods are assigned average ranks. The overall value of a method is measured by its average rank across all data sets.

The best 7 methods use a Random Oracle. The top ranks are for MultiBoost with a Random Oracle and re-weighting or re-sampling. The best method without an oracle is the re-sampling version of MultiBoost.

The last column of the table, the benefit, represents the difference between the average ranks of a method with an oracle and the corresponding method without the oracle. The length of the bars is proportional to that difference. For *all the methods*, the benefit is positive.

The table also shows that the random oracle can be used as the only heuristic for ensemble construction. The spherical oracle ensemble has a better rank than all the methods that do not use a random oracle. The linear oracle ensemble is not as good, but the only method, without a random oracle, with better rank than L-Ensemble is MultiBoost.

Table 2 also includes, for the methods with a random oracle, the number of data sets where that method is better, equal and worse than the corresponding version without the oracle. For all the methods, the versions with an oracle are better than the version without the oracle for at least 21 of 35 data sets.

When comparing two methods over 35 data sets, the differences are statistically significant, according to a sign test [5], for a level $\alpha = 0.05$, if the number

Table 2. Ensemble methods with and without Random Oracle sorted by their average ranks. The ensemble size, L, is 25.

Method	Total Rank	Win-tie -loss	Benefit	Method	Total Rank	Win-tie -loss	Benefit
S-MultiBoost (W)	8.41	•26-0-9		*S*-AdaBoostM1 (W)	15.00	•24-2-9	
S-MultiBoost (S)	8.43	•27-0-8		*L*-Rand. Subs. (75%)	15.03	•25-0-10	
L-MultiBoost (S)	8.64	•26-0-9		*L*-AdaBoostM1 (W)	15.56	•25-1-9	
L-MultiBoost (W)	9.33	•25-0-10		*L*-Rand. Subs. (50%)	15.73	•29-0-6	
S-Bagging	11.23	•31-0-4		*S*-Rand. Subs. (50%)	15.93	•27-1-7	
S-Wagging	12.44	•27-0-8		*L*-Decorate	18.03	23-0-12	
S-Ensemble	12.77			AdaBoostM1 (S)	18.63		
MultiBoost (S)	13.11			AdaBoostM1 (W)	19.03		
S-Rand. Subs. (75%)	13.63	•26-0-9		Bagging	20.19		
L-Bagging	13.93	•26-0-9		Rand. Subs. (75%)	20.44		
S-AdaBoostM1 (S)	14.09	23-0-12		*S*-Decorate	20.74	21-0-14	
MultiBoost (W)	14.33			Wagging	20.83		
L-Wagging	14.46	•24-0-11		Single	21.13		
L-Ensemble	14.63			Decorate	21.81		
L-AdaBoostM1 (S)	14.71	•24-1-10		Rand. Subs. (50%)	22.79		

Note 1: '*L*-' indicates that the linear oracle is present, '*S*-' that the spherical oracle is present.
Note 2: '•' indicates that the difference between the method with oracle and without oracle is statistically significant at $\alpha = 0.05$ (using sign test).

Table 3. Ensemble methods with and without Random Oracle sorted by their average ranks. The ensemble size, L, is 25 for the methods with oracle and 50 for the methods without oracle.

Method	Total Rank	Win-tie -loss	Benefit	Method	Total Rank	Win-tie -loss	Benefit
S-MultiBoost (W)	8.36	•28-1-6		MultiBoost (S)	15.37		
S-MultiBoost (S)	8.43	•27-0-8		*L*-AdaBoostM1 (W)	15.37	•26-1-8	
L-MultiBoost (S)	8.57	•26-0-9		*L*-Rand. Subs. (50%)	15.60	•25-0-10	
L-MultiBoost (W)	9.16	•28-0-7		*S*-Rand. Subs. (50%)	15.81	•28-0-7	
S-Bagging	11.11	•30-0-5		MultiBoost (W)	16.87		
S-Wagging	12.26	•27-1-7		*L*-Decorate	18.04	23-0-12	
S-Ensemble	12.54			AdaBoostM1 (S)	18.56		
S-Rand. Subs. (75%)	13.51	•28-0-7		AdaBoostM1 (W)	19.19		
L-Bagging	13.79	•25-0-10		Bagging	19.46		
S-AdaBoostM1 (S)	14.09	23-0-12		Wagging	19.97		
L-Wagging	14.23	•24-0-11		*S*-Decorate	20.66	21-0-14	
L-AdaBoostM1 (S)	14.61	23-1-11		Single	21.00		
L-Ensemble	14.67			Rand. Subs. (75%)	21.00		
L-Rand. Subs. (75%)	14.73	•28-0-7		Rand. Subs. (50%)	21.27		
S-AdaBoostM1 (W)	14.87	•24-2-9		Decorate	21.90		

Note: '*L*-' indicates that the linear oracle is present, '*S*-' that the spherical oracle is present.
Note 2: '•' indicates that the difference between the method with oracle and without oracle is statistically significant at $\alpha = 0.05$ (using sign test).

of wins (plus half the number of ties) is greater or equal than 24. Those cases are marked with a bullet in the table. From 18 tests, only in 3 cases the difference is not significant.

In the previous comparison, the number of base classifiers for all the ensembles was 25. It could be argued that the setting is favourable to the random oracle variants because these ensembles are formed by 50 NB classifiers (25 Random Oracles with 2 NB classifiers) while the variants without the oracle are formed by 25 NB classifiers. That setting was selected because in the variants with and

without the Random Oracle, each training instance was used to construct, at most, 25 NB classifiers and each testing instance was classified by, at most, 25 NB classifiers.

The experiments were repeated with all the ensembles without the random oracle using 50 NB classifiers. Table 3 shows the results. Interestingly, the results are even more favourable to the versions with the Random Oracle. This unexpected finding indicates that some classical methods performed worse with $L = 50$ classifiers than with $L = 25$ classifiers. One possible explanation is overtraining of the ensemble. As the results are based on ranking, the behaviour of one or two ensembles would affect the overall score for all methods. The suspect here is MultiBoost. In both tables, this is the best method without random oracle. In Table 2 this method was the 7th best method, with an average rank of 13.11. In Table 3 the method is at 16th place with an average rank of 15.37, showing that MultiBoost with $L = 50$ has been outperformed by more ensemble methods than MultiBoost with $L = 25$.

4 Conclusion

Here we study ensembles of NB classifiers with random oracle. Previously a random linear oracle was used to improve ensembles of decision trees [12]. Our results indicate that random oracles are even more suitable for NB classifiers than for decision trees.

Most ensemble methods rely on unstable base classifiers. It is known that NB are more stable than decision trees. The random oracle introduces the desired instability of NB, which makes random-oracle NB a good base classifier for constructing ensembles.

Nine ensemble models were considered (6 methods, 3 of them with 2 variants). For each of them, there were 3 variants: without random oracle, with the linear oracle and with the spherical oracle. 35 UCI data sets were used in this study. The spherical oracle ensemble method (based only on the random oracle heuristic) showed better results than any of the 9 ensemble models without oracle. Moreover the random oracle improved the performance of *all* nine ensemble models. Best method appeared to be MultiBoost with a spherical oracle. For NB base classifiers, the spherical oracle is generally better than the linear oracle.

There is further room for improvement; the 'best' random oracle to use can depend on the base classifier, the ensemble method and the data set. Also, the diversity of the classifiers in an ensemble could be improved using different random oracles in the same ensemble.

Acknowledgements. This work has been partially supported by the Spanish MCyT project DPI2005–08498, and the "Junta de Castilla y León" project VA088A05.

References

1. E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1–2):105–139, 1999.
2. R.R. Bouckaert. Naive Bayes classifiers that perform well with continuous variables. In *17th Australina Conference on AI (AI 04)*, Lecture Notes in AI. Springer, 2004.
3. L. Breiman. Bagging predictors. Technical Report 421, Department of Statistics, University of California, Berkeley, 1994.
4. L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
5. J. Demšar. Statistical comparison of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
6. T.G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems 2000*, pages 1–15, 2000.
7. C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.
8. P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.
9. Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
10. D.J. Hand and K. Yu. Idiot's bayes — not so stupid after all? *International Statistical Review*, 69:385–399, 2001.
11. T. K. Ho. The random space method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.
12. L. I. Kuncheva and J. J. Rodríguez. Classifier ensembles with a random linear oracle. *IEEE Transactions on Knowledge and Data Engineering*, 19(4):500–508, 2007.
13. L.I. Kuncheva. On the optimality of naïve bayes with dependent binary features. *Pattern Recognition Letters*, 27:830–837, 2006.
14. P. Melville and R. J. Mooney. Creating diversity in ensembles using artificial data. *Information Fusion*, 6(1):99–111, 2005.
15. G. I. Webb. Multiboosting: A technique for combining boosting and wagging. *Machine Learning*, 40(2):159–196, 2000.
16. I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2nd edition, 2005. `http://www.cs.waikato.ac.nz/ml/weka`.