

Bias and Variance of Rotation-based Ensembles^{*}

Juan José Rodríguez¹, Carlos J. Alonso², and Oscar J. Prieto²

¹ Lenguajes y Sistemas Informáticos
Universidad de Burgos, Spain
jjrodriguez@ubu.es

² Grupo de Sistemas Inteligentes
Departamento de Informática
Universidad de Valladolid, Spain
calonso@infor.uva.es

Abstract. In Machine Learning, ensembles are combination of classifiers. Their objective is to improve the accuracy. In previous works, we have presented a method for the generation of ensembles, named rotation-based. It transforms the training data set; it groups, randomly, the attributes in different subgroups, and applies, for each group, an axis rotation. If the used method for the induction of the classifiers is not invariant to rotations in the data set, the generated classifiers can be very different. In this way, different classifiers can be obtained (and combined) using the same induction method.

The bias-variance decomposition of the error is used to get some insight into the behaviour of a classifier. It has been used to explain the success of ensemble learning techniques. In this work the bias and variance for the presented and other ensemble methods are calculated and used for comparison purposes.

1 Introduction

One of the research areas in Machine Learning is the generation of ensembles. The basic idea is to use more than one classifier, in the hope that the accuracy will be better. It is possible to combine heterogeneous classifiers, where each of the classifiers is obtained with a different method. Nevertheless, it is also possible to combine homogeneous classifiers. In this case all the classifiers have been obtained with the same method. In order to avoid identical classifiers, it is necessary to change something, at least the random seed.

There are methods that alter the data set. *Bagging* [3] obtains a new data set by resampling the original data set. An instance can be selected several times, so some instances will not be present in the new data set. The *Random Subspaces* [8] method obtains a new data set deleting some attributes. *Boosting* [14] is a family of methods. The most prominent member is AdaBoost. In this case the data set is modified depending on the classification errors of the previously generated base classifier. The bad classified examples are assigned a greater weight, so the

^{*} This work has been supported by the Spanish MCyT project DPI2001-01809.

next classifier will give more importance to those examples. Another possibility, instead of modifying the data set, is to modify the base learning method. For instance, in the *Random Forest* method [4], decision trees are constructed in a way such as the selected decision for a node is, to some extent, random. Comparatives among ensemble generation methods are presented in [1, 5, 9].

These methods share the idea that it is necessary to modify the data set in a way that some information is lost (e.g., instances, attributes), or the learning method does not use all the information (i.e., random forest). None of the modifications would be considered if it was desired to obtain a unique classifier. They are used because ensemble methods need diverse base classifiers.

Rotation-based ensembles [13] transform the data set, but in a way that all the present information is preserved (although it is transformed). The idea is to group the attributes, and for each group to apply an axis rotation. Hence, all the available information (instance and attributes) in the data set is still available. Although there are a lot of learning methods that are invariant to rotations, one of the most used with ensembles, decision trees, are very sensitive to this kind of variations, because the generated decision surfaces are formed by parallels to the axis.

When the method was presented [13] it was compared with other ensemble methods using only the error. It is well known that the error of a classifier can be divided in bias and variance. In this work we calculate these measures for different ensemble methods and use them for comparison purposes.

The rest of the work is organized as follows. The proposed method is described in Sect. 2. The bias-variance decomposition of the error is presented in Sect. 3. Section 4 includes the experimental validation. Finally, section 5 concludes.

2 Rotation-based Ensembles

This method is introduced in [13]. It is based on transforming the data set, in a different way for each member of the ensemble. Then, the base learning method is used with the transformed data set. The results of the different classifiers are combined using majority vote.

The transformation process is based on Principal Component Analysis (PCA) [7]. These components are linear combinations of attributes. Normally, this technique is used for reducing the dimensionality of the data set, because the components are ordered according to their importance, so the last ones can be discarded. Nevertheless, we are going to consider all the components. In this case, the transformed data set has exactly the same information than the original one, with the only difference of axis rotation. This technique works with numeric attributes. If the data set has discrete attributes, they must be transformed to numeric.

One of the objectives in ensemble methods is that the combined classifiers were diverse, because nothing is gained if they are equal. Another objective, somewhat contradictory to the previous one, is that the combined classifiers were accurate. PCA is an adequate transformation for ensemble generation because,

first, no information is lost, so the accuracy of the base classifiers should not be worsened, as happens with other ensemble methods. And second, because the model generated by the base classifier can be rather different, given that the method was not invariant to rotations.

Nevertheless, for a given data set, the result of PCA is unique. For generating different classifiers it is necessary to obtain several transformed data sets. The possibilities are:

- To apply PCA to a subset of the instances. Unless the number of instances were rather small, the results will be rather similar.
- To apply PCA to a subset of the classes. In fact, this is a variant of the previous possibility, because when classes are selected, their instances are being selected. Nevertheless, the hope is that the results of the analysis will be more diverse.
- To apply PCA to a subset of the attributes. In this case only the selected attributes would be rotated. In order to modify all the attributes, it is possible to group the attributes in groups, and to apply PCA for each group.

The previous strategies are considered for the application of PCA, but once that the analysis has been obtained, then all the data set is reformulated using the components. If the previous strategies are combined, it is possible, except for very small data sets, to obtain a lot of different data sets.

The application of PCA to groups of attributes is another mechanism for diversification, but it also provides additional advantages. First, the execution time of the analysis depends mainly on the number of attributes, so it is much quicker to do the analysis in groups than doing it with all the attributes.

An algorithmic description of the method is presented in [13].

3 Bias-variance decomposition of the error.

The bias-variance decomposition of the error is a useful tool for analyzing learning algorithms. Originally it was proposed for regression, but there are several variants for classification [10, 6]. It decomposes the error in three terms [16], derived with reference to the performance of a learner when trained with different training sets drawn from some distribution of training sets:

- Squared bias: a measure of the error of the central tendency of the learner.
- Variance: a measure of the degree to which the learner's predictions as it is applied to learn models from different training sets.
- Intrinsic noise: a measure of the degree to which the target quantity is inherently unpredictable.

Given that it is infeasible to estimate the intrinsic noise from sample data, this term is usually aggregated to the bias term.

One of the possible explanations for the success of bagging and boosting uses this decomposition [11]. Although there is no general theory on the effects

of bagging and boosting on bias and variance, Bagging is assumed to reduce variance without changing the bias. Boosting in the early iterations primarily reduces bias and in the later iterations reduces mainly variance.

4 Experimental Validation

4.1 Data Sets

The used data sets appear in table 1. They are the ones available in the format of the WEKA library. All of them are from the UCI repository [2]. Some of the data sets were slightly modified. First, for the data sets “splice” and “zoo” an attribute was eliminated. They were instance identifiers. This kind of attributes are not useful for learning, and in the current implementation, they cause a considerably overhead, because continuous attributes are converted to numeric. Second, for the data set “vowel”, there was an attribute indicating if the instance was for training or for testing. This attribute was eliminated. Moreover, this data set includes information of the speakers and their sex. We consider two versions, “vowel-c” and “vowel-n” using and not using this context information, because there are works that use and not use this information.

4.2 Settings

The used method for constructing decision trees is one of the available in the WEKA library [17], named J48. It is a reimplementation of C4.5 [12]. The implementations of Bagging and AdaBoost.M1 are also from that library.

The parameters of the different methods were the default ones. The number of classifiers to combine was 10. For rotation-based ensembles, the number of attributes in each group was 3.

PCA is defined for numeric attributes. For the presented method, discrete attributes were converted to numeric ones, with as many attributes as possible values. This transformation was not applied for the methods used for comparison (bagging, boosting...) because they can deal directly with discrete attributes.

The bias and variance are calculated with the method proposed in [16]. It is available in WEKA. The default parameters were used. The exception was the number of times each instance is classified, that was set to 50 (the default value is 10). With these settings, it runs 50×2-fold cross validation.

4.3 Results

Table 2 shows the error, bias and variance results. It also includes, for each method, the mean for all the data sets, although it is a very gross measure of relative performance. Rotation-based ensembles have the minimum mean for the error and the variance. The minimum mean for the bias is obtained with boosting, although the result for rotation-based is very close.

data set	classes	examples	discrete	continuous
anneal	6	898	32	6
audiology	24	226	69	0
autos	7	205	10	16
balance-scale	3	625	0	4
breast-cancer	2	286	10	0
breast-w	2	699	0	9
colic	2	368	16	7
credit-a	2	690	9	6
credit-g	2	1000	13	7
diabetes	2	768	0	8
glass	7	214	0	9
heart-c	5	307	7	6
heart-h	5	294	7	6
heart-statlog	2	270	0	13
hepatitis	2	155	13	6
hypothyroid	4	3772	22	7
ionosphere	2	351	0	34
iris	3	150	0	4
labor	2	57	8	8
letter	26	20000	0	16
lymphography	4	148	15	3
primary-tumor	22	239	17	0
segment	7	2310	0	19
sonar	2	208	0	60
soybean	19	683	35	0
splice	3	3190	60	0
vehicle	4	846	0	18
vote	2	435	16	0
vowel-c	11	990	2	10
vowel-n	11	990	0	10
waveform	3	5000	0	40
zoo	7	101	16	2

Table 1. Characteristics of the used data sets.

The table also includes the geometric mean ratio [15]. For each data set, the ratio is the value for the alternative method divided by the value for rotation-based ensembles. Then, the geometric mean of the ratios is calculated. A geometric mean ratio greater than 1.0 represents an advantage of rotation-based and a lower value represents an advantage to the alternative algorithm. In the table, all the geometric rates are greater than 1.0.

Finally, the table also shows for each method and measure the number of times that the result is better and worst than the result for rotation-based. The only case that is not favourable to the presented method is the bias for boosting, that is better for 18 of 32 data sets. This is consistent with previous results, that indicate that boosting with few iterations reduces mainly the bias [11].

5 Conclusions and Future Work

Rotation-based ensembles is a novel approach for the generation of ensembles of classifiers. The method compares favorably to Bagging, Random Forest and AdaBoost.M1, when using decision trees as base classifiers. The proposed method has smaller mean values, favourable geometric mean ratios, and is more times better than worst when compared with the other methods. The only exception is that the bias is better for boosting than for rotation-based ensembles.

It is somewhat surprising that the results of the method were better than the results for boosting, because it appears to be rather more elaborated, and with a most solid theoretical basis. Nevertheless, one of the features of boosting, not present in neither our method nor Bagging is the ability of obtaining *strong* classifiers from *weak* ones, such as decision stumps. On the other hand, Bagging and our method allow the construction of the base classifiers in parallel.

Currently, only decision trees have been considered as base classifiers. Other methods that are not rotation-invariant can be considered. On the other hand, rotation-invariant methods can also be used if the number of attributes in the transformed data set is different than the number for the original data set.

The experimental validation has been limited to classification problems. Apparently, the method can be also used for regression problems, if the base regression method is not invariant to rotations, as is the case for regression trees.

The presented method is compatible with another ensemble methods. First, the base classifier for an ensemble method can be another ensemble method. For instance, it could be possible to use the presented method using bagging as base classifier. In this way, it could be possible to combine 100 decision trees, but applying the PCA procedure only 10 times. This can be useful because the PCA procedure is slower than resampling, the used strategy for Bagging.

Second, it is possible to apply several transformations to the original data set. For instance, resample and then the presented method. In this way, the two ensemble methods are not used hierarchically, but simultaneously. Hence, it is necessary to study the possible usefulness of some of these combinations.

	J48		Bagging J48		Boosting J48		Random Forest		Rotations J48						
	Error	Bias Variance	Error	Bias Variance	Error	Bias Variance	Error	Bias Variance	Error	Bias Variance					
anneal	0.0202	0.0098	0.0104	0.0195	0.0099	0.0096	0.0123	0.0051	0.0072	0.0141	0.0050	0.0091	0.0159	0.0082	0.0077
audiology	0.2925	0.1573	0.1352	0.2674	0.1462	0.1212	0.2372	0.1032	0.1340	0.3371	0.1219	0.2152	0.2665	0.1306	0.1359
autos	0.3385	0.1423	0.1962	0.2894	0.1265	0.1629	0.2617	0.0932	0.1685	0.2854	0.1030	0.1824	0.2822	0.1321	0.1501
balance-scale	0.2200	0.0826	0.1374	0.1813	0.0710	0.1103	0.2014	0.0862	0.1152	0.1927	0.0890	0.1037	0.1130	0.0513	0.0617
breast-cancer	0.2933	0.2242	0.0691	0.2860	0.2290	0.0570	0.3449	0.2126	0.1323	0.3264	0.2160	0.1104	0.2870	0.2133	0.0737
breast-w	0.0572	0.0299	0.0273	0.0445	0.0310	0.0135	0.0418	0.0263	0.0155	0.0437	0.0286	0.0151	0.0312	0.0258	0.0054
colic	0.1505	0.1367	0.0138	0.1516	0.1353	0.0163	0.1932	0.1226	0.0706	0.1670	0.1280	0.0390	0.1701	0.1261	0.0440
credit-a	0.1527	0.1102	0.0425	0.1434	0.1120	0.0314	0.1645	0.0984	0.0661	0.1550	0.1076	0.0474	0.1415	0.1092	0.0323
credit-g	0.2910	0.1969	0.0941	0.2713	0.1923	0.0790	0.2952	0.1698	0.1254	0.2731	0.1882	0.0849	0.2675	0.1838	0.0837
diabetes	0.2730	0.1792	0.0938	0.2555	0.1852	0.0703	0.2771	0.1734	0.1037	0.2589	0.1890	0.0699	0.2434	0.1949	0.0485
glass	0.3550	0.1641	0.1909	0.3157	0.1776	0.1381	0.3048	0.1553	0.1495	0.2894	0.1473	0.1421	0.2989	0.1711	0.1278
heart-h	0.2390	0.1383	0.1007	0.2135	0.1372	0.0763	0.2139	0.1199	0.0940	0.2090	0.1355	0.0735	0.1906	0.1259	0.0647
heart-c	0.2067	0.1522	0.0545	0.1988	0.1478	0.0510	0.2099	0.1428	0.0671	0.2076	0.1563	0.0513	0.1975	0.1443	0.0532
heart-statlog	0.2369	0.1334	0.1035	0.2131	0.1300	0.0831	0.2181	0.1374	0.0807	0.2104	0.1331	0.0773	0.1927	0.1276	0.0651
hepatitis	0.2074	0.1448	0.0626	0.1888	0.1415	0.0473	0.1914	0.1255	0.0659	0.1809	0.1243	0.0566	0.1764	0.1264	0.0500
hypothyroid	0.0058	0.0042	0.0016	0.0056	0.0043	0.0013	0.0052	0.0025	0.0027	0.0124	0.0046	0.0078	0.0066	0.0030	0.0036
ionosphere	0.1157	0.0583	0.0574	0.0929	0.0564	0.0365	0.0887	0.0500	0.0387	0.0783	0.0529	0.0254	0.0725	0.0450	0.0275
iris	0.0647	0.0473	0.0174	0.0617	0.0527	0.0090	0.0668	0.0493	0.0175	0.0548	0.0447	0.0101	0.0527	0.0363	0.0164
labor	0.2309	0.1274	0.1035	0.2193	0.1179	0.1014	0.1695	0.0667	0.1028	0.1821	0.0814	0.1007	0.1291	0.0484	0.0807
letter	0.1551	0.0325	0.1226	0.1013	0.0373	0.0640	0.0662	0.0182	0.0480	0.0798	0.0220	0.0578	0.0665	0.0176	0.0489
lymphography	0.2553	0.1363	0.1190	0.2249	0.1236	0.1013	0.2012	0.1055	0.0957	0.2091	0.1276	0.0815	0.1945	0.1078	0.0867
primary-tumor	0.6271	0.3023	0.3248	0.6067	0.3827	0.2240	0.6267	0.3016	0.3251	0.6194	0.3099	0.3095	0.5838	0.3168	0.2670
segment	0.0477	0.0163	0.0314	0.0401	0.0202	0.0199	0.0279	0.0111	0.0168	0.0313	0.0142	0.0171	0.0272	0.0138	0.0134
sonar	0.2914	0.1215	0.1699	0.2625	0.1510	0.1115	0.2396	0.1056	0.1340	0.2352	0.1025	0.1327	0.2072	0.1044	0.1028
soybean	0.1286	0.0558	0.0728	0.1042	0.0516	0.0526	0.0870	0.0475	0.0395	0.1164	0.0483	0.0681	0.0718	0.0417	0.0301
splice	0.0740	0.0472	0.0268	0.0666	0.0466	0.0200	0.0603	0.0334	0.0269	0.1147	0.0223	0.0924	0.0506	0.0345	0.0161
vehicle	0.3013	0.1551	0.1462	0.2760	0.1683	0.1077	0.2596	0.1487	0.1109	0.2689	0.1584	0.1105	0.2385	0.1428	0.0957
vote	0.0474	0.0387	0.0087	0.0440	0.0376	0.0064	0.0527	0.0317	0.0210	0.0463	0.0300	0.0163	0.0445	0.0314	0.0131
vowel-c	0.3036	0.0557	0.2479	0.1938	0.0555	0.1383	0.1585	0.0342	0.1243	0.1460	0.0101	0.1359	0.1057	0.0107	0.0950
vowel-n	0.2984	0.0504	0.2480	0.1968	0.0560	0.1408	0.1564	0.0339	0.1225	0.1511	0.0288	0.1223	0.1097	0.0147	0.0950
waveform	0.2534	0.1061	0.1473	0.1897	0.1187	0.0710	0.1900	0.0996	0.0904	0.1861	0.1055	0.0806	0.1625	0.0973	0.0652
zoo	0.1046	0.0450	0.0596	0.1053	0.0410	0.0643	0.0745	0.0295	0.0450	0.0667	0.0204	0.0463	0.0964	0.0343	0.0621
Mean	0.2075	0.1063	0.1012	0.1822	0.1092	0.0730	0.1781	0.0919	0.0862	0.1797	0.0955	0.0842	0.1592	0.0928	0.0663
Geo. mean ratio	1.3597	1.3065	1.4318	1.1904	1.3290	1.0577	1.1168	1.0441	1.3234	1.1708	1.0486	1.3061	1.1625	1.0973	1.0652
Win - Loss	2 - 30	3 - 29	6 - 26	4 - 28	2 - 30	11 - 21	6 - 26	18 - 14	5 - 27	4 - 28	13 - 15	6 - 26	4 - 28	13 - 15	6 - 26

Table 2. Error, bias and variance results

References

1. Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1-2):105–139, 1999.
2. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998. <http://www.ics.uci.edu/~mllearn/MLRepository.html>.
3. Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
4. Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
5. Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems 2000*, pages 1–15, 2000.
6. P. Domingos. A unified bias-variance decomposition and its applications. In *17th International Conference on Machine Learning*, pages 231–238, 2000.
7. Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.
8. Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.
9. Tin Kam Ho. A data complexity analysis of comparative advantages of decision forest constructors. *Pattern Analysis and Applications*, 5:102–112, 2002.
10. R. Kohavi and D. Wolpert. Bias plus variance decomposition for zero-one loss functions. In *13th International Machine Learning Conference (ICML96)*, 1996.
11. Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.
12. J. Ross Quinlan. *C4.5: programs for machine learning*. Machine Learning. Morgan Kaufmann, San Mateo, California, 1993.
13. Juan José Rodríguez and Carlos J. Alonso. Rotation-based ensembles. In *urrent Topics in Artificial Intelligence: 10th Conference of the Spanish Association for Artificial Intelligence*, pages 498–506. Springer, 2004.
14. Robert E. Schapire. The boosting approach to machine learning: An overview. In *MSRI Workshop on Nonlinear Estimation and Classification*, 2002. <http://www.cs.princeton.edu/~schapire/papers/msri.ps.gz>.
15. Geoffrey I. Webb, Janice R. Boughton, and Zhihai Wang. Not so naive bayes. *Machine Learning*, 58:5–24, 2005.
16. Geoffrey I. Webb and Paul Conilione. Estimating bias and variance from data, 2004. <http://www.csse.monash.edu.au/~webb/Files/WebbConilione03.pdf>.
17. Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999. <http://www.cs.waikato.ac.nz/ml/weka>.