International Journal of Innovative Computing, Information and Control
Volume **9**, Number **3**, March **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 1087–1094

ROTATION FOREST WITH LOGITBOOST

Sotiris Kotsiantis

Educational Software Development Laboratory
Department of Mathematics
University of Patras
Patras 25604, Greece
sotos@math.upatras.gr

Received January 2012; revised May 2012

Abstract. *According to experimental results, Logitboost and Rotation Forest may be the most powerful ensemble methods for classification problems. For this reason, in this work we combine rotation forest with logitboost ensemble. We performed a comparison with simple bagging, boosting, logitboost, rotation forest and random subspace methods ensembles, as well as other well known combining methods, on standard benchmark datasets and the presented technique had better accuracy in most cases.*

Keywords: Data mining, Machine learning, Pattern recognition, Ensembles of classifiers

1. **Introduction.** Multiple learner systems (an ensemble of classifiers) try to exploit the local different behaviour of the base classifiers to improve the accuracy and the reliability of the overall inductive learning system [1].

Boosting algorithms suffer from the over-fitting problem when dealing with very noisy data [2]. To cope with this situation, Friedman et al. suggest the use of LogitBoost, which could greatly reduce training errors and hence yield better generalization [3]. The main idea of Rotation Forest [4] is to simultaneously encourage diversity by using PCA to do feature extraction for each base classifier and accuracy is sought by keeping all principal components and also using the whole data set to train each base classifier. According to experimental results, Logitboost and Rotation Forest may be the most powerful ensemble methods for classification problems [5]. For this reason, in this work we combine rotation forest with logitboost ensemble techniques. We performed a comparison with simple bagging, boosting, logitboost and random subspace method ensembles as well as other known ensembles on standard benchmark datasets and the presented technique had better accuracy in most cases. For the experiments, decision stump was used as base learning algorithm [6].

Section 2 presents the most well known algorithms for building ensembles that are based on a single learning algorithm, while Section 3 discusses the presented ensemble method. Experiment results using a number of data sets and comparisons of the presented method with other ensembles are presented in Section 4. We conclude with summary and additional research topics in Section 5.

2. **Ensembles of Classifiers.** This section provides a short survey of methods for constructing ensembles using a single learning algorithm. The Bagging algorithm (Bootstrap aggregating) [7] votes classifiers generated by different bootstrap samples (replicates). Given the parameter T which is the number of repetitions, T bootstrap samples $S_1, S_2, \ldots, S_T$ are generated. From each sample S_i a classifier C_i is induced by the same

learning algorithm and the final classifier C* is formed by aggregating T classifiers. A final classification of object x is built by a uniform voting scheme on $C_1, C_2, \dots, C_T$, i.e., it is assigned to the class predicted most often by these base classifiers, with ties broken arbitrarily. Works in the literature focused on determining the ensemble size sufficient to reach the asymptotic misclassification rate, empirically showing that suitable values are between 10 and 20 depending on the particular data set and base classifier [8,9]. Fumera et al. [10] applied an analytical framework for the analysis of linearly combined classifiers to ensembles generated by bagging.

Dagging [11] produces a number of disjoint, stratified folds out of the data and feeds each chunk of data to a copy of the supplied base learner. Predictions are made via majority vote. Quite well known is Random Subspace Method [12], which consists of training several classifiers from input data sets constructed with a given proportion k of features picked randomly from the original set of features. The author of this method suggested in his experiment to select around half per cent of the original set of features. Firstly, this method obtains bootstrap instances. Then, it employs Information Gain (IG) based feature selection technique to identify and remove irrelevant or redundant features. Finally, base learners trained from the new sub data sets are combined via majority voting.

Random forest [13] is another method for constructing ensembles. They derive their strength from two aspects: using random subsamples of the training data (as in bagging) and randomizing the algorithm for learning base-level classifiers (decision trees). The base-level algorithm randomly selects a subset of the features at each step of tree construction and chooses the best among these. Cai et al. [14] took into account the diversity of classification margins in feature subspaces for improving the performance of bagging. Cai et al. [14] first studied the average error rate of bagging, convert the task into an optimization problem for determining some weights for feature subspaces, and then assigned the weights to the subspaces via a randomized technique in classifier construction.

The training set for each ensemble member of Boosting relies on the performance of the earlier trained classifiers. Thus, Boosting attempts to generate new classifiers that are able to better classify the hard instances for the previous ensemble members. There are several boosting variants; AdaBoost [15,16] is the most well-known. Schapire and Singer [17] identified two scenarios where AdaBoost is likely to fail: (i) when there is insufficient training data relative to the complexity of the base classifiers, and (ii) when the training errors of the base classifiers become too large too quickly. Schapire and Singer [18] proposed Real AdaBoost, a generalized version of AdaBoost, in which weak classifiers are piece-wise functions whose output is a real value representing the confidence-rated prediction. Normally, to construct such weak classifiers, one splits the input space X into non-overlapping blocks (or subspaces) $X_1, X_2, \dots, X_N$ so that the predictions of the weak classifier are the same for all instances falling into the same block. In the case of one-feature-based weak classifiers, this is equivalent to dividing the real line into intervals. Meanwhile, determining the appropriate number of bins for weak classifiers learned by Real AdaBoost is a challenging task because small ones might not accurately approximate the real distribution while large ones might cause over-fitting, increase computation time and waste storage space. Friedman et al. [3] imported an additive logistic regression model into AdaBoost and exactly explained the boosting algorithm from a statistical view. From the perspectives of additive regression model and exponential loss function, Friedman et al. [3] proposed some new boosting algorithms including GentleBoost and LogitBoost. Yin et al. [19] introduced a strategy of boosting based feature combination, where a variant of boosting is proposed for integrating different features. Different from the general boosting, at each round of this variant boosting, some weak classifiers are built on different feature sets, one of which is trained on one feature set. And then these classifiers are combined by weighted voting into a single one as the output classifier of this round.

Garca-Pedrajas and Ortiz-Boyer [20] proposed a boosting approach to random subspace method (RSM) to achieve an improved performance and avoid some of the major drawbacks of RSM. RSM is a successful method for classification. However, the random selection of inputs, its source of success, can also be a major problem. For several problems some of the selected subspaces may lack the discriminant ability to separate the different classes. These subspaces produce poor classifiers that harm the performance of the ensemble. Garca-Pedrajas and Ortiz-Boyer [20] search subspaces that optimize the weighted classification error given by the boosting algorithm, and then the new classifier added to the ensemble is trained using the obtained subspace.

MultiBoosting [21] is another technique of the same category that can be considered as wagging committees formed by AdaBoost. Wagging is a variant of bagging; bagging uses re-sampling to get the datasets for training and producing a weak hypothesis, whereas wagging uses re-weighting for each training instance, pursuing the effect of bagging in a different way. Rotation Forest is a successful ensemble classifier generation technique [4], in which the training set for each base classifier is formed by applying Principal Component Analysis (PCA) to rotate the original attribute axes. Specifically, to create the training data for a base classifier, the attribute set F is randomly split into K subsets and PCA is applied to each subset. All principal components are retained in order to preserve the variability information in the data. Thus, K axis rotations take place to form the new attributes for a base classifier.

Melville and Mooney [22] presented a new meta-learner (DECORATE, Diverse Ensemble Creation by Oppositional Re-labeling of Artificial Training Examples) that uses an existing strong learner (one that provides high accuracy on the training data) to construct a diverse committee. This is accomplished by adding different randomly constructed instances to the training set when building new committee members. These artificially constructed instances are given category labels that disagree with the current classification of the committee, thereby directly increasing diversity when a new learner is trained on the augmented data and added to the committee.

3. Presented Methodology.

According to experimental results, Logitboost and Rotation Forest may be the most powerful ensemble methods for classification problems [5]. For additional improvement of the prediction of a classifier, we suggest combining Rotation Forest with logitboost ensemble. We use Logitboost DS as base learner of Rotation Forest ensemble. The design choices and the parameter values of the Rotation Forest were picked in advance and not changed during the experiment. In detail, these were as follows:

1. Number of features in a subset: 3;
2. Number of classifiers in the ensemble: 5;
3. Extraction method: principal component analysis (PCA);
4. Base classifier model: Logitboost DS.

The approach is presented briefly in Figure 1. It has been observed that for bagging, boosting, rotation forest and random subspace method, an increase in committee size (sub-classifiers) usually leads to a decrease in prediction error, but the relative impact of each successive addition to a committee is ever diminishing. Most of the effect of each technique is obtained by the first few committee members [8,9]. We used 5 × 5 sub-classifiers for the presented algorithm.

The presented ensemble is effective owing to representational reason. The hypothesis space h may not contain the true function f (mapping each example to its real class),

```
(input LS learning set; $T(=5)$ number of bootstrap samples; LA learning algorithm
output C* classifier)
Begin
Let $E$ be an ensemble of learners, initially empty.
  for $j = 1$ to $T$ do
  begin
    The input variables are randomly grouped.
    For each group of input variables:
    – Consider a data set formed by this input variables.
    – Eliminate from the data set all the examples from a proper subset of the classes.
    – Eliminate from the data set a subset of the examples.
    – Apply PCA (Principal Component Analysis) with the remaining data set.
    – Consider the components of PCA as a new set of variables.
   $S_j$ := the training data set using as new variables the components selected by PCA
      for each group
   1. Starting with the instances of $Sj$ with equal weights $w_i = 1/N$, $i = 1, \dots, N$,
      function $F(x) = 0$ and sample probability estimates $p(x_i) = 0.5$.
   2. Repeat for $m = 1$ to $T$:
    a. Compute the working response and sample weights:
      $z_i = [y_i - p(x_i)]/[p(x_i)(1 - p(x_i))]$,
      $w_i = p(x_i)(1 - p(x_i))$.
    b. Fit the decision stump $f_m(x)$ by weighted least squares regression of $z_i$ to $x_i$
      using weights $w_i$.
    c. Update $F(x)$ and $p(x)$:
      $F(x) = F(x) + 0.5 f_m(x)$,
      $p(x) = \left(1 + e^{-2F(x)}\right)^{-1}$
    d. Add $f_m(x)$ to $E$.
    end of repeat
  end of for
begin
  Output C* = The most often predicted class of $E$
End
```

FIGURE 1. The rotation forest of Logitboost algorithm

but several good approximations. Then, by taking combinations of these approximations, classifiers that lie outside of h may be represented.

4. **Comparisons and Results.** For the comparisons of our study, we used 32 well-known datasets mostly from many domains from the UCI repository [23]. These data sets were selected so as to come from real-world problems and to vary in characteristics. Thus, we have used data sets from the domains of: pattern recognition (anneal, iris, zoo), image recognition (ionosphere, sonar), medical diagnosis (breast-cancer, colic, breast-w, diabetes, heart-c, heart-h, heart-statlog, hepatitis, lymphotherapy, primary-tumor) commodity trading (autos, credit-g) music composition (waveform), computer games (monk1, monk2, kr-vs-kp), various control applications (balance), language morphological analysis (dimin) [24] and prediction of student dropout (student) [25]. In order to calculate the classifiers accuracy, the whole training set was divided into ten mutually exclusive and equal-sized subsets and for each subset the classifier was trained on the union of all of

Table 1. Comparing the presented ensemble with well known ensembles that uses as base classifier the DS

Dataset	Rotation Forest	Bagging	Boosting	Random-	Dagging
	of Logitboost DS	DS	DS	Subspace DS	DS
anneal	98,33	82,96 *	83,63 *	82,26 *	83,64 *
audiology	79,21	46,46 *	46,46 *	46,46 *	32,35 *
autos	79,05	44,95 *	44,9 *	45,29 *	46,64 *
breast-cancer	73,45	73,38	71,55	73,9	72,22
breast-w	97,42	92,56 *	95,28	93,29 *	95,22
colic	83,41	81,52	82,72	81,78	80,84
credit-g	74,2	70 *	72,6	70 *	70,2 *
diabetes	76,7	72,45 *	75,37	72,35 *	75,25
dimin	94,3	59,31 *	59,31 *	62,92 *	59,57 *
haberman	73,82	73,07	74,06	73,23	73,16
heart-c	85,11	75,26 *	83,11	75,1 *	81 *
heart-h	80,98	81,41	82,42	81,71	82,89
heart-statlog	85,19	75,33 *	81,81	75,3 *	80,48
hepatitis	82,54	80,61	81,5	79,31	80,59
hypothyroid	97,88	95,39	92,97 *	93,84 *	94,99
ionosphere	92,61	82,66 *	92,34	84,82 *	81,68 *
iris	95,33	68,87 *	95,07	72 *	75,2 *
kr-vs-kp	93,18	66,05 *	95,08	82,97 *	67,15 *
lymphography	85,05	74,5 *	75,44 *	74,08 *	75,24 *
monk1	82,69	73,41 *	69,79 *	72,17 *	66,94 *
monk2	57,35	61,13	53,99	61,96	58,58
primary-tumor	48,98	28,91 *	28,91 *	27,38 *	26,81 *
segment	95,19	56,54 *	28,52 *	57,36 *	60,93 *
sick	96,51	96,55	97,07	94,03	96,52
sonar	80,71	73,21 *	81,06	72,64 *	70,89 *
soybean	93,11	27,83 *	27,96 *	38,02 *	44,71 *
students	86,67	87,22	87,16	86,51	87,19
titanic	78,65	77,6	77,83	77,6	77,6
vote	96,78	95,63	96,41	94,83	95,61
vowel	75,35	23,58 *	17,47 *	28,35 *	37,06 *
waveform	84,7	57,49 *	67,68 *	61,85 *	67,18 *
zoo	96,18	60,53 *	60,43 *	60,43 *	52,99 *
W/D/L		0/11/21	0/19/13	0/10/22	0/14/18
Average accuracy	84,39	69,26	71,25	70,43	70,35

the other subsets. Then, cross validation was run 10 times for each algorithm and the average value of the 10-cross validations was calculated [26].

For bagging, boosting and random subspace methodology, much of the reduction in error appears to have occurred after ten to fifteen classifiers. However, Adaboosting continues to measurably improve their test-set error until around 25 classifiers [1]. The time complexity of the presented ensemble is about the same with the remaining ensembles. This happens because we use $5 * 5$ sub-classifiers (totally 25).

Table 2. Comparing the presented ensemble with other well known ensembles that uses as base classifier the DS

Dataset	Rotation Forest of Logitboost DS	Multiboost DS	Decorate DS	Rotation Forest DS	Logitboost DS
anneal	98,33	83,63 *	76,89 *	84,07 *	98,55
audiology	79,21	46,46 *	46,46 *	46,46 *	84,92 v
autos	79,05	44,9 *	52,02 *	45,81 *	80,93
breast-cancer	73,45	71,9	75,16	73,81	72,4
breast-w	97,42	95,07	95,04	96,85	95,71
colic	83,41	83,13	83,01	82,06	81,51
credit-g	74,2	71,34	70 *	70 *	70,8 *
diabetes	76,7	75,22	76,08	74,6	74,09
dimin	94,3	59,31 *	64,75 *	84 *	96,3
haberman	73,82	73,09	71,86	75,14	74,82
heart-c	85,11	83,34	72,43 *	82,12	83,46
heart-h	80,98	82,26	81,78	82,01	77,57 *
heart-statlog	85,19	82,48	81,04 *	81,85	82,22
hepatitis	82,54	81,13	80,82	78,71 *	81,92
hypothyroid	97,88	92,97 *	92,97 *	92,89 *	99,58
ionosphere	92,61	87,69 *	90,61	86,05 *	91,17
iris	95,33	95,13	94	65,33 *	94
kr-vs-kp	93,18	93,94	90,43	88,62 *	93,8
lymphography	85,05	74,69 *	71,48 *	75,52 *	82,33 *
monk1	82,69	72,35 *	69,23 *	76,47 *	72,31 *
monk2	57,35	54,47	58,6	62,13	54,45
primary-tumor	48,98	28,91 *	28,6 *	24,79 *	46,91
segment	95,19	28,52 *	40,39 *	62,73 *	95,93
sick	96,51	96,81	96,81	95,04	97,91
sonar	80,71	78,29	75,45 *	74 *	79,29
soybean	93,11	27,96 *	39,96 *	34,72 *	92,97
students	86,67	87,22	87,24	87,24	86,37
titanic	78,65	77,6	77,6	78,55	77,83
vote	96,78	95,56	95,64	95,64	95,41
vowel	75,35	17,47 *	27,37 *	38,99 *	71,31 *
waveform	84,7	66,17 *	67,3 *	69,92 *	82,66
zoo	96,18	60,43 *	61,36 *	60,45 *	95,09
W/D/L		0/18/14	0/15/17	0/13/19	1/26/5
Average accuracy	84,39	70,92	71,64	72,71	83,27

We compare the presented ensemble with bagging, boosting, logitboost, RandomSubSpace and MultiBoost version of DS (using 25 sub-classifiers), as well as, with DECORATE, Dagging and Rotation Forest combining method using DS as base classifier. It must be also mentioned that we used the free available source code for these algorithms by [26] for our experiments. Decision stumps (DS) are one level decision trees [6] that classify instances by sorting them based on feature values. In the last raw of Table 1 and Table 2 one can see the aggregated results.

In the experiments, we represent with $*$ that the specific ensemble looses from the presented ensemble. That is, the presented algorithm performed statistically better than the specific ensemble according to t-test with $p<0.05$. In addition, in Table 1, we represent with v that the presented ensemble looses from the specific ensemble according to t-test with $p<0.05$. In all the other cases, there is no significant statistical difference between the results (Draws). In the last row of the Table 1 and Table 2 one can see the aggregated results in the form (a/b/c). In this notation, a means that the specific ensemble algorithm is significantly more accurate than the presented ensemble in a out of 32 data sets, c means that the presented ensemble is significantly more accurate than the specific ensemble in c out of 32 data sets, while in the remaining cases (b), there is no significant statistical difference between the results.

The presented ensemble is significantly more accurate than Bagging DS in 21 out of the 32 data sets, while it has significantly higher error rates in none data set. The presented ensemble is significantly more accurate than Boosting DS in 13 out of the 32 data sets whilst it has significantly higher error rates in none data set. Furthermore, Dagging DS has significantly lower error rates in 18 out of the 32 data sets than the presented ensemble, whereas it is significantly less accurate in none data set. What is more, Rotation Forest DS is significantly more accurate than the presented ensemble in none out of the 32 data sets whilst it has significantly higher error rates in 19 data sets. The presented ensemble is significantly more accurate than DECORATE DS in 17 out of the 32 data sets, while it has significantly higher error rates in none data set. The presented ensemble is significantly more accurate than Random-Subspace DS in 22 out of the 32 data sets whilst it has significantly higher error rates in none data set. What is more, Logitboost DS is significantly more accurate than the presented ensemble in one out of the 32 data sets whilst it has significantly higher error rates in 5 data sets. To sum up, the performance of the presented ensemble is more accurate than the other well-known ensembles that use only the DS algorithm.

5. Conclusions.

One of the most active areas of research in supervised machine learning has been to study methods for constructing good ensembles of learners [27]. In this work we built an ensemble combining rotation forest and logitboost ensembles. It was proved after a number of comparisons with other ensembles, that the presented methodology gives better accuracy in most cases. The presented ensemble has been demonstrated to (in general) achieve lower error than either boosting or bagging or random subspace or rotation forest method or other well known ensembles methods when applied to a base learning algorithm and learning tasks for which there is sufficient scope for both bias and variance reduction.

Nevertheless, there are still some interesting problems deserved to be investigated further, which include but are not limited to the following items: (a) Evaluation of the performance of the presented algorithm by adopting other algorithms such as rule learners and neural networks as the base learning algorithm; (b) How to automatically select the optimal number of learners in each sub-ensemble.

REFERENCES

[1] T. G. Dietterich, Ensemble methods in machine learning, *Multiple Classifier Systems, LNCS*, vol.1857, pp.1-15, 2001.

[2] T. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization, *Machine Learning*, vol.40, pp.139-157, 2000.

[3] J. Friedman, T. Hastie and R. Tibshirani, Additive logistic regression: A statistical view of boosting, *The Annals of Statistics*, vol.28, 2000.

[4] J. J. Rodrguez, L. I. Kuncheva and C. J. Alonso, Rotation forest: A new classifier ensemble method, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.28, no.10, pp.1619-1630, 2006.

[5] L. I. Kuncheva and J. J. Rodriguez, An experimental study on rotation forest ensembles, *Proc. of the 7th International Conference on Multiple Classifier Systems*, pp.459-468, 2007.

[6] W. Iba and P. Langley, Induction of one-level decision trees, *Proc. of the 9th International Machine Learning Conference*, Aberdeen, Scotland, 1992.

[7] L. Breiman, Bagging predictors, *Machine Learning*, vol.24, no.3, pp.123-140, 1996.

[8] E. Bauer and R. Kohavi, An empirical comparison of voting classification algorithms: Bagging, boosting, and variants, *Machine Learning*, vol.36, pp.105-139, 1999.

[9] D. Opitz and R. Maclin, Popular ensemble methods: An empirical study, *Artificial Intelligence Research*, vol.11, pp.169-198, 1999.

[10] G. Fumera, F. Roli and A. Serrau, A theoretical analysis of bagging as a linear combination of classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.30, no.7, 2008.

[11] K. M. Ting and I. H. Witten, Stacking bagged and dagged models, *The 14th International Conference on Machine Learning*, San Francisco, CA, USA, pp.367-375, 1997.

[12] T. K. Ho, The random subspace method for constructing decision forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.20, no.8, pp.832-844, 1998.

[13] L. Breiman, Random forests, *Machine Learning*, vol.45, no.1, pp.532, 2001.

[14] Q.-T. Cai, C.-Y. Peng and C.-S. Zhang, A weighted subspace approach for improving Bagging performance, *IEEE ICASSP2008*, pp.3341-3344, 2008.

[15] Y. Freund and R. E. Schapire, Experiments with a new boosting algorithm, *Proc. of ICML1996*, pp.148-156, 1996.

[16] Y. Freund and R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, *J. Comput. System Sci.*, vol.55, no.1, pp.119-139, 1997.

[17] R. E. Schapire, Y. Freund, P. Bartlett and W. S. Lee, Boosting the margin: A new explanation for the effectiveness of voting methods, *The Annals of Statistics*, vol.26, pp.1651-1686, 1998.

[18] R. E. Schapire and Y. Singer, Improved boosting algorithms using confidence-rated predictions, *Machine Learning*, vol.37, pp.297-336, 1999.

[19] X.-C. Yin, C.-P. Liu and Z. Han, Feature combination using boosting, *Pattern Recognition Letters*, vol.26, pp.2195-2205, 2005.

[20] N. Garca-Pedrajas and D. Ortiz-Boyer, Boosting random subspace method, *Neural Networks*, vol.21, pp.1344-1362, 2008.

[21] G. I. Webb, MultiBoosting: A technique for combining boosting and wagging, *Machine Learning*, vol.40, pp.159-196, 2000.

[22] P. Melville and R. Mooney, Constructing diverse classifier ensembles using artificial training examples, *Proc. of IJCAI2003*, Acapulco, Mexico, pp.505-510, 2003.

[23] A. Frank and A. Asuncion, *UCI Machine Learning Repository*, http://archive.ics.uci.edu/ml, University of California, Irvine, CA, 2010.

[24] A. Bosch and W. Daelemans, Memory-based morphological analysis, *Proc. of the 37th Annual Meeting of the ACL*, pp.285-292, 1999.

[25] S. Kotsiantis, C. Pierrakeas and P. Pintelas, Preventing student dropout in distance learning systems using machine learning techniques, *LNAI*, vol.2774, pp.267-274, 2003.

[26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, The WEKA data mining software: An update, *SIGKDD Explorations*, vol.11, no.1, 2009.

[27] J. H. Freidman and P. Hall, On bagging and nonlinear estimation, *Journal of Statistical Planning and Inference*, vol.137, no.3, pp.669-683, 2007.