Integrating Global and Local Boosting

Anastasia-Dimitra Lipitakis
Department of Mathematics
University of Patras
Greece
adlipitaki@gmail.com

Gerasimos S. Antzoulatos
Department of Mathematics
University of Patras
Greece
antzoulatos@upatras.gr

Sotiris Kotsiantis
Department of Mathematics
University of Patras
Greece
sotos@math.upatras.gr

Michael N. Vrahatis
Department of Mathematics
University of Patras
Greece
vrahatis@math.upatras.gr

***Abstract*—Several data analysis problems require investigations of relationships between attributes in related heterogeneous databases, where different prediction models can be more appropriate for different regions. A new technique of integrating global and local boosting is proposed. A comparison with other well known and widely used combining methods on standard benchmark datasets has shown that the proposed technique leads to more accurate results.**

Keywords—global boosting; local boosting; ensemble of classifiers; machine learning; classification; heterogeneous databases; inductive learning system

I. Introduction

Various multiple learner systems (an ensemble of classifiers) try to exploit the local different behavior of the base classifiers in order to improve the accuracy and reliability of the overall inductive learning system [23]. A useful and informal reasoning, from computational, statistical and representational viewpoints, of why ensembles can lead to better results can be found in [7].

A learning algorithm works as a global method if all training examples are considered when classifying a new test case. A learning algorithm works as a local method if only data local to the area around the testing instance contribute to the class probabilities [2]. Local methods have significant advantages when the probability measure defined on the space of objects for each class is complex, but can still be described by using a collection of less complex local approximations [1].

In the article at hand, an ensemble integrating global and local boosting algorithm is proposed. A comparison with other known ensembles has been performed on standard benchmark datasets and in most the cases the proposed technique has achieved better accuracy. For the experiments, representative learning algorithms, such as decision trees and Bayesian classifiers have been used. In section II, well known algorithms for building ensembles that are based on a single learning algorithm are presented, while in section III the proposed ensemble method is discussed. Experimental results using a number data sets and comparisons of the proposed method with other ensembles are presented in section IV. Finally, additional research topics and future research work are given in Section V.

II. ENSEMBLES OF CLASSIFIERS

Empirical studies have been showed that classification problem ensembles are often more accurate than the individual base learner that make them up [3], and recently different theoretical explanations have been proposed to justify the effectiveness of some commonly used ensemble methods [21]. In this research work a generative combining method is proposed and for this reason the most well-known generative methods for building ensembles of classifiers in the literature are presented.

The bagging method [6] samples the training set, generating random independent bootstrap replicates, constructs the classifier on each of these, and aggregates them by a simple majority vote in the final decision rule. Another related method that uses a different subset of training data with a single learning method is the boosting algorithmic approach [11]. This method assigns weights to the training instances, and these weight values are changed depending upon how well the associated training instance is learned by the classifier with the weights for misclassified instances being increased. After several iteration cycles, the prediction is performed by taking a weighted vote of the predictions of each classifier, with the weights being proportional to each classifier's accuracy on its training set. AdaBoost is considered to be a practical version of the boosting approach [11]. From the perspectives of additive regression model and exponential loss function, some other boosting algorithms including ''GentleBoost" and ''LogitBoost" have been proposed [12]. Boosting with bagging has been compared by Shirai et al.[22] using different base algorithms. MultiBoosting is another method that can be considered as wagging committees formed by AdaBoost [24]. Wagging is a variant of bagging; bagging uses resampling to get the datasets for training and producing a weak hypothesis, whereas wagging uses reweighting for each training example.

Another approach is the Random Subspace Method [14], which consists of training several classifiers from input data sets constructed with a given proportion around 50% of features picked randomly from the original set of features. Feature subset by the regularized version of Boosting [19], i.e., AdaboostReg has been introduced by Redpath and Lebart. Based on Principal Component Analysis (PCA), Rodríguez et al. in [20] proposed a new ensemble classifier generation technique Rotation Forest. Diversity is promoted by using PCA to do feature extraction for each base classifier.

Another meta-learner (DECORATE, Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples) that uses a learner (one that provides high accuracy on the training data) to build a diverse committee has been presented in [17]. This is accomplished by adding different randomly constructed examples to the training set when building new committee members.

III. THE PROPOSED ALGORITHM

It is known that boosting is an effective technique for improving prediction accuracy in many real life datasets [11]. However, several researches indicated that in heterogeneous databases, where several homogeneous regions exist, boosting does not enhance the prediction capabilities as well as for homogeneous databases [16].

Local learning typically depends on the notion of "neighborhood". The neighborhood can be based on some a-priori measure of locality such as the Euclidean distance in input space. Local learning [1] can be understood as a general principle that allows extending learning techniques, to the case of complex data for which the model's assumptions would not necessarily hold globally, but can be thought as valid locally. A simple example is the assumption of the linear separability, which, in general, is not satisfied globally in classification problems with rich data. Moreover, when the size of the training set is small enough compared to the complexity of the classifier, the learning machine usually overfits the noise in the training data. Thus effective control of complexity of a classifier plays a key role in achieving good generalization. Some related theoretical results and experimental results [24] indicate that a local learning algorithm (that is learning machine trained on the training subset) provides a feasible solution to this problem. A theoretical model of a local learning algorithm has been proposed in [5] and obtained bounds for the local risk minimization estimator.

The proposed model simple trains a boosting ensemble during the train process. For this reason, the training time of the model is that of simple boosting. During the classification of a test instance the model calculates the probabilities for each class and if the probability of the most possible class is at least two times the probability of the next possible class then the decision is that of global boosting model. However, if the global boosting is not appropriate e.g. the probability of the most possible class is less than two times the probability of the next possible class; the model finds the *k* nearest neighbors using the selected distance metric and train a local boosting model using these *k* instances. Finally, in this case the model averages the probabilities of global boosting with local boosting model for the classification of the testing instance. It must be mentioned that local boosting model is only used for a small number of test instances and for this reason classification time is not considered to be substantial deficiency for our model. In general, the proposed ensemble is described by pseudo-code in Fig 1. The proposed algorithm requires choosing the value of *K*. Note that there are several ways to do this. Firstly, a simple solution is to fix *K* a priori before the beginning of the learning process. Secondly, a more time-consuming solution is to determine the best *K* automatically through the minimization of a cost criterion. In the current implementation a fixed value for *K* (=50) has been used in order to i) keep the training time low and ii) since about this size of instances is appropriate for a simple algorithm, to build a precise model according to approach proposed by Frank et al. [9].

```
Build Global boosting model in all the
training set Classification:
1.   Obtain the test instance
2.   Calculate the probabilities of
belonging the instance in each class of
the dataset.
3.   If the probability of the most
possible class is at least two times the
probability of the next possible class
then the decision is that of global
boosting model else
a. Find the k(=50) nearest neighbors using
the selected distance metric
b. Using as training instances the k
instances train the local boosting model
c. Aggregate the decisions of global
boosting with local boosting model by
averaging of the probabilities for the
classification of the testing instance.
```

Fig. 1. Integrating Global and Local Application of Boosting (IGLB)

The proposed ensemble uses another free parameter such as the distance metric. In the numerical experiments, the most well known "Euclidean similarity" has been used as a distance metric. The performance of ADABoost.M1 has been shown to exceed or meet that of various other boosting algorithms [11], thus making it a good choice for this research work. Ten iterations for the boosting process have been used in order to reduce the time need for classification of a new instance.

IV. NUMERICAL EXPERIMENTS

In our numerical experiments several datasets from the UCI repository were used [10]. In order to calculate the classifiers' accuracy, the whole training set (27 datasets) was divided into ten mutually exclusive and equal-sized subsets, while for each subset the classifier was trained on the union of all of the other subsets. Then, cross validation was run 10 times for each algorithm and the median value of the 10-cross validations was calculated. It must be mentioned that for our numerical experiments the free available source code was used for most of the algorithms [26].

In the following subsection, the experimental results for different base classifiers are presented. For the experiments, representative algorithms of decision trees and Bayesian classifiers were used. We tried to minimize the effect of any expert bias by not attempting to tune any of the algorithms to the specific data set. Default values of learning parameters were used. This approach results in lower estimates of the true error rate, but it is a bias that affects all the learning algorithms uniformly.

A. Using Decision Stump (DS) as a base classifier

Decision stump (DS) are one level decision trees that classify instances by sorting them based on feature values [15]. Firstly, we compare the proposed ensemble methodology with Boosting DS [11], Logitboost DS [12], MultiBoost DS [24], Rotation Forest DS [20], Random Subspace DS [14] and Decorate DS [17] (using 10 sub-classifiers). The classification accuracy of each tested algorithm is presented in Table I.

Table I. COMPARING THE PROPOSED ENSEMBLE WITH OTHER ENSEMBLES THAT USE DS AS BASE LEARNER

Dataset	IGLB DS	Boosting DS	Logitboost DS	Multiboost DS	Rotation Forest DS	RS DS	Decorate DS
audiology	61.57	46.46	83.73	46.46	47.08	46.46	46.46
Autos	68.67	44.90	79.12	44.90	44.56	47.24	49.35
Badges	100.00	100.00	100.00	100.00	96.63	85.17	100.00
breast-cancer	74.95	71.62	71.42	71.19	73.28	72.67	75.26
wisconsin-breast-cancer	95.87	95.14	95.61	94.52	93.19	96.78	95.04
horse-colic	81.93	82.53	82.75	81.52	80.57	82.30	82.71
german_credit	73.85	71.27	71.68	70.11	70.00	70.00	70.00
credit-rating	86.46	84.80	85.72	85.51	84.75	85.49	84.42
pima_diabetes	74.61	74.92	74.54	73.38	71.94	74.47	75.65
Glass	70.28	44.89	70.99	44.89	49.41	50.23	53.00
cleveland-14-heart-diseas	82.84	83.47	81.59	82.61	75.97	79.60	72.43
hungarian-14-heart-diseas	81.31	81.41	81.47	81.31	81.51	80.96	81.78
heart-statlog	81.59	81.59	82.22	81.89	75.48	81.04	80.00
hepatitis	85.14	81.37	81.58	80.34	78.80	80.61	79.97
ionosphere	91.00	90.89	90.83	84.88	84.65	86.05	89.46
Iris	94.33	95.40	94.93	93.20	70.53	66.73	93.27
Labor	91.73	88.37	92.33	83.20	84.40	87.17	88.20
lymphography	82.85	75.44	82.36	73.73	73.81	76.25	74.68
monk1	84.23	70.29	71.63	73.41	71.78	73.88	72.26
monk2	56.56	53.95	55.60	57.61	61.96	61.83	61.13
monk3	92.87	92.06	93.37	90.90	85.24	89.33	93.45
Sonar	84.85	75.65	77.17	74.45	72.21	75.12	73.57
Relation	78.33	77.67	77.83	77.60	76.51	78.12	77.60
Vehicle	76.65	39.81	70.73	39.81	39.63	55.28	44.45
Vote	96.09	95.43	95.49	95.61	94.32	95.54	95.63
Wine	96.86	89.37	97.86	85.56	81.67	92.81	91.75
Zoo	83.55	60.43	95.06	60.43	60.33	61.44	60.71

In Table II the results of Friedman test are presented [13]. According to Holm/Hochberg Table [13] presented in Table III, the proposed ensemble is statistically significant more accurate than Boosting DS, Rotation Forest DS, DECORATE DS, Random-Subspace DS and Multiboost DS. On the other hand, the proposed method is more accurate but not statistically better than Logitboost DS according to our experiments.

Table II. Average Rankings of the algorithms (Friedman test)

Algorithm	Ranking
IGLB DS	2.185185185185184
Logitboost DS	2.592592592592593
Decorate DS	3.962962962962963
Boosting DS	4.111111111111111
RS DS	4.240740740740739
Multiboost DS	5.018518518518516
Rotation Forest DS	5.888888888888891

Table III. Holm/Hochberg Table for a = 0.05

Algorithm	$z=(R_0-R_i)/SE$	p	Holm/Hochberg /Hommel
Rotation Forest DS	6.29940788	2.98784E-10	0.0083333
Multiboost DS	4.81904703	1.44245E-6	0.01
RS DS	3.49617137	4.719854E-4	0.0125
Boosting DS	3.27569209	0.0010540	0.0166666
Decorate DS	3.02371578	0.0024969	0.025
Logitboost DS	0.69293486	0.4883504	0.05

B. Using C4.5 algorithm as base classifier

The most commonly used C4.5 algorithm [18] was the representative of the decision trees in our study. Subsequently, we compare the proposed ensemble methodology with Bagging C4.5 [6], Boosting C4.5 [11], MultiBoost C4.5 [24], Rotation Forest C4.5 [20], Random Subspace C4.5 [14] and Decorate C4.5 [17] algorithms (using 10 sub-classifiers). The classification accuracy of each tested algorithm is presented in Table IV. In Table V the results of Friedman test are presented.

Table IV. COMPARING THE PROPOSED ENSEMBLE WITH OTHER ENSEMBLE THAT USE C4.5 AS BASE LEARNER

Dataset	IGLB C4.5	Boosting C4.5	Bagging C4.5	Multiboost C4.5	Rotation Forest C4.5	RS C4.5	Decorate C4.5
Audiology	85.14	84.75	80.57	83.99	79.95	76.36	80.90
Autos	85.75	85.46	83.09	84.15	82.33	84.05	83.13
Badges	100.00	100.00	100.00	100.00	86.19	96.46	100.00
breast-cancer	70.49	66.89	72.45	69.73	72.19	73.23	70.13
wisconsin-breast-cancer	97.28	96.08	95.95	96.08	97.12	96.32	95.79
horse-colic	82.82	81.63	85.15	83.85	84.29	83.90	84.26
german_credit	74.49	70.75	73.36	73.75	74.45	73.91	72.21
credit-rating	86.19	84.01	85.78	86.17	86.01	85.36	85.49
pima_diabetes	75.36	71.69	75.72	75.14	76.21	73.96	74.88
Glass	76.50	75.15	72.89	75.67	74.62	74.89	71.34
cleveland-14-heart-diseas	80.07	78.79	78.98	79.74	82.61	81.26	77.95
hungarian-14-heart-diseas	79.14	78.68	79.94	80.34	81.74	81.20	78.75
heart-statlog	80.67	78.59	80.67	80.33	82.70	82.19	78.59
Hepatitis	84.31	82.38	81.64	82.24	84.28	82.36	81.48
Ionosphere	92.99	93.05	91.63	92.60	93.68	92.97	91.91
Iris	94.97	94.33	94.93	94.67	95.33	94.20	95.13
Labor	87.67	87.17	84.30	84.97	91.70	78.67	88.37
lymphography	84.08	80.87	77.80	81.84	82.57	79.03	78.21
monk1	93.01	94.10	81.56	84.65	94.93	83.16	89.35
monk2	60.42	60.82	60.66	58.57	75.87	62.19	57.15
monk3	90.16	90.01	91.56	91.91	92.14	90.69	87.54
Sonar	83.75	79.13	79.41	80.06	81.88	80.69	81.22
Relation	78.98	78.86	78.09	78.61	78.91	77.96	78.87
Vehicle	75.68	75.59	74.17	75.65	78.34	75.21	75.14
Vote	96.44	95.51	96.46	95.81	96.34	95.08	94.55
Wine	97.89	96.45	95.21	96.35	97.79	96.07	96.46
Zoo	94.38	95.18	93.51	94.19	92.72	94.95	93.11

According to Holm/Hochberg Table [13] presented in Table VI, the proposed ensemble is statistically significant more accurate than Boosting C4.5, DECORATE C4.5, Random-Subspace C4.5 and Multiboost C4.5. It must be mentioned that according to our experiments the proposed method is more accurate but not statistically better than Rotation Forest C4.5.

Table V. Average Rankings of the algorithms (Friedman test)

Algorithm	Ranking
IGLB C4.5	2.4999999999999999
Rotation Forest C4.5	2.518518518518517
Multiboost C4.5	4.092592592592593
Boosting C4.5	4.592592592592593
Bagging C4.5	4.759259259259259
RS C4.5	4.407407407407406
Decorate C4.5	5.129629629629629

Table VI. Holm/Hochberg Table for $a = 0.05$

Algorithm	$z=(R_0-R_i)/SE$	p	Holm/Hochberg /Hommel
Decorate C4.5	4.47257959	7.728161E-6	0.0083333
Bagging C4.5	3.84263880	1.217185E-4	0.01
Boosting C4.5	3.55916545	3.720351E-4	0.0166666
RS C4.5	3.24419505	0.00117783	0.0125
Multiboost C4.5	2.7087453898	0.00675381	0.05
Rotation Forest C4.5	0.0314970394	0.97487315	0.0083333

Table VII. COMPARING THE PROPOSED ENSEMBLE WITH OTHER ENSEMBLES THAT USE NB AS BASE LEARNER

Dataset	IGLB NB	Bagging NB	Boosting NB	Decorate NB	Rotation Forest NB	RS NB	Multiboost NB
audiology	79.21	72.46	79.26	72.69	71.34	67.14	74.36
Autos	78.14	57.63	57.12	57.76	62.76	58.83	58.49
Badges	98.78	99.66	99.66	96.87	84.80	96.99	99.66
breast-cancer	69.77	72.80	68.68	73.22	69.64	71.58	72.22
wisconsin-breast-cancer	96.15	96.12	95.55	96.02	95.82	95.95	96.05
horse-colic	78.65	78.92	77.62	78.21	71.53	79.87	80.20
german_credit	73.00	74.89	75.14	74.84	66.42	73.91	75.39
credit-rating	81.43	77.87	81.04	78.51	80.20	77.80	78.94
pima_diabetes	73.56	75.73	75.86	75.48	75.03	74.68	76.44
Glass	71.52	50.00	49.63	49.45	54.03	51.15	49.91
cleveland-14-heart-diseas	80.17	83.38	82.97	83.37	81.95	83.10	83.90
hungarian-14-heart-diseas	81.48	84.09	84.81	83.88	83.82	83.52	84.13
heart-statlog	79.15	83.74	82.59	83.81	84.07	83.59	84.19
Hepatitis	84.95	83.88	84.62	83.16	83.93	84.62	85.56
ionosphere	88.52	82.25	91.06	82.48	88.29	82.99	90.66
Iris	95.97	95.53	94.80	94.67	96.47	94.80	95.80
Labor	91.77	93.70	89.60	91.53	91.43	94.10	92.87
lymphography	86.44	83.56	81.27	82.92	82.89	82.33	83.86
monk1	83.98	73.55	72.68	75.51	75.10	71.60	71.97
monk2	58.04	55.59	56.83	56.98	55.25	60.06	56.59
monk3	91.29	93.45	90.90	92.72	90.76	89.04	92.75
Sonar	87.62	68.25	80.77	67.60	70.41	68.25	76.07
Relation	78.27	77.87	77.86	78.31	74.38	77.14	77.70
Vehicle	75.73	45.56	44.68	46.31	67.28	45.64	44.68
Vote	95.01	90.07	95.01	89.70	93.33	89.93	91.31
Wine	97.69	97.40	96.18	96.78	98.25	97.19	97.08
Zoo	97.03	94.77	97.23	94.97	92.16	94.47	96.63

C. *Using the Naive Bayes algorithm as a base classifier*

In this sub-section, a Bayesian method as a base classifier in the ensemble has been used. Naive Bayes algorithm was the representative of the Bayesian networks [8]. Subsequently, we compare the proposed ensemble methodology with Bagging NB [6], Boosting NB [11], MultiBoost NB [24], Rotation Forest NB [20], Random Subspace NB [14] and Decorate NB [17] algorithms (using 10 sub-classifiers). The classification accuracy of each tested algorithm is presented in Table VII. In Table VIII the results of Friedman test are presented.

Table VIII. Average Rankings of the algorithms (Friedman test)

Algorithm	Ranking
IGLB NB	2.907407407407408
Multiboost NB	3.018518518518518
Bagging NB	3.870370370370370
Boosting NB	4.222222222222222
Decorate NB	4.481481481481481
Rotation Forest NB	4.666666666666667
RS NB	4.833333333333333

According to Holm/Hochberg Table [13] presented in Table IX, the proposed ensemble is statistically significant more accurate than Boosting NB, DECORATE NB, Rotation Forest NB and Random-Subspace NB. On the other hand, according to our experiments the proposed method is more accurate but not statistically better than Multiboost NB and Bagging NB.

Table IX. Holm/Hochberg Table for $a = 0.05$

Algorithm	$z=(R_0-R_i)/SE$	p	Holm/Hochberg /Hommel
RS NB	3.2756920994	0.00105403	0.0083333
Rotation Forest NB	2.9922187446	0.00276957	0.01
Decorate NB	2.6772483504	0.00742295	0.0166666
Boosting NB	2.2362897986	0.02533279	0.0125
Bagging NB	1.6378460497	0.10145381	0.05
Multiboost NB	0.1889822365	0.85010673	0.0083333

V. Synopsis and Concluding Remarks

In the research work at hand, an ensemble integrating global and local boosting is presented and our experiment for some real datasets shows that the proposed combining method outperforms other well known and widely used combining methods.

It must be mentioned that local boosting model is only used for a small number of test instances and for this reason classification time is not an important problem for our model. However, local weighted learning algorithms must often decide what examples to be stored for use during generalization in order to avoid extreme storage and time complexity [2]. By removing a set of examples from a dataset the response time for some classification decisions will decrease, as fewer examples are examined when a query example is presented. This objective is primary when we are working with a large dataset and the storage is limited.

In a subsequent future research work we will focus on the problem of reducing the size of the stored set of instances while trying to maintain or even improve generalization accuracy by avoiding noise and over-fitting.

Numerous instance selection methods that can be combined with the proposed techniques can be found in [4], [25].

References

[1] C. G. Atkeson, A.W. Moore and S. Schaal, Locally weighted learning, Artificial Intelligence Review, 11(1-5), 11–73, 1997.

[2] C. G. Atkeson, A. W. Moore and S. Schaal, Locally weighted learning for control, Artificial Intelligence Review, 11(1-5), 75–113, 1997.

[3] E. Bauer and R. Kohavi, An empirical comparison of voting classification algorithms: Bagging, boosting and variants, Machine Learning, 36(1/2), 525–536, 1999.

[4] H. Brighton, C. Mellish, Advances in Instance Selection for Instance-Based Learning Algorithms, Data Mining and Knowledge Discovery, 6, 153–172, 2002.

[5] L. Bottou and V. Vapnik, Local learning algorithm, Neural Computation, 4(6), 888–901, 1992.

[6] L. Breiman, Bagging Predictors. Machine Learning, 24(3), 123–140, 1996.

[7] T. G. Dietterich, Ensemble methods in machine learning, LNCS, 1857, 1–15, 2001.

[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] E. Frank, M. Hall and B. Pfahringer, Locally weighted naive Bayes, Proc. of the 19th Conference on Uncertainty in Artificial Intelligence, Mexico, 2003.

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

[11] Y. Freund and R. Schapire, Experiments with a New Boosting Algorithm, Proc. of the ICML'96, pp. 148–156, 1996.

[12] J. Friedman, T. Hastie and R. Tibshirani, Additive logistic regression: A statistical view of boosting, Annals of Statistics, 28(2), 337–407, 2000.

[13] S. García and F. Herrera, An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons, Journal of Machine Learning Research, 9, 2677–2694, 2008.

[14] Tin Kam Ho, The Random Subspace Method for Constructing Decision Forests, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8), 832–844, 1998.

[15] W. Iba and P. Langley, Induction of one-level decision trees, Proc. of the Ninth International Machine Learning Conference, Scotland, 1992.

[16] T. Lazarevic and Z. Obradovic, Adaptive Boosting for Spatial Functions with Unstable Driving Attributes, Proc. of the Pacific-Asia Conf. on KDDM, pp. 329–340, 2000.

[17] P. Melville and R. Mooney, Constructing Diverse Classifier Ensembles using Artificial Training Examples, Proc. of the IJCAI'03, pp.505–510, Mexico, 2003.

[18] J. Quinlan, C4.5: Programs for machine learning, Morgan Kaufmann, San Francisco, 1993.

[19] D. B. Redpath, K. Lebart, Boosting feature selection, Proc. of the Third IC on Advances in Pattern Recognition, pp. 305–314, UK, 2005.

[20] J. J. Rodríguez, L. I. Kuncheva and C. J. Alonso, Rotation forest: A new classifier ensemble method, IEEE Trans. Pattern Anal. Machine Intell. 28(10), 1619–1630, 2006.

[21] L. Rokach, Ensemble-based classifiers, Artificial Intelligence Review, 33, 1–39, 2010.

[22] S. Shirai, M. Kudo and A. Nakamura, Bagging, Random Subspace Method and Biding, SSPR&SPR 2008, LNCS, 5342, 801–810, 2008.

[23] V. N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[24] G. I. Webb, MultiBoosting: A Technique for Combining Boosting and Wagging, Machine Learning, 40, 159–196, 2000.

[25] D. Wilson and T. Martinez, Reduction Techniques for Instance-Based Learning Algorithms, Machine Learning, 38, 257–286, 2000.

[26] I. H. Witten, E. Frank and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, January 2011.