Ensembles of Region Based Classifiers

Sungha Choi[0], Byungwoo Lee[1], Jihoon Yang[2]
Digital Media Research Lab., LG Electronics, Korea[0].
Dept. of Computer Science, Sogang University, Korea[1,2].
choish@sogang.ac.kr[0], elva1212@sogang.ac.kr[1], yangjh@sogang.ac.kr[2]

Abstract

In machine learning, ensemble classifiers have been introduced for more accurate pattern classification than single classifiers. We propose a new ensemble learning method that employs a set of region based classifiers. Since the distribution of data can be different in different regions in the feature space, we split the data and generate classifiers based on each region and apply a weighted voting among the classifiers. We used 11 data sets from the UCI Machine Learning Repository to compare the performance of our new ensemble method with that of individual classifiers as well as other ensemble methods such as bagging and boosting. As a result, we found that our method improve performance, particularly when the base learner is Naïve Bayes or SVM.

1. Introduction

Ensemble learning in the supervised learning is not to find the very best one hypothesis for explaining given training data but to create a set of hypotheses and predict the class of new data by combining the set of hypotheses like majority voting [7].

These ensemble classifiers often have been experimentally shown to be more effective than single classifiers [1,6,16]. The two conditions for an ensemble classifier to have higher accuracy than a single classifier is that each classifier should render more accurate classification than random prediction and have errors in different regions in the input space [13].

The most widely known methods of ensemble learning are bagging [4] and boosting [10]. Bagging and boosting may have two controversial points as follows: First, when a base learner shows poor performance on a set of data, even re-sampling might not make the classification any easier. In this case, especially in bagging, the regions of misclassification are similar, which does not generate a fine ensemble with diversity.

Second, even though the distribution of data differs in each region, it cannot tell whether the ensemble yields high accuracy over that specific area. For instance, in case of boosting, since preceding classifier weights on misclassified instances, it is learned to classify the region the preceding classifier missed. But when it tries to classify a new instance, it cannot tell where the region in which the instance lies is. If the specific instances on which each classifier is good at classifying is known, it is expected to improve the accuracy of classification.

2. The relationship between ensembles and base learner

When a minor change of training data causes a major change of classifier created from learning algorithm, it is said that the algorithm is unstable [4]. For example, the algorithm used for creating a decision tree such as C4.5 may cause a major change of the tree for change of data. Such unstable algorithms are known to have high variance [3].

Although ensemble learning is independent of the base learner, generally ensemble learning such as bagging or boosting works well with unstable algorithms. Breiman suggested that bagging improves performance of these unstable classifiers [4]. Meantime, Breiman argued that bagging and boosting do not show any remarkable improvement in Naïve Bayes or in a stable algorithm based on the linear discriminant analysis [3]. Dietterich also recommended boosting in an algorithm creating less expressive hypothesis including decision trees [8]. So many of the previous approaches to ensemble learning have been focused on the ensemble of decision trees [1,6,12,19].

The reason why the ensemble of unstable learning algorithms works well is related to the diversity in that

† This work was supported by grant No. R01-2004-000-10689-0 from the Basic Research Program of the Korea Science & Engineering Foundation

0-7695-2983-6/07 $25.00 © 2007 IEEE
DOI 10.1109/CIT.2007.74

each classifier should have errors in different areas of the input space [13]. This diversity is one of the major properties of ensemble learning, and there have been many studies on the relation between the diversity and the efficiency of an ensemble [7,13,15].

While there has been a study of ensemble learning (of bagging and boosting) with a stable learning algorithm, support vector machines (SVM), with improved performance, it was applied to only a small number of data sets compared to ensembles of decision trees or neural networks [14,21]. And another study demonstrated the negative effect of the SVM-based ensemble [5,9]. The significant fact is that generating an ensemble with stable base learning algorithms has been found less efficient than generating the ensemble with unstable algorithms.

3. Ensembles of Region Based Classifiers

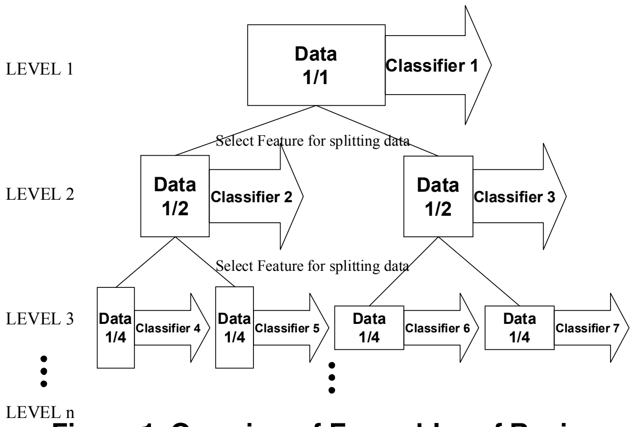

Figure 1. Overview of Ensembles of Region Based Classifiers

We propose an ensemble learning method represented as a *tree* in which each node corresponds to a base classifier that takes care of a particular region in the feature space (Figure 1). We consider two possibilities of using the ensemble in classification. First, only the leaf nodes (classifiers) in the tree are used. We call this a *region based classifier* (*RBC*). In RBC, only one classifier (among the leaf nodes) that is in charge of the region where the test instances lie is used for classification. Second, an ensemble of nodes in a specific path of the tree is used. We call these *ensembles of region based classifiers (RBE)*. In RBE, an instance is classified by voting of classifiers whose regions include the instance.

3.1. Algorithm

RBE generates a series of classifiers that correspond to regions in the feature space. The regions under consideration are iteratively split as learning progresses, and corresponding classifiers are generated using the patterns in each region. The detailed learning process of RBE for a sample data set is illustrated in Figure 2 and classification by the ensemble classifier is shown in Figure 3.

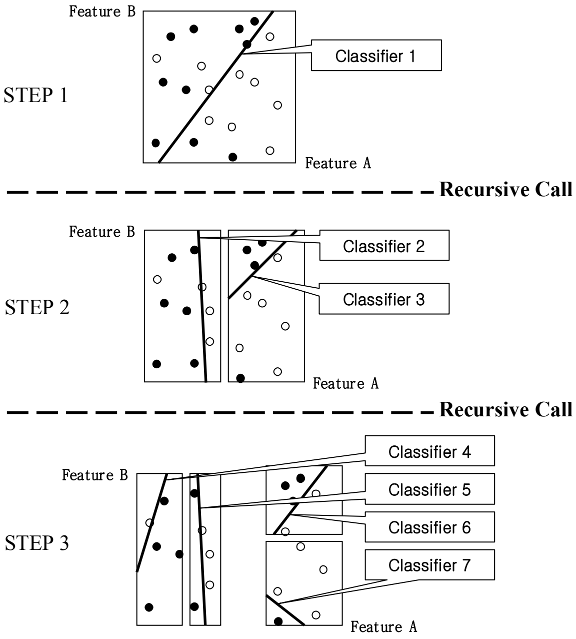

Figure 2. The learning process by RBE

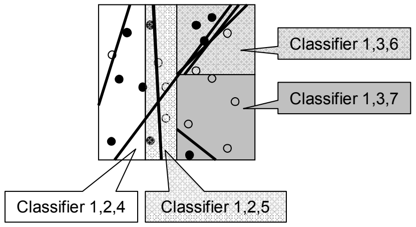

Figure 3. The classification by RBE

As shown in Figure 2, classifier C_1 (represented as "Classifier 1" in Figure 2) is constructed by training the entire data D_1 by the selected learning algorithm. Then the training data is divided into two regions containing the nearly same number of instances. This is to avoid generating imbalanced regions that might cause an inefficient ensemble. To divide the data in nearly half, we compute the information gain [18] of each feature in regard to the division. We then choose the feature with the maximum information gain, expecting this would induce subsets of data (corresponding to the newly generated regions) easier to classify. By the repetition of this process in the divided data the classifiers C_2, C_3 are generated. This can be implemented by recursive calls. We generated 7 classifiers (C_1, … , C_7) by going through the 3 regional splitting steps. When a test pattern is applied to the ensemble of the classifiers, it votes among the classifiers that include the test pattern in their region. In other words, a pattern is classified by the classifiers in a path of the tree each of which is in charge of the region that includes the test pattern. In case of a tie, the final decision is made by the leaf node in the path since it is the most accurate classifier (as proven by Theorem 1 in Section 3.2).

3.2. Analysis of RBE

Splitting the data in RBE has the following two significant aspects. First, as it goes on to split the data, the classifier becomes more accurate. Second, the classifiers learned from the data before the splitting can supplement the classifiers learned after the splitting. This is not only an advantage by going through the splitting but also the basis for practicing ensemble learning. The first issue is to be dealt with first.

In learning of RBE, going through the splitting simplifies the class distribution of the data and this can be led to an expectation that it will be easier to classify as it goes downwards to the leaf node. Therefore it can be anticipated that the classifiers of children nodes will have apparently less errors upon the training data than those of parent nodes do. This is to be presented and proved by Theorem 1.

Theorem 1] Assume that given training data D_i, the hypothesis produced by a learning algorithm L is h_i. And assume that for data D_0 and split data D_1, D_2 by halving D_0 based on a feature, hypotheses h_0, h_1, h_2 are produced by L, respectively. If the number of errors of h_0, h_1, h_2 upon individual training data are e_0, e_1, e_2 respectively, $e_0 \geq e_1 + e_2$.

Condition] The learning algorithm L chooses the hypothesis which has the least number of errors upon the data from the hypothesis space it can represent.

Proof] When h_0 creates e_{01} in D_1 and e_{02} in D_2, it is clearly $e_0 = e_{01} + e_{02}$. In order to prove $e_0 \geq e_1 + e_2$, it is sufficient to prove $e_{01} \geq e_1 \wedge e_{02} \geq e_2$.

In the hypothesis space the given learning algorithm L can express, the set of hypotheses generating the same value as the result h_0 has upon D_1 can be defined as H_{01}. And $h_0 \in H_{01}$ is proved; because the hypotheses in H_{01} can have any result value upon D_2 as long as it has the same result value as h_0 has upon D_1. That is, it demonstrates that it has e_{01} because every hypothesis included in H_{01} has the same result value as h_0 has upon D_1.

Now, suppose $e_{01} < e_1$ is true. Then, by the supposition, H_{01} which has e_{01} upon D_1 does not belong to the hypothesis space the learning algorithm L can express. If it does, by the given condition it comes to choose the hypothesis having e_{01} which is more accurate rather than choosing the hypothesis having e_1 which allows $e_{01} < e_1$ to be true. Therefore it has come to choose the hypothesis having e_1 which is no better than a hypothesis among H_{01}.

Hereby, it has been proved that hypotheses belonging to H_{01} do not exist in that hypothesis space. However, it is a contradiction because L found h_0 belonging to H_{01} when D_0 is given. Likewise, it also can be proved that $e_{02} < e_2$ is false. Consequently $e_{01} < e_1$ and $e_{02} < e_2$ are both proved to be false. Therefore $e_{01} \geq e_1 \wedge e_{02} \geq e_2$ is proved. ■

Table 1 is the results of the experiment for getting the actual application of Theorem 1. The learning algorithms tend to have a higher accuracy upon the training data in most data sets as it goes downwards. Exceptional cases are written in boldface. We think that these cases can't satisfy the Condition for Theorem 1. The experiment has been conducted with SMO [17] as the base learner. While SVM is known to meet the condition in Theorem 1, SVM takes too much time and therefore SMO, the approximately implemented one, was actually used.

Table 1. Training accuracy of SMO with linear kernel for each splitting level

Data Set	Level 1	Level 2	Level3
XOR shape	54.00	70.00	72.00
Balance-Scale	87.68	89.76	**89.44**
Glass	60.75	68.22	69.16
Ionosphere	91.45	92.88	**92.31**
Iris	96.67	97.33	98.67
Pima	77.44	77.71	**75.49**
SatImage	87.49	87.64	88.82
Segement	92.47	93.27	94.47
Spambase	90.76	91.28	92.46
Vehicle	76.24	79.55	84.16
Waveform	87.68	87.80	87.90
Wdbc	98.24	**97.89**	**97.72**

Now, the second issue needs to be addressed. In RBE, it is easier to classify as it goes downwards to the lower nodes because as it goes through the splitting the class distribution of the data becomes simplified. Therefore, as proved in Theorem 1, it is obvious that classifiers in the leaf nodes have fewer errors upon training data than the classifiers in the upper nodes do and that is proved by the results from most experiments. According to these results it is expected that deciding the class only with the classifiers in the leaf nodes will be more efficient when the pattern needed to be classified belongs to a specific area. Nevertheless the issue that the classifiers in the upper nodes supplement the classifiers in the lower nodes will be approached from the following two points of view.

First, even though the classifiers in the lower nodes have less errors upon the training data, not all the instances that are correctly classified by the classifiers in the upper nodes are always correctly classified by the classifiers in the lower nodes as well. This is related to the diversity that each classifier has errors in different parts of the feature space. This is formally represented and summarized in Theorem 2.

Theorem 2] Assume that h_i is the hypothesis obtained from a given data D_i with a learning algorithm L. And assume that for data $D_0 = \{\dots, (X_j, y_j), \dots\}$, ($j = 1, \dots, m$) and split data D_1, D_2 by halving D_0 based on a feature, hypotheses h_0, h_1, h_2 are produced by L, respectively. Then, for every X_j such that $X_j \in D_0$, $h_0(X_j) = y_j$, ($j = 1, \dots, m$), it is not always true that for $X_j \in D_1$, $h_1(X_j) = y_j$ and for $X_j \in D_2$, $h_2(X_j) = y_j$.

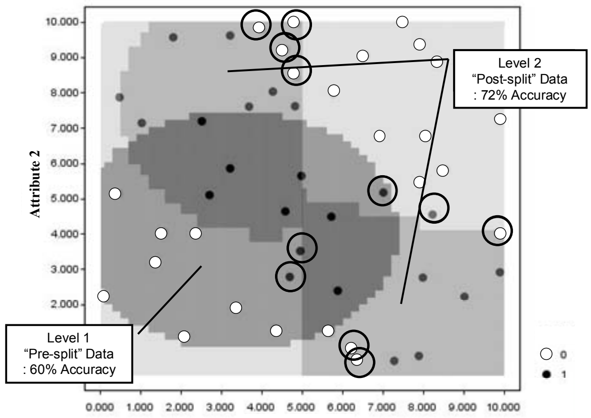

Figure 4. The decision boundary by Naïve Bayes in the upper node and lower nodes

Proof] Figure 4 is about the data following XOR shape in Table 1, which satisfies Theorem 1. This shows that there are certain instances which Naïve Bayes classifiers after the splitting are not able to classify while the classifier before the splitting classifies correctly. (The instances are marked by a circle.) ■

Second, even though it has high fitting upon the training data when this is classified with the classifiers in leaf nodes, it does not guarantee high efficiency when a new data pattern enters. That is because there can be difficulties in generating classifiers due to the lack of training data and the possibility of over-fitting. Over-fitting, a frequently presented problem in the field of machine learning, means it excessively fits into the training data and thus incurs poor generalization capability. It can be caused by the classifier which relies not on the distribution of the whole data but only on the regionally split area.

It is the classifiers in the upper nodes that prevent this in RBE. Such classifiers can lower the risk of over-fitting because they are not specialized to only a limited region, but they are generated in a bid to reflect the entire distribution of data.

In Section 1, we suggested that there should be diversity among classifiers generated from base learner [13]. From this point of view the learning of RBE guarantees the diversity between the classifiers. This connects up the second issue that the classifiers in the upper nodes supplement the classifiers in the lower nodes. Because RBE only deals with a region of the split data, each classifier learns different data. So, even though the learning algorithm creates a stable classifier with low variance, it can generate the classifier with relatively higher variance than other ensemble methods. This is because data created in RBE learning are largely different from one another, and thus the diversity is anticipated.

On the other hand, the classifier learned in this way might suffer from low generalization since it is trained only with partial data. However, the accuracy of the classifier also can be expected to be high because each classifier in RBE are specialists in its territory (i.e. classifies patterns in each region well). This is again connected to the first issue that as the splitting process goes on the accuracy of the classifier improves.

4. Experiments

4.1. Experimental Setup

In order to verify RBE to be a method which operates independently of the learning algorithm, we used Naïve Bayes, the decision tree by C4.5 [20], and SMO [17] with linear kernel algorithms as a base learner. 11 data sets from the UCI Machine Learning Repository [2] were used in our experiments.

In order to compare the performance of RBE, single classifier of base learner used for generating RBE, ensemble classifier with 7 classifiers generated by Adaboost.M1 [11], an algorithm embodied boosting, ensemble classifier with 7 classifiers generated by the bagging method, and RBC (mentioned in the Section 3) are compared in terms of classification accuracy. For fair comparison boosting and bagging also generated 7 classifiers because we generated 7 classifiers by learning RBE and splitting the data into three levels. Weka3 [22], open data mining software, is used for the base learning algorithms and our own programs are added for other ensemble methods.

Table 2. The accuracy comparison of RBE and other methods (Naïve Bayes Classifier)

Dataset	NB	vs RBE		Ada-7	vs RBE		Bagging	vs RBE		RBC	vs RBE	
BALANCE-SCALE	**89.92**	88.63		**89.76**	88.63		**89.60**	88.63		88.00	**88.63**	
GLASS	47.45	**56.32**	0	47.92	**56.32**	0	48.85	**56.32**	0	**61.06**	56.32	X
IONOSPHERE	82.91	**89.75**	0	**91.76**	89.75	X	82.91	**89.75**	0	89.21	**89.75**	
IRIS	96.00	**96.67**		96.67	96.67		95.33	**96.67**		95.33	**96.67**	
PIMA	75.62	**76.15**		75.88	**76.15**		75.75	**76.15**		72.37	**76.15**	0
SATIMAGE	79.60	**80.75**	0	79.85	**80.75**		79.70	**80.75**		78.30	**80.75**	0
SEGMENT	77.04	**89.63**	0	77.04	**89.63**	0	76.67	**89.63**	0	89.51	**89.63**	
SPAMBASE	79.50	**79.87**		79.50	**79.87**		**79.90**	79.87		79.39	**79.87**	
VEHICLE	46.28	**69.83**	0	46.28	**69.83**	0	46.52	**69.83**	0	67.21	**69.83**	0
WAVEFORM	80.02	**82.66**	0	80.02	**82.66**	0	80.06	**82.66**	0	80.68	**82.66**	0
WDBC	93.33	**93.50**		**95.96**	93.50	X	92.98	**93.50**		**94.03**	93.50	

Table 3. The accuracy comparison of RBE and other methods (SMO)

Dataset	SMO	vs RBE		Ada-7	vs RBE		Bagging	vs RBE		RBC	vs RBE	
BALANCE-SCALE	87.52	**89.28**	0	87.52	**89.28**	0	87.84	**89.28**	0	88.16	**89.28**	
GLASS	53.94	**61.51**	0	58.14	**61.51**	0	53.92	**61.51**	0	59.24	**61.51**	0
IONOSPHERE	88.03	**88.91**		87.78	**88.91**		87.47	**88.91**	0	87.77	**88.91**	
IRIS	**96.67**	95.33		**98.00**	95.33	X	**96.00**	95.33		**96.00**	95.33	
PIMA	**77.19**	76.81		**77.19**	76.81		76.80	**76.81**		74.19	**76.81**	0
SATIMAGE	85.05	**86.40**	0	85.30	**86.40**	0	84.95	**86.40**	0	86.15	**86.40**	
SEGMENT	**92.72**	92.59		**92.72**	92.59		**93.09**	92.59		**93.46**	92.59	
SPAMBASE	90.39	**91.87**	0	90.83	**91.87**		91.14	**91.87**		91.76	**91.87**	
VEHICLE	74.55	**77.16**	0	74.43	**77.16**	0	73.95	**77.16**	0	**77.40**	77.16	
WAVEFORM	**86.60**	86.56		**86.60**	86.56		86.34	**86.56**		85.74	**86.56**	
WDBC	**97.89**	97.54		**97.89**	97.54		**97.89**	97.54		95.97	**97.54**	0

Table 4. The accuracy comparison of RBE and other methods (C4.5)

Dataset	C4.5	vs RBE		Ada-7	vs RBE		Bagging	vs RBE		RBC	vs RBE	
BALANCE-SCALE	77.62	**79.37**	0	**79.54**	79.37		**82.26**	79.37	X	79.05	**79.37**	
GLASS	69.39	**70.82**	0	**74.70**	70.82	X	**74.70**	70.82	X	65.71	**70.82**	0
IONOSPHERE	88.05	**90.34**	0	**92.62**	90.34	X	**90.63**	90.34		88.06	**90.34**	0
IRIS	94.67	94.67		94.00	**94.67**		**95.33**	94.67		93.33	**94.67**	
PIMA	73.53	**74.57**		73.14	**74.57**	0	**76.40**	74.57	X	72.09	**74.57**	0
SATIMAGE	85.20	**87.95**	0	**88.20**	87.95		**88.90**	87.95		86.05	**87.95**	0
SEGMENT	96.17	**97.04**		**97.90**	97.04		96.17	**97.04**		95.80	**97.04**	
SPAMBASE	93.05	**94.31**		**95.26**	94.31		**94.35**	94.31		93.00	**94.31**	
VEHICLE	**75.02**	74.79		**76.80**	74.79	X	**76.21**	74.79		73.37	**74.79**	
WAVEFORM	75.36	**77.18**	0	**81.56**	77.18	X	**80.78**	77.18	X	75.10	**77.18**	0
WDBC	92.80	**94.03**		**96.49**	94.03	X	**94.56**	94.03		93.68	**94.03**	

4.2. Results

Table 2, 3 and 4 are the comparisons of accuracy between single classifier, other ensemble methods (bagging, boosting), and RBC and RBE classified by each base learner. Numbers in boldface represent higher accuracy. The O mark in RBE column means a significantly higher accuracy, and the X mark means a significantly lower accuracy.

In the comparison with single classifier, RBE exhibit higher accuracy regardless of the base learner. That means RBE plays its own role as a good ensemble classifier independent of the base learner.

As mentioned previously, boosting used AdaBoost.M1 algorithm which is applied also upon multiple class problems, and boosting and bagging both generated 7 classifiers. As seen in the tables, RBE was generally more accurate than other ensemble methods when Naïve Bayes or SVM are adopted as the main learning algorithm. For some data RBE was even more accurate than other ensemble approaches, and it hardly was less accurate than others in other data. With C4.5 as the base learner, ensemble method of boosting and bagging was generally more efficient than RBE. This is because the performance of the single classifier improved more conspicuously in boosting and bagging than RBE. As pointed in Section 2, boosting and bagging cause a leap of performance in unstable algorithms as the decision tree. We found that RBE was more accurate with stable algorithms such as Naïve Bayes or SVM and other existing ensemble method (e.g. boosting and bagging) were better with unstable algorithms like C4.5.

In addition, as seen in the tables, RBE was more accurate in most data sets than RBC regardless of the base learner, and the difference of the performance between those two was remarkably big in some data. This result experimentally supports the rationale behind ensemble learning in RBE.

5. Conclusion and future works

We have presented the ensembles of region based classifiers (RBE) as a new ensemble method. Compared to the existing methods such as bagging and boosting, our method demonstrated better performance, particularly when a stable algorithm is used as the base learner.

RBE generates an ensemble in the form of a tree of which the node represents an individual classifier. Each classifier is in charge of a specific region of the feature space, and trained by the patterns in the region only, which both entails diversity among the classifiers in the ensemble and guarantees more accurate classification in lower nodes as well. The criterion used in splitting the data and defining regions is the information gain. A feature with the maximum information gain is chosen to split the data. The classification of a pattern is by the majority voting among the nodes that contain the pattern in their region, giving preference to the leaf node in case of ties.

The feasibility of RBE is verified both theoretically with proven theorems and experimentally with real-world data sets. It outperformed a single classifier and a simple region based classifier. It either outperformed other ensemble methods or was comparable to them for different base learners.

Our current experiments are with three levels in RBE (i.e. use of complete binary tree with depth 3) and with seven classifiers in bagging and boosting. The number of levels is determined arbitrarily, and it needs to be examined further to develop a well-defined method to determine the level. Also, our algorithm needs to be expanded to handle various types of features (e.g. discrete values). In addition, other methods to combine the classifiers can be developed beyond the simple voting. These are left as topics for future research.

6. References

[1] Bauer, E., & Kohavi, R., "An Empirical Comparison of Voting Classification Algorithm: Bagging, Boosting, and Variants", Machine Learning, 36(1-2), 1999, pp. 105-142.

[2] Blake, C., & Merz, C., UCI Repository of Machine Learning Database, http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.

[3] Breiman, L., "Bias, Variance, and Arcing Classifiers", Technical Report TR, 460, UC Berkeley, 1996.

[4] Breiman, L., "Bagging Predictors. Machine Learning", 24(2), 1996, pp. 123-140.

[5] Buciu, I., Kotropoulos, C., & Pitas, I., "Combining Support Vector Machines for Accurate Face Detection", In Pr oc. of ICIP'01, 1, 2001, pp. 1054-1057.

[6] Dietterich, T., "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization", Machine Learning, 40(2), 2000, pp. 139-157.

[7] Dietterich, T., "Ensemble method in Machine learning", In J. Kittler and F. Roli (Ed.), First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science, 2000, pp. 1-15.

[8] Dietterich, T., "Ensemble Learning", In The Handbook of Brain Theory and Neural Networks, Second edition. The MIT Press, 2002, pp. 405-408.

[9] Evgeniou, T., Perez-Breva, L., Pontil, M., & Poggio, T., "Bound on the Generalization Performance of Kernel Machine Ensembles", In Proc. of the 17th International Conference on Machine Learning, 2000, pp. 271-278.

[10] Freund, Y., & Schapire, R., "Experiments with a new boosting algorithm", In Proc. of the 13th International Conference on Machine Learning, 1996, pp.148-156.

[11] Freund, Y., & Schapire, R., "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting", Journal of Computer and System Science, 55 , 1997, pp. 119-139.

[12] Friedman, J., Hastie, T., & Tibshirani, R., "Additive Logistic Regression: a Statistical View of Boosting", Annals of Statistics, 28(2), 2000, pp. 337-374.

[13] Hansen, L., & Salamon, P., "Neural Network Ensembles", IEEE Transaction on Pattern Analysis and Machine Intelligence, 12, 1990, pp. 993-1001.

[14] Kim, H., "Construction Support Vector Machine Ensemble", Pattern Recognition, 36, 2003, pp. 2757-2767.

[15] Kuncheva, L., & Whitaker, C., "Measures of diversity in classifier ensembles", Machine Learning, 51, 2003, pp. 181-207.

[16] Opitz, D., & Maclin, R., "Popular Ensemble Methods: An Empirical Study", Journal of Artificial Intelligence Research, 11, 1999, pp. 169-198.

[17] Platt, J., "Fast Training of Support Vector Machines using Sequential Minimal Optimization", chapter 12, The MIT Press, 1999, pp. 185-208.

[18] Quinlan, J., "Induction of Decision Tree", Machine Learning, 1(1), 1986, pp. 81-106

[19] Quinlan, J., "Bagging, Boosting, and C4.5", In Proc. of the Thirteenth National Conference on Artificial Intelligence, 1996, pp. 725-730.

[20] Quinlan, J., C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.

[21] Valentini, G., Muselli, M., & Ruffino, F., "Bagged Ensembles of SVMs for Gene Expression Data Analysis", The IEEE-INNS-ENNS International Joint Conference on Neural Networks, 2003, pp. 1844-1849.

[22] Witten, I., & Frank, E., Data Mining: Practical Machine Learning Tools and Techniques with Java Implementation, Second edition. Morgan Kaufmann, 2005.