Classifier Ensemble Using a Heuristic Learning with Sparsity and Diversity

Xu-Cheng Yin[1,⋆], Kaizhu Huang[2], Hong-Wei Hao[2],
Khalid Iqbal[1], and Zhi-Bin Wang[1]

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
[2] Institute of Automation, Chinese Academy of Sciences, Beijing 100090, China
xuchengyin@ustb.edu.cn, kzhuang@nlpr.ia.ac.cn,
hongwei.hao@ia.ac.cn, kik.ustb@gmail.com, wzb1818@yahoo.cn

Abstract. Classifier ensemble has been intensively studied with the aim of overcoming the limitations of individual classifier components in two prevalent directions, i.e., to diversely generate classifier components, and to sparsely combine multiple classifiers. Currently, most approaches are emphasized only on sparsity or on diversity. In this paper, we investigated classifier ensemble with learning both sparsity and diversity using a heuristic method. We formulated the sparsity and diversity learning problem in a general mathematical framework which is beneficial for learning sparsity and diversity while grouping classifiers. Moreover, we proposed a practical approach based on the genetic algorithm for the optimization process. In order to conveniently evaluate the diversity of component classifiers, we introduced the diversity contribution ability to select proper classifier components and evolve classifier weights. Experimental results on several UCI classification data sets confirm that our approach has a promising sparseness and the generalization performance.

Keywords: Classifier ensemble, Sparsity learning, Diversity learning, Bagging.

1 Introduction

An ensemble of multiple classifiers has been intensively studied and widely considered to be an effective technique for overcoming the limitations of individual classifiers' accuracy and stability [1–4]. Classifiers differing in feature representation, architecture, learning, or training data exhibit complementary behavior and the fusion of their decisions can yield higher performance than the best individual classifier. Generally speaking, besides the accuracies of classifier components, the performance relies on the diversity of the classifier components, and the combining strategy. Consequently, the research efforts in classifier ensemble have focused on two directions: how to generate diverse classifiers, and how to combine available multiple classifiers. In classifier ensemble, diversity learning is performed in two approaches such as seeking implicit and explicit diversity [5]. The common way for the prior approach is to train individual classifiers on different training sets, e.g., Bagging [6], Boosting [7, 8], SVM ensemble [9]

[⋆] Corresponding author.

T. Huang et al. (Eds.): ICONIP 2012, Part II, LNCS 7664, pp. 100–107, 2012.

and Random Forests [10]. As in the latter approach, the general way is to train multiple classifiers by using different classifier architectures or different feature sets [2, 11, 12]. Recently, Yu et al. proposed the diversity regularized machine, which efficiently generates an ensemble of assorted support vector machines [13]. Li et al. proposed the diversity regularized ensemble pruning method with PAC analysis [14].

In the combination learning, multiple classifiers with proper combination of rules or learning methods are grouped. Numerous methods such as an average, linear or nonlinear combination rules are employed [3, 11]. Given a number of available component classifiers, most conventional approaches employ all of these classifiers to constitute an ensemble. In the literature, many researchers suggested that ensemble of some parts of the available component classifiers may be better than ensemble as a whole. This leads to the sparse ensemble or pruned ensemble for the combination of multiple classifiers [15, 16], [17–20]. The sparsity learning seeks a sparse weight vector for combining the outputs of all classifiers. In general, a sparse model representation is expected to improve the generalization performance and computational efficiency. As described above, the diversity learning and the sparsity learning for classifier ensemble have different purposes and algorithmic treatments. Therefore, It is more rational for classifier ensemble with both sparsity and diversity learning strategies. With a similar idea, Chen and Yao et al. analyzed diversity and regularization in neural network ensembles for balancing diversity, regularization and accuracy of multi-objectives [21, 22]. Their methods were specifically designed for component classifier training and combination with neural network ensembles.

In this paper, considering a general classifier ensemble with numerous available component classifiers, we formulated the sparsity and diversity learning problem in a general mathematical framework with an optimized equation. Moreover, we proposed a practical approach based on the genetic algorithm (GA) with a direct evaluation of diversity for the optimization process. The rest of this paper is organized as follows. In Section 2, our sparsity and diversity learning for classifier ensemble is presented in details. Several comparative experiments with UCI classification data sets are demonstrated in Section 3. Finally, conclusion is shown in Section 4.

2 Sparsity and Diversity Learning

2.1 Problem Statement for Classifier Ensemble

In classifier ensemble, each instance a is associated with a label y. To classify one instance a into K classes $\{\omega_1, \ldots, \omega_K\}$, assume that we have N different classifiers (classification hypotheses) $\{h_1, \ldots, h_N\}$, each using a certain feature vector for a. On an input instance a, each classifier h_n outputs discriminant measures $x^n = h_n(a)$. With all classifiers we get $x = [x^1 \ldots x^N]^T$. We focus on the weighted combination. The combined similarity measures are computed by $H(a) = \sum_{n=1}^{N} w^n x^n = w^T x$, where w^n is the weight for the n^{th} classifier, and $w = [w^1 \ldots w^N]^T$.

Formally, given a sample set $\{(a_m, y_m)\}_{m=1}^{M}$, with N different classifiers, we have $\{(x_m, y_m)\}_{m=1}^{M}$, where x_m is a vector and $x_m = [x_m^1 \ldots x_m^N]^T$. The learning focuses on finding w for with which the empirical loss is small. In our work, we used the least squares loss. The classifier weights are estimated on the sample data set to the least squares loss. The general goal is to learn w with $\min_w \sum_{m=1}^{M} \frac{1}{2}(w^T x_m - y_m)^2$. With $X = [x_1, x_2, \ldots, x_M] \in \Re^{N \times M}$ and $y = [y_1, y_2, \ldots, y_M]^T \in \Re^M$, the classifier weights w are learned by solving the least squares (LS) problem,

$$\min_w \parallel X^T w - y \parallel_2 \qquad \text{s.t.} \quad w^n \geq 0 \tag{1}$$

2.2 Sparsity and Diversity

Sparsity Learning. In the literature of classifier ensemble, many researchers suggested that ensemble some parts of the available component classifiers may be better than entire ones. This leads to the sparsity learning for classifier combination. The classifier weights w in (1) are learned by incorporating the l_1-norm regularization,

$$\min_w \parallel X^T w - y \parallel_2 + \lambda \parallel w \parallel_1 \qquad \text{s.t.} \quad w^n \geq 0 \tag{2}$$

where λ is the l_1-norm regularization parameter.

Sparsity and Diversity Learning. To learn sparsity and diversity in classifier ensemble, the targeted learning should be set with one diversity loss in the regularization process,

$$\min_w \parallel X^T w - y \parallel_2 + \lambda \parallel w \parallel_1 + \beta s(w) \qquad \text{s.t.} \quad w^n \geq 0 \tag{3}$$

where, $0 \leq s(w) \leq 1$ is one measure inversely proportional to the average diversity of all component classifiers, and β is the diversity penalty parameter. The classifier ensemble will have more stronger diversity when $s(w)$ is less.

For robustness, we adopt the Yule's Q statistic diversity measure for the average diversity measure [4]. Suppose the Yule's Q statistic for two classifiers, h_{n_1} and h_{n_2}, is Q_{n_1,n_2}. For statistically independent classifiers, the expectation of Q_{n_1,n_2} is 0. Q varies between -1 and 1. Classifiers that tend to recognize the same objects correctly will have positive values of Q, and those which commit errors on different objects will render Q negative. That is to say, in classifier ensemble, if Q is small then the diversity of the ensemble is large. In this way, the average diversity measure of the ensemble, $g(w)$, is calculated with Yule's Q statistic by

$$g(w) = \frac{2}{N_1(N_1 - 1)} \sum_{n_1=1}^{N_1-1} \sum_{n_2=n_1+1}^{N_1} \frac{1 + Q_{n_1,n_2}}{2} \tag{4}$$

Then, the diversity semantic loss, $s(w)$ in (3), is simply calculated as

$$s(w) = g(w) \tag{5}$$

Obviously, $0 \leq s(w) \leq 1$.

2.3 Heuristic Learning Algorithm

The conventional optimization techniques of the l_1-norm regularization in (2) depend on the gradient computation with w. In sparsity and diversity learning with (3), the "diversity" semantic loss is indirectly calculated from w with (4). As a result, we can not directly compute the gradient of w from the "diversity" semantic loss with current l_1-norm regularization techniques.

Alternatively, we deal with the sparsity and diversity learning (3) in two steps which incorporate the l_1-norm regularization and the "diversity" semantic of ensemble classifiers in a heuristic and iterative way (see Figure 1). Firstly, the l_1-norm regularization is performed for the left part of (3), i.e. (2), and a sparse w is learned. Then, based on the learned w of the former step, we calculate the "diversity" measures of classifiers; moreover, we remove some included classifiers and add some excluded classifiers both with a probability proportional to **the diversity contribution ability**. The former two steps are repeated until some criterions (e.g., the maximum of iterations, the rate of change of the diversity measure) are satisfied. According to (3), the removal of included-classifiers and the addition of excluded classifiers will decrease the sparsity and diversity loss in (3). That is to say, in the second step, the new weight vector w' with classifier removal and addition will satisfied the following condition,

$$\parallel X^T w' - y \parallel_2 + \lambda \parallel w' \parallel_1 + \beta s(w') < \parallel X^T w - y \parallel_2 + \lambda \parallel w \parallel_1 + \beta s(w) \qquad (6)$$

The diversity contribution ability of a classifier in the ensemble is defined as follows. There are N_1 included component classifiers in the ensemble. The average diversity measure, $g(w)$ in (3), is calculated with Yule's Q statistic by (7),

$$f(w) = \sum_{n_1=1}^{N_1-1} \sum_{n_2=n_1+1}^{N_1} \frac{1+Q_{n_1,n_2}}{2} \qquad (7)$$

The average diversity measure without one included classifier, $h_n (n \in \{1, 2, \ldots, N_1\})$, can be calculated with

$$f_{/n}(w) = \sum_{n_1=1}^{N_1-1} \sum_{n_2=n_1+1}^{N_1} \frac{1+Q_{n_1,n_2}}{2} - \sum_{n_1=1, n_1 \neq n}^{N_1} \frac{1+Q_{n_1,n}}{2}$$

The diversity contribution ability of the included classifier h_n is represented by

$$\triangle f_{/n}(w) = 1 - \frac{1}{N_1 - 1}(f(w) - f_{/n}(w)) = 1 - \frac{1}{N_1 - 1} \sum_{n_1=1, n_1 \neq n}^{N_1} \frac{1+Q_{n_1,n}}{2} \qquad (8)$$

In the similar way, for **the diversity contribution ability** of the excluded classifier nh_l (i.e., its weight in (3) is equal to zero), we compute the diversity measure both with the N_1 included component classifiers and the excluded classifier nh_l,

$$f_{\backslash l}(w) = \sum_{n_1=1}^{N_1-1} \sum_{n_2=n_1+1}^{N_1} \frac{1+Q_{n_1,n_2}}{2} + \sum_{n_1=1}^{N_1} \frac{1+Q_{n_1,l}}{2}$$

```
INPUT: Training set $\{(x_m, y_m)\}_{m=1}^{M}$, regularization $\lambda$ and $\beta$ in (3), $T$ iterations
INITIALIZE: Calculate $X \in \Re^{M \times N}$ and $y$ , and set the weight vector $w$ with random values
FOR $t = 1, \ldots, T$
    (1) Learn sparsity weights $w$ via the $l_1$-norm regularization in (2)
    (2) Learn sparsity and diversity weights from the sparse weights $w$
        (2.1) Calculate the diversity contribution ability $\triangle f_{/n}(w)$ in (8)
        (2.2) Calculate the diversity contribution ability $\triangle f_{\setminus l}(w)$ in (9)
        (2.3) Sort $\triangle f_{/1}(w) \leq \triangle f_{/2}(w) \leq \cdots \leq \triangle f_{/N_1}(w)$
              FOR $n = 1, \ldots, N_{thres}$
                  Remove $h_n$ with a probability inversely proportional to $\triangle f_{/n}(w)$,
                      where $w^n < w_{thres}$
        (2.4) Sort $\triangle f_{\setminus 1}(w) \geq \triangle f_{\setminus 2}(w) \geq \cdots \geq \triangle f_{\setminus L}(w)$
              FOR $l = 1, \ldots, L_{thres}$
                  Remove $nh_l$ with a probability proportional to $\triangle f_{\setminus l}(w)$
        (2.5) Evolve the weights of added classifiers with (3) using the genetic algorithm
        (2.6) Update $w$
OUTPUT: The sparsity and diversity weights $w^*$
```

Fig. 1. Algorithm for the sparsity and diversity learning with a heuristic approach

Obviously, the number of excluded classifiers is $L = N - N_1$.

And **the diversity contribution ability** of the excluded classifier nh_l is

$$\triangle f_{\setminus l}(w) = 1 - \frac{1}{N_1}(f_{\setminus l}(w) - f(w)) = 1 - \frac{1}{N_1} \sum_{n_1=1}^{N_1} \frac{1 + Q_{n_1,l}}{2} \qquad (9)$$

which represents the increase of diversity while adding one excluded classifier.

Let's look around (6) and (4), where the changed weight vector w' is computed based on permutation and combination of all component classifiers, and moreover with varied weights of the excluded classifier addition. It is obvious that the computational cost for removing included-classifiers, adding excluded classifier, and computing the new weights are too extensive.

Consequently, we present a practical approach in the second step (see Figure 1) to find out the component classifier that should be excluded or included from the ensemble. The basic idea of this approach is a heuristic, i.e., assuming each included classifier can be removed with a probability inversely proportional to its diversity contribution ability. Also, each excluded classifier can be added with a probability proportional to its diversity contribution ability. Moreover, each excluded classifier for adding can be assigned a weight that could characterize the fitness of their inclusion in the ensemble. Thus, the included and excluded component classifiers with the new weights decrease the sparsity and diversity loss in (3) (satisfied (6)). With the help of (3) and (6), it could be viewed as defining an optimization problem. We develop a genetic algorithm based method for this optimization problem in Step (2) (see Figure 1). After Step (1) with the l_1-norm regularization in (2), a sparsity weight vector with included and excluded classifiers is available. Firstly, included and excluded classifiers are removed and added respectively according to their diversity contribution abilities (From Step (2.1)-(2.4) in

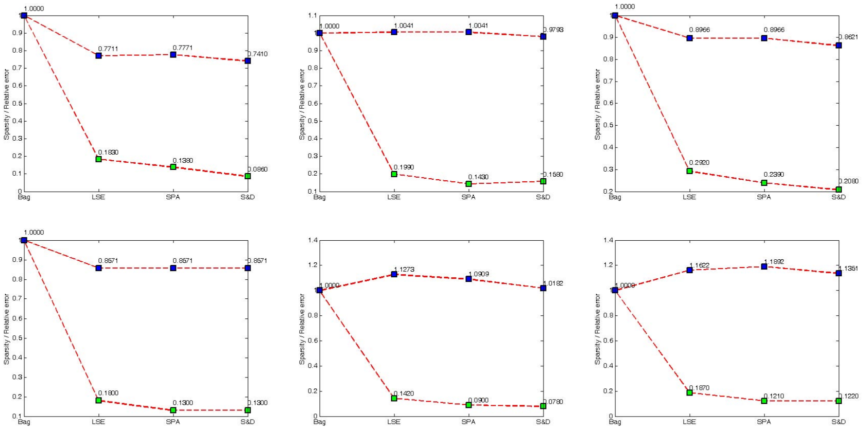

Fig. 2. The experimental results of sparsity (green grids) and relative errors (blue grids), from left/top to right/bottom: Chess, Credit, Ionosphere, Sick, Cancer Wisconsin, and Vote.

Figure 1). Subsequently, our approach employs genetic algorithm to evolve the weights of added classifiers (excluded classifiers).

In the second step of the algorithm in Figure 1, we add excluded classifiers with high diversity measures and remove included classifiers with low diversity measures in the ensemble in a sequence, where one excluded classifier with a higher diversity contribution ability will be added at first with a higher probability. In such a way, we hope the final ensemble will have a large diversity semantic.

3 Experiments

Six data sets (the Chess, Credit (German), Ionosphere, Sick, Breast Cancer Wisconsin, and Vote) for classification with 2 classes from UCI machine learning repository were used in our experiments, each of which contained at least 350 instances. In our experiments, 10-fold cross validation for the data sets was performed. We compared 4 ensemble methods: Bagging (Bag), LS Estimation Combination (LSE), Sparsity Learning (SPA), Sparsity and Diversity Learning (S&D). Each ensemble contained 100 neural network classifier components (with Back-propagation in Matlab), which were same to the components in Bagging.

The reported results were the average outcomes. The sparsities and relative errors are shown in Figure 2. The sparsity is the percentage of used component classifiers (which are with nonzero weights) in the ensemble. The relative error is the ratio against the error of the baseline ensemble (Bagging).

We compared the SPArsity learning algorithm (SPA) to the baseline algorithm Bagging (Bag) as shown in Figure 2. SPA has a better performance for the Chess, Ionosphere, and Sick datasets respectively. However, SPA has a worse performance for other data sets such as the credit, Cancer Wisconsin and Vote respectively. Furthermore, Least

Table 1. Comparison of test errors (%) of (S&D), Bag, LSE, SPA, and AB

Data sets	**S&D**	**Bag**	**LSE**	**SPA**	**AB**
Chess	**3.84**	5.19	4.00	4.03	4.06
Credit	**23.70**	24.20	24.30	24.30	26.50
Ionosphere	**7.14**	8.29	7.43	7.43	11.71
Sick	**1.85**	2.16	**1.85**	**1.85**	2.57
Cancer	8.00	**7.86**	8.86	8.57	**7.86**
Vote	9.77	**8.60**	10.00	10.23	9.53
Average rank	**1.67**	2.83	2.83	3.00	3.67

Squares Estimation method (LSE) and SPA have a similar performance. To some extent, these results show that the pure pursuit of sparsity or the focus on accuracy for combination will have little effect in classifier ensemble.

In contrast, our Sparsity and Diversity (S&D) learning approach has more impressive results. Compared to the sparsity learning (SPA), the relative errors of S&D are improved by 0.0361, 0.0248, 0.0345, 0.0000, 0.0727, and 0.0541 for all the data sets respectively as shown in Figure 2. At the same time, the sparsity performance of our S&D learning approach is not only competitive but also even better than the existing sparsity learning. In most cases, the S&D has a better performance compared to the baseline (Bag). As a result, we can conclude that our sparsity and diversity learning approach has taken more advantages for classifier ensemble using sparsity and diversity.

We also compared our sparsity and diversity learning method to the AdaBoost algorithm in [7] with 100 neural network components. The average classification errors for our (S&D) learning method, LSE, SPA, Bag, and AdaBoost (AB) are shown in Table 1. On each data set, we assign ranks to methods. The best method receives the rank 1, and the worst the rank 5.

From these experimental results in Table 1, we could find that our S&D learning method has a competitive performance with the Bagging (Bag) and Adaboost (AB) combination methods. In these five ensemble methods, our S&D learning method have the highest average ranks (1.67). Especially, in comparison with AdaBoost, our S&D learning method has improved the performance (in favor of decreasing the classification error) by 0.22%, 2.80%, 4.57%, and 0.72% using the Chess, Credit, Ionosphere and Sick datasets respectively. Moreover, our method has a very low sparsity, and only uses a small number (from 7% to 21%) of available classifiers in all experimental data sets.

4 Conclusion

Classifier ensemble is widely considered to be an effective technique for improving accuracy and stability for a classification system with numerous classifier components. We proposed a mathematical framework of classifier ensemble with a sparsity and diversity learning strategy, which captured advantages of features of both learning strategies. This framework can be implemented by an optimization procedure which is embedded with one diversity semantic loss incorporating the l_1-norm regularization. We also proposed a practical approach based on the genetic algorithm for the optimization process.

Experimental results on 2-class UCI data sets confirmed the validity of our classifier ensemble method with sparsity and diversity learning, which has a promising sparseness and generalization performance.

Acknowledgments. The research was partly supported by National Natural Science Foundation of China (61105018, 61175020), and the R&D Special Fund for Public Welfare Industry (Meteorology) of China (GYHY201106039, GYHY201106047).

References

1. Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
2. Ho, T., Hull, J., Srihari, S.: Decision Combination in Multiple Classifier Systems. IEEE T-PAMI 16(1), 66–75 (1994)
3. Kittler, J., Hatef, M., Duin, R., Matas, J.: On Combining Classifiers. IEEE T-PAMI 20(3), 226–239 (1998)
4. Kuncheva, L., Whitaker, C.: Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. Mach. Learn. 51(2), 181–207 (2003)
5. Tang, E., Suganthan, P., Yao, X.: An Analysis of Diversity Measures. Mach. Learn. 1, 247–271 (2006)
6. Breiman, L.: Bagging Predictors. Mach. Learn. 24(1), 123–140 (1996)
7. Freund, Y., Schapire, R.: A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting. J. Comput. Syst. Sci. 55(1), 119–139 (1997)
8. Schapire, R.: The Strength of Weak Learnability. Mach. Learn. 5(2), 197–227 (1990)
9. Kim, H.C., Pang, S., Je, H.M., Kim, D.: Constructing Support Vector Machine Ensemble. Pattern Recogn. 36(12), 2757–2767 (2003)
10. Breiman, L.: Random Forests. Mach. Learn. 45(1), 15–32 (2001)
11. Liu, C.L.: Classifier Combination Based on Confidence Transformation. Pattern Recogn. 38(1), 11–28 (2005)
12. Yin, X.C., Liu, C.P., Han, Z.: Feature Combination Using Boosting. Pattern Recogn. Lett. 26(13), 2195–2205 (2005)
13. Yu, Y., Li, Y.F., Zhou, Z.H.: Diversity Regularized Machine. In: IJCAI, pp. 1603–1608 (2011)
14. Li, N., Yu, Y., Zhou, Z.-H.: Diversity Regularized Ensemble Pruning. In: Flach, P.A., De Bie, T., Cristianini, N. (eds.) ECML PKDD 2012, Part I. LNCS, vol. 7523, pp. 330–345. Springer, Heidelberg (2012)
15. Grove, A., Schuurmans, D.: Boosting in the Limit: Maximizing the Margin of Learned Ensembles. In: AAAI, pp. 692–699 (1998)
16. Margineantu, D., Dietterich, T.: Pruning Adaptive Boosting. In: ICML, pp. 211–218 (1997)
17. Martinez-Munoz, G., Hernandez-Lobato, D., Suarez, A.: An Analysis of Ensemble Pruning Techniques Based on Ordered Aggregation. IEEE T-PAMI 31(2), 245–259 (2009)
18. Yao, X., Liu, Y.: Making Use of Population Information in Evolutionary Artificial Neural Networks. IEEE T-SMC Part B 28(3), 417–425 (1998)
19. Zhang, L., Zhou, W.D.: Sparse Ensembles Using Weighted Combination Methods Based on Linear Programming. Pattern Recogn. 44(1), 97–106 (2011)
20. Zhou, Z.H., Wu, J.X., Tang, W.: Ensembling Neural Networks: Many Could Be Better than All? Artif. Intell. 137(1-2), 239–263 (2002)
21. Chen, H., Tino, P., Yao, X.: Predictive Ensemble Pruning by Expectation Propagation. IEEE T-TDE 21(7), 999–1013 (2009)
22. Chen, H., Yao, X.: Multiobjective Neural Network Ensembles Based on Regularized Negative Correlation Learning. IEEE T-KDE 22(12), 1738–1751 (2009)