A WEIGHTED SUBSPACE APPROACH FOR IMPROVING BAGGING PERFORMANCE

Qu-Tang Cai[†], *Chun-Yi Peng*[‡], *Chang-Shui Zhang*[†]

[†] State Key Laboratory on Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology (TNList),
Department of Automation, Tsinghua University, Beijing, China.
[‡] Microsoft Research Asia, 49 Zhichun Road, Haidian District, Beijing 100084, China

ABSTRACT

Bagging is an ensemble method that uses random resampling of a dataset to construct models. In classification scenarios, the random resampling procedure in bagging induces some classification margin over the dataset. In addition, when perform bagging in different feature subspaces, the resulting classification margins are likely to be diverse. We take into account the diversity of classification margins in feature subspaces for improving the performance of bagging. We first study the average error rate of bagging, convert our task into an optimization problem for determining some weights for feature subspaces, and then assign the weights to the subspaces via a randomized technique in classifier construction. Experimental results demonstrate that our method is able to further improve the classification accuracy of bagging, and also outperforms several other ensemble methods including AdaBoost, random forests and random subspace method.

Index Terms— Bagging, Classifier ensemble, Probabilistic methods, Classification, Optimization

1. INTRODUCTION

Bagging [1] is a procedure for building an estimator by a resampling and combining technique. In classification tasks, a bagged classifier is produced by majority voting of several base classifiers trained on bootstrap samples. In many studies, bagging decision stumps, trees or neural networks tends to reduce classification error compared with the original predictor [1, 2]. In situations with large noise, bagging performs even better [2].

One way for characterizing the strength of the resulting classifiers is by classification margin, which has been used in some previous research [3, 4]. In the procedure of bagging, the training sets for growing base classifiers are created by drawing with replacement from the original training set. Accordingly, the trained base classifiers are inherently random. In other words, trained classifiers can be treated to be drawn based on some unknown underlying probability distribution from the base classifier space. Classification margin can then be viewed as the exceedance probability of correct classifiers. In practical applications, the classification margin of bagging usually can be estimated by an out-of-bag estimation [1].

As has been observed, classifiers grown from different feature subspaces behaves diversely. This has been explored by Ho [5] to improve classification accuracy. For bagging, when the base classifiers are grown in different feature subspaces, the classification margins in different subspaces are also likely to be diverse. Thus, it is hopeful to make use of the diversity to further improve the performance of bagging. The remaining parts of this paper are organized as follows. In section 2, we analyze the relationship between the average error rate of bagging and classification margin, after introducing some necessary definitions and notations. In section 3, we propose a weighted subspace approach for improve bagging performance. In section 4, we present experimental results of our approach. Conclusions are made in section 5.

2. CLASSIFICATION MARGIN OF BAGGING

Let X be the feature space and Y be the set of class labels. Let $\mathcal{D}$ denote the dataset, and every instance in $\mathcal{D}$ is represented by a feature-label pair $(x; y)$, where $x \in X, y \in Y$. In addition, we assume that samples are generated i.i.d. from an unknown underlying distribution D over $X \times Y$. For simplicity, we only consider two-class classification problems, i.e., $Y = \{-1, +1\}$. Throughout this paper, we use $I(\cdot)$, $P(\cdot)$ and $E(\cdot)$ as the indicator function, probability function and expectation, respectively.

A classifier can be viewed as a parameterized mapping from the feature space X to Y. For example, the Fisher linear classifier for binary classification problems can be parameterized by its projection vector and a separating point. Therefore, we can write every individual classifier as a parameterized mapping $h(x; \theta)$, abbreviated by h_θ for convenience, where θ is the corresponding parameter for current classifier, and x is the input feature. Moreover, we denote the majority voting ensemble of classifier $\theta_1, \ldots, \theta_k$ by $mv(x; \theta_1, \ldots, \theta_k)$.

In bagging, the classifier parameters of the base classifiers

Supported by National 863 project(No. 2006AA10Z210).

change with the bootstrapped training sets. However, the parameters are not allowed to take arbitrary value, and must be restricted to some space of the classifier parameters, denoted by Θ. We also use the same symbol to represent the base classifier space since it does not cause additional confusion. Furthermore, by the bootstrap procedure of bagging, the classifiers built for voting can be viewed to be drawn i.i.d. according to some unknown probability distribution over Θ, and we write this distribution by ϑ. Now we introduce the definition of classification margin for parameter space Θ, which coincides with the definition of Breiman for random forests [4].

Definition (margin function): *The margin function for the classifiers in parameter space Θ is a function from $X \times Y$ to $[-1, 1]$*

$$mr(x,y) \doteq P_{\vartheta}(h(x,\theta) = y) - \max_{j \neq y, j \in Y} P_{\vartheta}(h(x,\theta) = j). \quad (1)$$

The classification margin we mention below refers to (1). When (x, y) is randomly generated, then $mr(x, y)$ is a random variable taking value in $[-1, 1]$, and possesses a probability whose cumulative distribution function (cdf) is denoted by $F_m(\cdot)$. Thus $F_m(\alpha) = P_D(\{(x, y) : mr(x, y) \leq \alpha\})$. In bagging, F_m can be empirical calculated by an out-of-bag estimation. Once $F_m(\cdot)$ is known, we can immediately calculate the average error rate of bagging by Proposition 2.1.

Proposition 2.1. *When bagging k base classifiers, the average of the ensemble error rate is $\int_{-1}^{1} B(\alpha, k) dF_m(\alpha)$, where $B(\alpha, k) \doteq \sum_{i=\lceil k/2 \rceil}^{k} \binom{k}{i} (\frac{1-\alpha}{2})^i (\frac{1+\alpha}{2})^{k-i}$, $\lceil k/2 \rceil$ represents the minimal integer not less than $k/2$, and the integral is Lebesgue-Stieltjes integral.*

Proof. The classification error rate of majority voting of classifiers $h_{\theta_1}, \ldots, h_{\theta_k}$ is $P_{(x,y)\sim D}(mv(x; \theta_1, \ldots, \theta_k) \neq y)$. The k base classifiers' parameters $\theta_1, \ldots, \theta_k$ can be viewed to be drawn i.i.d. according to some underlying distribution ϑ. For each (x, y) that $mr(x, y) = \alpha, \alpha \in [-1, 1]$, the number of classifiers in $\{h_{\theta_1}, \ldots, h_{\theta_k}\}$ that correctly classified (x, y) is then a binomial random variable with parameters k and $\frac{1+\alpha}{2}$. Thus, the probability that (x, y) is misclassified by majority voting of $h_{\theta_1}, \ldots, h_{\theta_k}$ is

$$B(\alpha, k) \doteq \sum_{i=\lceil k/2 \rceil}^{k} \binom{k}{i} (\frac{1-\alpha}{2})^i (\frac{1+\alpha}{2})^{k-i}. \quad (2)$$

With the aid of Fubini's theorem,

$$\begin{aligned}
&E_{\theta_1,\ldots,\theta_k \sim \vartheta}(P_{(x,y)\sim D}(mv(x; \theta_1, \ldots, \theta_k) \neq y)) \\
&= E_{(x,y)\sim D; \theta_1,\ldots,\theta_k \sim \vartheta}(I(mv(x; \theta_1, \ldots, \theta_k) \neq y)) \\
&= E_{(x,y)\sim D}(E_{\theta_1,\ldots,\theta_k \sim \vartheta}(I(mv(x; \theta_1, \ldots, \theta_k) \neq y) | mr(x,y))) \\
&= E_{(x,y)\sim D}(B(mr(x,y), k)) = \int_{-1}^{1} B(\alpha, k) dF_m(\alpha).
\end{aligned}$$

□

Since $\int_{-1}^{1} B(\alpha, k) dF_m(\alpha)$ can be treated as the expectation of $B(\alpha, k)$, where α is a random variable with distribution $F_m(\alpha)$, we write $\int_{-1}^{1} B(\alpha, k) dF_m(\alpha)$ as $E_D(B(\alpha, k))$ for later use.

Bagging classifiers in different feature subspaces is likely to produce different classification margins. For example, as illustrated in Fig.1, there are a number of instances whose margins are notably different from each other, and moreover, there are even a number of instances that can be easily correctly classified in one feature subspace but obscure in the other subspace. Thus, utilizing the diverse classification power in feature subspaces is promising to improve the performance of bagging.

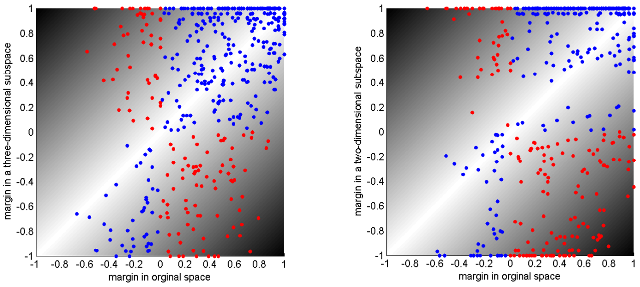

Fig. 1. Scatter plots of classification margin for bagging C4.5 classifiers on the UCI balance dataset in different feature spaces, including the original feature space (four-dimensional), a three-dimensional feature subspace and a two-dimensional feature subspace. Each point represents an instance. Red point means that one of the margins of current instance is positive while the other is negative. The lightness indicates the difference of margins in the spaces.

3. A WEIGHTED SUBSPACE APPROACH

Throughout this section, we assume that bagging can be cast in all feature subspaces, and all the classification margins have been obtained.

3.1. Combining Strategy

To improve the classification accuracy, our goal is to construct a new base classifier space based on some pre-selected feature subspaces, where the average error rate of bagging is minimized under the new distribution of classifier parameters. We will make the classification margin in the new base classifier space be a weighted combination of the classification margins in classifier spaces grown from different feature subspaces. More specifically, let the base classifier spaces be denoted by $\Theta_1, \ldots, \Theta_{\mathrm{n}}$, with their margin functions $mr_1(\cdot), \ldots, mr_n(\cdot)$ respectively, and then the new base classifier space is $\bigcup_{i=1}^{n} \Theta_{\mathrm{i}}$, where the margin function $mr(\cdot)$ is a linear combination of $mr_1(\cdot),\ldots,mr_n(\cdot)$:

$$mr(\cdot) = w_1 * mr_1(\cdot) + w_2 * mr_2(\cdot) + \cdots + w_n * mr_n(\cdot), \quad (3)$$

where w_i is the weight assigned to Θ_i. In (3), $w_i's$ can be further restricted to be nonnegative since one can reverse the output of all classifiers in Θ_i to make w_i nonnegative. In addition, $w_i's$ are made to meet the normalized condition that $\sum_{i=1}^n w_i = 1$. We use the randomized method as shown in Table 1 to achieve (3).

Table 1. Method for constructing new classifier space

For constructing each base classifier θ,

Step 1. Randomly draw one index s from $\{1,\ldots,n\}$ with $P(s=i) = w_i$.

Step 2. Draw θ randomly from Θ_s according to ϑ_s.

Proposition 3.1. *The classification margin in the new classifier space constructed as described in Table 1 is (3).*

Proof. By these two steps,

$$mr(x,y) = P(h(x;\theta)=y) - P(h(x;\theta) \neq y))$$
$$= \sum_{i=1}^{n} E[I(h(x;\theta)=y) - I(h(x;\theta) \neq y) | \theta \in \Theta_s] * P(s=i)$$
$$= \sum_{i=1}^{n} w_i * mr_i(x,y), \quad (4)$$

which is the desired margin function in (3). □

3.2. An Optimization Problem for Determining the Weights

We reformulate the previous ideas into an optimization task. Since we want to reduce the classification error rate of bagging, it is a natural way to use $E_D(B(\alpha,k))$ as the objective function. To construct the new classifier space is to solve the following optimization problem:

$$\min \int_{-1}^{1} B(\alpha,k) dF_m(\alpha), \quad (5)$$

where $F_m(\alpha) = P(\sum_{i=1}^{n} w_i * mr_i(x,y) \leq \alpha)$ and $\sum_{i=1}^{n} w_i = 1$.

Let the instances of the training set be denoted by (x_j, y_j), $j = 1,\cdots,m$, and then the discrete version of (5) is

$$\sum_{j=1}^{m} B(\sum_{i=1}^{n} w_i * mr(x_j,y_j),k) \text{ where } \sum_{i=1}^{n} w_i = 1. \quad (6)$$

Before we go further, we point out a useful alternate representation for $B(\alpha,k)$: $B(\alpha,k) = binc(\frac{1-\alpha}{2}, \lceil\frac{k}{2}\rceil, \lfloor\frac{k}{2}\rfloor + 1)$, where $binc$ is the normalized incomplete beta function:

$$binc(t,a,b) \doteq \int_0^t u^{a-1}(1-u)^{b-1}du \Big/ \int_0^1 u^{a-1}(1-u)^{b-1}du.$$

This can be shown via integration by part. The representation is useful for numerical computation purpose, since $\binom{k}{i}$ in (2) will be large when k is large and a direct computation of (2) will encounter floating point overflow problems.

3.3. A suboptimal algorithm

The summands in (6) does not posses "good" properties such as monotonicity or convexity for the free parameters $w_i's$, and (6) is difficult to be globally minimized. We use an approximate minimization technique instead.

By the $binc$ representation of $B(\alpha,k)$, for fixed k, as α increases, $B(\alpha,k)$ tends to 0. Thus, we expect that maximizing the number of instances whose classification margin exceeds some specified level is helpful for reducing (6). We carry out this by solving the following problem, for some $\gamma \geq 0$,

$$\min \sum_{j=1}^{m} \delta_j, \text{ s.t. } \sum_{i=1}^{n} w_i * mr_i(x_j,y_j) \geq \gamma - \delta_j, \quad (7)$$
$$\sum_{i=1}^{n} w_i = 1, \delta_j \geq 0, w_i \geq 0 \text{for } i = 1,\ldots,n, \ j = 1,\ldots,m.$$

In (7), δ_j's can be viewed as penalty if the resulting $mr(x_j,y_j)$ is lower than the prescribed value γ. The optimization problem is tractable since it is a linear optimization problem and can be globally minimized efficiently via linear programming. The solution for (7) only depends on γ, and we then tune γ by grid searching for minimizing (6). The procedure of our algorithm is given in Table 2.

Table 2. Algorithmic procedure

Estimating $mr_i(\cdot)$: For each $S_i \in \{S_1,\ldots,S_n\}$, calculate the empirical margin function $mr_i(\cdot)$ in the following way. For each instance (x,y), let the weak classifiers grown from S_i but not using (x,y) for training be denoted by $\theta_1',\ldots,\theta_t'$. Then $mr_i(x,y) = \frac{1}{t}\sum_{s=1}^{t}[I(h(x,\theta_s') = y) - I(h(x,\theta_s') \neq y)]$.

Training base classifiers: For each feature subspace $S_i, i = 1,\ldots,n$, train the base classifiers using bagging.

Optimization: Solving (7) based on $mr_i(\cdot)$'s for different γ's. Pick one solution $w_1^*,\ldots,w_n^*$ that minimizes (5).

Output: Grow k classifiers independently from S_i with probability w_i^*.

4. EXPERIMENTAL RESULTS

To demonstrate the effectiveness of the proposed algorithm, we compare the proposed algorithm with some other well-known related algorithms, including AdaBoost, bagging [1], random forest [4], and the random subspace method [5]. We use the C4.5 decision tree as the base classifier with classifier number 100. The datasets we used are from the UCI repository of machine learning databases [6], which have also been used extensively in related works. Since we only study the binary classification problem, we selected the two largest categories in each dataset for the classification task. For each dataset, we randomly draw 25 subspaces from the original feature space with dimension about 2/3 of the dimension of entire space, and use the classifier spaces grown from these subspaces as the base classifier spaces.

We use a ten-fold cross-validation for calculating the average classification error, and the experiments on each dataset are run 100 times independently. The experimental results are given in Table 3. We note that for most datasets, the average error rates of the proposed algorithm are lower than the others. Our algorithm achieves the lowest misclassification error in 13 out of 19 datsets. These validate that

- the classification margins in feature subspaces are diverse (otherwise, it is impossible to combine them to achieve a new better margin);
- our approximate optimization algorithm can successfully utilize the diverse classification ability in different base classifier spaces to achieve lower error rate.

Dataset	A	B	F	R	O
Balance	18.70	15.54	14.34	7.91	**5.89**
Breast Wisc	3.31	4.41	3.53	3.76	**2.79**
Bupa	30.43	26.66	28.09	28.06	**26.57**
Credit-g	25.23	25.70	24.61	24.06	**22.87**
Crx	13.72	13.75	14.28	13.49	**11.62**
Echocardio	11.10	9.42	9.59	10.17	**9.33**
Glass	**11.70**	17.27	12.83	13.47	12.27
Hayes Roth	23.11	21.0	22.09	23.46	**18.39**
Heart Cleve	19.06	21.21	18.81	17.76	**15.58**
Hepatitis	16.19	17.12	16.30	16.38	**13.06**
Horse Colic	17.12	**14.49**	15.52	20.97	14.60
Ionosphere	6.01	7.29	6.60	5.74	**5.49**
Pima	26.21	24.26	24.12	25.27	**23.73**
Promoters	8.53	12.55	9.39	8.55	**6.73**
Sonar	**13.52**	23.43	16.29	20.43	18.76
Tic-tac-toe	**0.89**	3.93	2.79	11.17	3.44
Vehicle	**1.72**	4.68	2.28	2.80	1.75
Votes	4.90	3.24	3.52	5.60	**2.96**
Yeast	37.26	**32.31**	32.53	33.01	32.42

Table 3. Experimental results, comparing the error rate of AdaBoost(A), bagging(B), random forests(F), random subspace methods(R), and our algorithm(O). For each dataset, we put in emphasis the **best** algorithm(s).

5. CONCLUSIONS

Motivated by the observation of the diversity of classification margins in feature subspaces, we have studied how to utilize different classification capability in classifier spaces for improving bagging performance. We have proposed a weighted subspace approach which constructs a new base classifier space, where the classification margin is a weighted linear combination of the classification margins of base classifier spaces grown from prescribed feature subspaces. The corresponding weights are determined by minimizing an objective function derived from classification margin.

The experimental results show that the proposed algorithm outperforms some other major ensemble algorithms. This verifies that the classifier spaces grown by bagging in feature subspaces behave diversely, and our approach can make use of the diversity for reducing classification error of bagging. Although we only consider classifier spaces constructed by bagging in different feature subspaces, a closer look at our algorithms reveals that our algorithm does not put restrictions on the method for constructing the base classifier spaces. Thus, the base classifier spaces can be produced not only by bagging, provided that there are probability distributions on the base classifier spaces. Therefore, the main results in this paper remain valid for a wider range of classifier spaces whenever they are endowed with probability distributions.

6. REFERENCES

[1] L. Breiman, “Bagging predictors,” *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[2] T. G. Dietterich, “An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization,” *Mach. Learn.*, vol. 40, no. 2, pp. 139–157, 2000.

[3] 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, no. 5, pp. 1651–1686, 1998.

[4] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[5] T. K. Ho, “The random subspace method for constructing decision forests,” *IEEE Trans. Patt. Anal. Mach.*, vol. 20, no. 8, pp. 832–844, 1998.

[6] C. B. D.J. Newman, S. Hettich and C. Merz, “UCI repository of machine learning databases,” 1998.