2012 IEEE 12th International Conference on Data Mining

Feature Weighting and Selection Using Hypothesis Margin of Boosting

Malak Alshawabkeh*, Javed A. Aslam†, Jennifer G. Dy*, and David Kaeli*
*Dept of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115
Email: {malshawa, kaeli, jdy}@ece.neu.edu
† College of Computer and Information Science, Northeastern University, Boston, MA 02115
Email: jaa@ccs.neu.edu

***Abstract*—Utilizing the concept of hypothesis margins to measure the quality of a set of features has been a growing line of research in the last decade. However, most previous algorithms have been developed under the large hypothesis margin principles of the 1-NN algorithm, such as Simba. Little attention has been paid so far to exploiting the hypothesis margins of boosting to evaluate features. Boosting is well known to maximize the training examples' hypothesis margins, in particular, the *average margins* which are known to be the first statistics that considers the whole margin distribution. In this paper, we describe how to utilize the training examples' mean margins of boosting to select features. A weight criterion, termed Margin Fraction (MF), is assigned to each feature that contributes to the average margin distribution combined in the final output produced by boosting. Applying the idea of MF to a sequential backward selection method, a new embedded selection algorithm is proposed, called SBS-MF. Experimentation is carried out using different data sets, which compares the proposed SBS-MF with two boosting based feature selection approaches, as well as to Simba. The results show that SBS-MF is effective in most of the cases.**

***Keywords*-Feature selection; boosting; average margin;**

I. Introduction

Boosting has attracted much attention in the machine learning community mainly because of its excellent performance and computational attractiveness for large datasets [25], [19]. The main breakthrough came with Freund and Schapire's most successful AdaBoost algorithm [10], [11]. The essence of AdaBoost is to train a number of simple weak classifiers that are linearly combined into a single strong classifier. A major advantage of the AdaBoost algorithm is the adaptive selection of discriminative and complementary features during the training process which most often yields better feature or variable selection while keeping or even increasing the prediction accuracy. AdaBoost has been used, in particular for feature selection, with great success in many applications like face recognition [27], [9], text mining [29] and intrusion detection [2], [15].

AdaBoost has the property that it does not often seem to suffer from overfitting, even after a large number of iterations [4], [22]. To understand this lack of overfitting, Breiman [4] first used the notion of variance and bias for classification to argue that AdaBoost could avoid overfitting by reducing variance, since it is in ways similar to bagging [3]. Schapire et.al. [24], however, explained to some extent a reasonable explanation to the success of AdaBoost by the *margin theory*. The margin of a boosted classifier is a number between -1 and 1, that according to the margin theory, can be thought of as a confidence measure of a classifier's predictive ability, or as a guarantee on the generalization performance. If the margin of a classifier is large, then it tends to perform well on test data. If the margin is small, then the classifier tends not to perform so well. Furthermore, Schapire et al. showed that AdaBoost has a tendency to increase the margins on the training examples. Thus, though not entirely complete, their theory and experiments strongly support the notion that margins are highly relevant to the behavior and generalization performance of AdaBoost.

Breiman [4], however, soon thereafter raised serious doubts on the margin theory by designing a boosting-type algorithm called arc-gv. Breiman's experiments indicated that his algorithm achieved higher margins than AdaBoost, and yet performed worse on test data. Reyzin and Schapire [23] reproduced Breiman's experiments and were able to reconcile his results with the margins explanation, noting that the weak classifiers found by arcgv are more complex than those found by AdaBoost. Although the empirical success of the AdaBoost algorithm depends on many factors (e.g., the type of data and how noisy it is, the capacity of the weak learning algorithm, the number of boosting iterations, regularization and the entire margin distribution over the training examples), it is well accepted that margin distribution is crucial to relate margin to the generalization of AdaBoost. Previously the *minimum margin* bound was established for AdaBoost, however, researchers believe that this is far from sufficient. Intuitively, Reyzin and Schapire [23] suggested to use the *average margin* as a measure to compare margin distributions, but provided

1550-4786/12 $26.00 © 2012 IEEE
DOI 10.1109/ICDM.2012.143

no bound prove for it; this was the focus of Gao and Zhou [12] who recently proved the average margin bound for AdaBoost and showed that a larger average margin implies stronger generalization.

In this work, we took an unusual approach for using boosting as an effective feature subset selection (FSS) technique. We introduce the idea of utilizing the training examples' average margin to measure the quality and relative importance of features. In each boosting iteration, a base hypothesis is learned with a prediction confidence for each example. The weights of misclassified instances are increased and those of correctly classified instances are decreased according to the confidence of the learned base hypothesis. Consequently, the learner is forced to search for base hypotheses which correctly classify these hard examples and thus increase their margins. Since the margin tends to give a strong indication of a learner's performance in practice, a natural goal is to find learners (features) that achieve a maximum margin. For this purpose, we propose an evaluation function which assigns weights to subsets of features according to the margin they induce. The weight, termed *Margin Fraction* (MF), is measured by computing the cumulative effect each feature has on the average margin associated with the weighted linear combination that boosting produces, i.e., the margin fraction that is due to a feature.

The problem of searching the "best" subset of features is solved by means of a greedy algorithm based on backward selection [17]. A backward sequential selection is used because of its lower computational complexity compared to randomized or exponential algorithms and its optimality in the subset selection problem [6]. For the sake of simplicity, the proposed algorithm will be referred to as SBS-MF (Sequential Backward Selection using Margin Fraction). Hence, the SBS-MF goal is to find a subset of size r among d features ($r < d$). The method starts with all the features and at each iteration an AdaBoost with *decision stumps* algorithm is trained, followed by removing one or more "bad" features from further consideration. The goodness of the features is determined by the margin fraction weights produced within AdaBoost. The features remaining after a number of iterations are deemed to be the most useful for discrimination, and can be used to provide insights into the given data.

Margin based feature selection is a growing line of research. There are two main ways to define margins [7]. The *sample-margin* (SM) measures the distance between the instance and the decision boundary induced by the classifier. SVM, for example, finds the separating hyper-plane with the largest SM. As an alternative definition, the *hypothesis-margin* (HM) requires the existence of a distance measure on the hypothesis class. The margin of a hypothesis with respect to an instance is the distance between the hypothesis and the closest hypothesis that assigns an alternative label to the given instance. Fig. 1 shows examples of both SM and HM. AdaBoost and 1-NN use the concept of HM. Various feature selection algorithms have been developed under the large margin (SM or HM) principles such as SVM-based feature selection [14] and Relief family (1-NN based) algorithms, such as Simba [13].

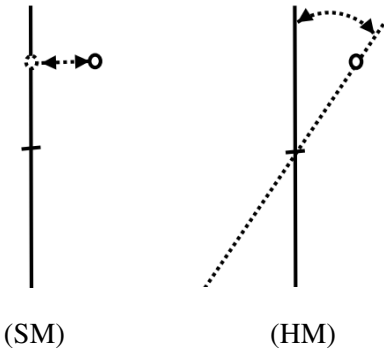
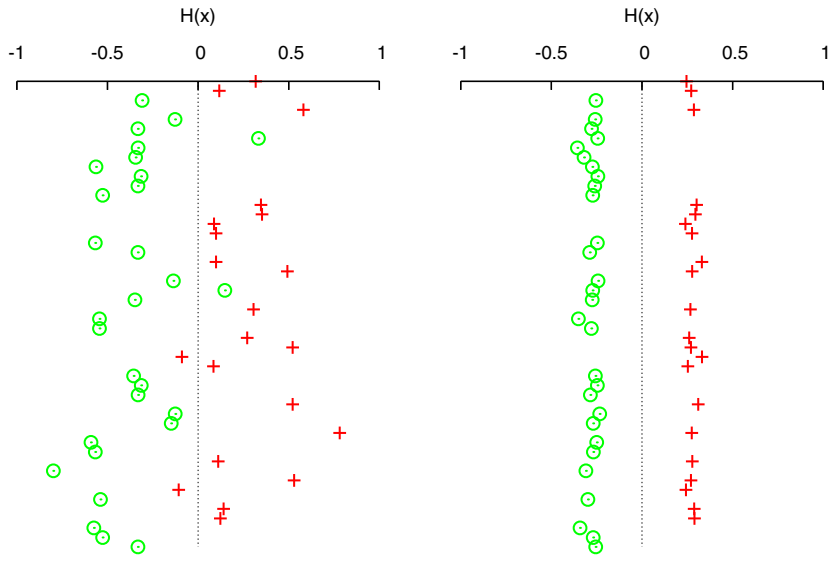


Figure 1: Sample Margin (SM) measures how much can an instance travel before it hits the decision boundary. On the other hand Hypothesis Margin (HM) measures how much can the hypothesis travel before it hits an instance [7].

To our knowledge, almost no previous work has exploited the characteristics of the hypothesis margins of boosting to determine the quality of features. The two main related works, particularly in boosting with decision stumps with a greedy search, are: 1) the work of Das [8], who proposed the BDSFS (Boosted Decision Stump Feature Selection) algorithm, and 2) Tsuchiya and Fujiyoshi [28] algorithm. Das applies a forward selection search strategy. The selection of the next feature to be considered is based on the information gain criterion and takes into consideration the weight of each dataset instance. This selected feature is then added to the set that will be returned and used to create a decision stump (used as the weak learner), which updates the weights of the dataset examples by assigning higher weights to examples that have often been misclassified in this round. The process repeats until a pre-specified number of features have been selected, in a process very similar to boosting. In Tsuchiya and Fujiyoshi [28] work the features are evaluated based on the *contribution ratio* (CR) criterion. The CR is defined as the relative importance of features based on the confidence ratio of



(a) 10 rounds (b) 100 rounds

Figure 2: Boosting $H(x)$ values for training instances of Prostate dataset. a) after 10 rounds, and b) after 100 rounds.

the learned base hypothesis. Similar to our approach, the Tsuchiya and Fujiyoshi algorithm starts with a given set of features, the CR of each feature from a feature set is estimated, and features that are not contributing and that have a low CRs are removed.

In this paper, however, we show that the dynamics inherent to boosting offer ideal means to evaluate features utilizing the training examples' margins distribution, and while a feature may have a large CR, it will not contribute to a good overall margin unless its "conditional" margin is also large. Thus, a better indicator is its fraction of the overall margin.

This paper is organized as follows. In Section 2, we review the AdaBoost margin concept. In Section 3, we present our proposed feature weighting using margin fraction. Our experimental evaluation of the approach is described and discussed in Section 4. We conclude in Section 5 with some pointers to future work.

II. Background

A. Boosting Margins

Definition Let $S = \{(x_i, y_i)\}_{i=\{1,\cdots,m\}}$ be a set of m instances, where x_i is a pattern vector $x \in \Re^F$ and y_i is a class label $y_i \in \{-1, 1\}$, drawn i.i.d. from D. And let $\mathcal{H}$ be a hypothesis space (in this paper we constrain $\mathcal{H}$ to be finite).

Boosting calls a given weak or base learning algorithm repeatedly in a series of rounds $t = 1, \ldots, T$. A base learner's goal is to find a weak hypothesis $h_t : X \rightarrow \{-1, 1\}$ appropriate for the distribution D_t. One of the main ideas of the algorithm is to maintain a distribution or set of weights over the training set. The weight of this distribution on training example i on round t is denoted $D_t(i)$. Initially, all weights are set equally, but in each round, the weights of incorrectly classified examples are increased so that the weak learner is forced to focus on the hard examples in the training set. After a hypothesis is received, the algorithm updates the weights of the distributions D_t. Combining the hypotheses generated by these weak learners will collectively produce the following weighted linear classifier:

$$H(x_i) = \sum_{t=1}^{T} \alpha_t h_t(x_i) \tag{1}$$

where

$$\alpha_t = \frac{1}{2} \ln \frac{1 + \gamma_t}{1 + \gamma_t} \tag{2}$$

where $\gamma_t = \sum_{i=1}^{m} D_t(i) y_i h_t(x_i)$ is called the edge of h_t, which is an affine transformation of the error rate of $h_t(x_i)$.

Boosting is particularly good at finding hypotheses with large margins, in which it concentrates on those examples whose margins are small (or negative) and forces the base learning algorithm to generate good classifications for those examples [10]. The margin of

boosting at T rounds associated with any instance i is defined as

$$\rho(x_i) = \frac{y_i H(x_i)}{\omega} = \frac{y_i \sum_{t=1}^{T} \alpha_t h_t(x_i)}{\omega} \quad (3)$$

while $\omega = \sum_{t=1}^{T} |\alpha_t|$ served as a normalization factor. It is easy to see that the margin is a number in the range $[-1, 1]$, and that an example is classified correctly if and only if its margin is positive, i.e., $H(x)$ classifies the example correctly as shown in Fig. 2. A large positive margin can be interpreted as a "confident" correct classification. The distribution of the margin can be visualized by plotting the cumulative distribution function (CDF) of margins, i.e., the fraction of examples whose margin is at most n as a function of $n \in [-1, 1]$. AdaBoost effectively maximizes the *minimum margin* ($\min yH(x)$), which leads to good generalization ability as Schapire et.al. proved in Theorem 1 [24].

However, recently, Gao and Zhou [12] show that compared to previous statistics on margin theory (i.e., minimum margin), the *average margin* is one of the statistics that considers the whole margin distribution and thus includes more information.

The *average margin* $E_S[yH(x)]$ across m examples can be defined as:

$$\bar{\rho} = \frac{1}{m} \sum_{i=1}^{m} \rho(x_i) \quad (4)$$

B. Average Margin Bound

Gao and Zhou [12] provided an upper bound for the generalization error of AdaBoost in term of average margin by Theorem 6. Which we briefly describe as follows.

Theorem 2.1: For constant $\gamma > 0$, suppose base learner h_t in each iteration has edge $\gamma_t \geq \gamma$ and set

$$\tau = \frac{-0.99 \ln(1-\gamma^2)}{\ln(1+\gamma) - \ln(1-\gamma)}$$

For any $\delta > 0$, if $\theta = E_S[yH(x)] > \sqrt{8/|\mathcal{H}|}$ and the iteration number

$$T \geq \left\lceil \frac{100}{\ln(1-\gamma^2)} \ln \left(\frac{1}{m} \left(\frac{16 \ln 2 |\mathcal{H}|}{\tau^2 \theta^2} (\ln 2m^2 - \ln \ln |\mathcal{H}|) + \in \ln \frac{|\mathcal{H}|}{\delta} \right) \right) \right\rceil$$

then with probability at least $1 - \delta$ over the random choice of sample S with size m, every voting classifier $H(x)$ by AdaBoost satisfies the following bound:

$$Pr_D[yH(x) < 0] \leq \left(\frac{\ln |\mathcal{H}|}{m} + \sqrt{\frac{8}{m} \left(\frac{8 \ln 2 |\mathcal{H}|}{\tau^2 \theta^2} \ln \frac{2m^2}{\ln |\mathcal{H}|} + \ln \frac{|\mathcal{H}|}{\delta} \right)} \right)$$

The Theorem states that the generalization of AdaBoost depends not only on the sample size and the complexity of the base learner, but also on the average margin, the number of iterations and the goodness of base learner. All these factors could affect the generalization error, and thus, completely explaining AdaBoost's resistance to overfitting is more difficult than what has been disclosed by previous theoretical results. In this sense, we would rather use average margin than other statistics to evaluate the quality of features.

III. Margin Fraction for Feature Weighting and Selection Algorithm

In this section, we present our new feature selection based on the concept of the average margin induced by boosting. Our method observes the training examples' mean margins to evaluate the quality of features. A weight criterion, termed Margin Fraction (MF), is assigned to each feature that contributes to the average margin distribution combined in the final output produced by boosting. Applying the idea of the MF to a sequential backward selection FSS method, a new selection algorithm is proposed, termed SBS-MF.

To perform feature selection we consider features as the weak learners for boosting. The choice of the weak learner is usually driven by optimizing the prediction performance. In addition, some structural properties can be another useful criterion as well. AdaBoost estimator is a linear combination of weak learners. Therefore, structural properties of the boosting function estimator are given by linear combination of structural characteristics of the weak leaner.

A. Stumps and larger trees as weak learners

Trees are among the most popular base procedures in machine learning. They have the advantage to be invariant under monotone transformations of predictor variables, i.e., we do not need to search for good data transformations. When using stumps [10], [30], i.e., a tree with two terminal nodes, the boosting estimate will be an additive model in the original predictor variables, because every stump-estimate is a function of a single predictor variable only. Similarly, boosting trees with (at most) $d + 1$ terminal nodes results in a nonparametric model having at most interactions of order $d - 1$:

e.g., for $d = 2$, we would pick up interaction terms between pairs of predictor variables. Thus, if we want to constrain the degree of interactions, we can easily do this by constraining the (maximal) number of nodes in the tree learner. For many real datasets, it seems that low-order interaction (or even additive) models are sufficiently rich for good prediction and interpretation. For example, the naive Bayes classifier works surprisingly well in many applications [16]. Also boosting with stumps, yielding an additive model, has proven to be successful in many areas, e.g. winning the performance prediction challenge of the IEEE World Congress on Computational Intelligence 2006 [18]. Thus, we often get good performance with trees having 2 or 3 terminal nodes ($d = 1$ or 2, respectively). With such small values of d, our proposed feature selection is computationally fast.

Therefore, we construct a set of weak classifiers by considering decision stumps. For each feature f and a given threshold θ, a decision stump h can be constructed as

$$h(x) \stackrel{def}{=} \begin{cases} -1, & x_f \leq \theta \\ +1, & x_f > \theta \end{cases}$$

where x_f denotes the component of feature vector x, which corresponds to feature f.

B. Hypothesis Margin Feature Weight

Definition Let F be the total number of unique features used across all T rounds i.e., decision stumps, and for any chosen feature f, let $h_{f,j}$ be the decision stump corresponds to the j-th use of feature f, and let N_f be the total number of times that feature f is used. We then have $\sum_{f=1}^{F} N_f = T$. Let $\alpha_{f,j}$ be the associated confidence.

Now for any individual feature f, one can consider the weighted linear combination associated with that feature and the "conditional" margin associated with just that weighted linear combination for any instance i.

$$H_f(x_i) = \sum_{j=1}^{N_f} \alpha_{f,j} h_{f,j}(x_i) \tag{5}$$

$$\rho_f(x_i) = \frac{y_i \sum_{j=1}^{N_f} \alpha_{f,j} h_{f,j}(x_i)}{\sum_{j=1}^{N_f} |\alpha_{f,j}|} \tag{6}$$

Consider the fraction of the absolute "confidence" weight associated with any feature f, defined as follows:

$$\Gamma_f = \frac{\sum_{j=1}^{N_f} \alpha_{f,j}}{\sum_{t=1}^{T} |\alpha_t|} = \frac{\sum_{j=1}^{N_f} \alpha_{f,j}}{\sum_{f=1}^{F} \sum_{j=1}^{N_f} |\alpha_{f,j}|} \tag{7}$$

We then have the following theorem.

Theorem 3.1: the overall margin associated with any instance "i" is the weighted linear combination of conditional margins, where Γ_f are the weights.

$$\sum_{f=1}^{F} \Gamma_f \rho_f(x_i) = \rho(x_i)$$

Proof:

$$\begin{aligned}
\sum_{f=1}^{F} \Gamma_f \rho_f(x_i) &= \sum_{f=1}^{F} \left(\frac{\sum_{j=1}^{N_f} \alpha_{f,j}}{\sum_{t=1}^{T} |\alpha_t|} \right) \rho_f(x_i) \\
&= \frac{\sum_{f=1}^{F} \left(\sum_{j=1}^{N_f} \alpha_{f,j} \right) \rho_f(x_i)}{\sum_{t=1}^{T} |\alpha_t|} \\
&= \left(\frac{\sum_{f=1}^{F} \left(\sum_{j=1}^{N_f} \alpha_{f,j} \right)}{\sum_{t=1}^{T} |\alpha_t|} \right. \\
&\quad \left. \cdot \frac{\left(\frac{y_i \sum_{j=1}^{N_f} \alpha_{f,j} h_{f,j}(x_i)}{\sum_{j=1}^{N_f} \alpha_{f,j}} \right)}{\sum_{t=1}^{T} |\alpha_t|} \right) \\
&= \frac{y_i \sum_{f=1}^{F} \sum_{j=1}^{N_f} \alpha_{f,j} h_{f,j}(x)}{\sum_{t=1}^{T} |\alpha_t|} \\
&= \rho(x_i)
\end{aligned}$$

■

This helps to support the use of Γ_f as an indicator of the utility of a given feature f, which is basically the *contribution ratio* (CR) weight criterion that Tsuchiya and Fujiyoshi [28] and Alshawabkeh et.al. [2] proposed (we will give more details about the CR weight criterion in Section III-C). However, while a feature f may have a large Γ_f, it will not contribute to a good overall margin unless ρ_f is also large. A better indicator is the fraction of the overall margin that is due to f:

$$\rho_f(x_i) = \frac{\Gamma_f \rho_f(x_i)}{\rho(x_i)} \tag{8}$$

Note, however, that this only deals with a single instance i. To consider the margin across all instances we use the average margin, which can be redefined as:

$$\bar{\rho} = \frac{1}{m} \sum_{i=1}^{m} \sum_{f=1}^{F} \Gamma_f \rho_f(x_i) \tag{9}$$

Thus, the MF due to feature f is computed as:

$$\begin{aligned} MF_f &= \frac{\Gamma_f \frac{1}{m}\sum_{i=1}^{m}\rho_f(x_i)}{\frac{1}{m}\sum_{i=1}^{m}\rho(x_i)} \\ &= \Gamma_f \frac{\sum_{i=1}^{m}\rho_f(x_i)}{\sum_{i=1}^{m}\rho(x_i)} \\ &= \frac{\sum_{j=1}^{N_f}\alpha_{f,j}}{\sum_{t=1}^{T}|\alpha_t|} \cdot \frac{\sum_{i=1}^{m}\left(\frac{y_i\sum_{j=1}^{N_f}\alpha_{f,j}h_{f,j}(x_i)}{\sum_{j=1}^{N_f}|\alpha_{f,j}|}\right)}{\sum_{i=1}^{m}\left(\frac{y_i\sum_{t=1}^{T}\alpha_t(x_i)}{\sum_{t=1}^{T}|\alpha_t|}\right)} \\ &= \frac{\sum_{i=1}^{m}\sum_{j=1}^{N_f}y_i\alpha_{f,j}h_{f,j}(x_i)}{\sum_{i=1}^{m}\sum_{t=1}^{T}y_i\alpha_t h_t(x_i)} \end{aligned} \quad (10)$$

We can use MF as an indicator of the utility of a given feature f. Typically, the higher the value of MF, the better the feature f.

C. Comparison between Margin Fraction MF and Contribution Ratio CR

Tsuchiya and Fujiyoshi has proposed a feature evaluation method based on AdaBoost with decision stumps [28]. They introduced a metric, called contribution ratio CR, that indicates how well the features "contribute" to the classification performance based on the performance weight α of the weak hypothesis h_t. We will refer to this method as AdaBoost-CR.

A contribution ratio CR_f for each feature f is defined by:

$$CR_f = \sum_{t=1}^{T}\bar{\alpha_t}\,\delta_K[P(h_t) - f] \quad (11)$$

where $\bar{\alpha_t}$ is the average confidence assigned to feature f, δ_K is the Kronecker delta; which is a function of two variables that is 1 if they are equal and 0 otherwise, and $P()$ is a function for outputting the feature chosen at round t in the AdaBoost training process. In other words, the CR_f is equal to the fraction of the absolute "confidence" weight associated with any feature f, thus, it is equal to Γ_f that we defined in Equation 7. However, while CR_f is a helpful evaluation metric of the quality of a given feature f, it is not an accurate indicator of its performance. A feature f may have a large CR_f value but will not contribute to a good overall margin. A better indicator is the margin fraction MF that is due to f.

D. Sequential Backward Selection using Margin Fraction

For a given dataset with the size of d, the goal for feature selection is to select r features ($r < d$) that provides the best results. Here the MF weight used as a ranking criterion for our proposed algorithm.

Algorithm 1 The SBS-MF algorithm

1. Initialization: feature ranked list $Rlist = [\,]$; subset of surviving features $Slist = [1, \cdots, d]$.
2. **r**epeat **u**ntil $Slist = [\,]$
   (a) Train an AdaBoost classifier with all the training data using the subset of surviving features $Slist$.
   (b) **f**or all features in $Slist$, do evaluate the weight criterion MF_f of feature f **e**ndfor.
   (c) Find the feature with smallest MF weight: $f = \arg\min_f(MF)$.
   (d) Update feature ranked list: $Rlist = [Slist(f), Rlist]$.
   (e) Eliminate the feature with smallest MF weight: $Slist = [1, \cdots, f-1, f+1, \cdots, length(Slist)]$.
3. **O**utput Feature ranked list $Rlist$.

The sequential backward selection search algorithm is used. The selection process starts from a full set of features then removes sequentially the most irrelevant ones. To find the most irrelevant feature of the current surviving subset, AdaBoost algorithm is trained on the training set with the current surviving features subset. The classification results of AdaBoost are then used to obtain the MF weights for each feature. The features are then ranked based on the MF weight criterion. Finally, the most irrelevant feature, which its MF weight is the smallest, is eliminated. The procedure is repeated until r features are removed or all of the features are ranked. The SBS-MF method is summarized in Algorithm 1.

IV. Experimental Setup

A. Data Sets

To validate performance fairly and to provide a comprehensive testing suite for feature selection methods under different conditions, two groups of benchmark datasets are adopted in our simulation experiments. The first group includes datasets with large number of samples. These datasets are all available from the UCI Machine Learning Repository [20] and most of them are frequently used in the literatures. Table I summarizes some general information about these datasets. Their full documentation can be obtained from the UCI website. Some of these datasets may embody missing values or continuous features, and so they would be processed during the preprocessing phases. For missing values, we

Table I: Data sets description.

Dataset	Features	Instances
Chess	32	3196
Ionosphere	34	351
Mushroom	22	8124
Musk clean1	166	476
Spambase	57	4601
Lymphoma	7129	77
Prostate	6000	89

replaced them with the most frequently used values and means for nominal and numeric features, respectively.

The second group includes two microarray datasets that contain a large number of features and a small number of samples: Lymphoma data [21] and Prostate cancer data [26], as described in Table I. The data sets were minimally preprocessed by trimming the range of inspected mass/charge ratios, normalizing and reducing the amount of noise.

B. Performance Assessment and Discussion

The performance of SBS-MF was assessed using the benchmark data sets and was compared against, AdaBoost-CR, BDSFS and Simba.

In simulation experiments, the datasets were firstly fed into these feature selectors, which will generate different feature ranking sets in which the features are sorted in a descending order according to their priorities. For UCI sets we chose 40% of top ranked features. After that, datasets with newly selected features were passed to external learning algorithms to assess classification performance. Currently, various outstanding learning algorithms are available. In our experiments, two classifiers, namely, 1-Nearest Neighbor (1-NN) and SVM, were chosen to test prediction capability of the selected subset. The reason to choose them is that these classifiers represent different approaches in learning and often used in data mining applications because of relatively high efficiency.

To achieve impartial results, 10-fold cross validations had been adopted for each algorithm-dataset combination in verifying classification capability. That is to say, for each dataset before and after feature selection, we run every classification algorithm on it 10 times and at each time a 10-fold cross validation was used, and the final results were their average values. To determine whether the difference is significant or not, pair t-tests between accuracies without feature selector and with each selector at a time had been performed. Throughout this paper, the difference of accuracies is considered significantly different if its p-value is less than 0.05 (i.e., confidence level greater than 95%) according to a paired t-test.

C. Performance Results of UCI Sets

The experimental results about classification performance on the UCI datasets for classifiers using all-features and using the four feature selection algorithms are presented in Table II. Notation "$\bullet$" (or "$\circ$") denotes that the performance of classifier with current selector is significantly better (or worse) than those without using selector (i.e., All-features) in statistical test. In addition, the bold value in entries means that it is the largest one among these three feature selectors in the same classifier. The average value of accuracies with the same selector is given in the "Ave." row.

The results in Table II show that the performance using SBS-MF are better than other approaches in the 1-NN classifier. SBS-MF has three maximal values of classification accuracy over 5 datasets. From the view of average performance SBS-MF is still relatively superior to other selectors.

For SVM classifier, one may also observe that our proposed method clearly surpasses others in many cases. As an illustration, for SBS-MF, the quantities of cases with significantly better and worse performance are three and zero, respectively. In addition, SBS-MF has the highest classification performance over three-quarter datasets, which is higher than other selection algorithms. Correspondingly, the average value of accurate ratios in SBS-MF is also the largest one.

D. Small Training Sets

To compare the performance of SBS-MF, AdaBoost-CR, BDSFS and Simba on small training sets, we examined two microarray datasets concerning two classification problems. These datasets have already been used for benchmarking feature selection algorithms (for example, see Chen et al. [5]). The Lymphoma data set contains 77 samples: 58 are diffuse large B-cell lymphoma, and 19 are follicular lymphomas, described by 7129 features, and the Prostate cancer data set is composed of 89 samples: 63 have no evidence of cancer, and 26 have prostate cancer, described by 6000 features.

In order to speed up the feature selection procedure, half of the features are removed at each step until 100 features remain still to be ranked. Then features are removed one at a time. For performance comparison, we rank all of the features using the four feature selection methods. We then take a prefix of this ranking and train

the learning algorithms with the prefix. The learning algorithms then are tested on the data.

Since the datasets are unbalanced, initial weights for AdaBoost were adjusted to account for the skewed class distributions [1]. We also applied stratified 10-fold cross validation to evaluate the methods. We repeated the stratified 10-fold cross validation 20 times and averaged results over each trial. Further, on imbalanced data sets, algorithms will be hard pressed to classify test samples as members of the minority class because the discriminant scores given by the classifier are often weighted toward the majority class. For that, we evaluated the results using the area under the ROC curve (AUC) metric, which is one of the statistics that researchers commonly use to focus on the minority class.

Sub-figures in Fig. 3 show the classification results in terms of AUC versus the number of features selected using an 1-NN and SVM classifiers. Lines with $\circ$ markers indicate classifiers using Simba-selected features, lines with $\star$ markers indicate classifiers using BDSFS-selected features, lines with $+$ markers indicate classifiers using AdaBoost-CR-selected features, and lines with $\times$ markers indicate classifiers using SBS-MF-selected features and the solid black line indicates the performance where all the features are used for classification.

The top subfigures in Fig. 3 show the AUC results for the Lymphoma data. The features selected by SBS-MF outperform other methods with 1-NN, and SVM classifiers. Further, note that SBS-MF is able to achieve this with as few as 20 features. What this indicates is that SBS-MF is more effective in selecting fewer high quality features, such that after keeping 70 features (with 1-NN), adding more features may not improve and may even degrade classifier performance. We observe the same trend when using 1-NN and SVM that SBS-MF achieves a higher AUC value compared to the other methods with fewer features on Prostate data (bottom subfigures in Fig. 3).

V. Conclusions and Future Work

In this paper, a new feature selection algorithm, called SBS-MF, is proposed based on the concept of the average margin induced by boosting. Our method observes the training examples' mean margins to evaluate the quality of features. A weight criterion, termed Margin Fraction (MF), is assigned to each feature that contributes to the average margin distribution combined in the final output produced by boosting, i.e., the margin fraction that is due to a feature. A backward sequential selection is used to search for the "best" subset of features since it has lower computational complexity compared to other search algorithms. Our method starts with all the features and at each iteration an AdaBoost with decision stumps algorithm is trained, followed by removing one or more "bad" features from further consideration. The goodness of the features is determined by the margin fraction weights produced in the AdaBoost. The features remaining after a number of iterations are deemed to be the most useful for discrimination.

We have validated the performance of our method on seven data sets. Five data sets were taken from UCI repository that contain relatively large number of samples. The remaining two datasets are microarray data. These data sets have a small number of samples with a large number of features. We compared our method to two boosting with decision stumps feature selection methods as well as to a margin-based feature selection. Our results show that our method is effective in most cases.

Our method might not be robust to noise since it is based on the *hard* margins of boosting. Part of the future work is to investigate how to overcome this issue by utilizing the *soft* margins of boosting instead.

References

[1] M. Alshawabkeh, J. A. Aslam, J. G. Dy, and D. Kaeli. Feature selection metric using AUC margin for small samples and imbalanced data classification problems. In *Proceedings of the Tenth International Conference on Machine Learning and Applications (ICMLA)*, pages 145–150, 2011.

[2] M. Alshawabkeh, M. Moffie, F. Azamandian, J. Aslam, J. Dy, and D. Kaeli. Effective virtual machine monitor intrusion detection using feature selection on highly imbalanced data. In *The ninth International Conference on Machine Learning and Applications (ICMLA)*, pages 823–825, 2010.

[3] L. Breiman. Bagging predictors. In *Machine Learning*, pages 123–140, 1996.

[4] L. Breiman. Arcing classifiers. *The Annals of Statistics*, 26(3):801–849, 1998.

[5] X. Chen and M. Wasikowski. Fast: a roc-based feature selection metric for small samples and imbalanced data classification problems. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 124–132, 2008.

[6] C. Couvreur and Y. Bresler. On the optimality of the backward greedy algorithm for the subset selection problem. *SIAM J. Matrix Anal. Appl.*, 21:797–808, 2000.

Table II: A comparison of classification accuracies of classifiers using four feature selection algorithms on UCI datasets.

Dataset	1-NN(%)				
	All-features	SBS-MF	AdaBoost-CR	BDSFS	Simba
Chess	89.73 ± 0.38	$\mathbf{94.20 \pm 0.49}^{\bullet}$	$91.79 \pm 4.41^{\bullet}$	89.50 ± 2.15	90.50 ± 1.05
Ionosphere	85.80 ± 0.14	$\mathbf{92.73 \pm 0.38}^{\bullet}$	$87.71 \pm 0.63^{\bullet}$	$84.98 \pm 1.03^{\circ}$	86.23 ± 0.71
Mushroom	97.84 ± 0.05	**99.91 ± 0.12**	93.1 ± 0.38	$91.34 \pm 0.22^{\circ}$	93.97 ± 0.08
Musk clean1	89.28 ± 4.41	$\mathbf{94.64 \pm 1.30}^{\bullet}$	$83.62 \pm 2.13^{\circ}$	$84.10 \pm 0.96^{\circ}$	87.62 ± 1.47
Spambase	91.58 ± 6.21	$\mathbf{94.56 \pm 0.61}$	91.97 ± 0.78	90.12 ± 1.40	91.02 ± 0.86
Ave.	90.85 ±2.24	95.21 ± 0.58	89.64± 1.66	88.01± 1.15	89.87 ± 0.83

	SVM(%)				
	All-features	SBS-MF	AdaBoost-CR	BDSFS	Simba
Chess	87.43 ± 1.42	$\mathbf{94.83 \pm 0.49}^{\bullet}$	89.73 ± 1.30	89.25 ± 0.64	85.79 ± 1.02
Ionosphere	86.04 ± 1.13	$\mathbf{92.52 \pm 0.31}^{\bullet}$	87.23 ± 0.81	86.15 ± 1.87	$82.21 \pm 1.69^{\circ}$
Mushroom	97.97 ± 0.26	**98.92 ± 0.11**	94.49 ± 0.81	95.73 ± 0.46	$90.95 \pm 0.89^{\circ}$
Musk clean1	86.27 ± 1.76	$\mathbf{92.89 \pm 0.52}^{\bullet}$	85.75 ± 0.56	$80.36 \pm 0.85^{\circ}$	$80.09 \pm 1.12^{\circ}$
Spambase	90.76 ± 4.11	**92.92± 0.25**	88.12 ± 0.69	$84.63 \pm 1.12^{\circ}$	$84.61 \pm 0.95^{\circ}$
Ave.	89.64 ± 1.74	94.42 ± 0.34	89.10 ± 0.83	87.22 ± 0.98	84.73 ±1.13

[7] K. Crammer, R. Gilad-bachrach, A. Navot, and N. Tishby. Margin analysis of the lvq algorithm. In *Advances in Neural Information Processing Systems 2002*, pages 462–469, 2002.

[8] S. Das. Filters wrappers and a boosting-based hybrid for feature selection export. *In Proceedings of the Eighteenth International Conference on Machine Learning*, pages 74–81, 2001.

[9] D. Dinh and S. Satoh. Feature selection by adaboost for svm-based face detection. *Information Technology Letters*, 3:183–186, 2004.

[10] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 148–156, 1996.

[11] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[12] W. Gao and Z. Zhou. The kth, median and average margin bounds for adaboost. *CoRR*, abs/1009.3613, 2010.

[13] R. Gilad-Bachrach, N. Amir, and N. Tishby. Margin based feature selection - theory and algorithms. In *In International Conference on Machine Learning (ICML)*, pages 43–50, 2004.

[14] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[15] W. Hu, We. Hu, and S. J. Maybank. Adaboost-based algorithm for network intrusion detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 38(2):577–583, 2008.

[16] A. Jamain and D. J. Hand. The naive bayes mystery: A classification detective story. *Pattern Recogn. Lett.*, 26:1752–1760, 2005.

[17] R. Kohavi and G. John. Wrappers for feature subset selection. *ARTIFICIAL INTELLIGENCE*, 97(1):273–324, 1997.

[18] R. W. Lutz. Logitboost with trees applied to the wcci 2006 performance prediction challenge datasets. In *IJCNN*, pages 1657–1660, 2006.

[19] P. Mitra, C. A. Murthy, and S. K. Pal. Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):301312, 2002.

[20] D. J. Newman, S. Hettich, C.L. Blake, and C.J. Merz. Uci repository of machine learning databases, 1998.

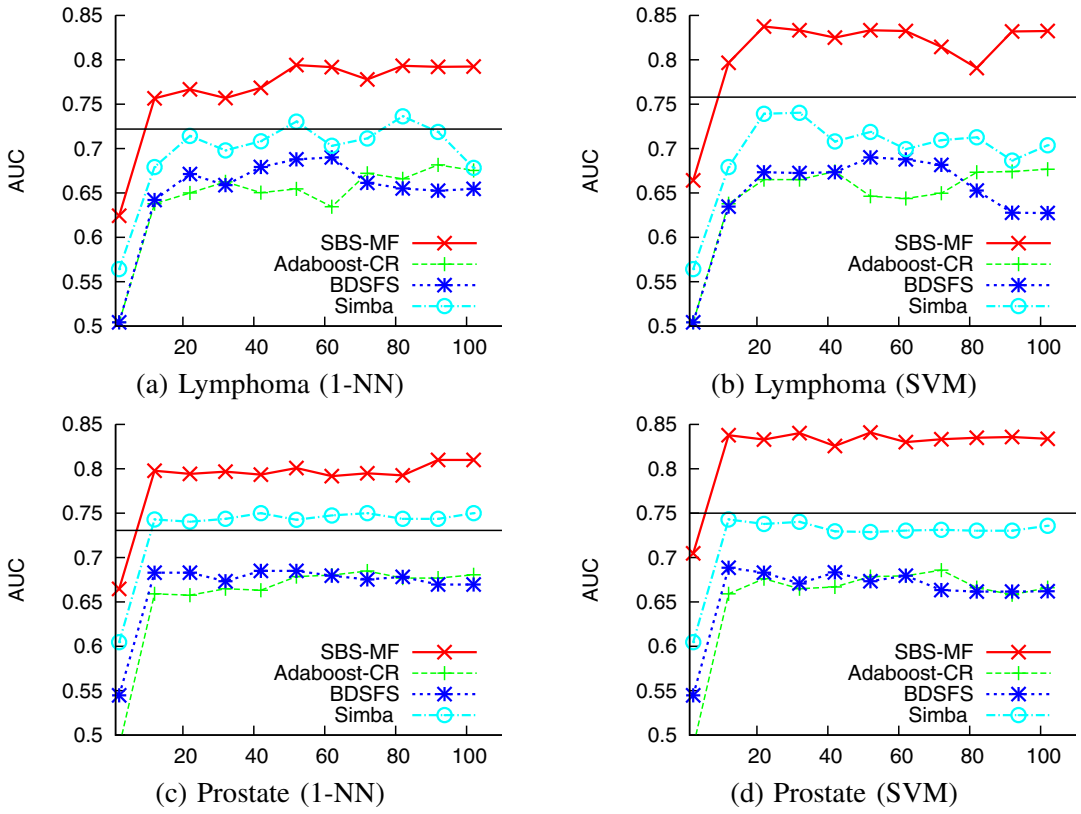


Figure 3: AUC vs. the number of ranked features used for training the two classifiers, with the four feature selection methods on Lymphoma data set a) with 1-NN and b) with SVM, and on Prostate data set c) with 1-NN and d) with SVM.

[21] E. Petricoin. Serum proteomic patterns for detection of prostate cancer. *Journal of the National Cancer Institute*, pages 1576–1578, 2002b.

[22] J. R. Quinlan. Bagging, boosting, and c4.5. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 725–730, 1996.

[23] L. Reyzin and R.E. Schapire. How boosting the margin can also boost classifier complexity. In *In Proceedings of the 23rd International Conference on Machine Learning*, pages 753–760, 2006.

[24] E. Schapire, Y. Freund, P. Bartlett, and W. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Ann. Stat.*, 26(5):1651–1686, 1998.

[25] R. Schapire. The boosting approach to machine learning: An overview. In *Nonlinear Estimation and Classification*. 2003.

[26] M. Shipp. Diffuse large b-cell lymphoma outcome prediction by gene expression profiling and supervised machine learning. *Nature Medicine*, pages 68–74, 2002.

[27] P. Silapachote, D. Karuppiah, and A. Hanson. Feature selecetion using adaboost for face expression recognistion. *Proceedings of the Fourth IASTED International Conference on Visualization, Imaging, and Imag Processing*, pages 84–89, 2004.

[28] M. Tsuchiya and H. Fujiyoshi. A method of feature selection using contribution ratio based on boosting. In *ICPR*, pages 1–4, 2008.

[29] N. Wiratunga, I. Koychev, and S. Massie. Feature selection and generalisation for retrieval of textual cases. In *Proceedings of 7th European Conference on Case-based Reasoning (ECCBR04), VOLUME 3155*, pages 806–820, 2004.

[30] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. second edition, 2005.