Semi-Supervised Boosting for Multi-Class Classification

Hamed Valizadegan, Rong Jin, and Anil K. Jain

Department of Computer Science and Engineering,
Michigan State University,
valizade@cse.msu.edu, rongjin@cse.msu.edu, jain@cse.msu.edu

Abstract. Most semi-supervised learning algorithms have been designed for binary classification, and are extended to multi-class classification by approaches such as one-against-the-rest. The main shortcoming of these approaches is that they are unable to exploit the fact that each example is only assigned to one class. Additional problems with extending semi-supervised binary classifiers to multi-class problems include imbalanced classification and different output scales of different binary classifiers. We propose a semi-supervised boosting framework, termed **Multi-Class Semi-Supervised Boosting (MCSSB)**, that directly solves the semi-supervised multi-class learning problem. Compared to the existing semi-supervised boosting methods, the proposed framework is advantageous in that it exploits both classification confidence and similarities among examples when deciding the pseudo-labels for unlabeled examples. Empirical study with a number of UCI datasets shows that the proposed MCSSB algorithm performs better than the state-of-the-art boosting algorithms for semi-supervised learning.

Key words: Semi-supervised learning, Multi-Class Classification, Boosting

1 Introduction

Semi-supervised classification combines the hidden structural information in the unlabeled examples with the explicit classification information of labeled examples to improve the classification performance. Many semi-supervised learning algorithms have been studied in the literature. Examples are density based methods [1, 2], graph-based algorithms [3–6], and boosting techniques [7, 8]. Most of these methods were originally designed for two class problems. However, many real-world applications, such as speech recognition and object recognition, require multi-class categorization. To adopt a binary (semi-supervised) learning algorithm to problems with more than two classes, the multi-class problems are usually decomposed into a number of independent binary classification problems using techniques such as one-versus-the-rest, one-versus-one, and error-correcting output coding [9]. The main shortcoming with this approach is that the resulting binary classification problems are *independent* binary class problems. As a result,

it is unable to exploit the fact that each example can only be assigned to one class. This issue was also pointed out in the study with multi-class boosting [10]. In addition, since every binary classifier is trained independently, their outputs may be on different scales, making it difficult to compare them [11]. Though calibration techniques [12] can be used to alleviate this problem in supervised classification, it is rarely used in semi-supervised learning due to the small number of labeled training examples. Moreover, techniques like one-versus-the-rest, where the examples of one class are considered against the examples of all the other classes, could lead to the imbalanced classification problem. Although a number of techniques have been proposed for supervised learning in multi-class problems [13, 14, 10], they have not addressed semi-supervised multi-class learning problems, which is the focus of this study.

Boosting is a popular learning method because it provides a general framework for improving the performance of any given learner by constructing an ensemble of classifiers. Several boosting algorithms have been proposed for semi-supervised learning [15, 7, 8]. They essentially operate like self-training where the class labels of unlabeled examples are updated iteratively: a classifier trained by a small number of labeled examples is initially used to predict the pseudo-labels for unlabeled examples; a new classifier is then trained by both labeled and pseudo-labeled examples; the processes of training classifiers and predicting pseudo-labels are altered iteratively till stopping criterion is reached. The main drawback with this approach is that it relies solely on the pseudo-labels predicted by the classifiers learned so far when generating new classifiers. Given the possibility that pseudo-labels predicted in the first few steps of boosting could be inaccurate, the resulting new classifiers may also be unreliable. This problem was addressed in [8] by introduction of a local smoothness regularizer. However, since all the existing semi-supervised boosting algorithms are designed for binary classification, they will still suffer from the aforementioned problems when applied to multi-class problems. In this paper, we develop a semi-supervised boosting framework, termed *Multi-Class Semi-Supervised Boosting (MCSSB)*, that is designed for multi-class semi-supervised learning problems. By directly solving a multi-class problem, we avoid the problems that arise when converting a multi-class classification problem into a number of binary ones. Moreover, unlike the existing semi-supervised boosting methods that only assign pseudo-labels to the unlabeled examples with high classification confidence, the proposed framework decides the pseudo labels for unlabeled examples based on both the classification confidence and the similarities among examples. It therefore effectively explores both the manifold assumption and the clustering assumption for semi-supervised learning. Empirical study with UCI datasets shows the proposed algorithm performs better than the state-of-the-art algorithms for semi-supervised learning.

2 Related Work

Most semi-supervised learning algorithms can be classified into three categories: graph-based, density-based, and boosting-based.

Semi-supervised SVMs (S^3VMs) or Transductive SVMs (TSVMs) are the semi-supervised extensions to Support Vector Machines (SVM). They are essentially density-based methods and assume that decision boundaries should lie in the sparse regions. Although finding an exact S^3VM is NP-complete [16], there are many approximate solutions for it [1, 17–19, 2]. Except for [19], these methods are designed for binary semi-supervised learning. The main drawback with [19] is its high computational cost due to the semi-definite programming formulation.

Graph-based methods aim to predict class labels that are smooth on the graph of unlabeled examples. These algorithms differ in how to define the smoothness of class labels over a graph. Example graph-based semi-supervised learning approaches include Mincut [3], Harmonic function [4], local and global consistency [5], and manifold regularization [6]. Similar to density based methods, most graph-based methods are mainly designed for binary classification.

Semi-supervised boosting methods such as SSMBoost [15] and Assemble [7] are direct extensions of Adaboost [20]. In [8], a local smoothness regularizer is introduced to improve the reliability of semi-supervised boosting. Unlike the existing approaches for semi-supervised boosting that solve 2-class problems, our study focuses on semi-supervised boosting for multi-class classification.

3 Multi-Class Semi-supervised Learning

3.1 Problem Definition

Let $\mathcal{D} = (x_1, .., x_N)$ denote the collection of N examples. Assume that the first N_l examples are labeled by $y_1, ..., y_{N_l}$, where $y_i = (y_i^1, ..., y_i^m) \in \{0, +1\}^m$ is a binary vector and m is the number of classes. $y_i^k = +1$ when x_i is assigned to the kth class, and $y_i^k = 0$, otherwise. Since we are dealing with a multi-class problem, we have $\sum_{k=1}^{m} y_i^k = 1$, i.e., each example x_i is only assigned to one and only one class. We denote by $\hat{y}_i = (\hat{y}_i^1, \ldots, \hat{y}_i{}^m) \in \mathbb{R}^m$ the predicted class labels (or confidence) for example x_i, and by $\hat{Y} = (\hat{y}_1^\top, \ldots, \hat{y}_N^\top)^\top$ the predicted class labels for all the examples[1]. Let $S = [S_{i,j}]_{N \times N}$ be the similarity matrix where $S_{i,j} = S_{j,i} \geq 0$ is the similarity between x_i and x_j. For the convenience of discussion, we set $S_{i,i} = 0$ for any $x_i \in \mathcal{D}$, a convention that is commonly used by many graph-based approaches. Our goal is to compute $\hat{y}_i$ for the unlabeled examples with the assistance of similarity matrix S and $Y = (y_1^\top, \ldots, y_{N_l}^\top)^\top$.

3.2 Design of Objective Function

The goal of semi-supervised learning is to combine labeled and unlabeled examples to improve the classification performance. Therefore, we design an objective function that consists of two terms: (a) F_u that measures the consistency between the predicted class labels $\hat{Y}$ of unlabeled examples and the similarity matrix S,

[1] $x^\top$ is the transpose of matrix(vector) x.

and (b) F_l that measures the consistency between the predicted class labels $\hat{Y}$ and true labels Y. Below we discuss these two terms in detail.

Given two examples x_i and x_j, we first define the similarity $Z^u_{i,j}$ based on their predicted class labels $\hat{y}_i$ and $\hat{y}_j$:

$$Z^u_{i,j} = \sum_{k=1}^{m} \frac{\exp(\hat{y}_i^k)}{\sum_{k'=1}^{m} \exp(\hat{y}_i^{k'})} \frac{\exp(\hat{y}_j^k)}{\sum_{k'=1}^{m} \exp(\hat{y}_j^{k'})} = \sum_{k=1}^{m} b_i^k b_j^k = b_i^\top b_j \tag{1}$$

where $b_i^k = \exp(\hat{y}_i^k)/(\sum_{k'=1}^{m} \exp(\hat{y}_i^{k'}))$ and $b_i = (b_i^1, \ldots, b_i^m)$. Note that b_i^k can be interpreted as the probability of assigning x_i to class k, and $Z^u_{i,j}$, the cosine similarity between b_i and b_j, can be interpreted as the probability of assigning x_i and x_j to the same class. We emphasize it is important to use b_i^k, instead of $\exp(\hat{y}_i^k)$, for computing $Z^u_{i,j}$ because normalization in b_i^k allows us to enforce the requirement that each example is assigned to a single class, a key feature of multi-class learning.

Let $Z^u = [Z^u_{i,j}]$ be the similarity matrix based on the predicted labels. To measure the inconsistency between this similarity and the similarity matrix S, we define F_u as the distance between the matrices Z^u and S using the Bregman matrix divergence [21], i.e.,

$$F_u = \varphi(Z^u) - \varphi(S) - \mathrm{tr}((Z^u - S)^\top \nabla\varphi(S)), \tag{2}$$

where $\varphi : \mathbb{R}^{N\times N} \rightarrow \mathbb{R}$ is a convex matrix function. By choosing $\varphi(X) = \sum_{i,j=1}^{N} X_{i,j}(\log X_{i,j} - 1)$ [21], F_u is written as [2]

$$F_u = \sum_{i,j=N_l+1}^{N} \left(S_{i,j} \log \frac{S_{i,j}}{Z^u_{i,j}} + Z^u_{i,j} - S_{i,j} \right) \tag{3}$$

By assuming that $\sum_{i,j=1+N_l}^{N} Z^u_{i,j} \approx \sum_{k=1}^{m} N_k^2$ and $\log x \approx x - 1$, where N_k is the number of examples assigned to class k, we simplify the above expression as $F_u \approx \sum_{i,j=N_l+1}^{N} S^2_{i,j}/Z^u_{i,j}$. Since $S^2_{i,j}$ could be viewed as a general similarity measurement, we replace $S^2_{i,j}$ with $S_{i,j}$ and simplify F_u as

$$F_u \approx \sum_{i,j=N_l+1}^{N} \frac{S_{i,j}}{Z^u_{i,j}} = \sum_{i,j=N_l+1}^{N} \frac{S_{i,j}}{\sum_{k=1}^{m} b_i^k b_j^k} \tag{4}$$

Remark We did not use $\varphi(X) = \sum_{i,j=1}^{N} X^2_{i,j}$ [21], which will result in $F_u = \sum_{i,j=N_l+1}^{N} (Z^u_{i,j} - S_{i,j})^2$. This is because the value of $Z^u_{i,j}$ and $S_{i,j}$ may be on different scales.

Similarly, we define the similarity between a labeled example x_i and an unlabeled example x_j based on their class assignments as follows

$$Z^l_{i,j} = \sum_{k=1}^{m} y_i^k b_j^k, \tag{5}$$

[2] We can only consider the sub-matrices related to unlabeled examples when defining F_u.

and the inconsistency measure F_l between the labeled and unlabeled examples as follows:

$$F_l = \sum_{i=1}^{N_l} \sum_{j=N_l+1}^{N} \frac{S_{i,j}}{Z^l_{i,j}} = \sum_{i=1}^{N_l} \sum_{j=N_l+1}^{N} \frac{S_{i,j}}{\sum_{k=1}^{m} y_i^k b_j^k} \tag{6}$$

Finally, we linearly combine F_l and F_u to form the objective function:

$$F = F_u + CF_l \tag{7}$$

where C weights the importance of F_l. It is set to $10,000$ in our experiments to emphasize F_l [3]. Given the objective function F in (7), our goal is to find solution $\hat{Y}$ that minimizes F.

3.3 Multi-Class Boosting Algorithm

In this section, we present a boosting algorithm to solve the optimization problem in (7). Following the architecture of boosting model, we incrementally add weak learners to obtain a better classification model. We denote by H_i^k the solution that is obtained for $\hat{y}_i^k$ so far, and by $h_i^k \in \{0, 1\}$ the prediction made by the incremental weak classifier that needs to be learned. Then, our goal is to find $h_i^k, i = N_l + 1, \ldots, N, k = 1, \ldots, m$ and a combination weight α such that the new solution $\widetilde{H}_i^k = H_i^k + \alpha h_i^k$ significantly reduces the objective function F in Equation 7. For the convenience of discussion, we use symbol $\widetilde{\ }$ to denote the quantities (e.g., $\widetilde{F}$) associated with the new solution $\widetilde{H}$.

The key challenge in optimizing F with respect to h_i^k and α is that these two quantities are coupled with each other and therefore the solution of one variable depends on the solution of the other. Our strategy to solve the optimization problem is to first upper bound F with a simple convex function in which the optimal solution for h_i^k can be obtained without knowing the solution to α. Given the solution to h_i^k, we then compute the optimal solution for α. Below we give details for these two steps.

First, the following lemma allows us to decouple the interaction between α and h_i^k within $Z^u_{i,j}$ and $Z^l_{i,j}$

Lemma 1.

$$\frac{1}{\widetilde{Z}^u_{i,j}} \leq \frac{1 + e^{6\alpha} + e^{-6\alpha}}{3Z^u_{i,j}} + \frac{e^{6\alpha} - 1}{3Z^u_{i,j}} \left(\sum_{k=1}^{m} (b_i^k - \tau_{i,j}^k) h_i^k \right) \tag{8}$$

$$\frac{1}{\widetilde{Z}^l_{i,j}} \leq \frac{1 + e^{6\alpha} + e^{-6\alpha}}{3Z^l_{i,j}} + \frac{e^{6\alpha} - 1}{6} \sum_{k=1}^{m} h_i^k \phi_{i,j}^k \tag{9}$$

where

$$\tau_{i,j}^k = \frac{b_i^k b_j^k}{\sum_{k'=1}^{m} b_i^{k'} b_j^{k'}}, \quad \phi_{i,j}^k = \sum_{k'=1}^{m} y_j^k \frac{b_i^k}{b_i^{k'}} - \frac{y_i^k}{b_i^k} \tag{10}$$

[3] The algorithm is quite stable with different values of C bigger than 1000 according to our experiment.

The proof of Lemma 1 can be found in Appendix A. Using Lemma 1, we derive an upper bound for $\widetilde{F}$ in the following theorem.

Theorem 1

$$\widetilde{F} \leq F\frac{1+\exp(6\alpha)+\exp(-6\alpha)}{3} + \frac{\exp(6\alpha)-1}{3}\sum_{i=N_l+1}^{N}\sum_{k=1}^{m} h_i^k(\alpha_i^k + C\beta_i^k) \quad (11)$$

where α_i^k *and* β_i^k *are defined as follows:*

$$\alpha_i^k = \sum_{j=N_l+1}^{N} \frac{S_{i,j}(b_i^k - \tau_{i,j}^k)}{Z_{i,j}^u}, \quad \beta_i^k = \frac{1}{2}\sum_{j=1}^{N_l} S_{i,j}\phi_{i,j}^k \quad (12)$$

Theorem 1 can be directly verified by replacing $1/\widetilde{Z}_{i,j}^u$ and $1/\widetilde{Z}_{i,j}^l$ in (7) with (8) and (9). Note that the bound in Theorem 1 is tight because by setting $\alpha = 0$, we have $\widetilde{H} = H$ and the inequality in Equation 11 is reduced to an equality. The key feature of the bound in Equation 11 is that the optimal solution for h_i^k can be obtained without knowing the solution for α. This is summarized by the following theorem.

Theorem 2 *The optimal solution for* h_i^k *that minimizes the upper bound of* $\widetilde{F}$ *in Equation 11 is*

$$h_i^k = \begin{cases} 1 \ k = \arg\max_{k'}(\alpha_i^{k'} + C\beta_i^{k'}) \\ 0 \ \textit{otherwise} \end{cases} \quad (13)$$

It is straightforward to verify the result in Theorem 2.

We then proceed to find solution for α given the solution for h_i^k. The following lemma provides a tighter bound for solving α in F [4].

Lemma 2.

$$\widetilde{F} - F \leq (e^{2\alpha}-1)(A_u + CA_l) + (e^{-2\alpha}-1)(B_u + CB_l) \quad (14)$$

where

$$A_u = \sum_{i,j=N_l+1}^{N} \frac{S_{i,j}}{Z_{i,j}^u}\sum_{k=1}^{m} h_i^k b_i^k \quad (15)$$

$$A_l = \frac{1}{2}\sum_{i=1}^{N_l}\sum_{j=N_l+1}^{N} S_{i,j}\sum_{k,k'=1}^{m}\frac{y_i^k}{b_j^k} b_j^{k'} h_j^{k'} \quad (16)$$

$$B_u = \sum_{i,j=N_l+1}^{N} \frac{S_{i,j}}{Z_{i,j}^u}\sum_{k=1}^{m} h_i^k \tau_{i,j}^k \quad (17)$$

$$B_l = \frac{1}{2}\sum_{i=1}^{N_l}\sum_{j=N_l+1}^{N} S_{i,j}\sum_{k=1}^{m} y_i^k \frac{h_j^k}{b_j^k} \quad (18)$$

[4] Note that this tighter bound can not be used to derive h_i^k

Algorithm 1 MCSSB: Multi-Class Semi-Supervised Boosting Algorithm

Input:
- D: The set of examples; the first N_l examples are labeled.
- s: the number of sampled examples from $(N - N_l)$ unlabeled examples
- T: the maximum number of iterations

for $i = 1$ **to** T
- Compute α_i^k and β_i^k for every example as given in Equation 12.
- Assign each unlabeled example x_i to class $k_i^* = \arg\max_k(\alpha_i^k + C\beta_i^k)$ and weight $w_i = \alpha_i^{k_i^*} + C\beta_i^{k_i^*}$
- Sample s unlabeled examples using a distribution that is proportional to w_i
- Train a multi-class classifier $h(x)$ using the labeled examples and the sampled unlabeled examples with assigned classes
- Predict h_i^k for unlabel examples using $h(x)$, and compute α using Equation 19. Exit the loop if $\alpha \leq 0$.
- $H(x) \leftarrow H(x) + \alpha h(x)$

The proof of Lemma 2 can be found in Appendix B. Using Lemma 2, Theorem 3 gives the optimal solution for α.

Theorem 3 *The optimal α that minimizes the upper bound of $\widetilde{F}$ in Equation 14 is*

$$\alpha = \frac{1}{4}\log\left(\frac{B_u + CB_l}{A_u + CA_l}\right) \tag{19}$$

Algorithm 1 summarizes the proposed boosting algorithm for multi-class semi-supervised learning. Several issues need to be pointed out: (a) w_i, the weight for the ith unlabeled example, is guaranteed to be non-negative. This is because $\sum_{k=1}^m \alpha_i^k + C\beta_i^k = 0$ and therefore $w_i = \max_k(\alpha_i^k + C\beta_i^k) \geq 0$; (b) we adopt the sampling approach to train a weak classifier. In our experiments, the number of sampled examples at each iteration is set as $s = \max(20, N/5)$; (c) the maximum number of iteration T is set to be 50 as suggested in [22][5].

Theorem 4 shows that the proposed boosting algorithm reduces the objective function F exponentially.

Theorem 4 *The objective function after T iterations, denoted by F^T, is bounded as follows:*

$$F^T \leq F^0 \exp\left(-\sum_{t=1}^{T} \frac{(\sqrt{A_u^t + CA_l^t} - \sqrt{B_u^t + CB_l^u})^2}{F^{t-1}}\right) \tag{20}$$

where A_u, A_l, B_u and B_l are defined in Lemma 2.

[5] We run the algorithm with much larger numbers of iterations and find that both the objective function and the classification accuracy remains essentially the same after 50 iterations. We, therefore, set the number of iterations to be 50 to save the computational cost.

Proof. Using Lemma 2 and Theorem 3, we have

$$\begin{aligned}\widetilde{F} - F &\leq \sqrt{\frac{B_u + CB_l}{A_u + CA_l}}(A_u + CA_l) + \sqrt{\frac{A_u + CA_l}{B_u + CB_l}}(B_u + CB_l) - (A_u + CA_l + B_u + CB_l) \\ &= -\left(\sqrt{A_u + CA_l} - \sqrt{B_u + CB_l}\right)^2,\end{aligned}$$

which is equivalent to

$$\begin{aligned}\frac{\widetilde{F}}{F} &\leq 1 - \frac{\left(\sqrt{A_u + CA_l} - \sqrt{B_u + CB_l}\right)^2}{F} \\ &\leq \exp\left(-\frac{\left(\sqrt{A_u + CA_l} - \sqrt{B_u + CB_l}\right)^2}{F}\right) \qquad (21)\end{aligned}$$

The above inequality follows from $\exp(x) \geq 1 + x$. We rewrite F^T as

$$F^T = F^0 \prod_{t=1}^{T} (F^t / F^{t-1})$$

By substituting F^t/F^{t-1} with the bound in Equation 21, we have the result in the theorem.

4 Experiments

In this section, we present our empirical study on a number of UCI data sets. We refer to the proposed semi-supervised multi-class boosting algorithm as **MCSSB**. In this study, we aim to show that (1) MCSSB can improve several available multi-class classifiers with unlabeled examples, (2) MCSSB is more effective than the existing semi-supervised boosting algorithms, and (3) MCSSB is robust to the model parameters and the number of labeled examples. It is important to note that it is not our intention to show that the proposed semi-supervised multi-class boosting algorithm always outperforms the other semi-supervised learning algorithms. Instead, our objective is to demonstrate that the proposed semi-supervised boosting algorithm is able to effectively improve the accuracy of different supervised multi-class learning algorithms using the unlabeled examples. Hence, the empirical study is focused on a comparison with the existing semi-supervised boosting algorithms, rather than a wide range of semi-supervised learning algorithms.

We follow [7] and use *Decision Tree* and *Multi-Layer Perceptron (MLP)* as the base multi-class classifiers in our study. In order to create weak classifiers as most boosting algorithms do, we restrict the levels of decision tree to be two, and the structure of MLP to be one hidden layer with two nodes. We create an instance of semi-supervised multi-class learning boosting algorithm for each base classifier, denoted by *MCSSB-Tree* and *MCSSB-MLP*, respectively. We compare

Table 1. Description of data sets.

	# samples	# attributes	# Classes
Balance	625	4	3
Glass	214	9	6
Iris	150	4	3
Wine	178	13	3
Car	1728	6	4
Vowel	990	14	11
Contraceptive	1473	9	3
Dermatology	358	34	6
Ecoli	336	7	8
Flag	194	28	8
Segmentation	2310	19	7
pendigit	3498	16	10
Optdigits	1797	64	10
Soybean	686	35	19
Waves	5000	21	3
Yeast	1484	8	10
Zoo	101	16	7

the proposed semi-supervised boosting algorithm to *ASSEMBLE*, a state-of-the-art semi-supervised boosting. Similar to MCSSB, two instances of classifiers are created for ASSEMBLE using decision tree and MLP base classifiers, denoted by *Assemble-Tree* and *Assemble-MLP*, respectively. A Gaussian kernel is used as the measure for similarity in *MCSSB-Tree* and *MCSSB-MLP* with kernel width set to be 15% of the range of the distance between examples [6] for all the experiments, as suggested in [23]. Table 1 summarizes seventeen benchmark data sets from the UCI data repository used in this study.

4.1 Evaluation of Classification Performance

Many binary semi-supervised learning studies assume a very small number of labeled examples, e.g. less that 1% of the total number of examples. This setup is difficult to be applied to multi-class cases since it may result in some classes with no labeled examples. As an example, consider Glass data set in Table 1, where 1% of the examples will provide us with only two labeled examples which will cover at most two classes. This motivated us to run two different sets of experiments to evaluate the performance of the proposed algorithm. In the first set of experiments, we assume that 5% of examples are labeled and in the second case we assume that 10% of examples are labeled [7]. We repeat each experiment

[6] i.e. $0.15 \times (d_{\max} - d_{\min})$, where $d_{\min}$ and $d_{\max}$ are minimum and maximum distance between examples

[7] Our experience with one labeled example per class shows similar results. We omit the result due to space limitation

Table 2. Classification accuracy with 5% of samples as the labeled set(N_l)

	Tree	MLP	Assemble-tree	Assemble-MLP	MCSSB-tree	MCSSB-MLP
Balance	65.0±0.9	82.0±1.4	65.0±0.9	82.2±1.0	72.5±1.0	**83.2±0.7**
Glass	39.7±1.6	**40.3±1.8**	39.7±1.6	**41.0±1.8**	**40.1±1.2**	**40.4±1.1**
Iris	32.3±0.1	71.6±2.5	32.4±0.1	74.3±3.8	**77.4±2.6**	74.0±3.0
Wine	33.4±1.3	70.0±2.7	62.4±3.6	66.0±2.8	**78.2±3.5**	75.0±3.1
Car	80.5±0.5	76.4±1.0	80.5±0.5	76.8±0.5	**81.6±0.3**	77.7±0.5
Vowel	**27.5±1.1**	17.9±0.6	26.0±1.1	18.8±0.6	**28.1±1.1**	19.3±0.6
Contraceptive	**47.3±0.8**	45.0±0.7	**47.2±0.7**	44.1±0.9	**47.3±0.8**	45.4±0.6
Dermatology	53.6±2.2	48.3±2.0	53.4±2.3	46.2±2.4	**77.0±1.2**	68.0±1.8
Ecoli	57.8±1.7	**61.5±1.8**	57.8±1.7	59.3±1.7	52.0±2.2	56.2±1.3
Flag	23.5±1.3	22.1±1.1	23.5±1.3	25.1±1.2	**30.3±1.1**	26.0±1.2
Segmentation	**47.6±2.1**	43.2±1.3	45.9±2.1	44.8±1.5	**47.6±2.1**	44.5±1.6
pendigit	33.8±1.5	30.0±1.0	32.5±1.5	29.8±0.8	**59.3±1.1**	54.7±1.7
Optdigits	**33.0±1.8**	23.3±1.0	30.6±1.4	23.3±0.7	**33.0±1.8**	21.9±0.6
Soybean	37.2±1.3	25.0±0.9	35.2±1.4	25.4±1.1	**42.4±1.1**	33.4±0.9
Waves	65.0±0.3	73.3±1.7	65.0±0.3	73.3±1.8	65.4±0.3	**74.8±0.8**
Yeast	**43.6±0.7**	40.2±0.6	**43.4±0.6**	40.4±0.9	**42.7±0.9**	39.4±1.2
Zoo	32.6±2.7	39.8±3.0	41.7±4.0	40.7±3.0	**59.0±2.7**	56.9±2.6

20 times and report both the mean and standard deviation of the classification accuracy.

Table 2 shows the result of different algorithms for the first experiment (5% labeled examples) with the performance of the best approach for each dataset highlighted by bold font. First, notice that MCSSB significantly[8] improves the accuracy of both decision tree and MLP for 10 of the 17 data sets. For six data sets, including 'Glass", "Vowel", "Contraceptive", "Segmentation", "Optdigits", and "Yeast", the classification accuracy remains almost unchanged after applying MCSSB to the base multi-class learning algorithm. Only for data set "Ecoli", MCSSB-MLP performs significantly worse than MLP. Note that for several data sets, the improvement made by the MCSSB is dramatic. For instance, the classification accuracy of decision tree is improved from 32.8% to 77.4% for data set "Iris", and from 33.4% to 78.2% for data set "Wine"; the classification accuracy of MLP is improved from 48.3% to 68.0% for data set "Dermatology", and from 30.0% to 54.7% for data set "pendigit". Second, when compared to ASSEMBLE, we found that the proposed algorithm significantly outperforms ASSEMBLE for 14 of the 16 data sets for both decision tree and MLP. Only for data set "Ecoli", ASSEMBLE performs better than MCSSB when using MLP as the base classifier. The key differences between MCSSB and ASSEMBLE is that MCSSB is not only specially designed for multi-class classification, it does not solely rely on the pseudo-labels obtained in the iterations of boosting algorithm. Thus, the suc-

[8] The variance reported in the table clearly shows the advantage of our method compared to the baseline.

Table 3. The accuracy of different methods with 10% labeled examples

	Tree	MLP	Assemble-tree	Assemble-MLP	MCSSB-tree	MCSSB-MLP
Balance	67.7±0.7	**86.0±1.1**	67.8±0.7	**87.0±0.5**	69.5±1.0	**86.6±0.6**
Glass	**46.9±1.7**	42.7±1.5	**46.8±1.7**	**45.4±1.6**	**45.3±1.5**	43.8±1.7
Iris	68.5±2.4	79.2±3.0	68.7±2.4	77.2±2.6	79.7±2.7	**84.1±2.3**
Wine	73.0±2.7	78.2±2.4	73.0±2.7	74.7±3.0	81.8±1.2	**83.2±1.2**
Car	**83.2±0.5**	77.1±0.6	83.1±0.5	78.4±0.7	**83.7±0.5**	78.0±0.4
Vowel	**27.2±1.1**	21.2±0.5	24.8±1.0	21.7±0.9	**27.6±1.0**	22.8±1.1
Contraceptive	42.6±0.0	31.9±3.2	42.6±0.0	28.6±3.5	**81.4±1.8**	71.0±2.8
Dermatology	64.6±1.6	48.7±2.0	63.9±1.5	50.4±2.7	**78.4±1.4**	65.6±2.1
Ecoli	**65.1±1.7**	**64.8±1.6**	**65.0±1.7**	**65.2±1.4**	61.4±1.9	**64.0±1.6**
Flag	**38.7±1.5**	29.2±1.3	**38.5±1.4**	30.3±1.5	**38.9±1.4**	28.3±1.1
Segmentation	**48.5±2.3**	46.3±1.5	46.4±2.0	42.9±1.3	**48.5±2.3**	46.8±1.7
Pendigits	36.5±1.5	31.7±0.8	34.0±1.4	31.3±0.9	**57.7±1.2**	52.2±1.4
Optdigits	**33.9±1.3**	26.3±0.8	31.9±1.1	27.5±0.6	**33.9±1.3**	27.6±1.1
Soybean	37.7±1.1	33.4±1.0	37.2±1.6	31.7±1.0	**42.8±1.1**	39.9±1.2
Waves	65.5±0.3	76.6±1.4	65.5±0.3	76.7±2.3	65.5±0.3	**79.3±0.9**
Yeast	**47.9±0.7**	41.0±1.1	**47.5±0.6**	41.7±0.9	**47.8±0.6**	41.6±1.1
Zoo	52.0±1.8	51.8±2.5	52.0±1.8	50.8±2.3	**74.3±1.9**	71.9±1.5

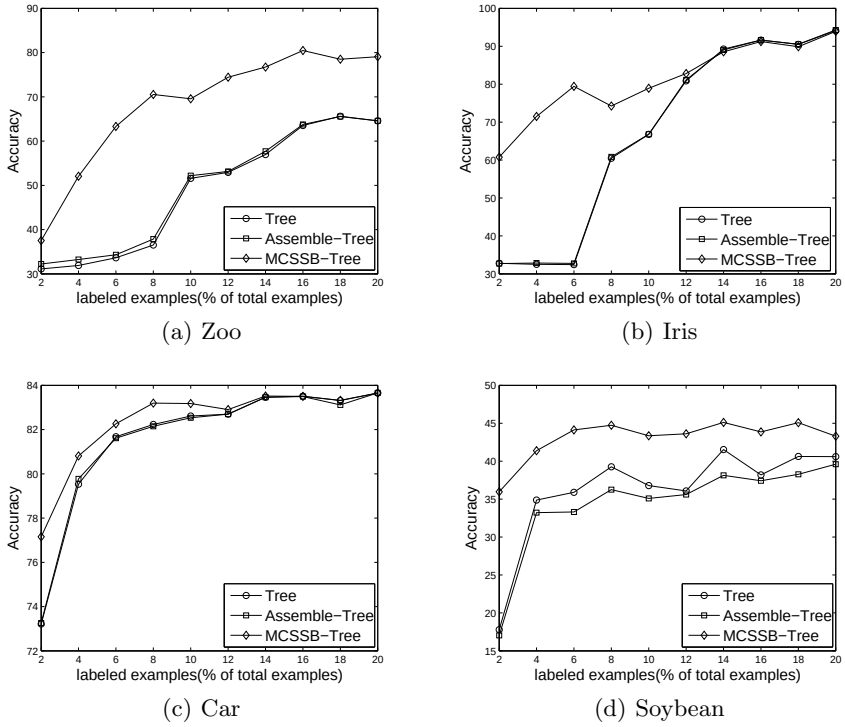
cess of MCSSB indicates the importance of designing semi-supervised learning algorithms for multi-class problems.

Table 3 shows the performance of different algorithms when 10% of the examples are labeled. Similar to the previous case, MCSSB outperforms both the base classifiers and the ASSEMBLE method for 8 of the 17 data sets. For the rest of data sets, including "Balance", "Glass","Car", "Vowel", "Ecoli", "Flag", "Segmentation", "Optdigits", "Waves ", and "Yeast", the classification accuracy remains unchanged after applying MCSSB to the base supervised learning algorithms. Noitce that the amount of improvement in this case is less than the case with 5% labeled examples. This is because as the number of labeled examples increases, the improvement gained by a semi-supervised learning algorithm decreases. Moreover, notice that similar to the case of 5% labeled examples, ASSEMBLE is not able to improve the performance of the base classifier. Based on the above observation, we conclude that the proposed semi-supervised boosting algorithm is able to effectively exploit the unlabeled data to improve the performance of supervised multi-class learning algorithms.

4.2 Sensitivity to the number of labeled examples

To study the sensitivity of MCSSB to the number of unlabeled examples, we run MCSSB and the baselines by varying the number of labeled examples from 2% to 20% of the total number of examples. Figure 1 shows the result of this experiment on 4 of the datasets when the base classifier is tree[9]. Notice that as

[9] We omit the result for other data sets and MLP as the base classifier due to space limitation.



(a) Zoo (b) Iris

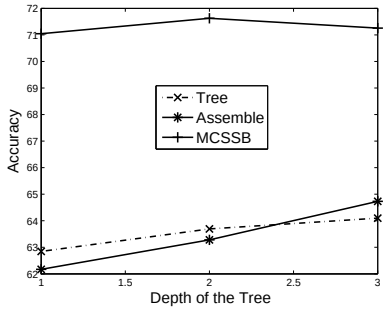
(c) Car (d) Soybean

Fig. 1. Sensitivity of MCSSB to number of labels

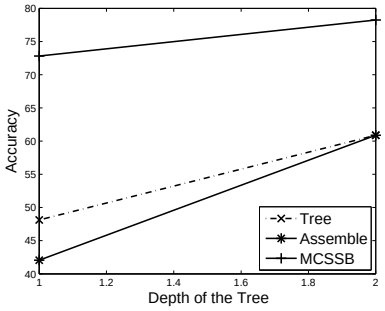
the number of labeled examples increases, the performance of difference methods improves. But MCSSB keeps its superiority for almost all the cases when compared to both the base classifier and the ASSEMBLE algorithm. We also observe that overall ASSEMBLE is unable to make significant improvement over the base classifier regardless of the number of labeled examples. More surprisingly, for data set "Soybean", ASSEMBLE performs worse than the base classifier. These results indicate the challenge in developing boosting algorithms for semi-supervised multi-class learning. Compared to ASSEMBLE that relies on the classification confidence to decide the pseudo labels for unlabeled examples, MCSSB is more reliable since it exploits both the classification confidence and similarities among examples when determining the pseudo labels.

4.3 Sensitivity to Base Classifier

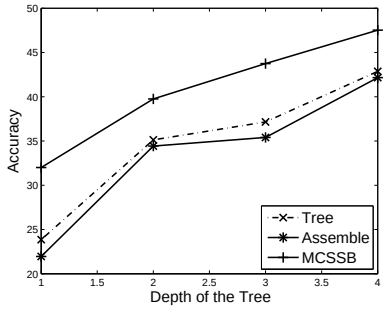
In this section, we focus on examining the sensitivity of MCSSB to the complexity of base classifiers. This will allow us to understand the behavior of the proposed semi-supervised boosting algorithm for both weak classifiers and strong classifiers. To this end, we use decision tree with varying number of levels as the



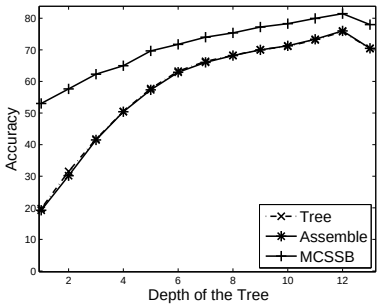
(a) Balance with 5% labeled examples



(b) Dermatology with 5% labeled examples



(c) soybean with 5% labeled examples



(d) Pendigits with 5% labeled examples

Fig. 2. Sensitivity of MCSSB to depth of the tree

base classifier. Only the results for datasets Balance, Dermatology, Soybean, and Pendigit are reported in this study because these were the only four data sets for which the fully grown decision tree had more than two levels.

Figure 2 shows the classification accuracy of Tree, ASSEMBLE-tree and MCSSB-tree when we vary the number of levels in decision tree. Notice that in each case, the maximum number of level in the plot for each data set is set to the tree fully grown for that data set. It is not surprising that overall the classification accuracy is improved with increasing number of levels in decision tree. We also observe that MCSSB is more effective than ASSEMBLE for decision trees with different complexity.

5 Conclusion

Unlike many existing semi-supervised learning algorithms that focus on binary classification problems, we address multi-class semi-supervised learning directly. We have proposed a new framework, termed multi-class semi-supervised boosting (MCSSB), that is able to improve the classification accuracy of any given base

multi-class classifier. We showed that our proposed framework is able to improve the performance of a given classifier much better than Assemble, a well-known semi-supervised boosting algorithm, on a large set of UCI datasets. We also show that MCSSB is very robust to the choice of base classifiers and the number of labeled examples.

6 Acknowledgements

The work was supported in part by the National Science Foundation (IIS-0643494) and Office of Naval Research (N00014-07-1-0255). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF and ONR.

Appendix A: Proof of Lemma 1

Proof. Bound in Equation (8) can be derived as follows:

$$\begin{aligned}
\frac{1}{\widetilde{Z}^u_{i,j}} &= \frac{1}{\sum_{k'=1}^m \widetilde{b}_i^k \widetilde{b}_j^k} = \frac{(\sum_{k'=1}^m b_i^{k'} \exp(\alpha h_i^{k'}))(\sum_{k'=1}^m b_i^{k'} \exp(\alpha h_j^{k'}))}{\sum_{k=1}^m b_i^k b_j^k \exp(\alpha(h_i^k + h_j^k))} \\
&\leq \left(\sum_{k'=1}^m b_i^{k'} \exp(\alpha h_i^{k'})\right)\left(\sum_{k'=1}^m b_i^{k'} \exp(\alpha h_j^{k'})\right) \qquad (22) \\
&\quad \times \frac{\left(\sum_{k=1}^m \tau_{i,j}^k \exp(-\alpha(h_i^k + h_j^k))\right)}{\sum_{k=1}^m b_i^k b_j^k} \\
&= \sum_{k_1,k_2,k_3=1}^m \frac{b_i^{k_1} b_j^{k_2} \tau_{i,j}^{k_3}}{Z^u_{i,j}} \exp\left(\alpha(h_i^{k_1} + h_j^{k_2} - h_i^{k_3} - h_j^{k_3})\right) \\
&\leq \frac{1 + \exp(6\alpha) + \exp(-6\alpha)}{3Z^u_{i,j}} - \frac{\exp(6\alpha) - 1}{3Z^u_{i,j}} \left(\sum_{k=1}^m (\tau_{i,j}^k - b_i^k) h_i^k\right) \qquad (23)
\end{aligned}$$

The inequality in (22) follows the convexity of reciprocal function, i.e.,

$$\begin{aligned}
\frac{1}{\sum_{k=1}^m b_i^k b_j^k \exp(\alpha(h_i^k + h_j^k))} &= \frac{1}{\sum_{k=1}^m b_i^k b_j^k} \frac{1}{\sum_{k=1}^m \tau_{i,j}^k \exp(\alpha(h_i^k + h_j^k))} \\
&\leq \frac{1}{\sum_{k=1}^m b_i^k b_j^k} \sum_{k=1}^m \tau_{i,j}^k \exp(-\alpha(h_i^k + h_j^k)
\end{aligned}$$

The inequality in (23) follows the convexity of exponential function, i.e.,

$$\exp(\alpha(h_i^{k_1} + h_j^{k_2} - h_i^{k_3} - h_j^{k_3})) = \exp\left(\begin{array}{l} 6\alpha \frac{h_i^{k_1} + h_j^{k_2} - h_i^{k_3} - h_j^{k_3} + 2}{6} + \\ 0 \times \frac{-h_i^{k_1} - h_j^{k_2} + h_i^{k_3} + h_j^{k_3} + 2}{6} + 6\alpha \frac{1}{3} \end{array}\right)$$

$$\leq \frac{h_i^{k_1} + h_j^{k_2} - h_i^{k_3} - h_j^{k_3} + 2}{6} \exp(6\alpha) + \frac{1}{3} \exp(6\alpha) + \frac{-h_i^{k_1} - h_j^{k_2} + h_i^{k_3} + h_j^{k_3} + 2}{6}$$

Bound in Equation 9 can be derived as follows

$$\frac{1}{\widetilde{Z}^l_{i,j}} = \sum_{k',k=1}^{m} y_i^k \exp(H_j^{k'} - H_j^k + \alpha(h_j^{k'} - h_j^k))$$
$$\leq \frac{1+\exp(6\alpha)+\exp(-6\alpha)}{3Z^l_{i,j}} + \frac{\exp(6\alpha)-1}{6}\sum_{k=1}^{m} h_i^k \left(\sum_{k'=1}^{m} y_j^{k'} \frac{b_i^k}{b_i^{k'}} - \frac{y_i^k}{b_i^k}\right)$$

The inequality used by the above derivation follows the convexity of exponential function, i.e.,

$$\exp(\alpha(h_i^{k'} - h_j^k)) \leq \exp\left(6\alpha\frac{h_i^{k'} - h_j^k + 2}{6} + 0 \times \frac{-h_i^{k'} + h_j^k + 2}{6} + 6\alpha\frac{1}{3}\right)$$
$$\leq \frac{h_i^{k'} - h_j^k + 2}{6}\exp(6\alpha) + \frac{1}{3}\exp(6\alpha) + \frac{-h_i^{k'} + h_j^k + 2}{6}$$

Using the definition of $\phi^k_{i,j}$, we have the result in Equation 9.

Appendix B: Proof of Lemma 2

Proof. Following the result in (22), we have

$$\frac{1}{\widetilde{Z}^u_{i,j}} \sum_{k_1,k_2,k_3=1}^{m} \frac{b_i^{k_1} b_j^{k_2} \tau_{i,j}^{k3}}{Z^u_{i,j}} \exp(\alpha(h_i^{k_1} + h_j^{k_2} - h_i^{k_3} - h_j^{k_3}))$$
$$\leq \frac{1}{Z^u_{i,j}} + \frac{\exp(2\alpha)-1}{2Z^u_{i,j}}\left(\sum_{k=1}^{m} h_i^k b_i^k + h_j^k b_j^k\right) + \frac{\exp(-2\alpha)-1}{2Z^u_{i,j}}\left(\sum_{k=1}^{m} a^k_{i,j}(h_i^k + h_j^k)\right)$$

The inequality in the above derivation follows the convexity of exponential function (similar to the proof of Lemma 1). For $Z^l_{i,j}$, we have

$$\frac{1}{\widetilde{Z}^l_{i,j}} = \sum_{k,k'=1}^{m} y_i^k \frac{b_j^{k'}}{b_j^k} \exp\left(2\alpha\frac{h_j^{k'}}{2} - 2\alpha\frac{h_j^k}{2} + 0\frac{2 - h_j^{k'} - h_j^k}{2}\right)$$
$$\leq \sum_{k,k'=1}^{m} y_i^k \frac{b_j^{k'}}{b_j^k}\left(\frac{h_j^{k'}}{2}\exp(2\alpha) + \frac{h_j^k}{2}\exp(-2\alpha)\right) + \sum_{k,k'=1}^{m} y_i^k \frac{b_j^{k'}}{b_j^k}\frac{2 - h_j^k - h_j^{k'}}{2}$$

Replacing $1/\widetilde{Z}^u_{i,j}$ and $1/\widetilde{Z}^l_{i,j}$ in (8) and (9) with the above bounds, we have the result in Lemma 2.

References

1. Bennett, K.P., Demiriz, A.: Semi-supervised support vector machine. In: NIPS. (1999)

2. Chapelle, O., Zien, A.: Semi-supervised classification by low density separation. In: 10th Int. Workshop on AI and Stat. (2005)
3. Blum, A., Chawla, S.: Learning from labeled and unlabeled data using graph mincuts. In: ICML. (2001)
4. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: ICML. (2003)
5. Zhou, D., Bousquet, O., Lal, T., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: NIPS. (2003)
6. Belkin, M., Matveeva, I., Niyogi, P.: Regularization and semisupervised learning on large graphs. In: COLT. (2004)
7. Bennett, K.P., Demiriz, A., Maclin, R.: Exploiting unlabeled data in ensemble methods. In: KDD. (2002)
8. Chen, K., Wang, S.: Regularized boost for semi-supervised learning. In: NIPS. (2007)
9. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. J. AI Res. **2** (1995) 263–286
10. Jin, R., Zhang, J.: Multi-class learning by smoothed boosting. Mach. Learn. **67**(3) (2007) 207–227
11. Scholkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press (2001)
12. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: KDD. (2002)
13. Eibl, G., Pfeiffer, K.P.: Multiclass boosting for weak classifiers. J. Mach. Learn. Res. **6** (2005) 189–210
14. Li, L.: Multiclass boosting with repartitioning. In: ICML. (2006)
15. d'Alche Buc, F., Grandvalet, Y., Ambroise, C.: Semi-supervised marginboost. In: NIPS. (2002)
16. Zhu, X.: Semi-supervised learning literature survey. Technical Report 1530, Computer Science, University of Wisconsin-Madison (2005)
17. Joachims, T.: Transductive inference for text classification using support vector machines. In: ICML. (1999)
18. Bie, T.D., Cristianini, N.: Convex methods for transduction. In: NIPS. (2004)
19. Xu, L., Schuurmans, D.: Unsupervised and semi-supervised multi-class support vector machines. In: AAAI. (2005)
20. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: ICML. (1996)
21. Higham, N.J.: Matrix nearness problems and applications. In Gover, M.J.C., Barnett, S., eds.: Applications of Matrix Theory, Oxford University Press (1989) 1–27
22. Opitz, D., Maclin, R.: Popular ensemble methods: An empirical study. J. AI Res. **11** (1999) 169–198
23. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8) (2000) 888–905