Regularized Multi-Class Semi-Supervised Boosting[*]

Amir Saffari   Christian Leistner   Horst Bischof
Institute for Computer Graphics and Vision
Graz University of Technology
{saffari,leistner,bischof}@icg.tugraz.at

Abstract

Many semi-supervised learning algorithms only deal with binary classification. Their extension to the multi-class problem is usually obtained by repeatedly solving a set of binary problems. Additionally, many of these methods do not scale very well with respect to a large number of unlabeled samples, which limits their applications to large-scale problems with many classes and unlabeled samples.

In this paper, we directly address the multi-class semi-supervised learning problem by an efficient boosting method. In particular, we introduce a new multi-class margin-maximizing loss function for the unlabeled data and use the generalized expectation regularization for incorporating cluster priors into the model. Our approach enables efficient usage of very large data sets. The performance and efficiency of our method is demonstrated on both standard machine learning data sets as well as on challenging object categorization tasks.

1. Introduction

Supervised learning algorithms requires a huge amount of labeled data which are often hard or costly to obtain. Semi-supervised methods offer an interesting solution to this requirement by learning from both labeled and unlabeled data. In the literature, one can find three main semi-supervised learning paradigms: 1) Some algorithms learn the cluster or manifold structure of the feature space with unlabeled samples and use it as an additional cue for the supervised learning process, for example *cluster kernels* [8], *label propagation* [28], *Laplacian SVMs* [3]. 2) Some methods, such as *Transductive Support Vector Machines* (TSVM) [12, 24], try to maximize the margin of the unlabeled samples by pushing away the decision boundary from dense regions of feature space. 3) In co-training [4] two initial classifiers are trained on some labeled data and then they label unlabeled data for re-training of the other one.

The computational complexity of many of state-of-the-art semi-supervised methods limits their application to large-scale problems [17]. This is specially counter-productive for computer vision tasks, such as object recognition or categorization, where a huge amount of unlabeled data is very easy to obtain, for example, via Web downloads.

Most of recent research has focused on binary classification problems, where multi-class problems are often tackled by applying the same algorithm to a set of decomposed binary tasks. Typical approaches are the 1-vs.-all, 1-vs.-1, and error correcting output codes [1]. However, multiple repetition of an already heavy-duty algorithm is not an attractive option for solving problems with many classes and samples. Also, in the case of the 1-vs.-all approach one introduces additional problems, such as producing unbalanced datasets or uncalibrated classifiers.

Hence, having an inherent multi-class semi-supervised algorithm with low computational complexity is very interesting for large-scale applications. Methods addressing both of these issues are very rare to find in the literature. Xu and Schuurmans [27] introduce a multi-class extension to the TSVM which, as stated in the paper, is computationally more intensive than the original TSVM formulation. Song *et al.* [25], and Rogers and Girolami [20] propose the use of Gaussian Processes, while Azran [2] use Markov random walks over a graph for solving the multi-class semi-supervised

problems. However, the computational complexity of these methods are in the order of $\mathcal{O}(n^3)$.

In this paper, we directly address this problem by developing a multi-class semi-supervised boosting method. There exist previous approaches to multi-class boosting, most notably the recent work of Zou *et al.* [29]. Other methods such as [10, 26] still decompose the multi-class problem to binary tasks. Also, there are semi-supervised boosting methods, such as [14, 22] and references therein. However, none of them propose a unified solution to the multi-class semi-supervised learning problem.

In our approach, we not only solve the multi-class semi-supervised problem directly and efficiently, but also we combine the ideas from both major trends of semi-supervised learning. We propose a novel margin maximizing loss function for the unlabeled samples and replace the structure assumption with a set of priors induced from the groups of similar samples. Our method is able to aggregate multi-class classifiers (without a need to solve several binary tasks) as its weak learners. Altogether, it enables us to reduce the complexity of the previous approaches while still being able to use such structural cues during the learning process. The experimental results on the challenging PASCAL 2006 [9] object categorization problems show that we can improve both the classification accuracy and the computation time compared to other methods.

2. Multi-Class Semi-Supervised Boosting

Let $\mathcal{X}_l = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{N_l}, y_{N_l})\}$ and $\mathcal{X}_u = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N_u}\}$ denote the set of labeled and unlabeled samples from a D-dim feature space. We denote the labels as $y \in \{1, \cdots, K\}$, where K is the number of classes. In this section, we briefly review the multi-class loss function we will use and some major semi-supervised learning paradigms.

2.1. A Review of Multi-Class Supervised Loss

Recently, Zou *et al.* [29] extended the concept of Fisher-consistent loss functions [16] from binary classification to the domain of multi-class problems. This concept explains the success of margin-based loss functions and their statistical characteristics.

Let $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \cdots, f_K(\mathbf{x})]^T$ be a multi-valued function, $p(y = i|\mathbf{x}), i = 1, \cdots, K$ be the unknown conditional class probabilities, and $\ell(f_i(\mathbf{x}))$

be a loss function. $\mathbf{f}(\mathbf{x})$ is called a margin vector, if

$$\forall \mathbf{x} : \sum_{i=1}^{K} f_i(\mathbf{x}) = 0. \tag{1}$$

The loss function $\ell(\cdot)$ is Fisher-consistent, if the minimization of the expected risk

$$\hat{\mathbf{f}}(\mathbf{x}) = \arg\min_{\mathbf{f}(\mathbf{x})} \int_{(\mathbf{x}, y)} \ell(f_y(\mathbf{x})) p(y, \mathbf{x}) d(\mathbf{x}, y) \tag{2}$$

has a unique solution and

$$C(\mathbf{x}) = \arg\max_i \hat{f}_i(\mathbf{x}) = \arg\max_i p(y = i|\mathbf{x}), \tag{3}$$

where $C(\mathbf{x})$ is the learnt multi-class classifier. Thus, by minimizing a Fisher-consistent margin-based loss function, one can approximate the unknown Bayes decision rule. This extends the notion of margin from binary classification to the multi-class case: the margin of the ith class, denoted as $f_i(\mathbf{x})$, is directly related to the class conditional probabilities $p(y = i|\mathbf{x})$. Note that because of the symmetry condition Eq.(1), maximizing the margin of a class is equivalent to reducing the margin of all other classes.

In this respect, the exponential loss $\ell(f(\mathbf{x})) = e^{-f(\mathbf{x})}$, is a Fisher-consistent loss [29] and its estimated conditional probabilities can be written as

$$\hat{p}(y = i|\mathbf{x}) = \frac{e^{f_i(\mathbf{x})}}{\sum_{j=1}^{K} e^{f_j(\mathbf{x})}}, \tag{4}$$

which is a symmetric multiple logistic transformation. In this paper, we use this loss function to obtain a multi-class boosting algorithm. It should be noted that usually the true conditional class probabilities in Eq.(2) are unknown. Therefore, given a set of i.i.d. samples, we use the empirical risk

$$\mathcal{L}(\mathbf{f}(\mathbf{x}), \mathcal{X}_l) = \sum_{(\mathbf{x}, y) \in \mathcal{X}_l} e^{-f_y(\mathbf{x})}. \tag{5}$$

2.2. Semi-Supervised Regularization

Many semi-supervised learning algorithms use the unlabeled samples to regularize the supervised loss function in the form of

$$\sum_{(\mathbf{x}, y) \in \mathcal{X}_l} \ell(y, h(\mathbf{x})) + \lambda \sum_{\mathbf{x} \in \mathcal{X}_u} \ell_u(h(\mathbf{x})), \tag{6}$$

where $h(\cdot)$ is a binary classifier, and $\ell_u(\cdot)$ encodes the penalty related to the unlabeled samples. For example, variants of TSVMs [12] maximize the margin for the unlabeled samples by using

$$\ell_u(h(\mathbf{x})) = \max(0, 1 - |h(\mathbf{x})|). \tag{7}$$

Another popular method is the manifold regularization used in the graph-based methods, *e.g.*, label propagation [28] and Laplacian SVM [3]

$$\ell_u(h(\mathbf{x})) = \sum_{\substack{\mathbf{x}' \in \mathcal{X}_u \\ \mathbf{x}' \neq \mathbf{x}}} s(\mathbf{x}, \mathbf{x}') \|h(\mathbf{x}) - h(\mathbf{x}')\|^2, \quad (8)$$

where $s(\mathbf{x}, \mathbf{x}')$ is a similarity function. Using this penalty term, one can enforce the classifier to predict similar labels if the samples are similar. While the graph-based methods are powerful, the pair-wise terms increase their computational complexity.

2.3. Our Method

In order to benefit from both of these ideas, we propose a multi-class multi-objective loss function as

$$\mathcal{L}(\mathbf{f}(\mathbf{x}), \mathcal{X}) = \sum_{(\mathbf{x}, y) \in \mathcal{X}_l} e^{-f_y(\mathbf{x})} + \qquad (9)$$

$$+ \alpha \sum_{\mathbf{x} \in \mathcal{X}_u} \ell_c(\mathbf{f}(\mathbf{x})) + \beta \sum_{\mathbf{x} \in \mathcal{X}_u} \ell_m(\mathbf{f}(\mathbf{x})).$$

The first regularization term $\ell_c(\cdot)$ penalizes the deviations of the model from the cluster assumption, while the second term $\ell_m(\cdot)$ is a multi-class margin maximizing loss. This formulation makes it possibile to tune the loss for a specific problem, where one or both of these penalties might be helpful.

2.3.1 Multi-Class Cluster Regularizer

Our method is a direct application of the cluster/manifold assumption: we want consistent labelings for similar samples. However, instead of a formulation like Eq.(8), we directly use the unlabeled data to identify groups of similar samples (clusters) with respect to the similarity function $s(\mathbf{x}, \mathbf{x}')$. Then, we relate the clusters with possible labelings of them, by estimating the label density in each of these regions. This results in a set of prior conditional probabilities in form of $p_p(y = i|\mathbf{x}), i \in \{1, \cdots, K\}$. Now, we replace the hard consistent labeling requirement by enforcing the learning model to produce a consistent probabilistic estimates over these regions.

Let $\mathcal{Q} = \{q_1, \cdots, q_V\}$ be the clusters returned by the unsupervised clustering method using both labeled and unlabeled samples, and $s(\mathbf{x}, \mathbf{x}')$ as a similarity function. We estimate the prior for all samples within a cluster as

$$\forall \mathbf{x} \in q_v : p_p(y = i|\mathbf{x}) = \frac{|q_v^i| + p(i)m}{\sum_{j=K}^{K} |q_v^j| + m}, \quad (10)$$

where $|q_v^i|$ is the number of samples from class i in cluster v, $p(i)$ is the label prior, and m is a positive number. Note that we use a M-estimation smoothing term in order to obtain a more robust estimate of the cluster priors [6]. If a cluster does not contain any labeled sample, we assign an equal probability to all classes for that partition. In practice, we can run the clustering algorithm a few times with different initial conditions and parameters, and average their results.

The next step is to enforce the model to be consistent with these priors. The Generalized Expectation (GE) criteria proposed by McCallum *et al.* [19] provides a nice framework to incorporate such a prior knowledge into the learning of a model. The generalized expectation describes our belief about how a model should generalize in some prefered directions.

We use the Kullback-Leibler (KL) divergence as the loss for cluster regularization term

$$\ell_c(\mathbf{f}(\mathbf{x})) = D(p_p\|\hat{p}) = -H(p_p) + H(p_p, \hat{p}), \quad (11)$$

where D is the KL-divergence, $H(p_p)$ is the entropy of the prior distribution, and $H(p_p, \hat{p})$ is the cross entropy between the prior and the learning model. Since $H(p_p)$ is a constant and does not depend on the optimized model, we can simply drop it. For notational brevity, let $p_{p,i} = p_p(y = i|\mathbf{x})$, $\hat{p}_i = \hat{p}(y = i|\mathbf{x})$, and $\mathbf{p}_p = [p_{p,1}, \cdots, p_{p,K}]^T$. Then by using Eq.(4) as the probabilistic model, we can further develop the cross entropy as

$$H(p_p, \hat{p}) = -\sum_{i=1}^{K} p_{p,i} \log \hat{p}_i = \qquad (12)$$

$$= -\sum_{i=1}^{K} p_{p,i} f_i(\mathbf{x}) + \log \sum_{j=1}^{K} e^{f_j(\mathbf{x})}.$$

Thus, the cluster regularizer can be written as

$$\ell_c(\mathbf{f}(\mathbf{x})) = -\mathbf{p}_p^T \mathbf{f}(\mathbf{x}) + \log \sum_{j=1}^{K} e^{f_j(\mathbf{x})}. \qquad (13)$$

2.3.2 Multi-Class Margin Regularizer

We also introduce a novel margin maximizing loss over the unlabeled samples, by extending the symmetric hinge loss function of the TSVM, Eq.(7) to the multi-class case in form of

$$\ell_m(\mathbf{f}(\mathbf{x})) = \max(0, M - \max_i (f_i(\mathbf{x}))). \qquad (14)$$

Here, the intuition is the fact that for the multi-class problems, the f_i provides the margin for the

class i and the approximated Bayes classification rule of Eq.(3) selects the class with the highest margin. Therefore, this loss function will try to maximize the margin of the unlabeled sample until it passes M.

Discussion In the absence of labels, semi-supervised methods often try to impose assumptions about the unlabeled data for the learning model and, possibly, their relation to the class labels. Hence, the validity of these assumptions over the unknown structure of the feature space determines the success of these methods. Our method is no exception to this. However, our framework provides an easy way to change the assumptions by replacing the cluster prior with any other source of useful information, like label priors [17], knowledge transfer priors [22], or human knowledge [23].

2.4. Learning

To optimize the regularized loss functions introduced in the previous section, we consider a general additive boosting approach in form of

$$\mathbf{f}(\mathbf{x}) = \nu \sum_{t=1}^{T} \mathbf{g}^t(\mathbf{x}), \qquad (15)$$

where $\nu \in (0, 1]$ is the *shrinkage* factor, and $\mathbf{g}^t(\mathbf{x})$ is the base function (or the weak learner). Note that the shrinkage factor replaces the α_t (weights of the weak learners) in the traditional boosting algorithms. As several researchers [21] suggested, using a shrinkage factor usually slows down the learning of the model, but improves the classification accuracy over using a line search to select the α_t.

We adopt the functional gradient descent view of boosting [18, 10] to iteratively learn the base functions. In this approach, boosting is used as a coordinate descent algorithm: at each iteration, we find a base function which provides the *steepest descent* in the loss. To accomplish this task, we try to find the best base function which has the highest correlation with the negative direction of the gradients of the loss for the current position of the model

$$\mathbf{g}^t(\mathbf{x}) = \underset{\mathbf{g}(\mathbf{x})}{\arg\max} \; -\nabla \mathcal{L}^T \mathbf{g}(\mathbf{x}), \qquad (16)$$

where $\nabla \mathcal{L}$ is the gradient vector of the loss at $\mathbf{f}(\mathbf{x}) = \nu \sum_{k=1}^{t-1} \mathbf{g}^k(\mathbf{x})$.

The gradients of the loss functions of Eq.(9) with respect to the current model, $\mathbf{f}(\mathbf{x})$ can be written as

$$\frac{\partial \mathcal{L}(\mathbf{f}(\mathbf{x}), \mathcal{X})}{\partial f_i(\mathbf{x})} = - \sum_{(\mathbf{x},y) \in \mathcal{X}_l} \mathbb{I}(y = i) e^{-f_i(\mathbf{x})} -$$

$$-\alpha \sum_{\mathbf{x} \in \mathcal{X}_u} \underbrace{(p_{p,i} - \frac{e^{f_i(\mathbf{x})}}{\sum_{j=1}^{K} e^{f_j(\mathbf{x})}})}_{\Delta p_i} -$$

$$\qquad\qquad\qquad\qquad (17)$$

$$-\beta \sum_{\mathbf{x} \in \mathcal{X}_u} \underbrace{\mathbb{I}(i = k)\mathbb{I}(f_i(\mathbf{x}) < M)}_{m_i},$$

where $\mathbb{I}(\cdot)$ is an indicator function, using Eq.(4) $\Delta p_i = p_{p,i} - \hat{p}_i$ is the residual error for estimating the prior of class i, and $k = \underset{j}{\arg\max} \, f_j(\mathbf{x})$ is the index of the class with largest margin. Let $\mathbf{y} = [\mathbb{I}(y = 1), \cdots, \mathbb{I}(y = K)]^T$, $\Delta \mathbf{p} = [\Delta p_1, \cdots, \Delta p_K]^T$ and $\mathbf{m} = [m_1, \cdots, m_K]^T$. Then the learning process of tth base classifier can be written as

$$\mathbf{g}^t(\mathbf{x}) = \underset{\mathbf{g}(\mathbf{x})}{\arg\max} \sum_{(\mathbf{x},y) \in \mathcal{X}_l} e^{-f_y(\mathbf{x})} \mathbf{y}^T \mathbf{g}(\mathbf{x}) +$$

$$+ \sum_{\mathbf{x} \in \mathcal{X}_u} \left(\alpha \Delta \mathbf{p} + \beta \mathbf{m}\right)^T \mathbf{g}(\mathbf{x}) \qquad (18)$$

The following Lemma provides a general solution for these learning problems using multi-class classifiers.

Lemma 2.1. *The solution of Eq.(18) using a multi-class classifier $C(\mathbf{x}) \in \{1, \cdots, K\}$ is*

$$C_t(\mathbf{x}) = \underset{C(\mathbf{x})}{\arg\min} \sum_{(\mathbf{x},y) \in \mathcal{X}_l} w_l \mathbb{I}(C(\mathbf{x}) \neq y) +$$

$$+ \sum_{\mathbf{x} \in \mathcal{X}_u} w_u \mathbb{I}(C(\mathbf{x}) \neq z) \qquad (19)$$

where $w_l = e^{-f_y(\mathbf{x})}$ is the weight for a labeled sample, $z = \underset{i}{\arg\max} \, (\alpha \Delta p_i + \beta m_i)$ and $w_u = \alpha \Delta p_z + \beta m_z$ are the pseudo-label and weight for an unlabeled sample, respectively.

Proof. Note that a multi-class classifier $C(\mathbf{x})$ can be represented as a margin vector $g_i(\mathbf{x}) = \mathbb{I}(C(\mathbf{x}) =$

$i) - \frac{1}{K}$. Thus, the Eq.(18) becomes

$$\sum_{(\mathbf{x},y)\in\mathcal{X}_l} w_l\big(\mathbb{I}(C(\mathbf{x}) = y) - \frac{1}{K}\big)+$$

$$+ \sum_{\mathbf{x}\in\mathcal{X}_u} \sum_{i=1}^{K} \big(\alpha\Delta p_i + \beta m_i\big)\big(\mathbb{I}(C(\mathbf{x}) = i) - \frac{1}{K}\big) =$$

$$= \sum_{(\mathbf{x},y)\in\mathcal{X}_l} w_l\mathbb{I}(C(\mathbf{x}) = y)+$$

$$+ \sum_{\mathbf{x}\in\mathcal{X}_u} \sum_{i=1}^{K} \big(\alpha\Delta p_i + \beta m_i\big)\mathbb{I}(C(\mathbf{x}) = i) + \text{const.}$$

Since $C(\mathbf{x})$ is a multi-class classifier

$$\sum_{i=1}^{K} \big(\alpha\Delta p_i + \beta m_i\big)\mathbb{I}(C(\mathbf{x}) = i) =$$

$$= \alpha\Delta p_{C(\mathbf{x})} + \beta m_{C(\mathbf{x})} \le w_u.$$

Therefore, if we choose $z = \arg\max_{i} (\alpha\Delta p_i + \beta m_i)$ as the target label of the unlabeled sample, we maximize the correlation between the base classifier and the gradients of the loss function. Hence, the learning problem becomes

$$C_t(\mathbf{x}) = \arg\max_{C(\mathbf{x})} \sum_{(\mathbf{x},y)\in\mathcal{X}_l} w_l\mathbb{I}(C(\mathbf{x}) = y)$$

$$+ \sum_{\mathbf{x}\in\mathcal{X}_u} w_u\mathbb{I}(C(\mathbf{x}) = z)$$

which is equivalent to minimizing the weighted mis-classification error rate shown in Eq.(19). $\qquad\square$

Discussion So far we have presented a Regularized Multi-class Semi-supervised Boosting algorithm, which we name *RMS-Boost*. This algorithm is able to directly aggregate multi-class weak learners and utilize unlabeled samples. The class of decision trees is a suitable candidate for the weak learners. Specifically, the random forests [5] produce an extremely fast solution.

3. Experiments

We conduct experiments on two machine learning datasets as well as challenging object category recognition dataset on Pascal VOC2006 [9]. The main goal of these evaluations is to compare our method with other semi-supervised methods which are proposed for large scale datasets. Note that the VOC2006 dataset offers a challenging benchmark

Dataset	# Train	# Test	# Class	# Feat.
Letter	15000	5000	26	16
SensIt (com)	78823	19705	3	100

Table 1. Data sets for the machine learning experiments.

where a simple representation already achieves good results without a need for complicated feature tunings. For sanity check, we also show the performance of the state-of-the-art supervised algorithms.

In these experiments, we compare to the following methods: 1) *AdaBoost.ML* [29]: a multi-class boosting algorithm based on minimizing the logit loss function. We will refer to this method as AML in the experiments. 2) *Kernel SVM*: for machine learning datasets we use the RBF kernel, while for object category recognition, we use the pyramid χ^2 kernel. 3) *MS-TSVM* [24]: Multi-Switch TSVM is probably the fastest version of the popular TSVM. 4) *SERBoost* [22]: a semi-supervised boosting algorithm based on the expectation regularization. We will denote this method as SER. 5) *RM-Boost*: the supervised version of our method (when $\alpha = 0$ and $\beta = 0$). We will refer to this method as RMB.

We apply the 1-vs.-all strategy to those methods which are not inherently multi-class. Preliminary evaluations of our method showed that setting $\beta = 0.1\alpha$ produces acceptable results. Thus, we use this setting for all experiments. We perform a 5-fold cross-validation to select the rest of the hyperparameters for all methods. We set the number of iterations T to be 10000 for all boosting methods and use extremely randomized forests [11] with 10 trees as weak learners. In order to obtain the cluster prior, we run the *hierarchical kmeans* clustering algorithm 10 times by setting the number of clusters to be $50K$ (where K is the number of classes) and average the prior for each sample. We also set the smoothing factor of the probability estimates to be $m = 50$.

3.1. Machine Learning Datasets

We use the *Letter* and *SensIt* datasets from the LibSVM repository [7]. A summary of these sets is presented in Table 1. We randomly partition the original training set into two disjoint sets of labeled and unlabeled samples. We randomly select 5% of the training set to be labeled and assign the rest (95%) to the unlabeled set. We repeat this procedure 10 times and report the average classification accuracy in Table 2. As can be seen from this table, our method achieves the best results over these datasets com-

Method	AML	SVM	TSVM	SER	RMB	**RMSB**
Letter	72.3	70.3	65.9	76.5	74.4	**79.9**
SensIt	79.5	80.2	79.9	81.9	79.0	**83.7**

Table 2. Classification accuracy (in %) for machine learning datasets. The RMSB stands for our method.

Method	AML	SVM	TSVM	SER	RMB	RMSB
Letter	22	25	74	3124	21	125
SensIt	28	195	687	1158	27	514

Table 3. Computation (train+test) time (in seconds) for machine learning datasets.

Figure 1. Classification accuracy with respect to the ratio of labeled samples.

pared to all other methods. Table 2 also shows the average computation time for these methods. Since our method processes 20 times more unlabeled data on these datasets, it is slower than the supervised boosting methods. However, compared to the other semi-supervised methods, our method is faster in the presence of large amount of data.

We also examined the relative contribution of each of the unlabeled regularizer terms. On the Letter dataset, using only the cluster regularizer results in 78.7% classification accuracy, while using only the margin term produces 75.1%. However, using both terms we can see that the performance is boosted to 79.9%. Similar to TSVMs, the margin term resembles a kind of self-learning strategy. Thus, its performance highly depends on the quality of the overall classifier. Therefore, in our case, the cluster prior helps to produce a boosted classifier while the margin term helps to improve the decision margins.

3.2. VOC2006 Dataset

For the VOC2006 dataset, we follow a fairly standard bag-of-words approach by extracting the SIFT descriptors on a regular dense grid of size 8 pixels at multiple scales of 8, 16, 24, and 32 pixels. We find the class-specific visual vocabulary by randomly selecting descriptors from 10 training images of each class, and forming 100 cluster centers using k-means. The final vocabulary is the concatenation of all class-specific cluster centers. We use the L_1-normalized 2-level spatial pyramids [13] to represent each image, and as a result, the feature space is 5000 dimensional. Note that since the VOC2006 presents a multi-label classification problem (some images contain more than one object), we duplicate the multi-label samples of the training set and assign a single label to each of them. Also during the test phase, we assign
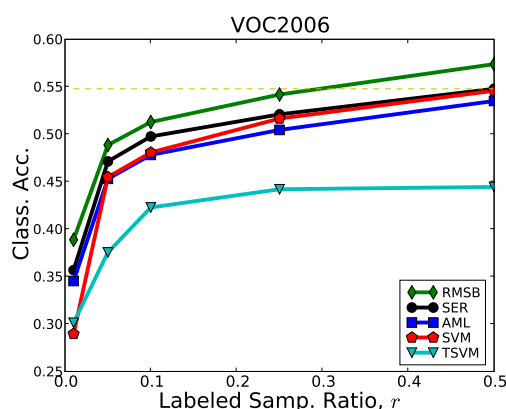
a correct classification if at least one of the labels is predicted correctly. We should emphasize that for the VOC2006 challenge the binary AUC was the performance measure. However, it is not trivial to extend the ROC curve analysis of binary classification to the multi-class problems directly (e.g., [15]). Therefore, we decided to use the other natural alternative which is the classification accuracy.

We randomly partition the training set of VOC2006 dataset into two disjoint sets of labeled and unlabeled samples. The size of the labeled partition is set to be $r = 0.01, 0.05, 0.1, 0.25$, and 0.5 times the number of all training samples. We repeat the procedure 10 times and measure the average classification accuracy over the test set. Note that for these experiments we do not show the results for the supervised version of our method as its performance is similar to that of Adaboost.ML.

Figure 1 shows how the classification accuracy evolves when the number of labeled samples changes in the training set. As can be seen, our model achieves the highest accuracy compared to others. The dashed yellow line shows the best results obtained by other methods (SERBoost at $r = 0.5$). We can see that our method surpasses the accuracy of the SERBoost by using only half of the labeled samples.

Our method not only obtains the best overall results, but also is the fastest compared to the other semi-supervised algorithms. This can be seen in Figure 2 where the computation (training and test) time is presented for two different ratios of labeled samples in the training set. Thus, using proper regularization and solving the multi-class problem directly
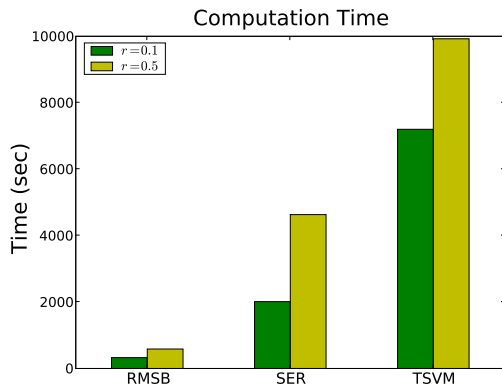
Figure 2. Computation (train+test) time for two different ratios of labeled samples for semi-supervised methods.

(a) Alpha

(b) Shrinkage

Figure 3. The effects of changing (a) α, and (b) the shrinkage factor ν over the classification accuracy.

(a) Class. Acc.

(b) Weights

Figure 4. (a) Classification accuracy (blue), and average gradients for labeled (green) and unlabeled (red) samples versus the number of iterations. (b) The average weights for the unlabeled samples where the prediction of the prior was correct (green) or wrong (red).

is essential in reducing the computation time.

Figure 3(a) shows how the accuracy changes with respect to α. As it can be seen, the performance does not vary considerably for a large range of α values. We found that setting α to the ratio of labeled samples, *i.e.*, $\alpha = \frac{N_l}{N_l + N_u}$, often produces acceptable results. Figure 3(b) presents the effects of the shrinkage factor ν. Here, it becomes clear that selecting a proper value for ν is essential in order to obtain a good result. This is no surprise as it is an established fact that selecting the proper step size for the gradient-based optimization methods is important for robustness of the overall optimization procedure.

Since we perform gradient descent by using multi-class classifiers, it is interesting to see how successful the optimization process is. Figure 4(a) plots the classification accuracy (blue), the average gradients for the labeled (green) and unlabeled (red) samples as a function of the boosting iterations. After an initial rise, the accuracy slowly improves over time. Accordingly, the gradients for the labeled samples also continue to converge to very small values which shows that the optimization over the labeled samples is also successful. However, the gradients for the un-labeled samples converge to a relatively small value. Further investigations of the weights and the cluster prior seem to clarify this behavior. Figure 4(b) shows the average weights of unlabeled samples in two groups: 1) those shown in green where the prior is correct about their true (hidden) label, and 2) those shown in red where the prior is wrong. In this example, the correct number is about 582 while the number of outliers is almost two times bigger: 1091. Therefore, we can see that our algorithm is able to
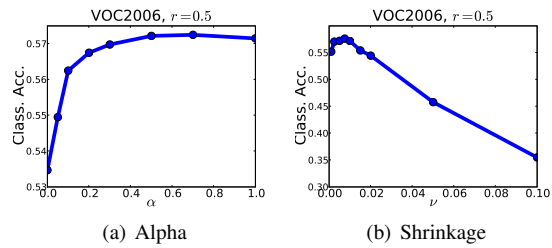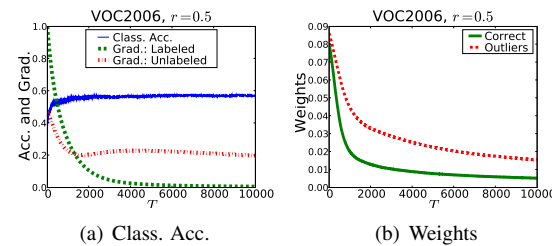
resist learning these outliers, to some extent, and hence, there is always a residual error in the unlabeled loss with respect to these samples. This shows that the optimization procedure is successful in producing a balance between learning from the labeled data and from a very noisy (and mostly wrong) prior.

4. Conclusion

In this paper, we presented a novel multi-class semi-supervised boosting method based on a new combined large margin and cluster regularization. In contrast to other semi-supervised approaches, we directly addressed the multi-class problem by developing an efficient semi-supervised boosting procedure. Furthermore, it is easy to incorporate any source of prior knowledge into our learning method. Our approach shows improved efficiency and accuracy on common machine learning datasets as well as on challenging visual object categorization tasks.

References

[1] E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying ap-

proach for margin classifiers. *JMLR*, 1:113–141, 2000.

[2] A. Azran. The rendezvous algorithm: multiclass semi-supervised learning with markov random walks. In *ICML*, pages 49–56, 2007.

[3] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7:2399–2434, 2006.

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100, 1998.

[5] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[6] B. Cestnik. Estimating probabilities: A crucial task in machine learning. In *ECAI*, pages 147–149, 1990.

[7] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines, 2001.

[8] O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In *NIPS*, pages 585–592, 2003.

[9] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The pascal visual object classes challenge 2006 results. Technical report, 2006.

[10] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 38(2):337–374, 2000.

[11] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. In *Machine Learning*, volume 63, pages 3–42, 2006.

[12] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, pages 200–209, 1999.

[13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, 2006.

[14] C. Leistner, H. Grabner, and H. Bischof. Semi-supervised boosting using visual similarity learning. In *CVPR*, 2008.

[15] J. Li and J. Fine. ROC analysis with multiple classes and multiple tests: methodology and its application in microarray studies. *Biostatistics*, 9(3):566–576, 2008.

[16] Y. Lin. A note on margin-based loss functions in classification. *Statistics & Probability Letters*, 68(1):73–82, 2004.

[17] G. S. Mann and A. Mccallum. Simple, robust, scalable semi-supervised learning via expectation regularization. In *ICML*, pages 593–600, 2007.

[18] L. Mason, J. Baxter, P. Bartlett, and M. Frean. *Functional gradient techniques for combining hypotheses*, pages 221–247. MIT Press, Cambridge, MA., 1999.

[19] A. Mccallum, G. Mann, and G. Druck. Generalized expectation criteria. Technical report, University of Massachusetts Amherst, August 2007.

[20] S. Rogers and M. Girolami. Multi-class semi-supervised learning with the e-truncated multinomial probit gaussian process. *JMLR*, 1:17–32, 2007.

[21] S. Rosset, J. Zhu, T. Hastie, and R. Schapire. Boosting as a regularized path to a maximum margin classifier. *JMLR*, 5:941–973, 2004.

[22] A. Saffari, H. Grabner, and H. Bischof. SERBoost: Semi-supervised boosting with expectation regularization. In *ECCV*, 2008.

[23] R. Schapire, M. Rochery, M. Rahim, and N. Gupta. Incorporating prior knowledge into boosting. In *ICML*, 2002.

[24] V. Sindhwani and S. S. Keerthi. Large scale semi-supervised linear svms. In *SIGIR*, pages 477–484, 2006.

[25] Y. Song, C. Zhang, and J. Lee. Graph based multi-class semi-supervised learning using gaussian process. In *IAPR workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 450–458, 2006.

[26] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *PAMI*, 29(5):854–869, 2007.

[27] L. Xu and D. Schuurmans. Unsupervised and semi-supervised multi-class support vector machines. In *AAAI*, 2005.

[28] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, 2002.

[29] H. Zou, J. Zhu, and T. Hastie. New multicategory boosting algorithms based on multicategory fisher-consistent losses. *Annals of Applied Statistics*, 2008.