Self-Paced Boost Learning for Classification

Te Pi[1]†, Xi Li[1]*, Zhongfei Zhang[1], Deyu Meng[2], Fei Wu[1], Jun Xiao[1]*, Yueting Zhuang[1]
[1]Zhejiang University, Hangzhou, China; [2]Xi'an Jiaotong University, Xi'an, China

Abstract

Effectiveness and robustness are two essential aspects of supervised learning studies. For effective learning, ensemble methods are developed to build a strong effective model from ensemble of weak models. For robust learning, self-paced learning (SPL) is proposed to learn in a self-controlled pace from easy samples to complex ones. Motivated by simultaneously enhancing the learning effectiveness and robustness, we propose a unified framework, Self-Paced Boost Learning (SPBL). With an adaptive from-easy-to-hard pace in boosting process, SPBL asymptotically guides the model to focus more on the insufficiently learned samples with higher reliability. Via a max-margin boosting optimization with self-paced sample selection, SPBL is capable of capturing the intrinsic inter-class discriminative patterns while ensuring the reliability of the samples involved in learning. We formulate SPBL as a fully-corrective optimization for classification. The experiments on several real-world datasets show the superiority of SPBL in terms of both effectiveness and robustness.

1 Introduction

Effectiveness and robustness are two essential principles of generic supervised learning studies. The effective learning focuses on the discriminativeness of the model to capture the intrinsic data patterns for an accurate prediction. The robust learning typically lies in a distinction of the reliable data from the noisy, confusing data, such that the learning is guided by the reliable samples and less influenced by the confusing ones. The efforts of most approaches for learning from the data generally come down to these two aspects.

For effective learning, the key issue lies in the complex distributions of data with local nonlinear structures. To effectively explore these patterns, the boosting scheme [Zhou, 2012] is developed. Generally, the boosting methods build a strong ensemble model as a combination of multiple weak models, where each weak model focuses on the samples mispredicted by the previous model ensembles. Through this, boosting performs an asymptotic piecewise approximation to the data distributions to fit each sample sufficiently. On the other hand, since only the mispredicted samples are considered in each step, the boosting is sensitive to the noisy and confusing data which greatly affect the optimization, especially at the later learning stage. Figure 1 shows a toy example of the decision boundary of boosting classifier for synthetic data with confusing data points. The boosting scheme is very discriminative while lacking a learning robustness.

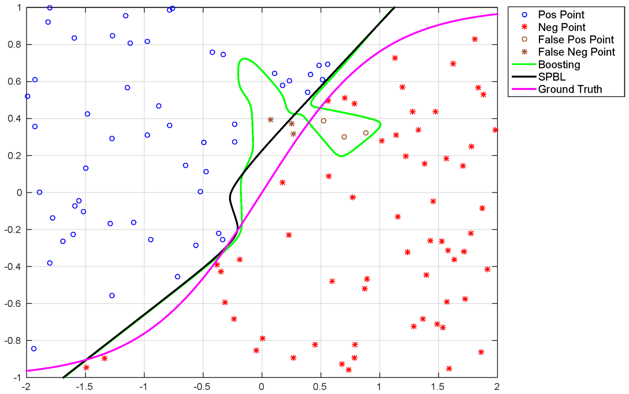


Figure 1: Decision boundaries of boosting and SPBL classifiers and the ground truth for synthetic data with confusing/noisy points. The decision boundary of SPBL is more robust and effective (closer to the ground truth) than that of boosting, since SPBL focuses on the misclassified samples with high reliability based on a self-paced boosting optimization.

For robust learning, the goal is to relieve the influence of the noisy and confusing data. The confusing data generally correspond to the highly nonlinear local patterns hardly learnable for the model space, and the noisy ones are the outliers that should not be learned. Typically, the learning robustness relies on a sample selection to distinguish the reliable samples from the confusing ones. The recently studied self-paced learning (SPL) [Zhao *et al.*, 2015] is such a representative effort. SPL is a learning paradigm that dynamically incor-

†{peterpite, xilizju, zhongfei}@zju.edu.cn;
dymeng@mail.xjtu.edu.cn;
{wufei, junx, yzhuang}@cs.zju.edu.cn.
*Corresponding authors

porates the samples into learning from easy ones to complex ones. With a self-controlled sample selection embedded in learning, the model is calibrated in a pace adaptively controlled by what it has already learned. Thus, SPL smoothly guides the learning to emphasize the patterns of the reliable discriminative data rather than those of the noisy and confusing ones, and obtains the learning robustness.

Based on the above analysis, we notice that boosting and SPL are consistent in basic principles and complementary in methodology. For consistency, both schemes are based on an asymptotic learning process from a weak/simple state to a strong/hard state. On the other hand, boosting and SPL are complementary in three aspects. First, the two schemes are respectively concerned on each of the two essential tasks of machine learning, the effectiveness and the robustness. Second, while boosting imposes a negative suppression on the insufficiently learned samples, SPL positively encourages the easily learned ones in a controlled pace. Third, boosting focuses more on the inter-class margins by striving to fit each sample, while SPL is more concerned with the intra-class variations by dynamically selecting easy samples with different patterns. Thus, boosting tends to reflect the local patterns and is more sensitive to the noisy data, while SPL tends to explore the data smoothly with more robustness. As a result, the two learning schemes are prone to benefit from each other.

To simultaneously enhance the learning effectiveness and robustness, in this paper, we propose a unified framework *Self-Paced Boost Learning* (SPBL). With an adaptive pace from easy to hard in boosting optimization, SPBL asymptotically guides the learning to focus on the insufficiently learned samples with high reliability. Through this, SPBL learns a model in both directions of positive encouragement (on reliable samples) and negative suppression (on misclassified samples), and is capable of capturing the intrinsic inter-class discriminative patterns while ensuring the reliability of the samples involved in learning. Figure 1 further shows the decision boundary of SPBL on the toy dataset, which demonstrates its robustness and effectiveness.

We formulate SPBL as a fully-corrective optimization for classification problem. Note that SPBL is a general framework for supervised learning and could be formulated for other supervised applications. The contributions of this paper are summarized as follows:

1. We propose a unified learning framework SPBL that learns in a joint manner from weak models to strong model and from easy samples to complex ones. To the best of our knowledge, this is the first work that reveals and utilizes the association of boosting and SPL to simultaneously enhance the effectiveness and the robustness for supervised learning.

2. We formulate SPBL as a fully-corrective max-margin boosting optimization with self-paced sample selection for classification task.

2 Related Work

We review the literature from the aspects of boost learning and self-paced learning.

Boosting is a family of supervised ensemble learning approaches which convert weak learners to strong ones [Zhou, 2012]. The boosting methods construct a strong (highly accurate) model by iteratively learning and combining many weak, inaccurate models, where each weak model focuses on the samples mispredicted by the previous models. The main variation among different boosting methods is their ways of weighting training samples and weak learners. Examples of boosting methods include Adaboost [Freund and Schapire, 1997], SoftBoost [Rätsch *et al.*, 2007], TotalBoost [Warmuth *et al.*, 2006], LPBoost [Demiriz *et al.*, 2002], LogitBoost [Friedman *et al.*, 2000], and MadaBoost [Domingo and Watanabe, 2000]. Boosting methods are applied in extensive applications, such as multi-class classification [Shen *et al.*, 2012b; Zhu *et al.*, 2009], regression [Duffy and Helmbold, 2002], metric learning [Shen *et al.*, 2012a], and statistical modeling[Tutz and Binder, 2006; Mayr *et al.*, 2014]. The effectiveness of boosting lies in its piecewise approximation of a nonlinear decision function to sufficiently fit the data patterns [Schapire and Freund, 2012]. However, [Long and Servedio, 2010] indicates that many boosting methods cannot withstand random classification noise.

First proposed by [Kumar *et al.*, 2010], the self-paced learning is inspired by the learning process of humans that gradually incorporates the training samples into learning from easy ones to complex ones. Different from the curriculum learning [Bengio *et al.*, 2009] that learns the data in a predefined order based on prior knowledge, SPL learns the training data in an order from easy to hard dynamically determined by the feedback of the learner itself, which is initially developed for avoiding the bad local minima. SPL is applied in different applications, such as image segmentation [Kumar *et al.*, 2011], multimedia reranking [Jiang *et al.*, 2014a], matrix factorization [Zhao *et al.*, 2015], and multiple instance learning [Zhang *et al.*, 2015]. Variants of SPL are also developed, such as self-paced curriculum learning [Jiang *et al.*, 2015], and SPL with diversity [Jiang *et al.*, 2014b]. Furthermore, [Meng and Zhao, 2015] provides a theoretical analysis of the robustness of SPL, which reveals the consistency of SPL with the non-convex regularization. Such regularization is upper-bounded to restrict the contributions of noisy examples to the objective, and thus enhances the learning robustness.

3 Self-Paced Boost Learning

3.1 Problem Formulation

Let $\{(x_i, y_i)\}_{i=1}^{n}$ be a set of n multi-class training samples, where $x_i \in \mathbb{R}^d$ is the feature of sample i, $y_i \in \{1, 2, \ldots, C\}$ is the class label of x_i, and C is the number of classes. Based on the standard supervised learning scheme, a classification model lies in learning a score function $F_r(\cdot) : \mathbb{R}^d \to \mathbb{R}$ for each class with which the prediction is made:

$$\tilde{y}(x) = \underset{r \in \{1,\ldots,C\}}{\operatorname{argmax}} \; F_r(x; \Theta), \tag{1}$$

where $F_r(x; \Theta)$ serves as the confidence score of classifying sample x to class r, parameterized by Θ. Following the max-margin formulation, the general objective function for multi-class classification is given by:

$$\min_{\Theta} \sum_{i=1}^{n} \sum_{r=1}^{C} L\left(\rho_{ir}\right) + \nu R\left(\Theta\right) \tag{2}$$
$$s.t.\ \forall i, r, \rho_{ir} = F_{y_i}\left(x;\Theta\right) - F_r\left(x;\Theta\right),$$

where ρ_{ir} is the score margin of x_i between its ground truth class y_i and class r; $L : \mathbb{R} \to \mathbb{R}^{+}$ is a loss function; $R\left(\Theta\right)$ is a regularization for Θ; $\nu > 0$ is a trade-off hyperparameter. Generally, the loss function $L\left(\cdot\right)$ should be convex as a convex surrogate of the 0-1 loss, and be monotonically decreasing for a large margin. The regularization $R\left(\Theta\right)$ is introduced to impose prior constraints on Θ to relieve overfitting.

The two key issues of learning the classifier lie in an effective formulation of the score function $F_r\left(\cdot\right)$, and a robust formulation of the loss function $L\left(\cdot\right)$. For an effective modeling, we adopt the boosting strategy that learns the classifier $F_r\left(\cdot\right)$ from weak models to a strong model. The effectiveness of boosting for classification lies in its asymptotic piecewise approximation for a nonlinear decision function to sufficiently fit the underlying data distributions. Specifically, a strong classifier $F_r\left(\cdot\right)$ is formulated as an ensemble of weak classifiers $\left\{h_j\left(\cdot\right) \in \mathcal{H}\right\}_{j=1}^{k}$ in the space of weak models $\mathcal{H}$:

$$F_r\left(x;W\right) = \sum_{j=1}^{k} w_{rj} h_j\left(x\right),\ r = 1, \ldots, C, \tag{3}$$

where each $h_j\left(\cdot\right) : \mathbb{R}^d \to \{0, 1\}$ is a binary weak classifier; $w_{rj} \geqslant 0$ is the weight parameter to be learned. Here Θ is specified as the weight matrix W, defined as $W = \left[w_1, \cdots, w_C\right] \in \mathbb{R}^{k\times C}$ with each $w_r = \left[w_{r1}, \cdots, w_{rk}\right]^T$.

On the other hand, the learning robustness relies on the formulation of the loss function $L\left(\cdot\right)$ to relieve the influence of noisy and confusing data. Instead of directly learning from the whole data batch, we aim to guide the boosting model to learn asymptotically from the easy/faithful samples to the complex/confusing ones in a smooth pace. Therefore, inspired by the self-paced learning (SPL) scheme [Kumar *et al.*, 2010], we reformulate the boosting model with a self-paced loss formulation, and propose a unified framework, Self-Paced Boost Learning (SPBL).

The general objective of SPBL is formulate as:

$$\min_{W,v} \sum_{i=1}^{n} v_i \sum_{r=1}^{C} L\left(\rho_{ir}\right) + \sum_{i=1}^{n} g\left(v_i;\lambda\right) + \nu R\left(W\right) \tag{4}$$
$$s.t.\ \forall i, r, \rho_{ir} = H_{i:}w_{y_i} - H_{i:}w_r;\ W \geqslant 0;\ v \in [0,1]^n,$$

where $H \in \mathbb{R}^{n\times k}$ is the weak classifiers' responses for the training data with $\left[H_{ij}\right] = \left[h_j\left(x_i\right)\right]$, and $H_{i:}$ is the i-th row of H; $v_i \in [0, 1]$ is the SPL weight of sample x_i that indicates its learning "easiness"; $g\left(\cdot;\lambda\right) : [0, 1] \to \mathbb{R}$ is the SPL function that specifies how the samples are selected (the reweighting scheme of v) controlled by the SPL parameter $\lambda > 0$.

In Eq. (4), a weight v_i is assigned to each sample as a measure of its "easiness". These SPL weights are tuned based on the current losses of samples and the SPL function $g\left(v_i;\lambda\right)$ to dynamically select the easily learned samples that are more reliable and discriminative. With a joint optimization of sample selection (for v) and boost learning (for W), the SPBL model gradually incorporates the training samples into learning from easy ones to complex ones, so as to control the pace of boost learning by what the model has already learned.

For a specific formulation of Eq. (4), we specify $L\left(\cdot\right)$ as a smooth loss function, the logistic loss, for the convenience of derivation, and specify $R\left(W\right)$ as the $l_{2,1}$-norm to exploit the group structure of the weak classifier ensembles:

$$\min_{W,v} \sum_{i,r} v_i \ln\left(1 + e^{-\rho_{ir}}\right) + \sum_{i} g\left(v_i;\lambda\right) + \nu\|W\|_{2,1} \tag{5}$$
$$s.t.\ \forall i, r, \rho_{ir} = H_{i:}w_{y_i} - H_{i:}w_r;\ W \geqslant 0;\ v \in [0,1]^n,$$

where $\|W\|_{2,1} = \sum_{j=1}^{k} \|W_{j:}\|_2$. Note that the above objective is $l_{2,1}$-norm regularized to impose a group sparsity constraint on the rows of W. The optimization would encourage the columns of W (each class) to select a relatively concentrated and shared subset of base classifiers, instead of learning them independently. We present the optimization of Eq. (5) and the specification of $g\left(v_i;\lambda\right)$ in the next subsection.

3.2 Optimization

We use an alternating optimization to solve Eq. (5), which optimizes each of the two variables with the other one fixed in an alternating manner. For the optimization of v, we have

$$v_i^* = \operatorname*{argmin}_{v_i} v_i l_i + g\left(v_i;\lambda\right),\ s.t.\ v_i \in [0, 1], \tag{6}$$

where $l_i = \sum_r \ln\left(1 + e^{-\rho_{ir}}\right)$ denotes the loss of sample x_i.

To solve v_i in Eq. (6), the self-paced function $g\left(v_i;\lambda\right)$ needs to be specified. [Jiang *et al.*, 2014a] has summarized the general properties of a self-paced function in three aspects. First, $g\left(v_i;\lambda\right)$ is convex w.r.t. $v_i \in [0, 1]$ to guarantee the uniqueness of v_i^*. Second, $v_i^*\left(l_i;\lambda\right)$ is monotonically decreasing w.r.t. l_i, which guides the model to select easy samples with smaller losses in favor of complex samples with larger losses. Third, $v_i^*\left(l_i;\lambda\right)$ is monotonically increasing w.r.t. λ, which means that a larger λ has a higher tolerance to the losses and can incorporate more complex samples. Several examples of the self-paced function have been listed in [Jiang *et al.*, 2014a], such as hard weighting, linear weighting, and mixture weighting. We specify the self-paced function as the one for mixture weighting, due to its overall better performance in the experiments:

$$g\left(v_i;\lambda,\zeta\right) = -\zeta \ln\left(v_i + \zeta/\lambda\right),\ \lambda, \zeta > 0, \tag{7}$$

where an extra SPL parameter ζ is introduced in addition to λ. The corresponding optimal v_i^* is given by:

$$v_i^* = \left\{ \begin{array}{cc} 1, & l_i \leqslant \zeta\lambda/(\zeta+\lambda) \\ 0, & l_i \geqslant \lambda \\ \zeta/l_i - \zeta/\lambda, & \text{otherwise} \end{array} \right., \tag{8}$$

which is a mixture of a hard 0-1 weighting and a soft real-valued weighting.

For the optimization of W, we have

$$W^* = \operatorname*{argmin}_{W} \sum_{i,r} v_i \ln\left(1 + e^{-\rho_{ir}}\right) + \nu\|W\|_{2,1}, \tag{9}$$
$$s.t.\ \forall i, r, \rho_{ir} = H_{i:}w_{y_i} - H_{i:}w_r;\ W \geqslant 0.$$

To solve W in Eq. (9), we adopt the column generation method [Demiriz *et al.*, 2002], due to the potentially infinite number of candidate weak models in the $\mathcal{H}$ space. The column generation is applied in the dual space of W to maintain a small set of weak models as the active dual constraints. This active set is augmented during optimization until it is sufficient to reach a solution within a tolerance threshold. We check the dual problem of Eq. (9):

$$\max_{U,Q} -\sum_{i,r}\{U_{ir}\ln U_{ir} + (v_i - U_{ir})\ln(v_i - U_{ir})\} \qquad (10)$$

$$s.t.\ \forall r,\ \sum_{i=1}^{n}[\delta_{ry_i}(\sum\nolimits_l U_{il}) - U_{ir}]H_{i:} \leqslant \nu Q_{:r}^{T};$$

$$\forall j,\ \|Q_{j:}\|_2 \leqslant 1,$$

where $\delta_{ry_i} = \mathbf{1}(r = y_i)$ is an indicator function. $U \in \mathbb{R}^{n\times C}$ is the Lagrangian multiplier of the equality constraints of Eq. (9), with a relation to the primal solution:

$$U_{ir} = \frac{v_i}{1+e^{\rho_{ir}}},\ i = 1,\cdots,n,\ r = 1,\cdots,C. \qquad (11)$$

The derivation of Eqs. (10) and (11) is similar to that of [Shen *et al.*, 2012b].

Based on the column generation, the set of active weak classifiers is augmented by a weak model $\hat{h}(\cdot)$ that most violates the current dual constraints in Eq. (10):

$$\{\hat{h}(\cdot), \hat{r}\} = \operatorname*{argmax}_{h(\cdot)\in\mathcal{H},r} \sum_{i=1}^{n}[\delta_{ry_i}(\sum\nolimits_l U_{il}) - U_{ir}]h(x_i). \quad (12)$$

Then the optimization continues with the new set of active weak models, until the violation score (objective value of Eq. (12)) reaches a tolerance threshold.

Eq. (12) indicates that the matrix U serves as the sample importance for learning a new weak classifier. Moreover, from Eq. (11) we see that U gives high weights to not only the misclassified samples with small margins ρ_{ir}, but also the easy samples with high SPL weights v_i. That means that U is actually a composite measure of learning insufficiency and learning easiness. Since the v_i weights are set in the previous iteration, based on Eq. (12), the future weak learners will put emphasis on samples that are both insufficiently learned currently and easily learned previously. The interactions of the update of the model parameters are summarized in Figure 2. As a balance and trade-off between boosting and SPL, the proposed SPBL performs learning in both directions of positive encouragement (on reliability) and negative suppression (on learning insufficiency), and takes both effectiveness and robustness into concern for learning a classification model.

Further, it is easily seen that the multi-class boosting classification model of [Shen *et al.*, 2012b] is a special case of SPBL with all SPL weights v fixed as $\mathbf{1}_n$. By replacing v_i in Eq. (11) with 1, the matrix U only emphasizes the misclassified samples with small margins, with the new weak classifier learned accordingly. Thus, the boosting method tends to be sensitive to the noisy and hardly learnable data by striving to correctly classify these samples. Therefore, the proposed SPBL is a robust generalization of boosting models.

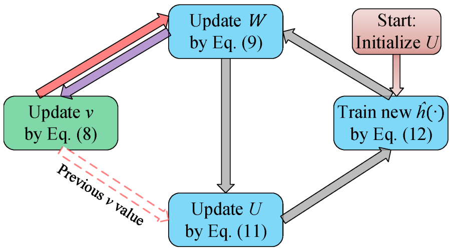


Figure 2: The interactions of the update of the model parameters. The blue blocks represent the boosting stage while the green block represents the SPL stage. The update of W and v are mutually interacted in successive iterations, while the current W and the previous v jointly influence the learning of the new weak classifier through U.

We summarize the optimization procedure in Algorithm 1. The algorithm alternates between learning new $\hat{h}(\cdot)$ (Line 5), updating W (Line 6), updating U (Line 7) and reweighting v in an SPL manner (Line 8). Note that the SPL parameters (λ, ζ) are iteratively increased (annealed) if they are small (Line 10 to 12), so as to introduce more (difficult) samples in the future learning. Furthermore, we adopt an early stopping criterion on a held-out validation set when the iteration number exceeds T_{ES} times to maintain a better generalization performance and a reasonable running time.

4 Experiments

We evaluate the performance of SPBL classification on three real-world datasets. The comparative methods include softmax regression (SR), multi-class SVM (MultiSVM), MultiBoost [Shen *et al.*, 2012b], and Multi-class Adaboost (AdaBoost) [Zhu *et al.*, 2009]. SR and MultiSVM are also embedded with a self-paced learning scheme for comparison, denoted as SR-S and MultiSVM-S.

Specifically, the two baseline methods, SR and MultiSVM, are formulated based on a linear classifier, where SR optimizes a log-likelihood and MultiSVM optimizes a hinge-loss. The MultiBoost is a fully-corrective formulation of multi-class boosting classification, which is a special case of SPBL with v fixed as $\mathbf{1}_n$. The Multi-class AdaBoost is a multi-class generalization of AdaBoost as a stagewise additive boosting model. It is worth comparing SPBL with the above methods to verify its effectiveness by learning a classifier in a joint boosting and self-paced manner.

4.1 Dataset Description

Three real-world image datasets are used. We choose the image data for experiments because the underlying patterns of image features tend to have rich nonlinear correlations. The three datasets are *Caltech256*[1], *AnimalWithAttributes* (*AWA*)[2], *Corel10k*[3]. All of them are publicly available and fully labeled with each sample belonging to only one class. The statistics of the datasets are summarized in Table 2.

[1] http://www.vision.caltech.edu/Image_Datasets/Caltech256/

[2] http://attributes.kyb.tuebingen.mpg.de/

[3] http://www.ci.gxnu.edu.cn/cbir/dataset.aspx

Algorithm 1: SPBL for Classification

```
Input  : Training set $\{(x_i, y_i)\}_{i=1}^{n}$; $\nu > 0$; initial
         SPL parameters $\lambda_0, \zeta_0 > 0$; initial SPL
         weights $v_0$; $\lambda_{max}$; $T_{ES}$; $\mu > 1$; $\epsilon > 0$.
Output : A set of $k$ weak classifiers $\{h_j(\cdot)\}_{j=1}^{k}$; $W$.
1  Initialize: $v^{(0)} \leftarrow v_0$; $(\lambda, \zeta) \leftarrow (\lambda_0, \zeta_0)$; $U \leftarrow v^{(0)} \mathbf{1}_C^T$;
2  $t \leftarrow 0$;
3  repeat
4      $t \leftarrow t + 1$;
       Boosting :
5      Learn a new weak classifier: solve Eq. (12) to obtain
         $\{h_t(\cdot), \hat{r}\}$ based on $U$;
6      Update $W$: solve Eq. (9) for $W^{(t)}$ based on $v^{(t-1)}$;
7      Update $U$: compute $U$ by Eq. (11) based on $v^{(t-1)}$;
       SPL :
8      Update $v$: compute $v^{(t)}$ by Eq. (8) based on $W^{(t)}$;
       Validation:
9      Test $\{h_j(\cdot)\}_{j=1}^{t}$ and $W^{(t)}$ on the validation set, to
         obtain the error rate $err^{(t)}$;
       Annealing:
10     if $\lambda < \lambda_{max}$ then
11         $\lambda \leftarrow \mu\lambda$; $\zeta \leftarrow \mu\zeta$;
12     end
13 until $\sum_i [\delta_{\hat{r}y_i} (\sum_l U_{il}) - U_{ir}] h_t(x_i) < \nu + \epsilon$ or
     $t \geqslant T_{ES}$ and $err^{(t)} > \min_{1 \leqslant s \leqslant t-1} err^{(s)}$;
14 $k \leftarrow \mathrm{argmin}_s \, err^{(s)}$;
Return : $\{h_j(\cdot)\}_{j=1}^{k}$, $W = W^{(k)}$.
```

We use the spatial pyramid features for *Caltech256* and *Corel10k* extracted based on [Lazebnik *et al.*, 2006], and use the available Decaf feature for *AWA*. We reduce the dimensions of all the features to 512 by PCA.

4.2 Experimental Settings

For a convenience of optimization, we first extend the output of a weak classifier $h(\cdot)$ to real value $[0, 1]$. We assume a logistic linear form for $h(\cdot)$:

$$h(x; \theta_h, b_h) = \left\{1 + \exp\left(-\left(\theta_h^T x + b_h\right)\right)\right\}^{-1}, \qquad (13)$$

where $\theta_h \in \mathbb{R}^d$, $b_h \in \mathbb{R}$ are the parameters of $h(\cdot)$.

We adopt the strategy in [Jiang *et al.*, 2014b] for the annealing of the SPL parameters (λ, ζ) (Line 10 to 12 in Algorithm 1). Specifically, at each iteration, we sort the samples in the ascending order of their losses, and set (λ, ζ) based on the number of samples to be selected by now. Instead of annealing the absolute values of (λ, ζ), we anneal the proportion of the number of selected samples. It is shown in [Jiang *et al.*, 2014b] that such annealing scheme is more stable.

We implement a grid search for the tuning of the hyperparameter ν. Further, in order to test the robustness of our model, we manually add label noise into the training set by randomly selecting and relabeling $s\%$ of the training samples with the other labels different from the true ones. We conduct experiments with $s \in \{0, 5, 10, 15\}$ for the three datasets.

4.3 Experimental Results

Table 1 shows the error rate performance of SPBL and the comparative methods on the three datasets, with different proportions of noisy samples. The best results are shown in bold face. To give a concise demonstration of the performances, we show in Figure 3 the error rates for three datasets w.r.t. the noise ratio. We see that SPBL has a better overall performance than the comparative methods.

Figure 3 further shows that the performances of boosting methods (MultiBoost, AdaBoost) are sensitive to the noisy data, and that the comparative methods embedded with SPL (SR-S, MultiSVM-S) are more robust than their original counterparts. It is expected, since the suppression effect to noise of a comparative method stems from the self-paced learning scheme. By effectively utilizing the complementarity of boosting and SPL, the proposed SPBL demonstrates a stable performance improvement over the SPL-embedded methods, and an increasing performance improvement over the other comparative methods.

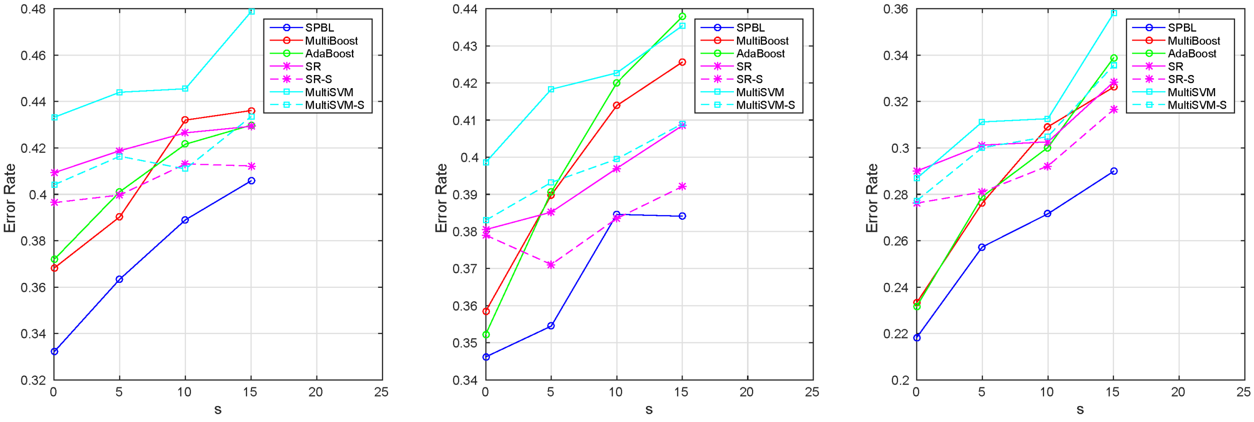
Further, we show in Figure 4 the change of the error rates on the training set and the test set w.r.t. the learning iterations of SPBL and MultiBoost, for $s = 0$. We see that the test and training error rate curves of SPBL are generally in between the corresponding curves of MultiBoost. Therefore, Figure 4 shows that SPBL relieves the overfitting problem of boosting methods, since it has a smaller gap between the training errors and the test errors. This is due to the smooth learning pace of SPBL based on a self-paced boosting optimization from easy samples to hard ones, instead of learning from the whole data batch as MultiBoost does. Through this, SPBL guides the model to focus on the samples not only insufficiently learned, but also with high confidence of reliability, and thus relieves the overfitting and obtains a better generalization performance.

5 Conclusions

In this work, we propose a unified learning framework, Self-Paced Boost Learning (SPBL), that learns in a joint manner from weak models to a strong model and from easy samples to complex ones, for both effective learning and robust learning. With an adaptive pace from easy to hard in boosting optimization, SPBL asymptotically guides the model to focus on the samples not only insufficiently learned but also with high reliability. Through this, SPBL learns a model in both directions of positive encouragement (on reliable samples) and negative suppression (on misclassified samples), and is capable of capturing the intrinsic inter-class discriminative patterns while ensuring the reliability of the samples involved in learning. To the best of our knowledge, this is the first work that reveals and utilizes the association of boosting and SPL to simultaneously enhance the effectiveness and the robustness for supervised learning. We formulate SPBL as a fully corrective optimization for classification task. The experiments on real-world datasets show the superiority of SPBL in terms of both effectiveness and robustness.

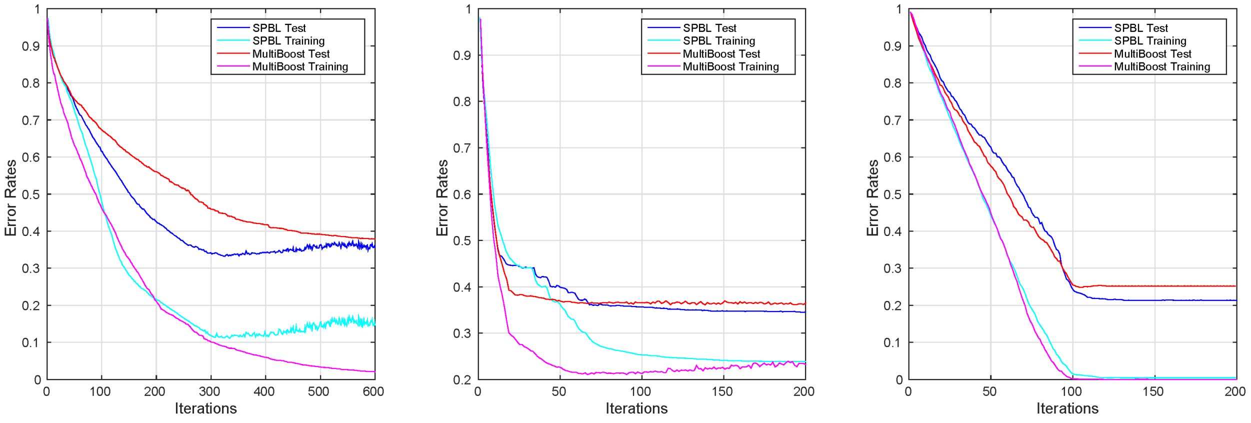
Table 1: The classification error rate performance of each approach on the three datasets

	Caltech101				*AWA*				*Corel10k*			
	$s = 0$	$s = 5$	$s = 10$	$s = 15$	$s = 0$	$s = 5$	$s = 10$	$s = 15$	$s = 0$	$s = 5$	$s = 10$	$s = 15$
SPBL	**0.3321**	**0.3634**	**0.3890**	**0.4058**	**0.3462**	**03545**	0.3846	**0.3841**	**0.2182**	**0.2573**	**0.2716**	**0.2899**
MultiBoost	0.3682	0.3903	0.4320	0.4361	0.3585	0.3898	0.4139	0.4256	0.2332	0.2762	0.3091	0.3262
AdaBoost	0.3719	0.4011	0.4216	0.4298	0.3573	0.3906	0.4200	0.4379	0.2316	0.2790	0.3002	0.3388
SR	0.4093	0.4188	0.4265	0.4293	0.3805	0.3852	0.3970	0.4085	0.2900	0.3012	0.3026	0.3284
SR-S	0.3964	0.3997	0.4131	0.4122	0.3790	0.3711	**0.3835**	0.3921	0.2762	0.2810	0.2922	0.3165
MultiSVM	0.4332	0.4440	0.4455	0.4787	0.3986	0.4183	0.4227	0.4354	0.2868	0.3112	0.3126	0.3580
MultiSVM-S	0.4041	0.4164	0.4109	0.4335	0.3830	0.3932	0.3995	0.4090	0.2772	0.3001	0.3049	0.3356



(a) *Caltech256* (b) *AWA* (c) *Corel10k*

Figure 3: The error rate results w.r.t. the noise ratio $s\%$ for the three datasets. The proposed SPBL has a better overall performance than the comparative methods.



(a) *Caltech256* (b) *AWA* (c) *Corel10k*

Figure 4: The error rates on the training and the test set of SPBL and MultiBoost w.r.t. the iterations for $s = 0$. The learning pace of the proposed SPBL is more smooth with a smaller gap between the training and the test performance. SPBL relieves the overfitting of boosting methods.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant U1509206, Grant 61472353, and Grant 61572431, the National Basic Research Program of China under Grant 2012CB316400 and Grant 2015CB352300, the Fundamental Research Funds for the Central Universities, Zhejiang Provincial Engineering Center on Media Data Cloud Processing and Analysis, and the equipment donation by Nvidia.

References

[Bengio *et al.*, 2009] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning.

Table 2: Statistics of the datasets

Dataset	Feature	Classes	Samples & Partition (training/validation/test)
Caltech256	SP	256	29780 (50%/20%/30%)
AWA	Decaf	50	30475 (50%/20%/30%)
Corel10k	SP	100	10000 (50%/20%/30%)

In *International Conference on Machine Learning*, pages 41–48. ACM, 2009.

[Demiriz *et al.*, 2002] Ayhan Demiriz, Kristin P Bennett, and John Shawe-Taylor. Linear programming boosting via column generation. *Machine Learning*, 46(1-3):225–254, 2002.

[Domingo and Watanabe, 2000] Carlos Domingo and Osamu Watanabe. Madaboost: A modification of adaboost. In *Annual Conference on Computational Learning Theory*, pages 180–189, 2000.

[Duffy and Helmbold, 2002] Nigel Duffy and David Helmbold. Boosting methods for regression. *Machine Learning*, 47(2-3):153–200, 2002.

[Freund and Schapire, 1997] Yoav Freund and Robert E 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.

[Friedman *et al.*, 2000] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 28(2):337–407, 2000.

[Jiang *et al.*, 2014a] Lu Jiang, Deyu Meng, Teruko Mitamura, and Alexander G Hauptmann. Easy samples first: Self-paced reranking for zero-example multimedia search. In *Proceedings of the ACM International Conference on Multimedia*, pages 547–556. ACM, 2014.

[Jiang *et al.*, 2014b] Lu Jiang, Deyu Meng, Shoou-I Yu, Zhenzhong Lan, Shiguang Shan, and Alexander Hauptmann. Self-paced learning with diversity. In *Advances in Neural Information Processing Systems*, pages 2078–2086, 2014.

[Jiang *et al.*, 2015] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. Self-paced curriculum learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[Kumar *et al.*, 2010] M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197, 2010.

[Kumar *et al.*, 2011] M Pawan Kumar, Haithem Turki, Dan Preston, and Daphne Koller. Learning specific-class segmentation from diverse data. In *International Conference on Computer Vision*, pages 1800–1807. IEEE, 2011.

[Lazebnik *et al.*, 2006] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition*, pages 2169–2178, 2006.

[Long and Servedio, 2010] Philip M Long and Rocco A Servedio. Random classification noise defeats all convex potential boosters. *Machine Learning*, 78(3):287–304, 2010.

[Mayr *et al.*, 2014] Andreas Mayr, Harald Binder, Olaf Gefeller, and Matthias Schmid. The evolution of boosting algorithms-from machine learning to statistical modelling. *arXiv preprint arXiv:1403.1452*, 2014.

[Meng and Zhao, 2015] Deyu Meng and Qian Zhao. What objective does self-paced learning indeed optimize? *arXiv preprint arXiv:1511.06049*, 2015.

[Rätsch *et al.*, 2007] Gunnar Rätsch, Manfred K Warmuth, and Karen A Glocer. Boosting algorithms for maximizing the soft margin. In *Advances in Neural Information Processing Systems*, pages 1585–1592, 2007.

[Schapire and Freund, 2012] Robert E Schapire and Yoav Freund. *Boosting: Foundations and algorithms*. MIT press, 2012.

[Shen *et al.*, 2012a] Chunhua Shen, Junae Kim, Lei Wang, and Anton Van Den Hengel. Positive semidefinite metric learning using boosting-like algorithms. *The Journal of Machine Learning Research*, 13(1):1007–1036, 2012.

[Shen *et al.*, 2012b] Chunhua Shen, Sakrapee Paisitkriangkrai, and Anton van den Hengel. A direct approach to multi-class boosting and extensions. *arXiv preprint arXiv:1210.4601*, 2012.

[Tutz and Binder, 2006] Gerhard Tutz and Harald Binder. Generalized additive modeling with implicit variable selection by likelihood-based boosting. *Biometrics*, 62(4):961–971, 2006.

[Warmuth *et al.*, 2006] Manfred K Warmuth, Jun Liao, and Gunnar Rätsch. Totally corrective boosting algorithms that maximize the margin. In *International Conference on Machine Learning*, pages 1001–1008. ACM, 2006.

[Zhang *et al.*, 2015] Dingwen Zhang, Deyu Meng, Chao Li, Lu Jiang, Qian Zhao, and Junwei Han. A self-paced multiple-instance learning framework for co-saliency detection. In *International Conference on Computer Vision*, pages 594–602, 2015.

[Zhao *et al.*, 2015] Qian Zhao, Deyu Meng, Lu Jiang, Qi Xie, Zongben Xu, and Alexander G Hauptmann. Self-paced learning for matrix factorization. In *AAAI Conference on Artificial Intelligence*, 2015.

[Zhou, 2012] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC Press, 2012.

[Zhu *et al.*, 2009] Ji Zhu, Hui Zou, Saharon Rosset, and Trevor Hastie. Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360, 2009.