Information Theoretic Regularization for Semi-Supervised Boosting

Lei Zheng
Kno.e.sis Center
Wright State University
lei.zheng@wright.edu

Shaojun Wang
Kno.e.sis Center
Wright State University
shaojun.wang@wright.edu

Yan Liu
Wright State University
yan.liu@wright.edu

Chi-Hoon Lee
Yahoo! Lab
chihoon@yahoo-inc.com

ABSTRACT

We present novel semi-supervised boosting algorithms that incrementally build linear combinations of weak classifiers through generic functional gradient descent using both labeled and unlabeled training data. Our approach is based on extending information regularization framework to boosting, bearing loss functions that combine log loss on labeled data with the information-theoretic measures to encode unlabeled data. Even though the information-theoretic regularization terms make the optimization non-convex, we propose simple sequential gradient descent optimization algorithms, and obtain impressively improved results on synthetic, benchmark and real world tasks over supervised boosting algorithms which use the labeled data alone and a state-of-the-art semi-supervised boosting algorithm.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Experimentation, Performance

Keywords

Ensemble method, semi-supervised learning

1. INTRODUCTION

Boosting, as one of the most powerful learning ideas introduced in early 1990s (Hastie et al. 2009), is a supervised machine learning and data mining technique that incrementally builds linear combinations of “weak” models to generate a “strong” predicative model and is proved to be one of the most successful and practical methods in machine learning. Schapire (1990) developed the first provable polynomial-time boosting algorithm, based on PAC learning, and showed how to improve a weak learner’s performance by training two additional classifiers. Freund and Schapire (1997) later invented the popular AdaBoost algorithm using the idea of adaptively resampling the data: that is, AdaBoost starts with a weak classifier and seeks its improvements iteratively based on its performance on the training data. Since its inception, AdaBoost algorithm for classification has attracted much attention in the machine learning community as well as in related areas in statistics. Various variants of AdaBoost algorithm have proven to be very competitive in prediction accuracy in a variety of applications. Boosting methods were originally proposed as ensemble methods, which rely on the principle of generating multiple predictions and majority voting (averaging) among the individual classifiers.

Semi-supervised learning (Chapelle et al. 2006) is a machine learning technique that uses both labeled and unlabeled data for training — typically a small amount of labeled data with a large amount of unlabeled data. Semi-supervised learning is touted as one of the most natural forms of training for prediction tasks, since unlabeled data is plentiful whereas labeled data is usually limited or expensive to obtain. Many approaches have been proposed for semi-supervised learning (Chapelle et al. 2006), including: generative models (Castelli and T. Cover 1996, Cohen and Cozman 2006, Nigam et al. 2000), self-learning (Celeux and Govaert 1992), co-training (Blum and Mitchell 1998), information-theoretic regularization (Grandvalet and Bengio 2004, Corduneanu and Jaakkola 2006) and graph-based transductive methods (Zhou et al. 2004 and Zhu et al. 2003).

Although highly desirable, semi-supervised boosting has not been studied as widely as the other semi-supervised settings mentioned above, with very few exceptions (Benett et al. 2002, Chen and Wang 2007, d’Alché-Buc et al. 2002, Valizadegan et al. 2008). These approaches are essentially a self-learning algorithm where the class labels of unlabeled data are updated iteratively; they essentially operate like self-training where the class labels of unlabeled examples are updated iteratively: first a classifier is constructed using a small amount of labeled data, then it is used to predict the pseudo-labels for unlabeled examples; a new classifier is then constructed using both labeled and pseudo-labeled examples; the processes of constructing classifiers and predict-

KDD’09, June 28–July 1, 2009, Paris, France.

ing pseudo-labels alternate iteratively until certain stopping criterion is reached. The main drawback of this approach is that it relies solely on the pseudo-labels predicted by the classifiers constructed so far when generating new classifiers. Since the pseudo-labels predicted by the constructed classifiers could be inaccurate, especially at the first few steps, the resulting new classifiers might also be unreliable. The errors might propagate due to this ripple effect, thus finally hurt the performance.

We propose semi-supervised boosting algorithms that use information-theoretic measures such as entropy and/or mutual information (Grandvalet and Bengio 2004, Corduneanu and Jaakkola 2006, Wang et al. 2009) as a vehicle for regularization on unlabeled data. The motivation is that minimizing conditional entropy or minimizing mutual information over unlabeled data encourages the algorithm to find putative labelings for the unlabeled data that are mutually reinforcing with the supervised labels; that is, greater certainty on the putative labelings coincides with greater conditional likelihood on the supervised labels, and vice versa. For a single classification variable, minimizing entropy criterion has been shown to effectively partition unlabeled data into clusters (Grandvalet and Bengio 2004, Roberts et al. 2000). Later on, this work was extended to semi-supervised learning for structured prediction such as sequence labeling (Jiao et al. 2006) and image segmentation (Lee et al. 2006), achieving very impressive improvements over using labeled data alone. Recently Wang et al. (2009) present a mutual information regularized semi-supervised learning as a data compression scheme that is formulated into a rate distortion (Cover and Thomas 1991) framework, and demonstrate encouraging results with two real-world problems to show the effectiveness of the proposed approach: text categorization as a multi-class classification problem, and hand-written character recognition as a sequence labeling problem. In this paper, we demonstrate how to use similar ideas for semi-supervised boosting. Different with existing self-learning type semi-supervised boosting algorithms, our approach is grounded on a firm information theoretic motivation. We propose simple sequential gradient descent optimization algorithms, and obtain impressively improved results on synthetic, benchmark and real world tasks over supervised boosting algorithms (which use the labeled data alone) and a state-of-the-art semi-supervised boosting algorithm, ASSEMBLE proposed in (Benett et al. 2002).

2. BOOSTING AS AN OPTIMIZATION METHOD

A fundamental theoretical issue for AdaBoost and its many variants is convergence, which is not addressed in the original AdaBoost paper (Freund and Schapire 1997). In fact, much work has been done to prove the convergence of boosting algorithm in terms of an optimization method. They can be categorized into two basic approaches: greedy function optimization and maximum entropy approach.

In the first approach, AdaBoost is viewed as a sequential gradient descent algorithm in function space, inspired by numerical optimization and statistical estimation. It was Breiman (1999) who made this path-breaking observation. This insight opened new perspectives and was extended to a variety of related objective functions. Many variant of Adaboost, such as logistic regression and least square (Friedman, et al, 2000, Mason et al. 1999) have also been developed for contexts other than classification, such as regression and density estimation. In this approach, statistical models are typically additive expansions in a set of basis functions and are fitted by minimizing a loss function averaged over the training data. For many loss functions or basis functions, this requires computationally intensive numerical optimization techniques. The boosting approach is a forward stagewise additive modeling (Friedman et al. 2000) that approximates the solution by sequentially adding new basis functions to the expansion without adjusting the parameters and coefficients of those that have already been added. At each iteration, one solves for the optimal basis function and corresponding coefficients to add to the current expansion. This produces new expansion and the process is repeated.

In the second approach (Collins et al 2002, Della Pietra et al. 1997, Lebanon and Lafferty 2002, Haffari et al. 2008), the boosting algorithm is cast in terms of maximizing generalized entropy subject to certain linear feature constraints, enforcing their expectations meet the empirical expectations. AdaBoost can be described as a greedy feature induction algorithm that incrementally builds random fields to solve the maxent problem. The greediness of the algorithm arises in steps that select the most informative feature. In these steps each feature in a pool of candidate features is evaluated by estimating the reduction in the Kullback-Leibler divergence that would result from adding that feature to the field. This reduction is approximated as a function of a single parameter and is equal to the exponential loss reduction. This approximation is one of the key elements making it practical to evaluate a large number of candidate features at each stage of the induction algorithm. By using an auxiliary function to bound the change in generalized K-L divergence from below, the iterative scaling algorithm can be derived and thus convergence to the global optimal solution is proved.

The first of these two methods searches the weak learner myopically. Thus it only approximately finds the best one and then obtains this weak learner's optimal voting parameter λ. The second method, in contrast, looks into weak learners one by one, then chooses the weak learner that induces largest loss reduction. However, it is more computationally expensive since the optimal voting parameter for each weak learner has to be calculated. Moreover, for general loss functions, it is in general very hard to construct auxiliary functions with closed form solutions. In this paper, therefore, we adopt the first approach.

3. GENERIC SEMI-SUPERVISED BOOSTING ALGORITHM

Let X be a random variable over data to be labeled, and Y be a random variable over corresponding labels ranging over a finite label alphabet $\mathcal{Y}$. Assume we have a set of labeled examples, $\mathcal{D}^l = \Big((x_1, y_1), \cdots, (x_N, y_N)\Big)$, and unlabeled examples, $\mathcal{D}^u = \Big(x_{N+1}, \cdots, x_M\Big)$. We would like to construct a discriminant function of the form

$$h_T(x) = \lambda_1 h(x;\theta_1) + \cdots + \lambda_T h(x;\theta_T) \qquad (1)$$

such that the prediction error is small. Here $h(x;\theta_t) : X \rightarrow Y$ denote weak learners, for example, decision stumps whose predictions are +1 and -1, from a fixed class $\mathcal{H}$, characterized by a set of parameters θ and $\lambda_t \in \Re$ are the weak learner weights. They also correspond to the features in random fields (Della Pietra et al. 1997) and sufficient statistics in an exponential model (Lebanon and Lafferty, 2002). Our goal is to learn such a model from the combined set of labeled and unlabeled examples, $\mathcal{D}^l \cup \mathcal{D}^u$.

Just as in supervised learning case for boosting, the estimation for the combination is simply minimization of the following surrogate risk functional over 0/1 loss;

$$J(h_t) = \sum_{i=1}^{N} \mathrm{L}_l(y_i h_t(x_i)) + \gamma \sum_{i=N+1}^{M} \mathrm{L}_u(h_t(x_i)) \qquad (2)$$

where the first term denotes the surrogate loss for labeled data, which is a monotonically decreasing and differentiable function of its argument $y_i h_t(x_i)$, such that the more the discriminant function agrees with the label y_i, the smaller the loss. The second term represents the surrogate loss for unlabeled data which behaves like a clustering criterion. γ is a trade-off parameter that controls the influence of the unlabeled data.

To derive the boosting algorithm that can accommodate any loss function, suppose that we have already included $t-1$ component classifiers

$$h_{t-1}(x) = \lambda_1 h(x; \hat{\theta}_1) + \cdots + \lambda_t h(x; \hat{\theta}_{t-1}) \qquad (3)$$

and we wish to add another $h(x;\theta)$. The estimation criterion for the overall discriminant function, including the new component with votes λ, is given by

$$\begin{aligned} J(\lambda,\theta) &= \sum_{i=1}^{N} \mathrm{L}_l\left(y_i h_{t-1}(x_i) + y_i \lambda h(x_i;\theta)\right) \qquad (4)\\ &\quad + \gamma \sum_{i=N+1}^{M} \mathrm{L}_u\left(h_{t-1}(x_i) + \lambda h(x;\theta)\right) \end{aligned}$$

Note that we explicate only how the objective depends on the choice of the last component and the corresponding votes, since the parameters of the $t-1$ previous components along with their votes have already been set and won't be modified further.

As in the case of supervised boosting, there are two parameters to optimize. We implement this optimization approximately in two steps. We first find the new component or parameters θ so as to maximize its potential in reducing the surrogate loss, "potential" in the sense that we can subsequently adjust the votes to actually reduce the surrogate loss. More precisely, we set θ so as to minimize the derivative

$$\begin{aligned} &\frac{d}{d\lambda} J(\lambda,\theta)|_{\lambda=0} \\ =\ & \sum_{i=1}^{N} \frac{d}{d\lambda} \mathrm{L}_l\left(y_i h_{t-1}(x_i) + y_i \lambda h(x_i;\theta)\right)|_{\lambda=0} \\ & + \gamma \sum_{i=N+1}^{M} \frac{d}{d\lambda} \mathrm{L}_u\left(h_{t-1}(x_i) + \lambda h(x;\theta)\right)|_{\lambda=0} \\ =\ & \sum_{i=1}^{N} \mathrm{dL}_l\left(y_i h_{t-1}(x_i)\right) y_i h(x_i;\theta) \\ & + \gamma \sum_{i=N+1}^{M} \sum_{y} \mathrm{dL}_u\left(y h_{t-1}(x_i)\right) y h(x_i;\theta) \end{aligned}$$

where $\mathrm{dL}(z) = \frac{\mathrm{dL}(z)}{dz}$. Note this derivative $\frac{d}{d\lambda} J(\lambda,\theta)_{|\lambda=0}$ precisely captures the amount by which we would start to reduce the surrogate loss if we gradually increase the votes for the new component with parameters θ. Minimizing this reduction seems like a sensible estimation criterion for the new component or θ. This strategy permits us to set θ and subsequently optimize λ to actually minimize the surrogate loss.

Define the following weights and normalized weights $\underline{w}$ on the training examples:

$$\begin{aligned} w_i^{(t-1)} &= -\mathrm{dL}_l\left(y_i h_{t-1}(x_i)\right) \text{ for } i = 1, \cdots, N \\ w_i^{(t-1)}(y) &= -\mathrm{dL}_l\left(y h_{t-1}(x_i)\right) \text{ for } i = N+1, \cdots, M \quad \forall y \end{aligned}$$

Note that for each piece of unlabeled data, since its label is unknown, we assign an individual weight for each possible label using the *derivative of loss functions for labeled data*. In fact, how to choose the weight on each labeled and unlabeled data is quite arbitrary, but the main purpose is to incorporate a scaling procedure for numerical consideration, since each time we add a weak classifier, the cost function becomes smaller and will exceed the precision range of essentially any machine (even in double precision). Thus defining a weight is the only reasonable way to perform the computation.

Then the normalized weights $\underline{w}$ on the training examples are

$$\begin{aligned} \tilde{w}_i^{(t-1)} &= \frac{w_i^{(t-1)}}{\sum_{i=1}^{N} w_i^{(t-1)} + \sum_{i=N+1}^{M} \sum_y w_i^{(t-1)}(y)} \\ &\quad \text{for } i = 1, \cdots, N \\ \tilde{w}_i^{(t-1)}(y) &= \frac{w_i^{(t-1)}(y)}{\sum_{i=1}^{N} w_i^{(t-1)} + \sum_{i=N+1}^{M} \sum_y w_i^{(t-1)}(y)} \\ &\quad \text{for } i = N+1, \cdots, M \quad \forall y \end{aligned}$$

These weights are guaranteed to be non-negative since the loss function for labeled data is a decreasing function of its argument (its derivative has to be negative or zero). By ignoring the multiplicative constant (constant at iteration t) we will estimate θ by minimizing

$$\begin{aligned} & -\sum_{i=1}^{N} \tilde{w}_i^{(t-1)} y_i h(x_i;\theta) \qquad (5)\\ & + \gamma \sum_{i=N+1}^{M} \sum_{y} \frac{\mathrm{dL}_u\left(y h_{t-1}(x_i)\right)}{\tilde{w}_i^{(t-1)}(y)} \tilde{w}_i^{(t-1)}(y) y h(x_i;\theta) \end{aligned}$$

After we find $\hat{\theta}$, we solve the minimization problem for λ_t over the following objective function,

$$\begin{aligned} J(\lambda, \hat{\theta}_t) &= \sum_{i=1}^{N} \mathrm{L}_l\left(y_i h_{t-1}(x_i) + y_i \lambda h(x_i; \hat{\theta}_t)\right) \qquad (6)\\ &\quad + \gamma \sum_{i=N+1}^{M} \mathrm{L}_u\left(h_{t-1}(x_i) + \lambda h(x_i; \hat{\theta}_t)\right) \end{aligned}$$

This can be done by one-dimensional numerical line search [1].

[1] The search is quite expensive since it is to be performed at each round t. In fact, we can compute from the data a finite interval to which we know that λ belongs. This gives us a formula for a worst case search of approximate λ (Janodet et al. 2004).

We are now ready to cast the steps of the semi-supervised boosting algorithm as function gradient descent in a form similar to AdaBoost.

Generic function gradient descent semi-supervised boosting algorithm

1. Initialize the observation weights $\underline{w}$
2. For $t = 1$ to T;
   (a) Compute the negative Gateaux derivative $dJ(\cdot)$ of the functional $J(\cdot)$,
   $$-dJ(f)(x) = -\frac{\partial}{\partial\lambda} J(f + \lambda\delta_x)|_{\lambda=0}$$
   then fit a classifier $h_t(x, \theta)$ to this gradient using weights $\underline{w}$, Thus $h_t(x, \theta)$ can be viewed as an approximation of the negative vector.

   (b) Set the votes λ_t for the new component by minimizing the overall surrogate loss (6).

   (c) Update the weights on the training examples, labeled and unlabeled, based on the new base learner:
   $$\begin{aligned} w_i^{(t)} &= -\alpha \, \mathrm{dL}_l \left(y_i h_t(x_i))\right) \\ & \quad \text{for } i = 1, \cdots, N \\ w_i^{(t)}(y) &= -\alpha \, \mathrm{dL}_l \left(y h_t(x_i))\right) \\ & \quad \text{for } i = N+1, \cdots, M, \;\; \forall y \end{aligned}$$
   where $h_t(x_i) = h_{t-1}(x_i) + \lambda_t h(x_i; \hat{\theta}_t)$ and α is chosen such that $\sum_{i=1}^{N} \tilde{w}_i^{(t-1)} + \sum_{i=N+1}^{M} \sum_y \tilde{w}_i^{(t-1)}(y)$ $= 1$, this ensures that the new weights sum to one after the update.
3. Output final classifier
   $$h_T(x) = \sum_{t=1}^{T} \lambda_t h_t(x; \hat{\theta}_t)$$

The following result shows the convergence of the above algorithm to a local minimum or stop early at round T. Let L_l and L_u be any lower bounded Lipschitz differentiable cost functionals. Either the sequence of combined classifiers generated by the algorithm above halts on round T with its gradient being positive, or the combined loss function converges to a local minimum J^*, in which case $\lim_{t\to\infty} < \nabla J(h_t), h_t >= 0$. A similar result has been shown in (Mason et al. 1999), where $J(\cdot)$ is a convex function to guarantee convergence to a global minimum, here we relax $J(\cdot)$ to be non-convex, thus only local minimum can be reached. The proof technique is similar to that in (Mason et al. 1999) with minor modifications (Bertsekas 1999).

4. INFORMATION THEORETIC REGULARIZATION APPROACH

We use information-theoretic measures, entropy and mutual information, as regularization for the use of unlabeled data. The rationale of using these terms has been explained in Grandvalet and Bengio (2004), Jiao et al. (2006) and Wang et al. (2009). Unfortunately both measures on unlabeled data are not convex over λ, mainly because they are composition functions of convex/concave functions over the parameters λ. Jiao et al. (2006) explained one simple case, the entropy of exponential models; justification for general situations can be found in (S. Boyd and L. Vandenberghe 2004).

For ease of exposition, we first consider binary classification, that is, $y \in \{-1, 1\}$. We then extend to classification with multiple classes. In both cases, we use normalized log-linear models $p(y|x) = \frac{e^{(-yh(x))}}{\sum_y e^{(-yh(x))}}$[2].

4.1 Binary Classification

Consider normalized log-linear models $p(y|x) = \frac{e^{(-yh(x))}}{\sum_y e^{(-yh(x))}}$, we use the logistic loss, that is, negative log-probability, for labeled data,

$$\mathrm{L}_l(y_i h_t(x_i)) = -\log p(y_i|x_i) = \log(1 + e^{(-y_i h_t(x_i))}) \qquad (7)$$

Note that there should exist a constant 2 inside the exponential but we omit it by scaling the weaker learners by half.

Let $\mathrm{L}_l(z) = \log(1 + e^{-z})$, then $\mathrm{dL}_l(z) = -\frac{e^{-z}}{\log(1+e^{-z})}$, thus the weights are given by

$$\begin{aligned} \hat{w}_i^{(t-1)} &= \alpha \, \frac{e^{-y_i h_{t-1}(x_i)}}{1 + e^{-y_i h_{t-1}(x_i)}} \;\; \text{for } i = 1, \cdots, N \\ \hat{w}_i^{(t-1)}(y) &= \alpha \, \frac{e^{-y h_{t-1}(x_i)}}{1 + e^{-y h_{t-1}(x_i)}} \;\; \text{for } i = N+1, \cdots, M, \; \forall y \end{aligned}$$

where $\alpha = \sum_{i=1}^{N} \frac{e^{-y_i h_{t-1}(x_i)}}{1+e^{-y_i h_{t-1}(x_i)}} + \sum_{i=N+1}^{M} \sum_y \frac{e^{-y h_{t-1}(x_i)}}{1+e^{-y h_{t-1}(x_i)}}$.

For unlabeled data, again we have two options. We can minimize either the negative conditional Kullback-Leibler divergence

$$-D(p(y|x), \mathcal{U}(y|x)) = -\sum_{x \in \mathcal{D}^u} \tilde{p}(x) \sum_y \left(p(y|x) \log \frac{p(y|x)}{\mathcal{U}(y|x)} \right) \quad (8)$$

where $\mathcal{U}(y|x)$ is the uniform distribution and $\tilde{p}(x)$ denotes the empirical distribution of X, or the mutual information of unlabeled data

$$\begin{aligned} I(\tilde{p}(x), p(y|x)) &= D(\tilde{p}(x)p(y|x), (\tilde{p}(x)p(y)) \\ &= \sum_{x \in \mathcal{D}^u} \tilde{p}(x) \sum_y \left(p(y|x) \log \frac{p(y|x)}{p(y)} \right) \quad (9) \end{aligned}$$

where $p(y) = \sum_x \tilde{p}(x) p(y|x)$. In the following, we illustrate the derivation for using both entropy and mutual information as regularization on unlabeled data.

Entropy regularization Minimizing negative conditional Kullback-Leibler divergence is equivalent to minimizing the sum of the conditional entropy of unlabeled data,

$$\begin{aligned} \mathrm{L}_u(h_t(x_i)) &= \sum_y L_u(y h_t(x_i)) = H(p(y|x_i)) \\ &= -\sum_y p(y|x_i) \log p(y|x_i) \\ &= \sum_y \frac{1}{1 + e^{(-y h_t(x_i))}} \log(1 + e^{(-y h_t(x_i))}) \end{aligned}$$

Looking at the above formula, clearly entropy regularization is merely plain boosting with the unlabeled examples

[2] At the first sight, it seems that similar idea can be applied to unnormalized model once we use generalized conditional Kullback-Leibler divergence (Lebanon and Lafferty 2002) to define the entropy and mutual information for unnormalized model, however it doesn't work for entropy case, the surrogate loss in Step 2(b) in this case is lowerly unbounded.

replaced by labeled examples with all classes, where each class is assigned a weight by the class conditional probability given unlabeled example.

Let $\mathrm{L}_u(z) = \frac{1}{1+e^{-z}} \log(1+e^{-z})$, then $\mathrm{dL}_u(z) = \frac{e^{-z}}{(1+e^{-z})^2}$ $(-1+\log(1+e^{-z}))$.

The loss function for the weaker learner in step 2(a) is

$$\begin{aligned}
& \frac{d}{d\lambda} J(\lambda,\theta)_{|\lambda=0} \qquad (10) \\
= & \sum_{i=1}^{N} \mathrm{dL}_l \left(y_i h_{t-1}(x_i)\right) y_i h(x_i;\theta) \\
& + \gamma \sum_{i=N+1}^{M} \sum_y \mathrm{dL}_u \left(y h_{t-1}(x_i)\right) y h(x_i;\theta) \\
= & -\sum_{i=1}^{N} \tilde{w}_i^{(t-1)} y_i h(x_i;\theta) - \gamma \sum_{i=N+1}^{M} \sum_y \\
& \frac{(-1+\log(1+e^{-y h_{t-1}(x_i)}))}{(1+e^{(-y h_{t-1}(x_i))})} \tilde{w}_i^{(t-1)}(y) y h(x_i;\theta)
\end{aligned}$$

We look over all of the weak learners and choose the one $h(\cdot;\hat{\theta})$ which has the lowest value of this loss function.

The minimization over the surrogate loss in Step 2(b) is used to determine the optimal value of λ.

Mutual information regularization Minimizing mutual information of unlabeled data is equivalent to minimizing the sum of the difference between the entropy of unlabeled data and the conditional entropy of unlabeled data,

$$\begin{aligned}
\mathrm{L}_u(h_t(x_i)) &= \sum_y L_u(y h_t(x_i)) \\
&= H(p(y)) - H(p(y|x_i)) \\
&= -\sum_y \sum_x \tilde{p}(x) p(y|x_i) \log \sum_x \tilde{p}(x) p(y|x_i) \\
&\quad + \sum_y p(y|x_i) \log p(y|x_i) \\
&= -\frac{\log(M-N)}{M-N} \sum_y \sum_{i=N+1}^{M} \frac{1}{1+e^{(-y h_t(x_i))}} \\
&\quad + \frac{1}{M-N} \sum_y \sum_{i=N+1}^{M} \frac{1}{1+e^{(-y h_t(x_i))}} \cdot \\
&\qquad \log\Big(\sum_{i=N+1}^{M} \frac{1}{1+e^{(-y h_t(x_i))}}\Big) \\
&\quad - \sum_y \frac{1}{1+e^{(-y h_t(x_i))}} \log(1+e^{(-y h_t(x_i))})
\end{aligned}$$

Let $\mathrm{L}_u(z) = -\frac{\log(M-N)}{M-N} \sum_{i=N+1}^{M} \frac{1}{1+e^{(-z)}} + \frac{1}{M-N} \sum_{i=N+1}^{M}$ $\frac{1}{1+e^{(-z)}} \log(\sum_{i=N+1}^{M} \frac{1}{1+e^{(-z)}}) + \frac{1}{1+e^{(-z)}} \log(1+e^{(-z)})$, then $\mathrm{dL}_u(z)$ can be computed easily.

The loss function for the weaker learners in step 2(a) is

$$\begin{aligned}
& \frac{d}{d\lambda} J(\lambda,\theta)_{|\lambda=0} \qquad (11) \\
= & \sum_{i=1}^{N} \mathrm{dL}_l \left(y_i h_{t-1}(x_i)\right) y_i h(x_i;\theta)
\end{aligned}$$

$$\begin{aligned}
& + \gamma \sum_{i=N+1}^{M} \sum_y \mathrm{dL}_u \left(y h_{t-1}(x_i)\right) y h(x_i;\theta) \\
= & -\sum_{i=1}^{N} \tilde{w}_i^{(t-1)} y_i h(x_i;\theta) \\
& -\gamma \log(M-N) \sum_{i=N+1}^{M} \sum_y \frac{\tilde{w}_i^{(t-1)}(y) y h(x_i;\theta)}{(1+e^{(-y h_{t-1}(x_i))})} \\
& + \gamma \sum_{i=N+1}^{M} \sum_y \frac{(-1+\log(1+e^{-y h_{t-1}(x_i)}))}{(1+e^{(-y h_{t-1}(x_i))})} \cdot \\
& \qquad \tilde{w}_i^{(t-1)}(y) y h(x_i;\theta) \\
& -\gamma \sum_y \sum_{i=N+1}^{M} \frac{\log(\sum_{i=N+1}^{M} \frac{1}{1+e^{-y h_{t-1}(x_i)}} + 1)}{(1+e^{(-y h_{t-1}(x_i))})} \cdot \\
& \qquad \tilde{w}_i^{(t-1)}(y) y h(x_i;\theta)
\end{aligned}$$

Again we look over all of the weak learners and choose the one $h(\cdot;\hat{\theta})$ which has the lowest value of this loss function.

The minimization over the surrogate loss in Step 2(b) is used to determine the optimal value of λ.

4.2 Multi-class Classification

In the multi-class classification setting, we use the approach proposed in (Zhu et al. 2005) and recode the class label $y \in \mathcal{Y} = \{1, \cdots, K\}$ with a K-dimensional vector $\mathbf{c}$, with all entries equal to $-\frac{1}{K-1}$ except a 1 in position k if $y = k$, i.e. $\mathbf{c} = (c_1, \cdots, c_K)^T$, and

$$c_k = \begin{cases} 1, & \text{if } y = k, \\ -\frac{1}{K-1}, & \text{if } y \neq k. \end{cases}$$

and there is a one-to-one correspondence between y and $\mathbf{c}$. We use $\mathbf{c}(y)$ to denote the vector corresponding to class y.

A generalization of the exponential loss function for the labeled data to the multi-class case then naturally follows:

$$\mathrm{L}_l(\mathbf{c}(y), \mathbf{h}(x)) \quad = \quad e^{(-\frac{1}{K}\mathbf{c}(y)^T \mathbf{h}(x))}$$

where $\mathbf{h}(x) = (h^1(x), \cdots, h^K(x))^T$ and $h^k(x)$ correspond to class k and $\mathbf{h}(x)$ satisfies symmetric constraint:

$$h^1(x) + \cdots + h^K(x) = 0$$

So when $K = 2$, this multi-class exponential loss function reduces to the binary exponential loss.

We require the weak learner $\mathbf{h}(x;\theta)$ to satisfy the symmetric constraints:

$$h^1(x;\theta) + \cdots + h^K(x;\theta) = 0 \qquad (12)$$

Specifically, at a given x, $\mathbf{h}(x)$ maps x onto $\mathcal{C}, \mathbf{h} : x \to \mathcal{C}$, where $\mathcal{C}$ is the set containing K K-dimensional vectors:

$$\mathcal{C} = \left\{ \begin{array}{c} (1, -\frac{1}{K-1}, \cdots, -\frac{1}{K-1})^T \\ (-\frac{1}{K-1}, 1, \cdots, -\frac{1}{K-1})^T \\ \cdot \\ \cdot \\ \cdot \\ (-\frac{1}{K-1}, \cdots, -\frac{1}{K-1}, 1)^T \end{array} \right\}$$

It is easy to check that all the derivation for binary classification will remain the same for multi-class classification with simple modular modification: substitute $yh(x)$ with $\frac{1}{K}\mathbf{c}(y)^T \mathbf{h}(x)$.

It is natural to see that the normalized log-linear model is $p(y|x) = \frac{e^{(-\frac{1}{K}\mathbf{c}(y)^T\mathbf{h}(x))}}{\sum_y e^{(-\frac{1}{K}\mathbf{c}(y)^T\mathbf{h}(x))}}$, the logistic loss, that is, negative log-probability, for labeled data, is

$$\mathrm{L}_l(y, \mathbf{h}(x)) = \log(1 + \sum_{y'} e^{(-\frac{1}{K}(\mathbf{c}(y)-\mathbf{c}(y'))^T\mathbf{h}(x))}) \qquad (13)$$

The entropy of unlabeled data is then

$$\begin{aligned}\mathrm{L}_u(\mathbf{h}(x)) &= \sum_y L_u(y, \mathbf{h}(x)) = -\sum_y p(y|x)\log p(y|x) \\ &= \sum_y \frac{1}{1+e^{(-\frac{1}{K}\mathbf{c}(y)^T\mathbf{h}(x))}} \cdot \qquad (14) \\ &\qquad \log(1 + \sum_{y'} e^{(-\frac{1}{K}(\mathbf{c}(y)-\mathbf{c}(y'))^T\mathbf{h}(x))})\end{aligned}$$

Again it is easy to check that all the derivation for binary classification will remain the same for multi-class classification with simple modular modification: substitute $e^{(-yh_{t-1}(x))}e^{(-y\lambda h(x;\hat{\theta}_t))})$ with $\sum_{y'} e^{(-\frac{1}{K}(\mathbf{c}(y)-\mathbf{c}(y'))^T\mathbf{h}_{t-1}(x)))}$ $e^{(-\lambda\frac{1}{K}(\mathbf{c}(y)-\mathbf{c}(y'))^T\mathbf{h}(x;\hat{\theta}_t))})$; substitute $e^{(-yh_{t-1}(x))}$ with $\sum_{y'} e^{(-\frac{1}{K}(\mathbf{c}(y)-\mathbf{c}(y'))^T\mathbf{h}_{t-1}(x)))}$ and substitute $yh(x)$ with $\frac{1}{K}\mathbf{c}(y)^T\mathbf{h}(x))$.

Similar derivations can be obtained for mutual information case.

The final classifier is given by

$$\mathbf{h}_T(x) = \sum_{t=1}^{T} \lambda_t \mathbf{h}_t(x : \hat{\theta}_t) \qquad (15)$$

and for new sample data x, we assign its class label as

$$y = \arg\max_{y'} \mathbf{c}(y')^T\mathbf{h}_T(x) \qquad (16)$$

5. EXPERIMENTAL EVALUATION

In this section, we report experimental results on synthetic, benchmark, and real world data. It is important to note that it is not our intention to show that the proposed semi-supervised boosting algorithm always outperforms a variety of 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 the well-known supervised boosting algorithms using the unlabeled examples, and it is more effective than the existing semi-supervised boosting algorithms. Hence, the empirical study is focused on a comparison with the existing supervised and semi-supervised boosting algorithms, rather than a wide range of semi-supervised learning algorithms. To be more specific, we evaluate empirically supervised AdaBoost and LogitBoost, and a state-of-the-art semi-supervised boosting algorithm, Assemble (Bennett et al. 2002), (Assemble.LogitBoost) with our proposed semi-supervised boosting methods. The weak learner we used in all of our experiments is the decision stump FindAttrTest as described in (Freund and Schapire, 1996). All of the experiments are repeated 10 times, the results are expressed by the mean and its standard deviation.

5.1 Synthetic Data

We first consider a 2-class problem with a 10-dimensional input space. Each class is generated by a normal distribution with equal mixing weight probability. The mean of the first class is $(4, \cdots, 4)^T$ and the mean of the second is $(-4, \cdots, -4)^T$. The covariance matrix is diagonal matrix with standard deviation being 20. We set the number of labeled data $N = 50$ and we increase the number of unlabeled data from 50 to 1500. We use development data to choose the best value of regularization parameter, i.e. γ in our proposed method and α and β in Assemble method. The size of development and test data is 450.

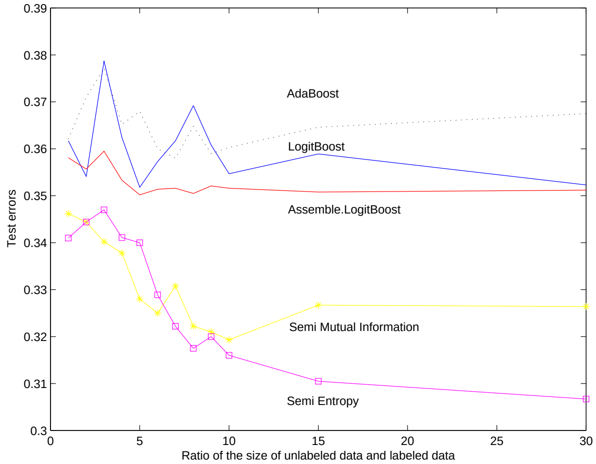

Figure 1: Test errors on data generated by two mixtures of 10-dimensional Gaussian distribution when we increase the size of unlabeled data.

Figure 1 and Table 1 show the results of the classification error rate on test data using Assemble.LogitBoost and both entropy and mutual information on unlabeled data as regularization term when we increase the size of unlabeled data. Clearly this result shows that our semi-supervised boosting algorithms overcomes both AdaBoost and LogitBoost as well as Assemble.LogitBoost. Especially, unlabeled data does help improve performance. With the increment of unlabeled data, the error rates of our algorithms tend to decrease.

In the experiment, we also find that, in terms of error rate on test data, both entropy based semi-supervised boosting algorithm and mutual information based semi-supervised boosting algorithm converge more quickly than AdaBoost and LogitBoost. Figure 2 shows how the error rate on test data changes at each iteration when the ratio of unlabeled data and labeled data is set to 5. Clearly both entropy and mutual information regularization terms on unlabeled data behave as excellent data dependent regularizers, and entropy based approach has better regularization effect than mutual information based approach. We've also used l_2 norm of the parameters suggested in (Lebanon and Lafferty 2002) as a regularization term, but unfortunately it has almost no affect on the performance.

Since the minimization problem in Step 2(b) is simply one-dimensional one, we study the objective function being minimized when we use conditional entropy on unlabeled data as a regularization term. Figures 3 and Figure 4 illustrate the loss curves at the 5th iteration where we set $\gamma = 0.2$ and $\gamma = 1$ respectively, and the ratio of unlabeled data and labeled data is 5. The loss function on labeled data is convex and the loss function on unlabeled data is non-convex. When the regularization parameter γ is small,

Ratio	AdaBoost	LogitBoost	Assemble.LogitBoost	γ (MI)	MI	γ (Entropy)	Entropy
1	36.21(4.59)	36.17(3.31)	35.81(3.99)	.0070	34.62(3.45)	.0025	34.10(2.52)
2	37.10(4.23)	35.41(4.34)	35.57(3.80)	.0050	34.44(3.31)	.0023	34.44(2.68)
3	37.72(2.45)	37.87(3.80)	35.95(3.26)	.0030	34.02(4.24)	.0030	34.70(4.77)
4	36.52(3.23)	36.24(6.27)	35.33(3.63)	.0025	33.78(2.91)	.0025	34.11(2.82)
5	36.80(2.34)	35.18(5.12)	35.02(3.20)	.0018	32.80(3.22)	.0024	34.00(2.97)
6	36.00(3.13)	35.73(4.31)	35.14(2.33)	.0019	32.50(3.40)	.0022	32.89(2.27)
7	35.80(2.31)	36.12(5.01)	35.16(2.39)	.0018	33.08(3.01)	.0022	32.22(2.44)
8	36.48(3.67)	36.92(4.27)	35.05(2.43)	.0012	32.22(2.48)	.0020	31.75(2.34)
9	35.90(2.31)	36.08(4.33)	35.21(2.34)	.0013	32.10(2.56)	.0015	32.00(2.78)
10	36.03(2.52)	35.47(4.64)	35.16(2.43)	.0011	32.93(2.83)	.0011	31.60(1.95)
15	36.46(3.35)	35.89(4.89)	35.08(2.26)	.0010	32.67(2.17)	.0009	31.05(1.83)
30	36.75(3.23)	35.23(5.50)	35.12(1.61)	.0005	32.64(2.06)	.0006	30.67(1.90)

Table 1: Error rates (%) on Gaussian mixture test data and the corresponding values of regularization parameter for both entropy and mutual information (MI) based semi-supervised boosting when varying the ratio of the size of unlabeled and labeled Gaussian mixture data.

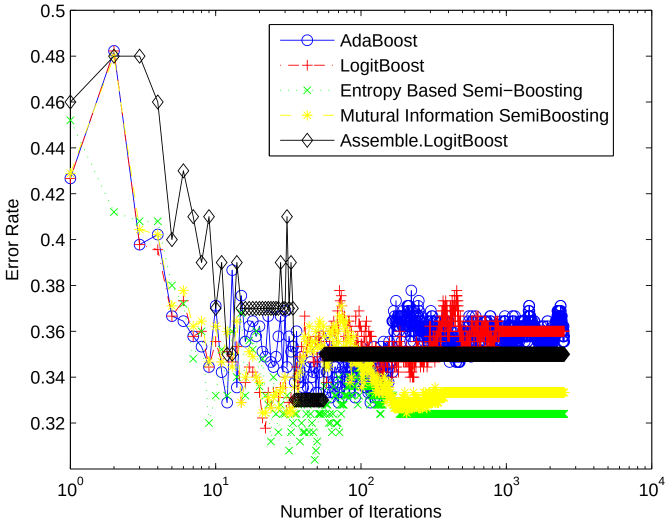

Figure 2: Test errors vary at each iteration with maximum iteration being 2500 where the ratio of unlabeled data and labeled data is set to 5.

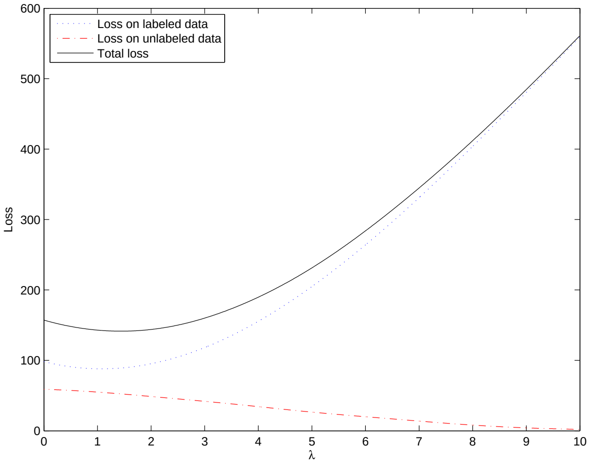

Figure 3: Loss curves to be minimized at the 5th iteration with regularization parameter $\gamma = 0.2$.

the loss function on labeled data dominates to force the total loss function to be convex, this guarantees that we indeed find a global minimum solution using line search; When the regularization parameter γ is large, the loss function on labeled data is not able to dominate, so the total loss function is non-convex, as Figure 4 illustrates, there are one minimum and several saddle points exist. When the iteration gets larger, the loss functions look more like in Figure 3 and the total loss function becomes convex, this is true even for large γ. In all the experiments on synthetic, benchmark and real data, usually smaller regularization parameter γ gives better test error, thus non-convexity of the objective function is not a big issue at all in practice. We have similar observation when we use mutual information on unlabeled data as a regularization term.

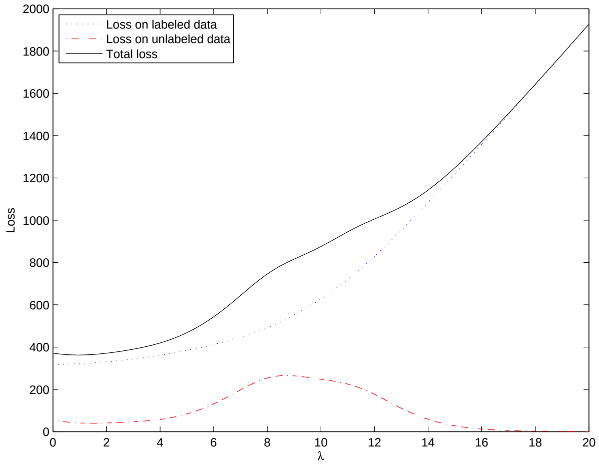

Figure 4: Loss curves to be minimized at the 5th iteration with regularization parameter $\gamma = 1$.

Finally we apply our algorithms to 3-class synthetic data. Each class is still generated by a 10-dimensional normal distribution with equal mixing weight probability. The means of each class are $(-8, \cdots, -8)^T$, $(0, \cdots, 0)^T$ and $(8, \cdots, 8)^T$ respectively. The covariance matrix is diagonal with standard deviation being 10. The number of labeled data is $N = 30$, and the number of unlabeled data is 210. We also use unlabeled data as test data. In this experiment, the error rate of LogitBoost is 33.81% (2.71), the error rate of Assemble.LogitBoost is 32.86% (4.49), while error rate of the entropy based semi-supervised boosting is 30.47% (2.04) with $\gamma = 0.05$, and error rate of mutual information semi-supervised boosting is 29.50% (2.82) with $\gamma = 0.05$.

5.2 Benchmark Data

In this experiment, we use several benchmark datasets: Balance scale weight & distance, Pima, Wisconsin diagnostic breast cancer and BUPA liver disorders, in UCI Machine Learning Repository to test the performance of our proposed semi-supervised boosting algorithms. We use 15% as labeled data and 85% as unlabeled data. These unlabeled data are used as the test data in the experiment. Table 2 shows the results of the classification error rates on test data.

Data	Logit	Assemble	MI	Entropy
Bala	27.43(1.52)	25.76(1.47)	24.80(1.72)	24.10(2.02)
Pima	22.50(2.52)	20.87(3.47)	20.44(3.75)	19.87(3.03)
Wins	5.14(0.74)	4.15(1.12)	2.92(0.77)	3.77(1.07)
BUPA	37.24(5.59)	36.17(3.40)	29.84(3.79)	31.77(2.31)

Table 2: Error rates (%) on four benchmark UCI data sets where the corresponding values of regularization parameter γ for entropy based method are 0.01, 0.001, 0.10 and 0.01 and those for mutual information based method are 0.007, 0.001, 0.10 and 0.10.

This experiment shows that our semi-supervised learning algorithms can get helpful information from unlabeled data and significantly improve the classification results.

5.3 Human Mental Workload Data

We now report results on a real-world problem: human mental workload classification task in modern aviation systems. Mental workload refers to the information processing demands imposed on the operator by the performance of cognitive tasks, accurate and reliable real-time assessment of operators' cognitive states, i.e. their mental abilities to carry out the jobs in time is the key for successful implementation of adaptive human-aiding techniques in modern aviation systems, such as uninhabited air vehicles (UAVs) and uninhabited combat air vehicles (UCAVs). One of the measures of mental workload is the Electroencephalogram (EEG), a measurement of electrical activity produced by the brain as recorded from electrodes placed on the scalp, and it is used to predict human workload.

In this experiment, we use real EEG data to model human work load. The input variable X is a 105 dimensional vector and the class label Y has three states "Low", "Medium" and "High".

First we combine the "Low" class and "Medium" class together as one class and solve a binary classification problem. We set the number of labeled data to be $N = 30$, we then increase the number of unlabeled data from 60 to 240. We use development data to choose the best value of regularization parameter. The size of development data is 50. We repeat this process 10 times.

Figure 5 and Table 3 show the results of the classification error rate on EEG test data using both entropy and mutual information on unlabeled EEG data as regularization terms when we increase the size of unlabeled EEG data.

Finally we consider 3-class cases of EEG data, in which we do not combine "Low" class and "Medium" class. In this experiment, the number of labeled data is 30, and the number of unlabeled data is 70. Unlabeled data are still the test data. We repeat this process 10 times. The error rate of LogitBoost is 32.94% (2.47); the error rate of Assemble.LogitBoost is 31.01% (1.97); the error rate of our entropy semi-supervised boosting algorithm is 29.43% (2.44) when $\gamma = 0.07$, and the error rate of mutual information based semi-supervised boosting algorithm is 30.58% (2.51) when $\gamma = 0.05$.

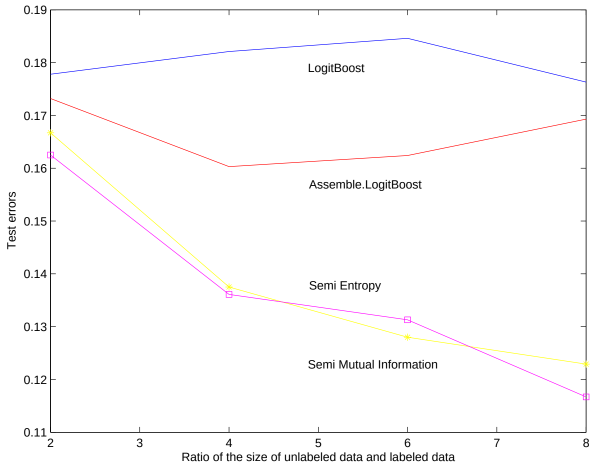

Figure 5: Test errors for LogitBoost (blue), Assemble.LogitBoost (red), semi-supervised entropy based boosting (magenta) and semi-supervised mutual information based boosting (yellow) on EEG data when we increase the size of unlabeled data.

Ratio	Logit	Assemble	MI	Entropy
2	17.78(3.01)	17.32(2.32)	16.67(2.60)	16.25(3.70)
4	18.21(3.23)	16.03(2.17)	13.75(2.59)	13.61(3.70)
6	18.46(1.63)	16.24(2.25)	12.80(2.31)	13.13(2.51)
8	17.63(1.78)	16.93(1.84)	12.29(1.68)	11.67(1.10)

Table 3: Error rates on EEG test data for both entropy and mutual information based semi-supervised boosting when varying the ratio of the size of unlabeled and labeled EEG data, where the corresponding values of regularization parameter γ for both entropy based method are 0.015, 0.007, 0.015 and 0.009 and mutual information based method are 0.0015, 0.0009, 0.0005 and 0.0005.

6. CONCLUSION

In this paper, we present semi-supervised boosting learning where information theoretic terms, both entropy and mutual information, are used to encode the information provided by unlabeled data and behave as data dependent priors. The combined loss functions are non-convex, we derive simple sequential gradient descent optimization algorithms and test these algorithms on synthetic, benchmark and real world tasks. Experimental results show that by exploiting the availability of auxiliary unlabeled data, our proposed semi-supervised boosting algorithms can impressively improve the performances of both supervised boosting algorithms and a state-of-the-art semi-supervised boosting algorithm. We are working on a formal analysis to give some theoretical justifications on why these information measures can be used to improve classification performance.

7. ACKNOWLEDGEMENT

The authors wish to thank researchers at the 711th HPW/ RHCP lab of the Wright Patterson Air Force Base for providing them the EEG human mental workload classification dataset.

8. REFERENCES

[1] K. Benett, A. Demiriz and R. Maclin. Exploiting unlabeled data in ensemble methods. *The 8th International Conference on Knowledge Discovery and Data Mining*, 289-296, 2002.

[2] D. Bertsekas. *Nonlinear Programming*, 2nd Edition, Athena Scientific, 1999.

[3] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. *The Workshop on Computational Learning Theory*, 92-100, 1998.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*, Cambridge University Press, 2004.

[5] L. Breiman. Prediction games and arcing classifiers. *Neural Computation*, 11:1493-1517, 1999.

[6] V. Castelli and T. Cover. The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Trans. on Information Theory*, 42(6):2102-2117, 1996.

[7] G. Celeux and G. Govaert. A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, 14:315-332, 1992.

[8] O. Chapelle, B. Scholköpf and A. Zien. *Semi-Supervised Learning*, MIT Press, 2006.

[9] K. Chen and S. Wang. Regularized boost for semi-supervised learning. *Advances in Neural Information Processing Systems 20*, 2007.

[10] I. Cohen and F. Cozman. Risks of semi-supervised learning. *Semi-Supervised Learning*, O. Chapelle, B. Scholköpf and A. Zien, 55-70, MIT Press, 2006.

[11] M. Collins, R. Schapire and Y. Singer. Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, 48(1-3):253-285, 2002.

[12] A. Corduneanu and T. Jaakkola. Data dependent regularization. *Semi-Supervised Learning*, O. Chapelle, B. Scholköpf and A. Zien, 163-182, MIT Press, 2006.

[13] T. Cover and J. Thomas. *Elements of Information Theory*, John Wiley & Sons, 1991.

[14] F. d'Alché-Buc, Y. Grandvalet and C. Ambroise. Semi-supervised marginBoost. *Advances in Neural Information Processing Systems 14*, 553-560, 2002.

[15] S. Della Pietra, V. Della Pietra and J. Lafferty. Inducing features of random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(4):380-393, 1997.

[16] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. *The 13th International Conference on Machine Learning*, 148-156, 1996.

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

[18] J. Friedman, T.Hastie and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28(2):337-407, 2000.

[19] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. *Advances in Neural Information Processing Systems*, 17:529-536, 2004.

[20] G. Haffari, Y. Wang, S. Wang, G. Mori and F. Jiao. Boosting with incomplete information. *The 25th International Conference on Machine Learning*, 368-375, 2008.

[21] T. Hastie, R. Tibshirani, J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd Edition, Springer, 2009.

[22] J. Janodet, R. Nock, M. Sebban and H. Suchier. Boosting grammatical inference with confidence oracles. *The 21st International Conference on Machine Learning*, 54-61, 2004.

[23] F. Jiao, S. Wang, C. Lee, R. Greiner and D. Schuurmans. Semi-supervised conditional random fields for improved sequence segmentation and labeling. *The Joint 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 209-216, 2006.

[24] G. Lebanon and J. Lafferty. Boosting and maximum likelihood for exponential models. *Advances in Neural Information Processing Systems 14*, 447-454, 2002.

[25] C. Lee, S. Wang, F. Jiao, D. Schuurmans and R. Greiner. Learning to model spatial dependency: Semi-supervised discriminative random fields. *Advances in Neural Information Processing Systems 19*, 793-800, 2007.

[26] L. Mason, J. Baxter, P. Bartlett and M. Frean. Functional gradient techniques for combining hypotheses. In *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Scholköpf and D. Schuurmans, editors, 221-246, MIT Press, 2000.

[27] K. Nigam, A. McCallum, S. Thrun and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning.* 39(2/3):135-167, 2000.

[28] S. Roberts, R. Everson and I. Rezek. Maximum certainty data partitioning. *Pattern Recognition*, 33(5):833-839, 2000.

[29] R. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197-227, 1990.

[30] H. Valizadegan, R. Jin and A. Jain. Semi-supervised boosting for multi-class classification. *The European Conference on Machine Learning and Knowledge Discovery in Databases*, 522-537, 2008.

[31] Y. Wang, G. Haffari, S. Wang and G. Mori. Rate distortion based semi-supervised discriminative learning. *Technical Report*, 2009.

[32] D. Zhou, O. Bousquet, T. Navin Lal, J. Weston and B. Schölkopf. Learning with local and global consistency. *Advances in Neural Information Processing Systems*, 16:321-328, 2004.

[33] J. Zhu, S. Rosset, H. Zhou and T. Hastie. Multiclass AdaBoost. *Technical Report*, 2005.

[34] X. Zhu, Z. Ghahramani and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. *The 20th International Conference on Machine Learning*, 912-919, 2003.