Efficient large margin semisupervised learning*

Junhui Wang
School of Statistics
University of Minnesota
Minneapolis, MN 55455

Abstract

In classification, semisupervised learning involves a large amount of unlabeled data with only a small number of labeled data. This imposes great challenge in that the class probability given input can not be well estimated through labeled data alone. To enhance predictability of classification, this article introduces a large margin semisupervised learning method constructing an efficient loss to measure the contribution of unlabeled instances to classification. The loss is iteratively refined, based on which an iterative scheme is derived for implementation. The proposed method is examined for two large margin classifiers: support vector machines and ψ-learning. Our theoretical and numerical analyses indicate that the method achieves the desired objective of delivering higher performances over any other method initializing the scheme.

* This research is supported by NSF grants IIS-0328802 and DMS-0604394.

1. Introduction

Semi-supervised learning occurs in the context of classification, where only a small number of labeled data is available but with a large amount of unlabeled data, particularly when it is costly to query labeled data. For instance, in webpage classification, a small number of manually labeled texts (web-pages) is usually available because of impracticability of manually labeling, together with a huge amount of unlabeled texts (web-pages) because of the speed of collecting texts by a machine; c.f., Blum and Mitchell (1998). This occurs also in spam email detection and face recognition, c.f., Baluja (1998); Amini and Gallinari (2003); Balcan, et al. (2005). In a situation as such, the primary goal is how to enhance predictability of classification by utilizing unlabeled and labeled data jointly. As a result, a machine's generalization ability is enhanced by integrating humans' intelligence with machine's processing speed.

In semisupervised learning, a labeled sample $(X^l, Y^l) = \{(X_i, Y_i)_{i=1}^{n_l}\}$ is observed according to an unknown distribution $P(x, y)$, together with an independent unlabeled sample $X^u = \{X_j\}_{j=n_l+1}^{n}$ according to distribution $P(x)$, where $Y = \pm 1$ indicates labeling, $n = n_l + n_u$, and n_l is usually much smaller than n_u. Here $P(x)$ may not be necessarily the marginal distribution of $P(x, y)$.

Two major approaches have been proposed in the literature; one is distributional while the other is margin-based. For a distributional approach, some assumptions are made to relate the marginal distribution of X to the conditional distribution $P(Y = 1|X = x)$. Distributional approaches include, among others, co-training (Blum and Mitchell, 1998), the EM method (Nigam, Mc-

Callum, Thrun and Mitchell, 1998), the bootstrap method (Collin and Singer, 1999), the cluster-and-label method (Dara, Kremer and Stacey, 2002), Bayesian Network (Cozman, Cohen and Cirelo, 2003), Gaussian random fields (Zhu, Lafferty and Ghahramani, 2003), and discriminative-generative models (Ando and Zhang, 2004). For a margin approach, a concept of margins is used in the form of regularization, including Transductive SVM (TSVM, Vapnik, 1998; Chapelle and Zien, 2005; Astorino and Fuduli, 2005; Wang, Shen and Pan, 2006) and Wang and Shen (2006).

The distributional approach makes various assumptions to relate $P(Y = 1|X = x)$ to the marginal distribution of X for an improvement to occur when the assumptions are met. However, these assumptions are usually not verifiable or satisfiable in practice. As a consequence, any departure from the assumptions is likely to degrade the "alleged" gain, and may even perform worse than its supervised counterpart; c.f., Zhang and Oles (2000).

The margin approach makes no distributional assumptions, utilizing unlabeled data directly to approximate the classification boundary. However, all existing margin methods use the unlabeled data in a weak way that relies purely on the notion of separation.

This article develops a large margin semisupervised learning method, with most effort focused towards utilizing unlabeled data more efficiently to deliver high performance of classification. Toward this end, we construct an efficient loss for unlabeled data by incorporating the knowledge of classification. This allows the loss to provide information about the optimal Bayes rule when a reasonably good initial estimate of the conditional class probability is given. On this basis, an iterative scheme is developed for implement. This is an analogy of Fisher's efficient scoring method with a consistent initial estimate, yielding an improvement over the initial estimate. The proposed method has been implemented for support vector machines (SVM) and ψ-learning. Numerical analysis indicates that the method improve better than or the same as the state-of-the-art methods.

This article is organized in five sections. Section 2 introduces the proposed semisupervised learning method. Section 3 develops an iterative algorithm. Section 4 presents some numerical examples, followed by a discussion in Section 5. The appendix contains technical proofs.

2. Methodology

2.1 Large margin classification

We begin with our discussion on classification with labeled data $(X_i, Y_i)_{i=1}^{n_l}$ alone. In the linear case, given a class of linear decision functions $\mathcal{F}$ of the form $f(x) = \tilde{w}_f^T x + w_{f,0} \equiv (1, x^T)w_f$, a cost function

$$C \sum_{i=1}^{n_l} L(y_i f(x_i)) + J(f)$$

is minimized over $f \in \mathcal{F}$ to obtain the minimizer $\hat{f}$ yielding a classifier $\mathrm{Sign}(\hat{f})$. Here $J(f) = \|\tilde{w}_f\|^2/2$ is the reciprocal of the L_2 geometric margin, and $L(\cdot)$ is a margin loss defined by functional margin $z = yf(x)$. In the nonlinear case, a kernel $K(\cdot, \cdot)$ mapping from $S \times S$ to $\mathcal{R}^1$ is introduced to give a flexible representation: $f_j(x) = \sum_{i=1}^{n_l} \alpha_{ij} K(x, x_i) + b_j$. For this reason, it is also referred to as kernel-based learning, where the reproducing kernel Hilbert spaces (RKHS) are useful, c.f., Gu (2000) and Wahba (1990).

Different margin losses correspond to different learning methodologies. Margin losses include, among others, the hinge loss $L(z) = (1-z)_+$ for SVM with its variants $L(z) = (1-z)_+^q$ for $q > 1$; c.f., Lin (2002); the ψ-losses $L(z) = \psi(z)$, with $\psi(z) = 1-\mathrm{Sign}(z)$ if $z \geq 1$ or $z < 0$, and $2(1-z)$ otherwise, c.f., Shen, Tseng, Zhang and Wong (2003), the logistic loss $V(z) = \log(1 + e^{-z})$, c.f., Zhu and Hastie (2005); the ρ-hinge loss $L(z) = (\rho-z)_+$ for nu-SVM (Schölkopf, Smola, Williamson and Bartlett, 2000) with $\rho > 0$ need to be optimized; the sigmoid loss $L(z) = 1 - \tanh(cz)$; c.f., Mason, et al. (2000). A margin

loss $L(z)$ is said to be large margin if $L(z)$ is non-increasing in z, penalizing small margin values.

2.2 Construction of cost function for unlabeled data

In margin classification (1), $f^* = \text{argmin}_{f\in\mathcal{F}} EL(Yf(X))$ is the target and is estimated from labeled data. In presence of a large amount of unlabeled data, the focus is how to leverage unlabeled data to improve upon (1). Toward this end, we construct a margin loss U (mapping: $R^1 \to R^1$) to measure the performance of unlabeled data with respect to estimating f^* for classification. Ideally, such a loss U needs to satisfy a requirement $\text{argmin}_{f\in\mathcal{F}} EU(f(X)) = \text{argmin}_{f\in\mathcal{F}} EL(Yf(X))$. To construct a loss satisfying this requirement, we seek the optimal loss U from a class of candidate losses of the form $T(f(x))$, which minimizes the L_2-distance between the target classification loss $L(yf(x))$ and $T(f(x))$. The expression of this optimal loss U is given in Lemma 1.

Lemma 1 *For any margin loss $L(z)$,*

$$\begin{aligned}&\underset{T}{\text{argmin}} E(L(Yf(X)) - T(f(X)))^2 \\ &\qquad = p(x)L(f(x)) + (1-p(x))L(-f(x)),\end{aligned}$$

where $p(x) = P(Y = 1|X = x)$. Moreover, $\text{argmin}_{f\in\mathcal{F}} E(L(Yf(X))|X) = \text{argmin}_{f\in\mathcal{F}} EL(Yf(X))$.

This optimal loss depends on unknown p that is a function of f^* depending on L and $\mathcal{F}$. For instance, when $L(yf(x)) = \log(1+\exp(-yf(x)))$ is the logistic loss for Import Vector Mahince (Zhu and Hastie, 2004), $p = \exp(f^*)/(1+\exp(f^*))$ provided that $\mathcal{F}$ is sufficiently rich. In the literature, several methods have been proposed to estimate the relationship between $\hat{p} = \hat{p}(\hat{f})$ and $\hat{f}$ for large margin classification, including Platt (1999) and Wang, Shen and Liu (2006). For construction, we define our loss as $U(f(x)) = \hat{p}(x)L(f(x)) + (1-\hat{p}(x))L(-f(x))$ with p in the optimal loss being replaced by its estimate $\hat{p}$ to be specified in Section 3.1. Evidently, U is nearly optimal provided that $\hat{p}$ is a reasonably good estimate of p.

The forgoing discussion leads to our cost function for semisupervised learning

$$\begin{aligned}&s(f) = J(f) + C\Big(\frac{1}{n_l}\sum_{i=1}^{n_l} L(y_i f(x_i)) + \\ &\frac{1}{n_u}\sum_{j=n_l+1}^{n}\big(\hat{p}(x_j)L(f(x_j)) + (1-\hat{p}(x_j))L(-f(x_j))\big)\Big).\end{aligned} \tag{1}$$

Minimization of (1) with respect to $f \in \mathcal{F}$ gives our estimated decision function $\hat{f}$ for classification.

3. Computation

3.1 Iterative scheme

As discussed in Section 2.2, the effectiveness of U depends largely on the accuracy of $\hat{p}$ for p. As argued early, p can be not well estimated through a small amount of labeled data, suggesting that additional unlabeled data must be utilized. For this purpose, the method of Wang and Shen (2006) can be used to extract information about f from both labeled and unlabeled data in absence of knowledge about p, which uses $L(|f(x)|)$. From an estimate $\hat{f}$, we may explore the relationship between f and p to yield an estimate $\hat{p}$ that is more precise than the one using labeled data alone. Wang, Shen and Liu (2006) provides a robust probability estimation method for margin based classifier by designing a sequence of weighted classifications, corresponding to a refined partition of $[0,1]$, to locate which subinterval contains $p(x)$ for any fixed x. Given this more accurate estimate $\hat{p}$, minimizing (1) with respect to f yields a more accurate estimate $\hat{f}$, and this in turn leads to better $\hat{p}$. This suggests a scheme by iterating the process of estimating p given f and that of estimating f from (1). This iterative scheme is expected to outperform minimizing (1) without iteration because loss U converges to the optimal loss $E(L(Yf(X))|X)$ as iteration continues, provided that the initial $\hat{p}$ is sufficiently accurate. This can be thought of as an analogy of Fisher scoring method: given a good

initial estimate, a more efficient estimate can result through efficient scoring. In a sense, the iterative scheme combines the advantage of the loss $L(z)$ with the optimal loss $E(L(Yf(X))|X)$. A detailed implementation of this scheme is summarized as follows.

Algorithm 1:
Step 1. (Initialization) Set $\hat{f}^{(0)}$ to be solution of the large margin semisupervised methodology, and compute $\hat{p}^{(0)}$ using $\mathrm{Sign}(\hat{f}^{(0)})$ through the probability estimation method of Wang, Shen and Liu (2006). Set an initial precision tolerance level $\epsilon > 0$.

Step 2. (Iteration) At iteration $k+1$, given $\hat{p}^{(k)}$, solve (1) for $\hat{f}^{(k+1)}$. This is achieved through either convex or difference convex programming. When L is the hinge loss with the L_2 penalty, quadratic programming is used. When L is a ψ-loss with the L_2 penalty, sequential quadratic programming is applicable, as described in Section 3.2. Then assign labels to unlabeled data with $\mathrm{Sign}(\hat{f}^{(k+1)})$ and compute $\hat{p}_t^{(k+1)}$ through Wang, Shen and Liu (2006). Define $\hat{p}^{(k+1)} = \max(\hat{p}^{(k)}, \hat{p}_t^{(k+1)})$ when $\hat{f}^{(k+1)} \geq 0$ and $\min(\hat{p}^{(k)}, \hat{p}_t^{(k+1)})$ otherwise.

Step 3. (Stopping rule) Terminate when $|s(\hat{f}^{(k+1)}) - s(\hat{f}^{(k)})| \leq \epsilon$.
The final solution $\hat{f}$ is the best solution among $\hat{f}^{(k)};\ k = 0, 1, \cdots$, which yields the estimated decision function.

Theorem 1 *(Monotonicity)* **Algorithm 1** *has a monotone property such that $s(\hat{f}^{(k)})$ is non-increasing in k. As a consequence,* **Algorithm 1** *converges to a stationary point $s(\hat{f}^{(\infty)})$ in that $s(\hat{f}^{(k)}) \geq s(\hat{f}^{(\infty)})$.*

One key aspect of the scheme is the monotone property, which is assured by the choice of $\hat{p}^{(k+1)}$ such that $(\hat{p}^{(k+1)} - \hat{p}^{(k)})\hat{f}^{(k+1)} \geq 0$. In this sense, Algorithm 1 differs from the EM algorithm in that the monotone property is guaranteed by the property likelihood with missing at random, which is in contrast to our situation that neither L is a likelihood nor label missing at random is assumed. In addition, it differs from the variant MM algorithm (Hunter and Lange, 2000) in that the MM algorithm solves optimization of upper or low brackets of the original cost function.

In Step 2 of **Algorithm 1**, given $\hat{p}^{(k)}(x)$, minimization in (1) is convex when $L(z)$ is the hinge loss or the logistic loss, but involves nonconvex minimization when $L(z)$ is ψ-loss. It then requires a nonconvex minimization technique to solve (1) for $\hat{f}^{(k+1)}$, which will be discussed in next section.

3.2 Nonconvex minimization

This section develops a nonconvex minimization technique based on difference convex (DC) programming (An and Tao, 1997) for semisupervised ψ-learning, which has also been used in Liu, Shen and Wong (2005) for supervised ψ-learning.

Key idea to DC programming is to decompose the cost function $s(f)$ in (1) with $L(z) = \psi(z)$ into a difference of two convex functions as follows:

$$s = s_1 - s_2, \tag{2}$$

where $s_1 = C\big(\frac{1}{n_l}\sum_{i=1}^{n_l}\psi_1(y_i f(x_i)) + \frac{1}{n_u}\sum_{j=n_l+1}^{n}\big(\hat{p}^{(k)}(x_j)\psi_1(f(x_j)) + (1-\hat{p}^{(k)}(x_j))\psi_1(-f(x_j))\big)\big) + J(f)$ and $s_2 = C\big(\frac{1}{n_l}\sum_{i=1}^{n_l}\psi_2(y_i f(x_i)) + \frac{1}{n_u}\sum_{j=n_l+1}^{n}\big(\hat{p}^{(k)}(x_j)\psi_2(f(x_j)) + (1-\hat{p}^{(k)}(x_j))\psi_2(-f(x_j))\big)\big)$ with $\psi_1 = 2(1-z)_+$ and $\psi_2 = 2(-z)_+$. Here ψ_1 and ψ_2 are obtained through a convex decomposition of $\psi = \psi_1 - \psi_2$ as in Figure 1.

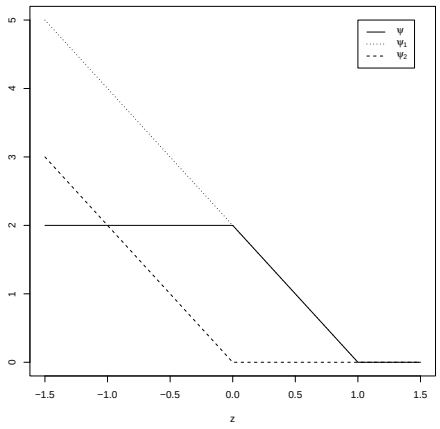
With these decompositions, we treat the nonconvex minimization in (1) with given $\hat{p}^{(k)}(x)$ by solving a sequence of quadratic programming (QP) problems as in **Algorithm 2**.

Algorithm 2: (Sequential QP)

Step 1. (Initialization) Set initial values $\hat{f}^{(k+1,0)}$ as the solution of SVM with labeled data alone, and an precision tolerance level $\epsilon > 0$.

Step 2. (Iteration) At iteration $l+1$, compute

Figure 1: Plot of ψ, ψ_1 and ψ_2 for the DC decomposition of $\psi = \psi_1 - \psi_2$. Solid, dotted and dashed lines represent ψ, ψ_1 and ψ_2, respectively.



$\hat{f}^{(k+1,l+1)}$ by solving a subproblem

$$\min_f \left\{ s_1(f) - \langle w_f, \nabla s_2(\hat{f}^{(k+1,l)}) \rangle \right\}, \quad (3)$$

where $\nabla s_2(f^{(k+1,l)})$ is a gradient vector of $s_2(f)$ at $w_{\hat{f}^{(k+1,l)}}$.

Step 3. (Stopping rule) Terminate when $|s(\hat{f}^{(k+1,l+1)}) - s(\hat{f}^{(k+1,l)})| \leq \epsilon$.
Then the estimate $\hat{f}^{(k+1)}$ is the best solution among $\hat{f}^{(k+1,l)};\ l = 0, 1, \cdots$.

In (3), gradient $\nabla s_2(f^{(k+1,l)})$ is defined as the sum of derivatives of its components with $\nabla\psi_2(z) = 0$ if $z > 0$ and $\nabla\psi_2(z) = -2$ otherwise. Using the definition of $\nabla s_2(f^{(k+1,l)})$ and convexity of $s_2(f^{(k+1,l)})$, the subproblems in (2) provide a sequence of non-increasing upper envelops to (1), which can be solved through its dual form.

The convergence property of **Algorithm 2** can be derived in similar fashion as in Theorem 3 of Liu, Shen and Wong (2005) for ψ-learning, which along with Theorem 1 of Section 3.1 yields the convergence speed of **Algorithm 1** in Theorem 2.

Theorem 2 *(Convergence rate of* **Algorithm 1***)* **Algorithm 1** *converges superlinearly in the sense that* $\lim_{k\to\infty} \|s(f^{(k+1)}) - s(f^{(\infty)})\| / \|s(f^{(k)}) - s(f^{(\infty)})\| = 0$ *provided that there does not exist an instance* $\tilde{x}$ *such that* $f^{(\infty)}(\tilde{x}) = 0$*, where* $f^{(\infty)} = (1, K(x, x_1), \cdots, K(x, x_n))w_f^{(\infty)}$.

Theorem 2 implies that the number of iterations required to achieve the precision ϵ is $o(\log(1/\epsilon))$.

Remarks: A faster convergence rate can be achieved by **Algorithm 1** due to the piecewise linearity of the objective function in (1). In fact, a similar treatment as in proof of Theorem 2 yields that the convergence rate of **Algorithm 1** is faster than any polynomial rate.

4. Numerical results

This section examines the effectiveness of the proposed methodology, as well as the effect of initial estimates, through two simulated and five benchmark examples. The performance is examined for four different initial estimates: SVM with labeled data alone (SVM), TSVM (Joachims, 1999), and the method of Wang and Shen (2006) with the hinge loss (SSVM) and the ψ-loss (SPSI), denoted as ESVM, ETSVM, ESSVM and ESPSI, respectively.

A test error, averaged over 100 independent replications, is used to measure the generalization performance. For simulation comparison, the amount of improvement of the test over the initial estimate $\hat{f}^{(0)}$ is defined as the percent of improvement in terms of the Bayesian regret,

$$\frac{(T(\text{Before}) - Bayes) - (T(\text{After}) - Bayes)}{T(\text{Before}) - Bayes}, \quad (4)$$

where $T(\text{Before})$, $T(\text{After})$ and $Bayes$ are the test errors of the initial estimate $\hat{f}^{(0)}$, the proposed method equipped with $\hat{f}^{(0)}$, and the Bayes error. The Bayes error serves as a baseline for comparison or the best performance over all methods, and is approximated by the test error of the Bayes rule over a test sample of large size, say 10^5. For benchmark comparison, the amount of improvement over the initial estimate $\hat{f}^{(0)}$ is

defined as

$$\frac{T(\text{Before}) - T(\text{After})}{T(\text{Before})}, \tag{5}$$

which underestimates the amount of improvement in absence of the Bayes rule.

Numerical analyses are performed in R2.1.1. In the linear case, $K(x, y) = \langle x, y\rangle$; in the Gaussian kernel case, $K(x, y) = \exp(-\frac{\|x-y\|^2}{\sigma^2})$, where σ is set to be the median distance between the positive and negative classes to reduce computational cost for tuning σ^2.

Some details are given below concerning the numerical examples.

Simulated examples: Examples 1 and 2 were used in Wang and Shen (2006), where 200 instances are randomly selected for training, and the remaining 800 instances are retained for testing. Among the 200 instances, 190 unlabeled instances (X_{i1}, X_{i2}) are obtained by removing labels at random, whereas the remaining 10 instances are treated as labeled data. The Bayes errors for Examples 1 and 2 are 0.162 and 0.089 respectively.

Benchmarks: Five benchmark examples include Wisconsin Breast Cancer (WBC), Pima Indians Diabetes (Pima), Heart, Mushroom and Spam email, each available in the UCI Machine Learning Repository (Blake and Merz, 1998). Instances in the WBC, Pima, Heart and Mushroom examples are randomly divided into halves with 10 labeled and 190 unlabeled instances for training, and the remaining instances for testing. Instances in the Spam email example are randomly divided into halves with 20 labeled and 380 unlabeled instances for training, and the remaining instances for testing.

In each example, the smallest test errors of ESVM, ETSVM, ESSVM and ESPSI are computed over 60 grid points for the tuning parameter C in (1) through a grid search over interval $[10^{-3}, 10^3]$ with ten equally-spaced points within each interval $(10^k, 10^{k+1}]$; $k = -3, \cdots, 2$. The results are summarized in Tables 1-2.

As suggested in Tables 1-2, the proposed method yields an improvement over any initializing method in almost all the examples except ESVM in the spam email example. That is, ESVM, ETSVM, ESSVM and ESPSI outperform their counterparts SVM, TSVM, SSVM and SPSI respectively. The amount of improvement, however, varies over examples and types of classifiers. In the linear case, the improvements of the proposed method are from 1.9% to 67.8% over the initializing methods except in the spam email example where ESVM performs slightly worse than SVM. In the nonlinear case, the improvements range from 0.0% to 23.2% over their initializing methods in the all the examples. With regard to initializing methods, SPSI seems to be preferable, leading to the best performances across all the examples.

5. Summary

This article proposes a novel iterative semisupervised learning method that that is applicable to a class of semisupervised classifiers, leading to an improvement over these methods. The method constructs an efficient loss to measure the contribution of unlabeled instances to classification. An iterative scheme DCA is derived for implementation. Our numerical analysis suggests that the proposed method compares favorably against the existing semisupervised methods.

Appendix

Proof of Lemma 1: Let $U(f(x)) = E(L(Yf(X))|X = x)$. Using the orthogonality property, we have $E(L(Yf(X)) - T(f(X)))^2 = E(L(Yf(X)) - U(f(X)))^2 + E(U(f(X)) - T(f(X)))^2$, which implies that $U(f(x))$ minimizes $E(L(Yf(X)) - T(f(X)))^2$ over any T. Furthermore, it is easy to verify that $\text{argmin}_{f\in\mathcal{F}} E(L(Yf(X))|X) = \text{argmin}_{f\in\mathcal{F}} EL(Yf(X))$. This completes the proof.

Proof of Theorem 1: For clarity, write $s(\hat{f})$ as $s(\hat{f}, \hat{p})$ in what follows. Then it suffices to

Table 1: **Linear learning.** Averaged test errors as well as the estimated standard errors (in parenthesis) of our proposed methodology with three different initial estimates: TSVM, large margin semisupervised methodology with SVM and ψ-learning, denoted as ETSVM, ESSVM, ESPSI, over 100 pairs of training and testing samples, in the simulated and benchmark examples. Here TSVM, SSVM and SPSI stand for the three initial estimates. The amount of improvement is defined in (4) or (5).

Data	Example 1	Example 2	WBC	Pima	Heart	Mushroom	Spam
SVM	.344(.0104)	.333(.0129)	.053(.0071)	.351(.0070)	.284(.0085)	.232(.0135)	.216(.0097)
ESVM	.281(.0143)	.297(.0177)	.031(.0007)	.320(.0059)	.214(.0066)	.172(.0084)	.217(.0178)
Improv.	53.8%	19.8%	41.5%	8.8%	24.6%	25.9%	-0.5%
TSVM	.249(.0121)	.222(.0128)	.077(.043)	.315(.0067)	.270(.0082)	.204(.113)	.227(.0120)
ETSVM	.190(.0074)	.147(.0131)	.029(.0009)	.309(.0063)	.211(.0062)	.153(.0054)	.179(.0101)
Improv.	67.8%	56.4%	62.3%	1.9%	21.9%	25.0%	21.1%
SSVM	.188(.0084)	.129(.0031)	.032(.0025)	.307(.0054)	.240(.0074)	.186(.0095)	.191(.0114)
ESSVM	.182(.0065)	.124(.0034)	.028(.0006)	.293(.0029)	.205(.0059)	.162(.0054)	.169(.0107)
Improv.	23.1%	12.5%	12.5%	4.6%	14.6%	12.9%	11.5%
SPSI	.184(.0084)	.128(.0084)	.029(.0022)	.291(.0032)	.232(.0067)	.184(.0095)	.189(.0107)
ESPSI	.182(.0065)	.123(.0029)	.027(.0006)	.284(.0026)	.181(.0052)	.137(.0067)	.167(.0107)
Improv.	9.1%	12.8%	6.9%	4.5%	22.0%	25.5%	10.1%

Table 2: **Nonlinear learning with Gaussian kernel.** Averaged test errors as well as the estimated standard errors (in parenthesis) of our methodology with three different initial estimates TSVM, large margin semisupervised methodology with SVM and ψ-learning respectively, over 100 pairs of training and testing samples, in the simulated and benchmark examples. The amount of improvement is defined in (4) or (5).

Data	Example 1	Example 2	WBC	Pima	Heart	Mushroom	Spam
SVM	.385(.0099)	.347(.0119)	.047(.0038)	.342(.0044)	.331(.0094)	.217(.0135)	.226(.0108)
ESVM	.368(.0077)	.322(.0109)	.039(.0067)	.335(.0035)	.308(.0107)	.187(.0118)	.212(.0104)
Improv.	7.6%	9.7%	17.0%	2.0%	6.9%	13.8%	6.2%
TSVM	.267(.0132)	.258(.0157)	.037(.0015)	.353(.0073)	.331(.0087)	.217(.0117)	.275(.0158)
ETSVM	.236(.0090)	.235(.0084)	.030(.0005)	.323(.0028)	.303(.0094)	.201(.0093)	.198(.0106)
Improv.	11.6%	13.6%	18.9%	8.5%	8.5%	7.4%	28.0%
SSVM	.201(.0072)	.175(.0092)	.030(.0005)	.304(.0044)	.226(.0063)	.173(.0126)	.189(.0120)
ESSVM	.201(.0072)	.170(.0083)	.030(.0005)	.304(.0042)	.223(.0054)	.147(.0105)	.170(.0103)
Improv.	0.0%	5.8%	0.0%	0.0%	1.3%	15.0%	10.1%
SPSI	.200(.0069)	.175(.0092)	.030(.0005)	.295(.0037)	.215(.0057)	.164(.0123)	.189(.0112)
ESPSI	.198(.0072)	.169(.0082)	.030(.0005)	.294(.0033)	.215(.0054)	.126(.0083)	.169(.0091)
Improv.	1.0%	7.0%	0.0%	0.3%	0.0%	23.2%	10.6%

prove that $s(\hat{f}^{(k)}, \hat{p}^{(k)}) \geq s(\hat{f}^{(k+1)}, \hat{p}^{(k+1)})$. Note that $s(\hat{f}^{(k)}, \hat{p}^{(k)}) \geq s(\hat{f}^{(k+1)}, \hat{p}^{(k)})$ since $\hat{f}^{(k+1)}$ minimizes $s(f, \hat{p}^{(k)})$. Furthermore, $s(\hat{f}^{(k+1)}, \hat{p}^{(k)}) - s(\hat{f}^{(k+1)}, \hat{p}^{(k+1)}) = \sum_{j=n_l+1}^{n} (\hat{p}^{(k)} - \hat{p}^{(k+1)})(L(\hat{f}^{(k+1)}(x_j)) - L(-\hat{f}^{(k+1)}(x_j)))$, which is always nonnegative by the definition of $\hat{p}^{(k+1)}$.

Proof of Theorem 2: It follows from Theorem 1 that there exists a stationary point $s(f^{(\infty)})$ such that $\lim_{k\to\infty} s(f^{(k)}) = s(f^{(\infty)})$. By assumption, there does not exist an instance $\tilde{x}$ such that $f^{(\infty)}(\tilde{x}) = 0$. This property holds in a small neighborhood of $f^{(\infty)}$ by continuity. Then for any f in the neighborhood, the corresponding classifier $\mathrm{Sign}(f)$ is identical to $\mathrm{Sign}(f^{(\infty)})$ for all labeled and unlabeled instances, implying that $\hat{p} = \hat{p}^{(\infty)}$. Therefore, the convergence speed of **Algorithm 1** is equivalent to that of **Algorithm 2** up to a constant. The desired result then follows through an argument similar to that in Liu, Shen and Wong (2005).

References

[1] AMINI, M., AND GALLINARI, P. (2003). Semi-supervised learning with an explicit label-error model for misclassified data. *IJCAI2003*.

[2] AN, L., AND TAO, P. (1997). Solving a class of linearly constrained indefinite quadratic problems by D.C. algorithms. *J. Global Optimization*, **11**, 253-285.

[3] ANDO, R., AND ZHANG, T. (2004). A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR*, **6**, 1817-1853.

[4] ASTORINO, A., AND FUDULI, A. (2005). Nonsmooth optimization techniques for semisupervised classification. Preprint.

[5] BALCAN, M., BLUM, A., CHOI, P., LAFFERTY, J., PANTANO, B., RWEBANGIRA, M., AND ZHU, X. (2005). Person identification in webcam images: an application of semi-supervised learning. *ICML Workshop on Learning with Partially Classified Training Data*.

[6] BALUJA, S. (1998). Probabilistic modeling for face orientation discrimination: Learning from labeled and unlabeled data. *NIPS1998*.

[7] BLAKE, C.L., AND MERZ, C.J. (1998). UCI repository of machine learning databases. http://www.ics.ci.edu/∼mlearn/MLRepository.html. University of California, Irvine, Department of Information and Computer Science.

[8] BLUM, A. AND MITCHELL, T. (1998). Combining labeled and unlabeled data with co-training. *Proc. of the 11th Annual Conf. on Computational Learning Theory (COLT98)*, 92-100.

[9] BOSER, B., GUYON, I., AND VAPNIK, V. N. (1992). A training algorithm for optimal margin classifiers. *Fifth Annual Conference on Computational Learning Theory*, Pittsburgh ACM, 142-152.

[10] BROWN, M., GRUNDY, W., LIN, D., CRISTIANINI, N., SUGNET, C., FUREY, T., ARES, M., AND HAUSSLER, D. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci.*, **92**, 262-67.

[11] CHAPELLE, O., AND ZIEN, A. (2005). Semi-supervised classification by low density separation. *AISTAT2005*, 57-64.

[12] COZMAN, F.G., COHEN, I., AND CIRELO, M.C. (2003) Semi-Supervised Learning of Mixture Models and Bayesian Networks. *ICML2003*.

[13] COLLINS, M., AND SINGER. Y. (1999). Unsupervised models for named entity classification. In *Empirical Methods in Natural Language Processing and Very Large Corpora*.

[14] CORTES, C., AND VAPNIK, V. N. (1995). Support-vector networks. *Machine Learning*, **20**, 273-97.

[15] DARA, R., KREMER, S., AND STACEY, D. (2002) Clustering unlabeled data with SOMs improves classification of labeled real-world data. *Proc. of the IEEE Int. Joint Conf. on Neural Networks (IJCNN)*.

[16] DENNIS. J.E., AND MORÉ, J.J. (1974). A characterization of superlinear convergence and its application to quasi-Newton methods. *Math. Comput.*, **28**, 549-60.

[17] FUNG, G., AND MANGASARIAN, O.L. (2002). Data selection for support vector machine classifiers. Manuscript.

[18] FALK, J. E., AND HOOEMAN, K. L. (1976). A successive underestimation method for concave minimization problems. *Mathematics of Operations research*, **1**. 251-59.

[19] GU, C. (2000). Multidimension smoothing with splines. *Smoothing and Regression: Approaches, Computation and Application*, edited by M.G. Schimek.

[20] HORST, R., AND TUY, H. (1989). *Global optimization- deterministic approaches*. Springer-Verlag.

[21] HUNTER, D., AND LANGE, K. (2000). Quantile regression via an MM algorithm, *J. of Comp. and Graphic. Statist.*, **9**, 60-77.

[22] JOACHIMS, T. (1999). Transductive inference for text classification using support vector machines. *ICML1999*.

[23] LIU, S., SHEN, X., AND WONG, W. (2005). Computational development of ψ-learning. *SIAM2005*, P1-12.

[24] NIGAM, K., MCCALLUM, A., THRUN, S., AND MITCHELL T. (1998). Text classification from labeled and unlabeled documents using EM. *AAAI1998*.

[25] PLATT, J. C. (1999). *Advances in large margin classifiers.* chapter Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, MIT press.

[26] RÄTSCH. G., ONODA, T., AND MÜLLER, K.R (2001). Soft Margins for AdaBoost. *Machine Learning*. **42**,287-320.

[27] SHEN, X., TSENG, G., ZHANG, X., AND WONG, W. H. (2003). On ψ-learning. *J. Ameri. Statist. Assoc.*. **98**, 724-34.

[28] WAHBA, G. (1990). *Spline models for observational data*. CBMS-NSF Regional Conference Series, Philadelphia.

[29] WANG, J., AND SHEN, X. (2007). Large margin semi-supervised learning. *Journal of Machine Learning Research*, tentatively accepted.

[30] WANG, J., SHEN, X., AND LIU, Y.F. (2007). Probability estimation for large margin classifiers. Submitted

[31] WANG, J., SHEN, X., AND PAN, W. (2007). On transductive support vector machines. *Proceedings of the Joint Summer Research Conference on Machine and Statistical Learning: Prediction and Discovery* , accepted.

[32] ZHANG, T., AND OLES, F. (2000). A probability analysis on the value of unlabeled data for classification problems. *ICML2000*.

[33] ZHU, J. AND HASTIE, T. (2005). Kernel logistic regression and the import vector machines. *J. Comput. Graph. Statist.*, **14**, 185-205.

[34] ZHU, X., LAFFERTY, J., AND GHAHRAMANI, Z. (2003). Combining active learning and semi-supervised learning using Gaussian fields and Harmonic functions. *ICML2003*.