A Gaussian Latent Variable Model for Large Margin Classification of Labeled and Unlabeled Data

Do-kyum Kim, Matthew Der and Lawrence K. Saul
Department of Computer Science and Engineering, University of California, San Diego
{dok027, mfder, saul}@cs.ucsd.edu

Abstract

We investigate a Gaussian latent variable model for semi-supervised learning of linear large margin classifiers. The model's latent variables encode the signed distance of examples to the separating hyperplane, and we constrain these variables, for both labeled and unlabeled examples, to ensure that the classes are separated by a large margin. Our approach is based on similar intuitions as semi-supervised support vector machines (S^3VMs), but these intuitions are formalized in a probabilistic framework. Within this framework we are able to derive an especially simple Expectation-Maximization (EM) algorithm for learning. The algorithm alternates between applying Bayes rule to "fill in" the latent variables (the E-step) and performing an unconstrained least-squares regression to update the weight vector (the M-step). For the best results it is necessary to constrain the unlabeled data to have a similar ratio of positive to negative examples as the labeled data. Within our model this constraint renders exact inference intractable, but we show that a Lyapunov central limit theorem (for sums of independent, but non-identical random variables) provides an excellent approximation to the true posterior distribution. We perform experiments on large-scale text classification and find that our model significantly outperforms existing implementations of S^3VMs.

1 Introduction

The goal of semi-supervised learning is to build predictive models from small collections of labeled examples but large collections of unlabeled ones. Semi-supervised learning offers the most promise in domains where collecting data is cheap but labeling it is expensive. Important applications include text and web page classification (Nigam et al., 2000), protein classification (Weston et al., 2005), surveillance (Balcan et al., 2005), real-time traffic classification (Erman et al., 2007), gene function prediction (Wang et al., 2009), and many others.

Among the most popular models for semi-supervised learning are transductive support vector machines (TSVMs) (Joachims, 1999a), also known as semi-supervised support vector machines (S^3VMs) (Bennett and Demiriz, 1998). These models extend the original framework of support vector machines (SVMs) (Cortes and Vapnik, 1995) to handle partially labeled data. The decision boundaries in these models attempt to satisfy two criteria: first, to separate the positively and negatively labeled examples by a large margin; second, to cross through regions of low density in the unlabeled examples. The models work well when the data satisfies the so-called *cluster assumption*—that is, when points in the same cluster are likely to share the same label (Chapelle and Zien, 2005). The optimizations for S^3VMs, however, involve inherently non-convex loss functions; as a result, they are notoriously more difficult than those for ordinary SVMs. Researchers have explored a vast arsenal of techniques for training S^3VMs, making use (for example) of combinatorial search (Joachims, 1999a; Sindhwani and Keerthi, 2006), convex-concave procedures (Collobert et al., 2006) and annealing (Sindhwani and Keerthi, 2006; Ogawa et al., 2013); see Chapelle et al. (2008) for a comprehensive review.

In this paper we propose a latent variable model for the semi-supervised learning of linear large margin classifiers. Our model shares the same intuitions as S^3VMs but encodes them in a fully probabilistic framework. Within this framework, we are able to derive a simple Expectation-Maximization (EM) algorithm (Dempster et al., 1977) that alternately computes the posterior means of the model's latent variables (the E-step) and updates the model's weight vector by performing an unconstrained least-squares re-

Appearing in Proceedings of the 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33.

gression (the M-step). Our approach has certain especially attractive properties. It scales extremely well to sparse, high-dimensional data sets because we can leverage the highly specialized solvers currently available for sparse least-squares problems (Barrett et al., 1994). It also handles unlabeled examples as transparently as labeled examples; these examples differ only in the formula used to compute the posterior means of their latent variables.

Our approach borrows key insights from null category noise models (NCNMs) (Lawrence and Jordan, 2005) of semi-supervised learning. As in NCNMs, we use latent variables to encode the signed distance of examples to the model's decision boundary, and we use magnitude constraints on these variables to enforce large margin criteria. However, our approach differs from NCNMs in two important ways. First, we focus on large-scale linear classification as opposed to classification via Gaussian processes. While our models cannot parameterize nonlinear decision boundaries, they scale much better to large data sets. Second, beyond previous work on NCNMs, we show how to incorporate a class-balancing constraint into our latent variable model. This is a critical constraint for avoiding uninformative solutions that assign all unlabeled examples to the same class. Within our model the class-balancing constraint renders exact inference intractable, but as a further technical contribution, we show that a Lyapunov central limit theorem (Billingsley, 1995)—for sums of independent, but non-identical random variables—provides an excellent working approximation to the true posterior distribution.

We evaluate our latent variable models on six large problems in text classification and compare them to three leading implementations of S^{3}VMs. Here we find significant gains in speed from the specialized handling of sparse least-squares problems, as well as significant gains in accuracy from the use of unlabeled examples. A seeming advantage of our probabilistic framework is the more principled inference of target classes and "margins" for unlabeled examples. This advantage translates into consistently lower error rates than the other implementations of S^{3}VMs; we see these improvements across all the data sets in our study and over a wide range of experimental settings.

2 Model for labeled data

We begin by describing our model in the case of *fully labeled* data. Here we assume that the data consists of n examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \Re^d$ and $y_i \in \{-1, +1\}$. The model can be viewed as a variant of ℓ_2-regularized probit regression (McCullagh and Nelder, 1989) with large margin constraints; Fig. 1 shows its representation as a Bayesian network. In typical fashion, the model parameterizes a linear decision boundary $y = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$ by a weight vector $\mathbf{w} \in \Re^d$ and bias $b \in \Re$, and the magnitude $\|\mathbf{w}\|$ is regularized by a hyperparameter $\lambda \geq 0$. We use $\Theta = (\mathbf{w}, b)$ to denote the model's joint parameters.

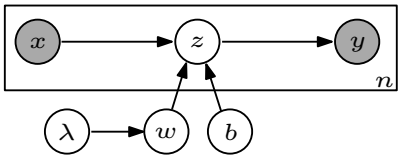

Figure 1: Bayesian network for large margin variant of ℓ_2-regularized probit regression. See text for details.

2.1 Notation and preliminaries

The model's observed variables are the inputs $\mathbf{x} \in \Re^d$ and the labels $y \in \{-1, 0, +1\}$; these are represented by the shaded nodes in Fig. 1. A label $y = \pm 1$ indicates that an example is positively or negatively classified *by a large margin*; a label $y = 0$ indicates that the example lies *close (i.e., within one unit of distance) to the decision boundary*. Although we never observe the value $y = 0$ in the labeled data, the potential for this prediction still plays an important role in the model's development (Lawrence and Jordan, 2005).

The model's observed variables $\mathbf{x}$ and y are connected by the latent variable z. This variable z follows the simple Gaussian distribution:

$$P(z|\mathbf{x}, \Theta) \;=\; \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(z - \mathbf{w}\cdot\mathbf{x} - b)^2}. \tag{1}$$

The Gaussian latent variable z in turn *determines* the label $y \in \{-1, 0, +1\}$; note that in this dependence there is no uncertainty. In particular we have:

$$y \;=\; \begin{cases} \text{sign}(z) & \text{if } |z| \geq 1 \\ 0 & \text{if } |z| < 1. \end{cases} \tag{2}$$

The dependence in eq. (2) incorporates a key insight of large margin classification—namely, that correctly labeled examples should lie at least one unit of distance away from the decision boundary.

Together eqs. (1–2) reveal the model's relation to probit regression: we obtain the standard model of probit regression by restricting the domain of y to $\{-1, +1\}$ and replacing eq. (2) by the simpler dependence $y = \text{sign}(z)$ for all z. As in ℓ_2-regularized probit regression, we also adopt a symmetric Gaussian prior $\mathbf{w} \sim \mathcal{N}(0, \lambda^{-1}\mathbf{I}_d)$ on the weight vector $\mathbf{w}$, where $\mathbf{I}_d$ is the $d \times d$ identity matrix.

2.2 Inference

Inference in this model is entirely tractable: no more is required than integrating out the Gaussian latent variable z. As shorthand, we use

$$\xi \;=\; \mathbf{w} \cdot \mathbf{x} + b \tag{3}$$

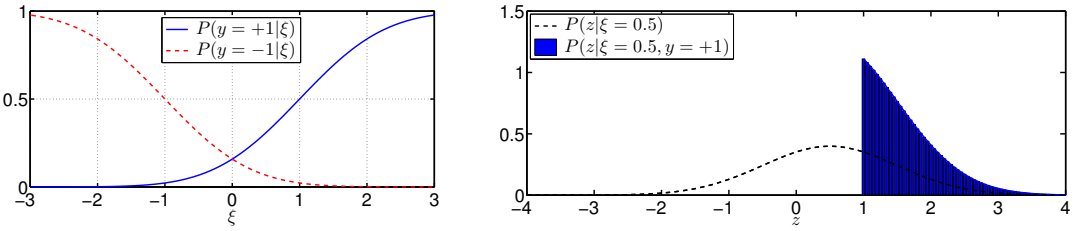

Figure 2: *Left*: dependence of the marginal probabilities $P(y|\mathbf{x},\Theta)$ in eq. (4) on the linear score of the classifier, $\xi = \mathbf{w}\cdot\mathbf{x}+b$. *Right*: comparison of the Gaussian prior $P(z|\mathbf{x},\Theta)$ and the truncated Gaussian posterior $P(z|\mathbf{x}, y=+1, \Theta)$ for a positive example with $\xi = 0.5$.

to denote the (signed) linear score of the classifier. Then for *non-zero* values of y, we obtain the marginal probabilities:

$$P(y=\pm 1|\mathbf{x},\Theta) \;=\; \frac{1}{2}\mathrm{erfc}\left(\frac{1-y\xi}{\sqrt{2}}\right), \qquad (4)$$

where erfc denotes the complementary error function. Likewise, by normalization, it follows that:

$$P(y=0|\mathbf{x},\Theta) = 1 - P(y=+1|\mathbf{x},\Theta) - P(y=-1|\mathbf{x},\Theta).$$

Note that these probabilities depend on $\mathbf{x}$, $\mathbf{w}$, and b only through the classifier score ξ. The left panel of Fig. 2 plots the probabilities $P(y=\pm 1|\xi)$ as a function of this score; note that for $|\xi| < 1$, neither label $y = \pm 1$ has probability greater than 0.5.

Also of interest is the posterior distribution $P(z|\mathbf{x}, y, \Theta)$, which we obtain in a straightforward fashion from Bayes rule and conditional independence:

$$P(z|\mathbf{x}, y, \Theta) \;=\; \frac{P(y|z)P(z|\mathbf{x},\Theta)}{P(y|\mathbf{x},\Theta)}\,. \qquad (5)$$

Recall that the label y is *determined* by the latent variable z; in particular, for $y = \pm 1$, the first term in the numerator $P(y|z)$ equals unity if $yz \geq 1$ and vanishes otherwise. It follows that the posterior distribution in eq. (5) takes the form of a *truncated* Gaussian. The right panel of Fig. 2 illustrates the truncating effect of conditioning the latent variable z on the label $y=+1$.

The statistics of the posterior distribution in eq. (5) are required for the E-step of this model's EM algorithm. Of special importance are the posterior means, $\mathrm{E}[z|\mathbf{x}, y, \Theta] = \int dz\, z\, P(z|\mathbf{x}, y, \Theta)$. For labeled examples, this calculation gives

$$\mathrm{E}[z|\mathbf{x}, y=\pm 1, \Theta] = \xi + y\sqrt{\frac{2}{\pi}}\left[\frac{\exp(-\frac{1}{2}(1-y\xi)^2)}{\mathrm{erfc}(\frac{1}{\sqrt{2}}(1-y\xi))}\right], \qquad (6)$$

where the last term on the right hand side gives the *correction* (either positive or negative) to the posterior mean from the prior mean $\mathrm{E}[z|\mathbf{x},\Theta] = \xi$. The right panel of Fig. 2 shows that this posterior mean may shift considerably from the prior mean due to the model's large margin constraints.

2.3 Supervised learning

The parameters $\mathbf{w}$ and b of this model can be learned by an especially simple EM algorithm. (As we shall see in the next section, the same algorithm extends transparently to the problem of semi-supervised learning.) The goal of learning from labeled data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is to maximize the regularized log-likelihood of the model shown in Fig. 1. This is given by:

$$\mathcal{L}_{\mathrm{labeled}}(\Theta) \;=\; \sum_{i=1}^{n} \log P(y_i|\mathbf{x}_i,\Theta) - \frac{\lambda}{2}\|\mathbf{w}\|^2. \qquad (7)$$

It can be verified for $\lambda \geq 0$ that the log-likelihood in eq. (7) is a strictly concave function of the parameters $\mathbf{w}$ and b. Though eq. (7) cannot be maximized in closed form, we can derive iterative EM updates that converge monotonically in the log-likelihood. As shorthand, let

$$\bar{z}_i \;=\; \mathrm{E}[z_i|\mathbf{x}_i, y_i, \Theta] \qquad (8)$$

denote the posterior means computed from eq. (6). Then at each iteration of EM, the model parameters are updated by solving the unconstrained least-squares problem:

$$\min_{\mathbf{w},b}\left\{\sum_{i=1}^{n}(\bar{z}_i - \mathbf{w}\cdot\mathbf{x}_i - b)^2 \;+\; \lambda\|\mathbf{w}\|^2\right\}. \qquad (9)$$

In a nutshell, the algorithm simply alternates between the *E-step* of computing the posterior means in eq. (6) and the *M-step* of solving the least-squares problem in eq. (9). Henceforth we refer to this **EM** algorithm for **B**inary **L**arg**e** **M**argin classification as **EMBLEM**.

Two properties of EMBLEM make it highly scalable. First, we note that the quadratic terms in eq. (9) do not change from iteration to iteration. Thus, for small d, it is possible to perform one $O(d^3)$ matrix inverse at the beginning of EMBLEM after which each solution to eq. (9) involves only a single $O(d^2)$ matrix-vector multiplication. Second, we note that in the opposite regime of large d, there exist highly efficient solvers for *sparse* least-squares problems (Barrett et al., 1994). We can leverage these solvers

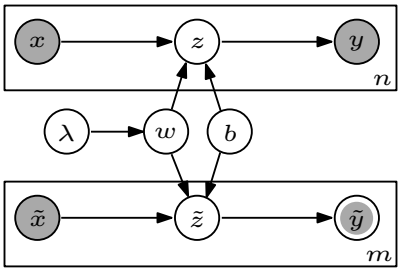

Figure 3: Bayesian network for semi-supervised classification with n labeled and m unlabeled examples.

for data that is extremely high-dimensional but contains many zero-valued features (e.g., word-document counts). This latter property will prove especially important for semi-supervised applications of EMBLEM to text data.

Finally we note a clever trick to accelerate EMBLEM without sacrificing its guarantee of monotonic convergence. Let $\mathbf{w}'$ denote the updated value of the weight vector from the least-squares problem in eq. (9). We can take a larger step in the direction of this value using the method of successive overrelaxation (Yu, 2011):

$$\mathbf{w} \leftarrow (1+\eta)\mathbf{w}' - \eta\mathbf{w}, \tag{10}$$

where $\eta \in [0, 1]$. It is straightforward to show that for latent variable models with quadratic auxiliary functions—for example, eq. (9)—these overrelaxed updates also converge monotonically in the log-likelihood. We achieved our fastest results for EMBLEM by setting $\eta = 1$, running until near-convergence, then setting $\eta = 0$ (the standard EM update) and running until convergence.

3 Extension to unlabeled data

The algorithm in the previous section extends in a straightforward way to the problem of semi-supervised learning. In this case, we have m unlabeled examples $\{\tilde{\mathbf{x}}_j\}_{j=1}^m$ in addition to the n labeled ones $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$. In this section we show how to treat the unknown labels $\{\tilde{y}_j\}_{j=1}^m$ as missing data which can be "filled in" and modeled by an EM algorithm. Naturally this use of unlabeled examples requires certain assumptions about the data. The assumptions we make here are similar to those in TSVMs; however, the algorithm that results is quite different. In particular, missing labels are inferred from Bayes rule (as opposed to, say, an integer program), and the weight vector is updated by a least-squares regression (as opposed to, say, a quadratic program). These features of the algorithm make it highly scalable.

3.1 Large margin constraints

Our first assumption is that the unlabeled examples, like the labeled ones, should also lie at least one unit of distance away from the separating hyperplane. This assumption leads to the "twinned" Bayesian network shown in Fig. 3. In the top plate of the model, for the n labeled examples, the node y is shaded to represent an instantiated label of $+1$ for positive examples and -1 for negative examples. In the bottom plate of the model, for the m unlabeled examples, the node $\tilde{y}$ is *partially* shaded to indicate that *its label equals either +1 or -1 but is never equal to zero*. Both plates share the same parameters $\mathbf{w}$ and b for large margin variants of ℓ_2–regularized probit regression. Note from eq. (2) that the excluded label $\tilde{y} \neq 0$ implies $|\tilde{z}| \geq 1$, thus encoding the assumption that each unlabeled example lies at least one unit of distance from the decision boundary.

Without labels, we cannot compute the probability that an unlabeled example $\tilde{\mathbf{x}}_j$ is *correctly* classified. However we can compute the marginal probability

$$P(\tilde{y}\neq 0|\tilde{\mathbf{x}}, \Theta) = P(\tilde{y}=+1|\tilde{\mathbf{x}}, \Theta) + P(\tilde{y}=-1|\tilde{\mathbf{x}}, \Theta) \tag{11}$$

that one way or another the example is classified by a large margin $|\tilde{z}| \geq 1$. Fig. 4 plots the log of this probability, whose shape recalls the loss function for unlabeled examples in TSVMs (Chapelle et al., 2008). From Bayes rule in eq. (5), it is similarly straightforward to compute the posterior distribution $P(\tilde{z}|\tilde{\mathbf{x}}, \tilde{y}\neq 0, \Theta)$. In this case the posterior takes the form of a doubly truncated Gaussian, as illustrated in the right panel of Fig. 4. As shorthand, let

$$\rho_{\pm}(\tilde{\mathbf{x}}, \Theta) = \frac{P(\tilde{y}=\pm 1|\tilde{\mathbf{x}}, \Theta)}{P(\tilde{y}\neq 0|\tilde{\mathbf{x}}, \Theta)} \tag{12}$$

denote the posterior probability that the label $\tilde{y}$ of an unlabeled input $\tilde{\mathbf{x}}$ is positive or negative given that the example is classified by a large margin. Then the posterior mean of $\tilde{z}$ for an unlabeled example is equal to the weighted sum:

$$\begin{aligned}\mathrm{E}[\tilde{z}|\tilde{\mathbf{x}}, \tilde{y}\neq 0, \Theta] = {} & \rho_+(\tilde{\mathbf{x}}, \Theta)\,\mathrm{E}[\tilde{z}|\tilde{\mathbf{x}}, \tilde{y}=+1, \Theta] + \\ & \rho_-(\tilde{\mathbf{x}}, \Theta)\,\mathrm{E}[\tilde{z}|\tilde{\mathbf{x}}, \tilde{y}=-1, \Theta],\end{aligned} \tag{13}$$

where the expected values on the right hand side are given by eq. (6). The calculation of this posterior mean is the only additional form of inference required for semi-supervised learning.

Finally we present the model's EM algorithm for parameter estimation. The model in Fig. 3 is learned by maximizing the regularized log-likelihood of both the labeled *and* unlabeled examples:

$$\mathcal{L}_{\mathrm{ss}}(\Theta) = \mathcal{L}_{\mathrm{labeled}}(\Theta) + \sum_{j=1}^{m} \log P(\tilde{y}_j \neq 0|\tilde{\mathbf{x}}_j, \Theta), \tag{14}$$

where the first term on the right hand side is the regularized log-likelihood of labeled examples from eq. (7). This *semi-supervised* objective function, unlike the log-likelihood in

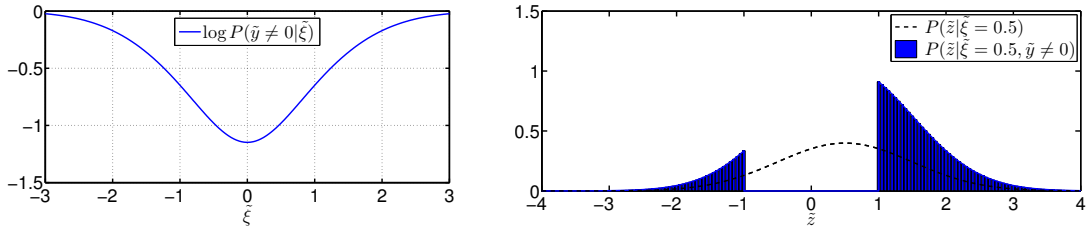

Figure 4: *Left*: dependence of the marginal log-probability $\log P(\tilde{y} \neq 0|\tilde{\mathbf{x}}, \Theta)$ on the linear score of the classifier, $\tilde{\xi} = \mathbf{w}\cdot\tilde{\mathbf{x}} + b$. *Right*: comparison of the Gaussian prior $P(\tilde{z}|\tilde{\mathbf{x}}, \Theta)$ and the truncated Gaussian posterior $P(\tilde{z}|\tilde{\mathbf{x}}, \tilde{y} \neq 0, \Theta)$ for an unlabeled example with $\tilde{\xi} = 0.5$.

eq. (7), is not concave in the parameters $\mathbf{w}$ and b. However, the EM algorithm for maximizing eq. (14) takes the same form as the one in the previous section. In particular, analogous to the shorthand in eq. (8), we use

$$\hat{z}_j \;=\; \mathrm{E}[\tilde{z}_j|\tilde{\mathbf{x}}_j, \tilde{y}_j \neq 0, \Theta] \tag{15}$$

to denote posterior means of the model's latent variables for *unlabeled* examples; these are computed by eq. (13). Then each iteration of the EM algorithm updates $\mathbf{w}$ and b by solving the least-squares problem:

$$\min_{\mathbf{w},b}\left\{\sum_{i=1}^{n}(\bar{z}_i - \mathbf{w}\cdot\mathbf{x}_i - b)^2 + \sum_{j=1}^{m}(\hat{z}_j - \mathbf{w}\cdot\tilde{\mathbf{x}}_j - b)^2 + \lambda\|\mathbf{w}\|^2\right\}. \tag{16}$$

Once again the algorithm simply alternates between an *E-step* of computing posterior means and an *M-step* of performing a least-squares regression. The update in eq. (16) has the same appealing properties as the update in eq. (9); convergence (in this case, to a *local* maximum) can again be accelerated by successive overrelaxation. Henceforth we refer to this semi-supervised version of the algorithm as $\mathbf{EMBLEM_{ss}}$.

3.2 Class-balancing constraint

The large margin constraints in the previous section enforce that unlabeled examples lie at least one unit of distance from the separating hyperplane. With enough labeled examples, these constraints may suffice to learn an improved model from the unlabeled ones. When there are very few labeled examples, however, many studies have shown that additional constraints are needed to avoid uninteresting models in which all the unlabeled examples are assigned to the same class (Joachims, 1999a; Chapelle and Zien, 2005; Chapelle et al., 2006; Collobert et al., 2006; Chapelle et al., 2008).

The Bayesian network in Fig. 5, with one extra node $\tilde{\mu}$, incorporates a class-balancing constraint into the model of the previous section. The extra node has an especially simple dependence on its m parents (which encode the class

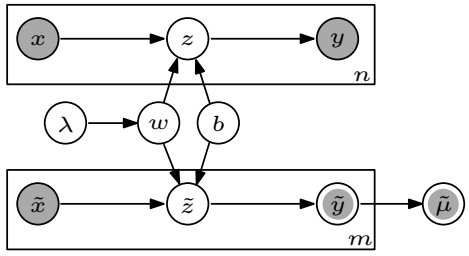

Figure 5: Bayesian network for semi-supervised learning with a class-balancing constraint.

labels of the unlabeled examples); in particular, it deterministically computes their mean

$$\tilde{\mu} \;=\; \frac{1}{m}\sum_{j=1}^{m}\tilde{y}_j. \tag{17}$$

We assume that a target range $\tilde{\mu} \in [\tilde{\mu}_{\min}, \tilde{\mu}_{\max}]$ is known for the class balance of unlabeled examples, and once again, we use a *partially shaded* node to indicate that the variable $\tilde{\mu}$ is only partially observed—i.e., not fully specified, but instead restricted to a subset of its complete domain of possible values. The interval $[\tilde{\mu}_{\min}, \tilde{\mu}_{\max}]$ may either be available from prior knowledge, or it can be estimated from the statistics of the labeled examples.

Our use of an interval constraint differs crucially from existing implementations of S^{3}VMs, which enforce an *equality* constraint on the class balance of unlabeled examples (Joachims, 1999a; Collobert et al., 2006; Sindhwani and Keerthi, 2006). In practice, the exact ratio of positive to negative examples is never known for unlabeled data, and an *interval* constraint gives the model more flexibility to fit the data. The experimental results in Section 4 suggest strongly that the interval constraint leads to better models than the equality constraint of previous implementations.

The model in Fig. 5 inherits much of its basic structure

from the previous model in Fig. 3. The goal of learning in this model is to maximize the regularized log-likelihood

$$\mathcal{L}_{\mathrm{ss}}^{\mathrm{bal}}(\Theta) \;=\; \mathcal{L}_{\mathrm{ss}}(\Theta) + \mathcal{L}_{\mathrm{bal}}(\Theta), \tag{18}$$

where the first term is given by eq. (14) and the second term is given by:

$$\mathcal{L}_{\mathrm{bal}}(\Theta) = \log P\big(\tilde{\mu} \in [\tilde{\mu}_{\min}, \tilde{\mu}_{\max}] \big| \{\tilde{\mathbf{x}}_j, \tilde{y}_j \neq 0\}_{j=1}^m, \Theta\big). \tag{19}$$

Note that the marginal probability in eq. (19) does not have a simple closed-form expression. However, for large numbers of unlabeled examples—the typical regime for semi-supervised learning—we obtain an excellent approximation from the central limit theorem.

Our use of the central limit theorem is motivated directly by eq. (17). In particular, for large m we expect the node $\tilde{\mu}$, which simply computes the mean of the m independent random variables $\tilde{y}_j$, to be approximately Gaussian distributed. The discrete labels $\tilde{y}_j$ are not identically distributed, but still a Lyapunov central limit theorem (Billingsley, 1995) applies to their sum. Assuming that the sum is Gaussian, it is possible to compute the log-likelihood in eq. (19) and to derive an EM algorithm for learning. To begin, we compute the conditional mean and variance of $\tilde{y}_j$ under the constraint $\tilde{y}_j \neq 0$. We denote these by:

$$\tilde{\gamma}_j = \mathrm{E}[\tilde{y}_j | \tilde{\mathbf{x}}_j, \tilde{y}_j \neq 0, \Theta] \quad = \rho_+(\tilde{\mathbf{x}}_j, \Theta) - \rho_-(\tilde{\mathbf{x}}_j, \Theta), \tag{20}$$

$$\tilde{\sigma}_j^2 = \mathrm{Var}[\tilde{y}_j | \tilde{\mathbf{x}}_j, \tilde{y}_j \neq 0, \Theta] = 1 - \tilde{\gamma}_j^2 \,. \tag{21}$$

For large m we expect the node $\tilde{\mu}$ to be approximately Gaussian distributed with mean $\tilde{\mu}_* = \frac{1}{m}\sum_j \tilde{\gamma}_j$ and variance $\tilde{\sigma}_*^2 = \frac{1}{m^2}\sum_j \tilde{\sigma}_j^2$. As shorthand, let

$$u \;=\; \frac{\tilde{\mu}_{\max} - \tilde{\mu}_*}{\tilde{\sigma}_* \sqrt{2}}, \quad \ell \;=\; \frac{\tilde{\mu}_{\min} - \tilde{\mu}_*}{\tilde{\sigma}_* \sqrt{2}}, \tag{22}$$

denote the scaled deviations of $\tilde{\mu}_{\max}$ and $\tilde{\mu}_{\min}$ from $\tilde{\mu}_*$. Then the Gaussian approximation for the posterior distribution $P(\tilde{\mu} \mid \{\tilde{\mathbf{x}}_j, \tilde{y}_j \neq 0\}_{j=1}^m, \Theta)$ gives the simple result:

$$\mathcal{L}_{\mathrm{bal}}(\Theta) \;\approx\; \log \frac{1}{2}[\mathrm{erf}(u) - \mathrm{erf}(\ell)]. \tag{23}$$

We expect this approximation to be highly accurate for large numbers of unlabeled examples, m.

In practice, the marginal probability in eq. (19) may be extremely small, and in these cases, the calculation of the log-likelihood $\mathcal{L}_{\mathrm{bal}}(\Theta)$ from eq. (23) is numerically unstable. (The instability occurs when the error functions are nearly equal, with a difference that is less than machine precision.) A stable computation may be obtained from asymptotic expansions of the complementary error function. In particular, we have:

$$\mathcal{L}_{\mathrm{bal}}(\Theta) \approx \begin{cases} -\ell^2 - \frac{1}{2}\log(4\pi\ell^2) & \text{if} \;\; 0 \ll \ell < u, \\ -u^2 - \frac{1}{2}\log(4\pi u^2) & \text{if} \;\; \ell < u \ll 0, \end{cases} \tag{24}$$

in the regimes where both error functions in eq. (23) evaluate to the same answer (either plus or minus unity) within machine precision.

Finally we consider the EM algorithm for parameter estimation in this model. For the E-step of the model we must compute the posterior means $\hat{z}_j = \mathrm{E}[\tilde{z}_j | \{\tilde{\mathbf{x}}_k, \tilde{y}_k \neq 0\}_{k=1}^m, \tilde{\mu} \in [\tilde{\mu}_{\min}, \tilde{\mu}_{\max}], \Theta]$ under *both* the large margin and class-balancing constraints. It can be shown that these posterior means are given by:

$$\hat{z}_j \;=\; \mathrm{E}[\tilde{z}_j | \tilde{\mathbf{x}}_j, \tilde{y}_j \neq 0, \Theta] \;+\; \frac{\partial}{\partial \tilde{\xi}_j} \mathcal{L}_{\mathrm{bal}}(\Theta), \tag{25}$$

where the first term on the right-hand side is given by eq. (13), and the second term gives corrections from the class-balancing constraint. It is straightforward (though tedious) to differentiate $\mathcal{L}_{\mathrm{bal}}(\Theta)$ on the right hand side of eq. (25) with respect to the classifier score $\tilde{\xi}_j = \mathbf{w} \cdot \tilde{\mathbf{x}}_j + b$ of the jth unlabeled example. Once again the EM algorithm simply alternates between computing posterior means and solving a least-squares regression. In particular, the M-step update with the class-balance constraint takes the same form as eq. (16), but with the posterior means in eq. (25) substituting for the target values $\hat{z}_j$. Henceforth we refer to this semi-supervised version of the EM algorithm with the class-balancing constraint as $\mathbf{EMBLEM}_{\mathbf{ss}}^{\mathbf{bal}}$.

4 Experiments

In this section, we evaluate $\mathrm{EMBLEM}_{\mathrm{ss}}$ and $\mathrm{EMBLEM}_{\mathrm{ss}}^{\mathrm{bal}}$ on several large problems in text classification and compare their performance to existing implementations of $\mathrm{S}^3\mathrm{VMs}$. Text classification is one of the oldest applications of semi-supervised learning (Nigam et al., 2000); moreover, it lends itself to linear models, as we consider here, due to the sparse, high-dimensional nature of word-document counts.

4.1 Setup

We experimented on six tasks in binary text classification, of which four were adopted from previous work in semi-supervised learning (Sindhwani and Keerthi, 2006). The `aut-avn` and `real-sim` tasks were derived from a collection of UseNet articles (McCallum, 2001). After removing zero vectors, the `aut-avn` task has 71066 documents with a 20707-term vocabulary, and the `real-sim` task has 72201 documents with a 20958-term vocabulary. The `ccat` and `gcat` tasks were created from the top-level categories in the RCV1 data set (Lewis et al., 2004); both tasks have 23149 documents with a 47236-term vocabulary. We created the fifth task from the 20-Newsgroups data set (Rennie, 2008), combining the `comp` and `sci` topics to form one class and collapsing the remaining documents into the other class, which resulted in 18774 documents with a 61188-term vocabulary. The last task was

derived from a collection of job postings on a crowdsourcing site Freelancer.com (Motoyama et al., 2011; Kim et al., 2011). We considered the simple binary task of distinguishing postings for benign versus abusive jobs. The data set contains 355386 documents with a 27600-term vocabulary; however, only a small subset of 5489 documents were manually labeled by the researchers. We preprocessed all the data sets by tf-idf weighting and normalized all the resulting document-vectors to have unit length.

Each experiment was done in a transductive setting: we trained on a partially labeled data set then tested on its unlabeled examples. All reported results were averaged over our own twelve random labeled/unlabeled splits. We ensured that both classes were represented among the labeled examples but otherwise did *not* balance the class labels across splits. For both $\text{EMBLEM}_{\text{ss}}$ and $\text{EMBLEM}_{\text{ss}}^{\text{bal}}$, we initialized the model parameters $\mathbf{w}$ and b by a linear SVM trained by LIBLINEAR (Fan et al., 2008) on the labeled examples. The M-step — sparse least-squares regression — was done by a preconditioned conjugate gradients method (Barrett et al., 1994). The regularization parameter λ was set to unity. For $\text{EMBLEM}_{\text{ss}}^{\text{bal}}$, we estimated the class balance of unlabeled examples from the class balance of labeled ones. In particular, we constrained $|\tilde{\mu} - \mu| \leq 0.1\sigma/\sqrt{n}$, where μ and σ^2 were respectively the sample mean and variance of the n training labels. We made an exception for the Freelancer data set, however, where Motoyama et al. (2011) manually estimated the proportion of abusive job postings as 29.2% from a collection of 2000 samples. We used this prior knowledge to constrain the balance of classes in all the Freelancer experiments.

We compared our models to three popular implementations of S^3VMs: SVM-Light (Joachims, 1999b), UniverSVM (Collobert et al., 2006), and SVMlin (Sindhwani and Keerthi, 2006). SVM-Light initializes labels for unlabeled examples using a classifier trained from labeled examples; it then alternates between solving a quadratic program and switching labels for individual pairs of unlabeled examples. The hyperparameter C^{-1} for SVM-Light was set to the mean squared ℓ_2-norm of all examples (equal to unity in all our experiments). The optimization in UniverSVM is based on a convex-concave procedure. All the hyperparameters of UniverSVM were set to the default values. SVMlin implements a modified finite newton method for ℓ_2-SVMs (Keerthi and DeCoste, 2005). As in SVM-Light, SVMlin swaps the labels of unlabeled examples, but it does so for multiple pairs at a time (yielding a considerable speedup). The hyperparameters λ and λ' of SVMlin were set to 0.001 and 1, respectively.

4.2 Results

Our first set of experiments compared the two versions of the latent variable model in Section 3, one with class-balancing constraints ($\text{EMBLEM}_{\text{ss}}^{\text{bal}}$), and one without ($\text{EMBLEM}_{\text{ss}}$). Does the class-balancing constraint lead to more accurate classification of unlabeled examples, and if so, by how much? Fig. 6 plots the gain from the class-balancing constraint on three different tasks as a function of the number of labeled examples. In these experiments, we see that the class-balancing constraint was universally helpful (i.e., the gain was always positive); it was especially critical for semi-supervised learning from very few labeled examples. As expected, though, the class-balancing constraint had a smaller effect on learning in the experiments with larger numbers of labeled examples.

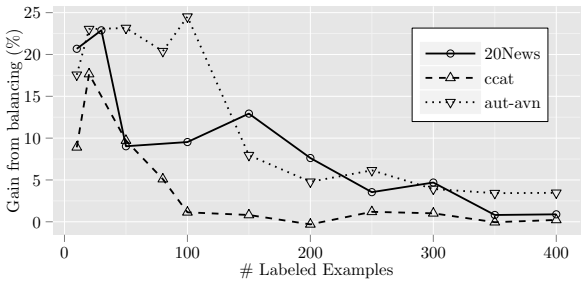

Figure 6: Gain of $\text{EMBLEM}_{\text{ss}}^{\text{bal}}$ over $\text{EMBLEM}_{\text{ss}}$ in classification accuracy on unlabeled examples.

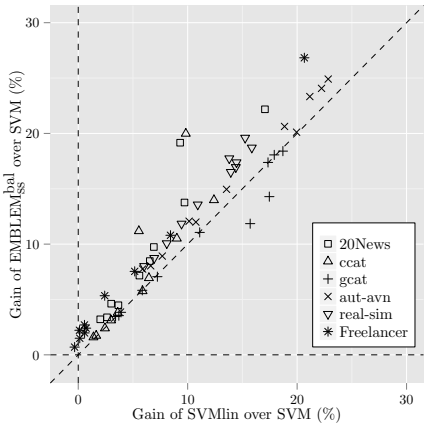

Figure 8: Gain in accuracy of $\text{EMBLEM}_{\text{ss}}^{\text{bal}}$ over SVM vs. gain in accuracy of SVMlin over SVM.

Our second set of experiments compared $\text{EMBLEM}_{\text{ss}}^{\text{bal}}$ against other implementations of S^3VMs, focusing again on classification accuracies for unlabeled examples. Fig. 7 plots the average test error rates as a function of the number of labeled examples. (Some results are missing for data sets that were too large for particular implementations: on the Freelancer task, SVM-Light terminated without con-

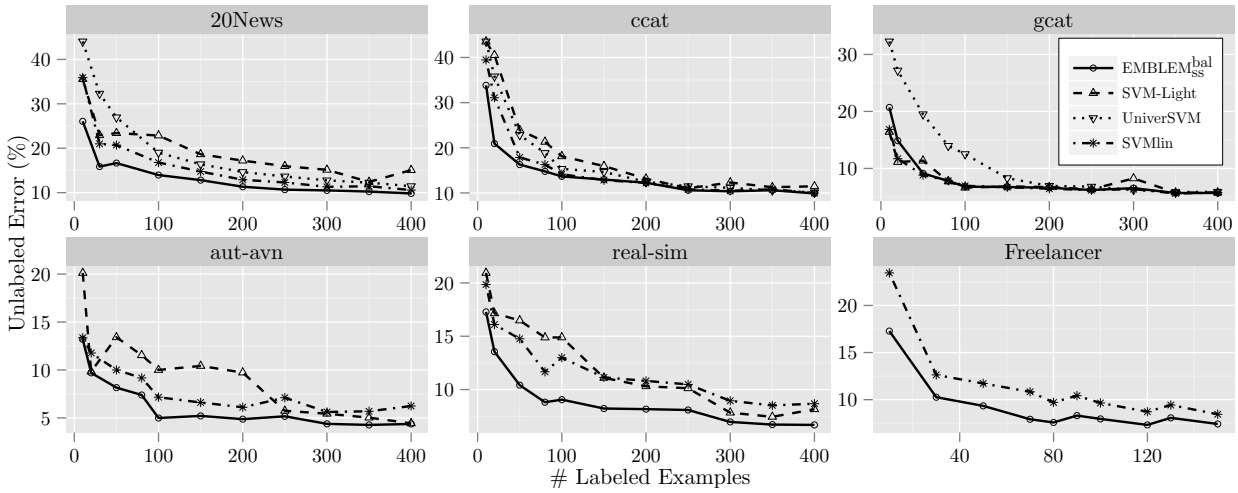

Figure 7: Average test error rates on six data sets for EMBLEM$_{ss}^{bal}$, SVM-Light, UniverSVM, and SVMlin. Results are shown in the regime where the use of unlabeled data improves performance over a baseline SVM.

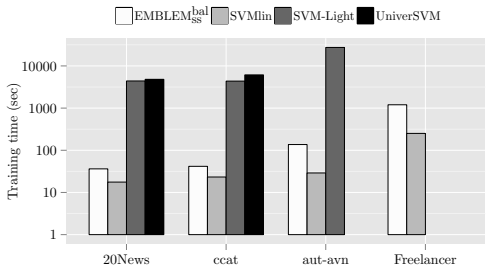

Figure 9: Comparison of average training times.

vergence, and on the three largest tasks, UniverSVM was unable to store the kernel matrix.) The subplot for each task shows results in the regime where semi-supervised methods (i.e., those making use of the unlabeled data) actually lead to better performance than a baseline SVM. On four out of the six tasks (except `ccat` and `gcat`), EMBLEM$_{ss}^{bal}$ performs better in this regime than all the other algorithms over the full range of labeled examples. On `ccat` and `gcat` tasks, EMBLEM$_{ss}^{bal}$ does not always perform best but only trails by a small amount. The most competitive alternative to EMBLEM$_{ss}^{bal}$ is SVMlin. Fig. 8 collapses all the results for EMBLEM$_{ss}^{bal}$ and SVMlin in Fig. 7 into a single scatterplot. The axes in this plot record the gains of EMBLEM$_{ss}^{bal}$ (vertical) and SVMlin (horizontal) over a baseline SVM. We see from the large concentration of points above the diagonal that with only a few exceptions EMBLEM$_{ss}^{bal}$ improves more over baseline SVMs than SVMlin.

Finally, Fig. 9 compares average training times. EMBLEM$_{ss}^{bal}$ is orders-of-magnitude faster than SVM-Light and UniverSVM, though still somewhat slower than SVMlin. This is partly due to the fact that EMBLEM$_{ss}^{bal}$ is implemented in MATLAB, whereas SVMlin is implemented in C.

5 Conclusion

We have introduced a Gaussian latent variable model for semi-supervised learning of large margin classifiers. The EM algorithm for this model—EMBLEM—is especially well geared to sparse, high-dimensional data and generally outperforms existing implementations of S^{3}VMs. Our experiments suggest that EMBLEM can scale to even larger problems than we considered here: for example, we can easily parallelize the E-step (which simply computes the posterior mean for each example), and we can also parallelize the M-step (least-squares regression) using MapReduce (Chu et al., 2007). Though no single model of semi-supervised learning is likely to provide a "silver bullet" for all applications (Chapelle et al., 2008, 2006), EMBLEM has many attractive properties, including a crisp probabilistic semantics, transparent handling of unlabeled data, and a simple core optimization (sparse least-squares). Our implementation of EMBLEM is available online at `https://github.com/dokyum/EMBLEM`, and we hope that others will find it equally useful in their work.

Acknowledgements

This work was supported in part by NSF grant NSF-1237264 and ONR MURI grant N000140911081. Kim was also supported by a Samsung Scholarship.

References

M.-F. Balcan, A. Blum, P. P. Choi, J. Lafferty, B. Pantano, M. R. Rwebangira, and X. Zhu. Person identification in webcam images: An application of semi-supervised learning. In *ICML 2005 Workshop on Learning with Partially Classified Training Data*, 2005.

R. Barrett, M. Berry, T. F. Chan, and et al. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, 1994.

K. P. Bennett and A. Demiriz. Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems*, 1998.

P. Billingsley. *Probability and measure*. John Wiley & Sons, 3rd edition, 1995.

O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, 2005.

O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006. URL `http://www.kyb.tuebingen.mpg.de/ssl-book`.

O. Chapelle, V. Sindhwani, and S. S. Keerthi. Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9:203–233, June 2008.

C.-T. Chu, S. K. Kim, Y.-A. Lin, Y. Yu, G. Bradski, A. Y. Ng, and K. Olukotun. Map-reduce for machine learning on multicore. In *Advances in Neural Information Processing Systems 19*. 2007.

R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive svms. *Journal of Machine Learning Research*, 7:1687–1712, September 2006.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–37, 1977.

J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson. Offline/realtime traffic classification using semi-supervised learning. *Perform. Eval.*, 64(9-12): 1194–1213, Oct. 2007.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, 1999a.

T. Joachims. Making large-scale svm learning practical. In *Advances in Kernel Methods – Support Vector Learning*. MIT Press, 1999b.

S. S. Keerthi and D. DeCoste. A modified finite newton method for fast solution of large scale linear svms. *Journal of Machine Learning Research*, 6:341–361, Dec. 2005.

D.-k. Kim, M. Motoyama, G. M. Voelker, and L. K. Saul. Topic modeling of freelance job postings to monitor web service abuse. In *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, 2011.

N. D. Lawrence and M. I. Jordan. Semi-supervised learning via gaussian processes. In *Advances in Neural Information Processing Systems 17*. 2005.

D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, Dec. 2004.

A. McCallum. Simulated/real/aviation/auto usenet data, 2001. URL `http://people.cs.umass.edu/~mccallum/data.html`.

P. McCullagh and J. A. Nelder. *Generalized linear model*, volume 37. Chapman & Hall/CRC, 1989.

M. Motoyama, D. McCoy, K. Levchenko, G. M. Voelker, and S. Savage. Dirty Jobs: The Role of Freelance Labor in Web Service Abuse. In *Proceedings of the USENIX Security Symposium*, 2011.

K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2-3):103–134, May 2000.

K. Ogawa, M. Imamura, I. Takeuchi, and M. Sugiyama. Infinitesimal Annealing for Training Semi-Supervised Support Vector Machines. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013.

J. Rennie. The 20 newsgroups data set, 2008. URL `http://qwone.com/~jason/20Newsgroups/`.

V. Sindhwani and S. S. Keerthi. Large scale semi-supervised linear svms. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, 2006.

J. Wang, X. Shen, and W. Pan. On efficient large margin semisupervised learning: Method and theory. *Journal of Machine Learning Research*, 10:719–742, June 2009.

J. Weston, C. Leslie, E. Ie, D. Zhou, A. Elisseeff, and W. S. Noble. Semi-supervised protein classification using cluster kernels. *Bioinformatics*, 21(15):3241–3247, 2005.

Y. Yu. Monotonically overrelaxed em algorithms. *Journal of Computational and Graphical Statistics*, 2011.