Incorporating Prior Knowledge with Weighted Margin Support Vector Machines *

Xiaoyun Wu
Dept. of Computer Science and Engineering
University at Buffalo
Amherst, NY 14260
xwu@cse.buffalo.edu

Rohini Srihari
Dept. of Computer Science and Engineering
University at Buffalo
Amherst, NY 14260
rohini@cse.buffalo.edu

ABSTRACT

Like many purely data-driven machine learning methods, Support Vector Machine (SVM) classifiers are learned exclusively from the evidence presented in the training dataset; thus a larger training dataset is required for better performance. In some applications, there might be human knowledge available that, in principle, could compensate for the lack of data. In this paper, we propose a simple generalization of SVM: Weighted Margin SVM (WMSVMs) that permits the incorporation of prior knowledge. We show that Sequential Minimal Optimization can be used in training WMSVM. We discuss the issues of incorporating prior knowledge using this rather general formulation. The experimental results show that the proposed methods of incorporating prior knowledge is effective.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.5.4 [**Pattern Recognition**]: Design Methodology—*classifier design and evaluation*

General Terms

Algorithms,Performance

Keywords

Text Categorization, Support Vector Machines, Incorporating Prior Knowledge

1. INTRODUCTION

Support Vector Machines (SVM) have been successfully applied in many real-world applications. However, little work [4, 14] has been done to incorporate prior knowledge into SVMs. Schölkopf [14] showed that the prior knowledge can be incorporated with the appropriate kernel function, and Fung [4] showed prior knowledge in the form of multiple polyhedral sets can be used with a reformulation of SVM. In this paper, we describe a generalization of SVM that allows for incorporating prior knowledge of any form, as long as it can be used to estimate the conditional in-class probabilities. The proposed Weighted Margin Support Vector Machine (WMSVM) can generalize on imperfectly labeled training dataset because each pattern in the dataset associates not only with a category label but also a confidence value that varies from 0.0 to 1.0. The confidence value measures the strength of the corresponding label. This paper provides the geometrical motivations for generalized WMSVM formulation, its primary and dual problem, and a modification of Sequential Minimum Optimization (SMO) training algorithm for WMSVM. We can then incorporate the prior human knowledge through generating the "pseudo training dataset" from an unlabeled dataset, using the estimation of conditional probability $P(y|x)$ over the possible label values {-1,+1} as the confidence value.

In this paper we use text classification as a running example not only because empirical studies [7, 18] suggest SVM is well suited for the application and often produces better results, but also because the keyword based prior knowledge is easy to obtain [13] in the text domain. For example, it is intuitive that words like "NBA" are indicative of sports category. Therefore it is of interest to see whether the ability of incorporating fuzzy prior knowledge can offer further improvement over this already highly effective method.

The rest of this paper is structured as follows. We introduce related work in section 2. Section 3 discusses the generalized WMSVM, its geometrical motivation, formulation, primary and dual optimization problem. Section 4 briefly describes how to use the modified SMO for WMSVM training. The general issues faced when combining the true training dataset and pseudo training dataset are analyzed in the section 5. In Section 6, we present experimental results on a popular text categorization dataset. We conclude in Section 7 with some discussion of potential use of WMSVMs.

2. RELATED WORK

Most machine learning methods are statistical based. They are usually considered as data-driven methods, since prediction models are generalized from some labeled training

*(Produces the permission block, copyright information and page numbering). For use with ACM_PROC_ARTICLE-SP.CLS V2.6SP. Supported by ACM.

KDD'04, August 22–25, 2004, Seattle, Washington, USA.

datasets. Different learning methods usually use different hypothesis space, and thus can result in different performance on the same application. The common theme however is that an adequate number of labeled training examples is required to guarantee the performance of generalized model, and the more labeled training data the better the performance.

However, labeling the data is usually time consuming and expensive and therefore having enough labeled training data is rare in many real-world applications. The lack of labeled data has been addressed in many recent studies [15, 1, 8, 3, 13]. To reduce the need for the labeled data, these studies are usually conducted on a learning problem that is slightly different from the standard settings. For example, while the training set is chosen to be a random sampling of instances, in active learning [15], the learner can actively choose the training data. By always picking the most informative data points to be labeled, it is hoped that the learner's need for large quantities of labeled data can be reduced. This paper is, however, developed based on the following two approaches: learning with prior knowledge and transductive learning.

In some applications, while labeled data can be limited, there may be human knowledge that might compensate for the lack of labeled data. Schapiro et al. showed in [13] that logistic regression can be modified to allow the incorporation of prior human knowledge. Note that although the training method is a boosting style algorithm, the modified logistic regression can also be trained by other methods such as Gauss-Seidel [5]. In their approach, rough prior human knowledge is represented as a prediction rule π that maps each instance x to an estimated conditional probability distribution $\pi(y|x)$ over the possible label values $-1, +1$. Given this prior model and training data, they seek a logistic model $\sigma(x)$ that fits not only the labeled data but also the prior model. They measure the fit to the data by log conditional likelihood, the fit to the prior model by the relative entropy(Kullback-Leibler divergence). Let $\pi_+ = \pi(y = +1|x)$, the objective function for the modified logistic regression is given:

$$\sum_i [\ln(1 + \exp(-y_i f(x_i)) + \eta RE(\pi_+(x_i)||\sigma(x_i))] \qquad (1)$$

where $RE(\cdot)$ is the binary relative entropy, and η is used to control the relative importance of two terms in the objective function. While using the relative entropy to measure the fit to the prior model is a natural solution for the logistic model, it is not applicable to SVM since the prediction model generalized by SVM is discriminant in nature.

Transductive learning was first introduced in [16]. In [1, 8], transductive support vector machine was proposed and its application in text categorization was demonstrated. The difference between the standard SVM and transductive SVM is whether the unlabeled test set is used in the training stage. In particular, the position information of unlabeled test set is used by transductive SVM to decide the decision hyperplane. Transductive SVM is depicted in figure 1. Positive/negative examples are marked as +/-, test examples as circles. The dashed line is the solution of the standard SVM. The solid line shows the transductive classification. The problem of transductive SVM is that its training is much more difficult. For example, integer programming was used in [1], and an iterative method with one SVM training on each step was used in [8]. Although the exact time complexity analysis for these training algorithms are not available, the general impression is that they are significantly slower than the standard SVM training.

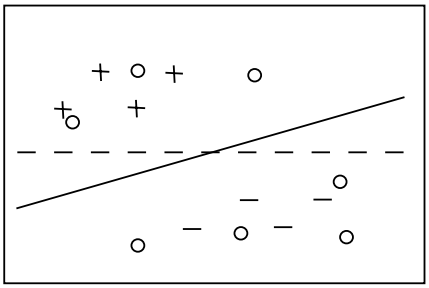

Figure 1: Transductive SVM

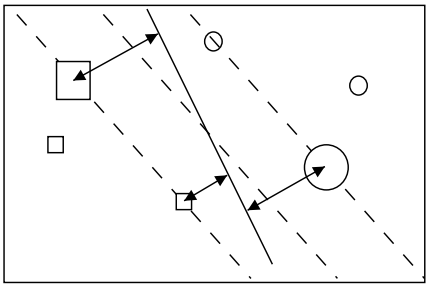

Figure 2: Weighted Margin SVM

3. WEIGHTED MARGIN SUPPORT VECTOR MACHINES

We now describe some notations and definitions from which we developed weighted margin support vector machines. Given a set of vectors $(x_1, ..., x_n)$, along with their corresponding labels $(y_1, ..., y_n)$ where $y_i \in \{+1, -1\}$, the SVM classifier defines a hyperplane (w, b) in kernel mapped feature space that separates the training data by a maximal margin.

DEFINITION 1. *We define the functional margin of a sample (x_i, y_i) with respect to a hyperplane (w, b) to be $y_i(\langle w \cdot x_i \rangle + b)$. We define the geometric margin of a sample (x_i, y_i) with respect to a hyperplane (w, b) to be $y_i(\langle w \cdot x_i \rangle + b)/\|w\|_2$, where $\|w\|_2$ is the L_2 norm of w. Furthermore, we define the geometric margin of a set of (x_i, y_i) with respect to a hyperplane (w, b) to be the quantity of* $\min_{0 \le i < n}(y_i(\langle w \cdot x_i \rangle + b)/\|w\|_2)$.

The maximum margin hyperplane for a training set S is the hyperplane with respect to which the training set has maximal margin over all hyperplanes defined in the feature space. Typically the maximum margin hyperplane is pursued by fixing the functional margin of the training set to be 1 and minimizing the norm of the weight vector w. Those samples with minimum geometric margin with respect to maximum margin hyperplane are called support vectors because the maximum margin hyperplane is supported by these vectors: deleting support vectors will result in different maximum margin hyperplane.

We consider the problem settings where, besides the vectors $(x_1, ..., x_n)$ and their corresponding labels $(y_1, ..., y_n)$, we also have confidence values $(v_1, ..., v_n)$. Each v_i, where $v_i \in (0, 1]$, indicates the confidence level of y_i's labeling. Intuitively, the larger the confidence we have on a label, the larger the margin we want to have on that sample. But in the standard SVM, there is no provision for this confidence value to be useful. The difference between the WMSVM and SVM is illustrated in figure 2. There, positive examples are depicted as circles and negative examples squares. The size of the squares/circles represents their associated confidence value. The dashed line in the middle is the hyperplane derived based on the standard SVM training, and the solid line is the solution to the transductive SVM learning.

DEFINITION 2. *We define the effective weighted functional margin of weighted sample (x_i, y_i, v_i) with respect to a hyperplane (w, b) and a margin normalization function f to be $f(v_i)y_i(\langle w \cdot x_i\rangle + b)$, where f is a monotonically decreasing function.*

3.1 Weighted Hard Margin Classifier

The simplest model of support vector machine is the maximal hard margin classifier. It only works on a data set that is linearly separable in feature space thus, it can not be used in many real-world situations. But it is the easiest algorithm to understand, and it forms the foundation for more complex Support Vector Machines. In this subsection, we will generalize this basic form of Support Vector Machines so that it can be used on fuzzy truthing data.

When each label is associates with a confidence value, intuitively one wants support vectors that are labeled with higher confidence to assert more force on the decision plane, or equivalently one wants those support vectors to have bigger geometric margin to the decision plane. So, to train a maximal weighted hard margin classifier, we fix the effective weighted functional margin instead of fixing the functional margin of support vectors. Then we try to minimize the norm of weight vector. We thus have the following proposition.

PROPOSITION 1. *Given a linearly separable (in feature space if kernel function is used) training sample set*

$$S = ((x_1, y_1, v_1), \cdots, (x_n, y_n, v_n)) \tag{2}$$

the hyperplane (w, b) that solves the following optimization problem

$$\begin{aligned} &\textit{minimize} \quad : \quad \langle w \cdot w\rangle \\ &\textit{subject to} \quad : \quad f(v_i)y_i(\langle w \cdot x_i\rangle + b) \geq 1, i = 1, \cdots, n \end{aligned}$$

realizes the maximal weighted hard margin hyperplane, where f is a monotonically decreasing function such that $f(\cdot) \in (0, 1]$.

The corresponding dual optimization problem can be found by differentiating the primal Lagrangian with respect to w and b, imposing stationarity:

PROPOSITION 2. *Given a linearly separable (in feature space if kernel function is used) training sample set*

$$S = ((x_1, y_1, v_1), \cdots, (x_n, y_n, v_n)) \tag{3}$$

and suppose the parameters α^* solve the following optimization problem

$$\begin{aligned} \textit{maximize} \quad & W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i y_i f(v_i)\alpha_j y_j f(v_j)\langle x_i \cdot x_j\rangle \\ \textit{subject to} \quad & \sum_{i=1}^{n} y_i \alpha_i f(v_i) = 0 \\ & \alpha_i \geq 0, i = 0, \cdots n \end{aligned}$$

then the weight vector $w^* = \sum_{i=1}^{n} y_i \alpha_i^* x_i f(v_i)$ realizes the maximal weighted hard margin hyperplane.

The value of b does not appear in the dual problem so b^* must be found in the primal constraints:

$$b^* = -\frac{f(v_i)\langle w^* \cdot x_i\rangle + (f(v_j)\langle w^* \cdot x_j\rangle}{f(v_i) + f(v_j)} \tag{4}$$

Where

$$\begin{aligned} i &= \arg\max f(v_n)\langle w^* \cdot x_n\rangle, \quad y_n = -1, \\ j &= \arg\min f(v_n)\langle w^* \cdot x_n\rangle, \quad y_n = +1, \end{aligned}$$

The Karush-Kuhn-Tucker condition states that optimal solutions α^*, (w^*, b^*) must satisfy

$$\alpha_i^*[y_i f(v_i)(\langle w_i^* \cdot x_i\rangle + b^*) - 1] = 0, i = 0, \cdots n \tag{5}$$

This condition implies that only inputs x_i for which the functional margin is $f^{-1}(v_i)$ have their corresponding α_i^* non-zero. These are the support vectors in the WMSVM. All other training samples will have their α_i^* equal to zero. In the final expression of weight vector w^*, only these support vectors will be needed. Thus we have decision plane $h(x)$:

$$h(x, a^*, b^*) = \langle w^* \cdot x\rangle + b^* = \sum_{i \in sv} \alpha_i^* y_i f(v_i)\langle x_i \cdot x\rangle + b^* \tag{6}$$

3.2 Weighted Soft Margin Classifier

The hard maximal margin classifier is an important concept, but it has two problems. First, hard margin classifier can be very brittle, since any labeling mistake on support vectors will result in significant change in decision hyperplane. Second, training data is not always linearly separable, and when it is not, we are forced to use a more powerful kernel, which might result in over-fitting. To be able to tolerate noise and outliers, we need to take into consideration the positions of more training samples than just those closest to the boundary. This is done generally by introducing slack variables and soft margin classifier.

DEFINITION 3. *Given a value $\gamma > 0$, we define the margin slack variable of a sample (x_i, y_i) with respect to the hyperplane (w, b) and target margin γ to be*

$$\xi_i = \max(0, \gamma - y_i(\langle w \cdot x_i\rangle + b)) \tag{7}$$

This quantity measures how much a point fails to have a margin of γ from the hyperplane (w, b). If $\xi_i > 0$, then x_i is misclassified by (w, b). As a more robust measure of margin distribution, $\sum_{i=0}^{n} \|\xi_i\|_p$ measures the amount by which the training set fails to have margin γ, and it takes into account any misclassification of the training data. The soft margin classifier is typically the solution that minimizes the regularized norm of $\langle w \cdot w\rangle + C\sum_{i=0}^{n} \|\xi_i\|_p$. To generalize the soft margin classifier to weighted soft margin classifier, we first define a weighted version of slack variable.

DEFINITION 4. *Given a value $\gamma > 0$, we define the effective weighted margin slack variable of a sample (x_i, y_i, v_i) with respect to the hyperplane (w, b) and margin normalization function f, slack normalization function g and target margin γ as*

$$\xi_i^w = g(v_i)\max(0, \gamma - y_i f(v_i)(\langle w \cdot x_i\rangle + b)) = g(v_i)\xi_i \quad (8)$$

where f is a monotonically decreasing function such that $f(\cdot) \in (0, 1]$, g is a monotonically increasing function.

The primal optimization problem of maximal weighted soft margin classifier can thus be formulated as:

PROPOSITION 3. *Given a training sample set*

$$S = ((x_1, y_1, v_1), \cdots, (x_n, y_n, v_n)) \quad (9)$$

the hyperplane (w, b) that solves the following optimization problem

$$\begin{aligned} \textit{minimize} \quad & : \quad \langle w \cdot w\rangle + C\sum_{i=1}^{n} g(v_i)\xi_i \\ \textit{subject to} \quad & : \quad y_i(\langle w \cdot x_i\rangle + b)f(v_i) \geq 1 - \xi_i, i = 1, \cdots, n \\ & \quad \xi_i \geq 0, i = 1, \cdots, n \end{aligned}$$

realizes the maximal weighted soft margin hyperplane, where f is a monotonically decreasing function such that $f(\cdot) \in (0, 1]$, g is a monotonically increasing function such that $g(\cdot) \in (0, 1]$.

Here the effective weighted margin slack variable is used to regulate $\langle w \cdot w\rangle$. This implies that the final decision plane will be more tolerant on these margin violating samples with low confidence than others. This is exactly what we want: samples with high confidence label to contribute more to final decision plane.

The corresponding dual optimization problem can be found by differentiating the corresponding Lagrangian with respect to w, b, ξ_i, imposing stationarity:

PROPOSITION 4. *Given a training sample set*

$$S = ((x_1, y_1, v_1), \cdots, (x_n, y_n, v_n)) \quad (10)$$

and suppose the parameters α^* solve the following optimization problem

$$\begin{aligned} \textit{maximize} \quad & W(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i y_i f(v_i)\alpha_j y_j f(v_j)\langle x_i \cdot x_j\rangle \\ \textit{subject to} \quad & \sum_{i=1}^{n} y_i\alpha_i f(v_i) = 0 \\ & g(v_i)C \geq \alpha_i \geq 0, i = 0, \cdots n \end{aligned}$$

then the weight vector $w^* = \sum_{i=1}^{n} y_i\alpha_i x_i f(v_i)$ realize the maximal weighted soft margin hyperplane, where f is a monotonically decreasing function such that $f(\cdot) \in (0, 1]$, g is a monotonically increasing function such that $g(\cdot) \in (0, 1]$.

Notice the dual objective function is curiously identical to that of the weighted hard margin case. The only difference is the constraint $g(v_i)C \geq \alpha_i \geq 0$, where the first part of the constraint comes from the conjunction of $Cg(v_i) - \alpha_i - r_i = 0$ and $r_i \geq 0$.

The KKT conditions in this case are therefore

$$\begin{aligned} \alpha_i[y_i f(v_i)(\langle x_i \cdot w\rangle + b) - 1 + \xi_i] = 0, \quad & i = 1, \cdots, n \\ \xi_i(\alpha_i - g(v_i)C) = 0, \quad & i = 1, \cdots, n \end{aligned}$$

This implies that samples with non-zero slack can only occur when $\alpha_i = g(v_i)C$, they are bounded support vectors. Samples for which $g(v_i)C > \alpha_i > 0$ (unbounded support vectors) have an effective weighted margin of $1/\|w\|$ from the hyperplane (w^*, b^*). The threshold b^* can be calculated in the same way as before. The $h(x)$ will also have the same expression as before.

3.3 Discussion on WMSVM Formulation

In [9], SVM with different misclassification costs is introduced to battle the imbalanced dataset where the number of negative examples is overwhelming. In particular, the primal optimization problem is given:

$$\langle w \cdot w\rangle + C_{y_i}\xi_i$$

Let m_{-1}, m_{+1} denote the number of negative and positive examples, and assume $m_{-1} \geq m_{+1}$, one typically wants to have $C_{+1} \geq C_{-1}$. This amounts to penalize more on an error made on positive example in training process.

Both WMSVM and SVM with different misclassification cost for each example can result in the same box constraint for each α when we have $C_i^{SVM} = C_i^{WMSVM} g(v_i)$. However, there is some intrinsic difference between them. To see this, let $C_i = 0$ for both formulations. As shown in Fig. 2, these two different formulations can result in different decision hyperplanes. The difference between these two formulations is also readily revealed in their respective dual objective functions. For example, attempt to replace $\alpha_i f(v_i)$ with α_i^* in dual objective function for WMSVM results in

$$\sum_{i=1}^{n}\frac{\alpha_i^*}{f(v_i)} - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i^* y_i \alpha_j^* y_j\langle x_i \cdot x_j\rangle$$

which is different from that of standard SVM:

$$\sum_{i=1}^{n}\alpha_i^* - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i^* y_i \alpha_j^* y_j\langle x_i \cdot x_j\rangle$$

.

4. SEQUENTIAL MINIMAL OPTIMIZATION FOR WMSVM

The Sequential Minimal Optimization (SMO) algorithm is first proposed by Platt [12], and later enhanced by Keerthi [10]. It is essentially a decomposition method with working set of two examples. The optimization problem can be solved analytically; thus SMO is one of the easiest optimization algorithms to implement. There are two basic components of SMO: analytical solution for two points and working set selection heuristics. Since the selection heuristics in Keerthi's improved SMO implementation can be easily modified to work with WMSVM, only the analytical solution is briefly described here.

Assume that x_1 and x_2 are selected for current optimization step. To observe the linear constraint, the values of their multipliers (α_1, α_2) must lie on a line:

$$y_1\alpha_1^{new} f(v_1) + y_2\alpha_2^{new} f(v_2) = y_1\alpha_1^{old} f(v_1) + y_2\alpha_2^{old} f(v_2) \quad (11)$$

where a box constraint applies: $g(v_1)C \geq \alpha_1 \geq 0, g(v_2)C \geq \alpha_2 \geq 0$. A more restrictive constraint on the feasible value for α_2^{new}, $U \leq \alpha_2^{new} \leq V$, can be derived by the box con-

straint and linear equality constraint, where

$$U = \max(0, \frac{\alpha_2^{old} g(v_2) - \alpha_1^{old} g(v_1)}{g(v_2)})$$
$$V = \min(g(v_2)C, \frac{g^2(v_1)C - \alpha_1^{old} g(v_1) + \alpha_2^{old} g(v_2)}{g(v_2)})$$

if $y_1 \neq y_2$, and

$$U = \max(0, \frac{\alpha_1^{old} g(v_1) + \alpha_2^{old} g(v_2) - g^2(v_1)C}{g(v_2)})$$
$$V = \min(g(v_2)C, \frac{\alpha_2^{old} g(v_2) + \alpha_1^{old} g(v_1)}{g(v_2)})$$

if $y_1 = y_2$.

Let $h(x)$ denote the decision hyperplane $\langle w \cdot x \rangle + b$ represented as $\sum_{j=0}^{n} \alpha_j y_j f(v_j) \langle x \cdot x_j \rangle + b$, let E_i denote the scaled difference between the function output and target classification on the training samples:

$$E_i = \frac{y_i f(v_i) h(x_i) - 1}{y_i f(v_i)}, i = 1, 2 \quad (12)$$

then it is easy to prove the following theorem.

THEOREM 1. *The maximum of the objective function for the soft margin optimization problem, when only α_1, α_2 are allowed to change, is achieved by first computing the quantity*

$$\alpha_2^{new,unc} = \alpha_2^{old} + \frac{E_1 - E_2}{y_2 f(v_2)(K(x_1, x_1) + K(x_2, x_2) - 2K(x_1, x_2))}$$

and then clipping it to enforce the constraint $U \leq \alpha_2^{new,unc} \leq V$:

$$\alpha_2^{new} = \begin{cases} V & \text{if } \alpha_2^{new,unc} \leq V \\ \alpha_2^{new,unc} & \text{if } U \leq \alpha_2^{new,unc} \leq V, \\ U & \text{if } \alpha_2^{new,unc} \geq U \end{cases}$$

The value of α_1^{new} is obtained from α_2^{new} as follows:

$$\alpha_1^{new} = \alpha_1^{old} + \frac{y_2 f(v_2)(\alpha_2^{old} - \alpha_2^{new})}{y_1 f(v_1)} \quad (13)$$

5. INCORPORATING PRIOR KNOWLEDGE

The proposed Weighted Margin Support Vector Machine is a general formulation. It is useful for incorporating any confidence value attached to each instance in the training dataset. However, along with added generality, there are some issues which need to be addressed to make it practical. For example, it is not clear from the formulation how to choose margin normalization function f and slack normalization function g. One also needs to determine the confidence value v_i for each example. In this section, we will address these issues for the application of incorporating prior knowledge into SVM using WMSVM.

We propose a two-step approach. First, rough human prior knowledge is used to derive a rule π, which assigns each unlabeled pattern x a confidence value that indicates the likelihood of pattern x belonging to category of interest. A "pseudo training dataset" is generated by applying these rules on a set of unlabeled documents. Second, the true training dataset and the pseudo training dataset are concatenated to form a training dataset, and a WMSVM classifier can then be trained from it.

5.1 Creating Pseudo Training Dataset

In [4], Fung et al introduce a SVM formulation that can incorporate prior knowledge in the form of multiple polyhedral sets. However, in practice, it is rare to have prior knowledge available in such closed function form. In general, human prior knowledge is fuzzy in nature, the rules resulting from it thus have two problems. First, the coverage of these rules are usually limited since they may not be able to provide prediction for all the patterns. Second, these rules are usually not accurate and precise.

We will have to defer the discussion on how to derive prediction rules to the next section as it is largely an application dependent issue. Given such prediction rules, we generate a "pseudo training dataset" by applying these rules on a set of unlabeled dataset, in our case, the test set. This amounts to using the combined evidence from the human knowledge and labeled training data at both training and testing stage. Similar to transductive SVM, the idea of using unlabeled test set is a direct application of Vapnik's principle of never solving a problem which is more general than the one we actually need to solve [17]. However, the proposed approach differs from the transductive learning in two aspects. First, in determining the decision hyperplane, the proposed approach relies on both the prior knowledge and the distribution of these testing examples, while transductive SVM relies only on the distribution. Second, contrast to one iteration of SVM training needed by the proposed approach, multiple iterations of SVM training are needed and the number of iterations is dependent on the size of test set. For a large test-set, transductive SVM is significantly slower.

The proposed way of incorporating such fuzzy prior knowledge is mainly influenced by the approach introduced in [13]. However, there are some noticeable differences between these two approaches. First, while the proposed approach can work on rules with limited coverage, the approach in [13] needs to work on rules with complete coverage. In another words, the rules needed there have to make a prediction on every instance. This requirement can be too restrictive sometimes and reenforcement of such a requirement can introduce unnecessary noise. Second, the proposed approach has an integrated training and testing phase, thus classification is based on the evidence from both the training data and prior knowledge. However, the prediction power of human knowledge on testing data is thus lost in their approach.

5.2 Balancing Two Conflicting Goals

Given the true training dataset and pseudo training dataset, we now have two possibly conflicting goals in minimizing the empirical risk when constructing a predictor: (1) fit the true training dataset, and (2) fit the pseudo training dataset and thus fit the prior knowledge. Clearly, the relative importance of the fitness of the learned hyperplane to these two training datasets needs to be controlled so that they can be appropriately combined.

For SVM, it is easier to measure the unfitness of the model to these training datasets. In particular, one can use the sum of the weighted slack over the dataset to measure the unfitness of the learned SVM model to these two training sets. Let the first m training examples be the labeled examples, and the rest be the pseudo examples, the objective function

of primal problem is given:

$$\langle w \cdot w \rangle + C \sum_{i=1}^{m} \xi_i + \eta C \sum_{i=m+1}^{n} g(v_i)\xi_i$$

Here the functionality of the parameter C is the same as the standard SVM to control the balance between the model complexity and training error. The parameter η is used to control the relative importance of the evidence from these two different datasets. Intuitively, one wants to have a relative bigger η when the number of the true labeled examples is small. When the number of true training examples increases, one typically wants to reduce the influence of the "pseudo training dataset" since the evidence embedded in the true training dataset is of better quality. Because we do not have access to the exact value of ξ_i before training, in practice, we approximate the unfitness to these two datasets by mC and $\sum_{m+1}^{n} \eta C g(v_i)$.

The solution of WMSVM on the concatenated dataset depends on a number of issues. The most important factor is v_i, the confidence value of each test example. The influence of margin/slack normalization function f/g is highly dependent on v_i. Since the value of v_i is just a rough estimation in this particular application, and there is no theoretical justification for the more complex function form, we choose to use the simplest function form for both f and g. Precisely, we use $f(x) = 1/x$ and $g(x) = x$ in this paper. Experiments show this particular choice of function form is appropriate.

6. EXPERIMENTS

To test the effectiveness of the proposed way of incorporating prior knowledge, we compare the performance of WMSVM with prior knowledge against SVM without such knowledge, particularly when the true labeled dataset is small. We use text categorization as a running example as prior knowledge is readily available in this important application.

We conduct all our experiments on two standard text categorization datasets: Reuters-21578 and OHSUMED. Reuters-21578 was compiled by David Lewis from Reuters newswire. The ModApte split we used has 90 categories. After removing all numbers, stop words and low frequency terms, there are about 10,000 unique stemmed terms left. OHSUMED is a subset of Medline records collected by William Hersh [6]. Out of 50,216 documents that have abstract in year 1991, the first 2/3 is used in training and the rest is used in testing. This corresponds to the same split used in [11]. After removing all numbers, stop words and low frequency terms, there are about 26,000 unique stemmed terms left. Since we are studying the performance of the linear classifier under different data representation, we split the classification problem into multiple binary classification problems in a one-versus-rest fashion. The 10 most frequent categories are used for both datasets. No feature selection is done, and a modification of libSVM [2] based on description described in section 4 is used to train WMSVM.

6.1 Constructing the Prior Model

The proposed approach permits prior knowledge of any kind, as long as it provides estimates, however rough, of the confidence values of *some* test examples belonging to the class of interest.

For each category, one of the authors, with access to the training data (not the testing data that will later form the "pseudo training dataset"), comes up with a short list of indicative keywords. Ideally, one could come up with such a short list with only an appropriate description of the category, but such description is not available for the datasets we use. These keywords are produced through a rather subjective process based on only the general understanding of what the categories are about. The idea of using keywords to capture the information needs is considered to be practical in many scenarios. For example, the name for each category can be used as keyword for OHSUMED with little exception(ignoring the common words such as "disease"). Keywords used for both dataset are listed in table 1, 2.

Table 1: Keywords used for 10 most frequent categories in Reuters

category	keywords
earn	cents(cts), net, profit, quarter(qtr), revenue(rev), share(shr)
acq	acquire, acquisition, company, merger, stake
money-fx	bank, currency, dollar, money
grain	agriculture, corn, crop, grain, wheat, usda
crude	barrel, crude, oil, opec, petroleum
trade	deficit, import, surplus, tariff, trade
interest	bank, money, lend, rate
wheat	wheat
ship	port, ship, tanker, vessel, warship
corn	corn

We next use these keywords to build a very simple model to predict the confidence value of an instance. To see how the proposed approach performs in practice, we used a model that is, while far from perfect, a natural solution given the very limited information we processed. Given a document x, the confidence value of x belonging to the class of interest is simply: $|x|_w/|c|_w$, where $|x|_w$ denotes the number of keywords appearing in documents x, and $|c|_w$ the total number of keywords that describe category c. To make sure that SVM training is numerically stable, a document will be ignored if it does not contain at least one of keywords that characterize the category of interest. This suggests the prior model we use has an incomplete coverage, it is thus significantly different from the prior model used in [13]. We think such a partial coverage prior model is a closer match to the fact that the keywords have only limited coverage, particularly when the category is broad (thus there are many indicative keywords). Inducing a full coverage prior model like [13] from the keywords with limited coverage, in principle, will introduce noise.

Nine true datasets are created by taking the first m_i examples, where $m_i = 16 * 2^i, i \in [0, 8]$. We then train standard SVM classifiers on these true datasets, WMSVM classifiers on the concatenations of these true datasets and the pseudo datasets. The pseudo datasets are always generated by applying the prior model on the testing sets. The test examples are then used to measure their performance. No experiments were conducted to determine whether better performance could be achieved with wiser choice of C (for SVM), we set it to 1.0 for all experiments.

We set the parameter η using the heuristic formula $400/m$, where m is the number of true labeled training examples

Table 2: Keywords used for 10 most frequent categories in Ohsumed

category	keywords
coronary disease	coronary
myocardial-infarction	myocardial, infarction
heart failure, congestive	congestive, failure
arrhythmia	arrhythmia
heart defects, congential	fontan, congential
heart disease	cardiac, heart
tachycardia	tachycardia
angina pectoris	angina, pectoris
heart arrest	arrest
coronary arteriosclerosis	arteriosclerosis, arteriosclerotic

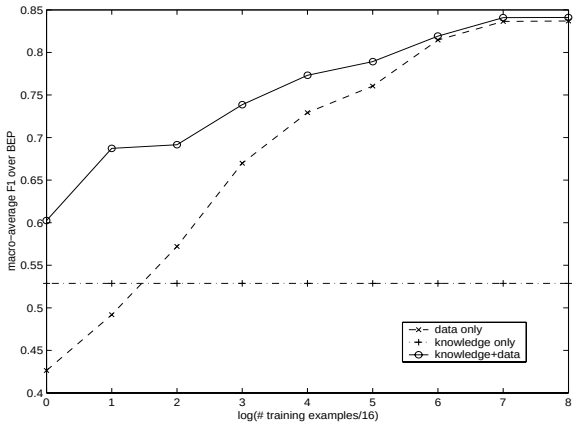

Figure 3: Comparison of macro-average Break-Even-Point using prior knowledge and data separately or together on the Reuters 25718, 10 most frequent categories, measured as a log function of the number of training examples divided by 16, macro-average F1 over BEP.

used. Making η an inverse function of m is based on two common understanding: first, SVM performs very well when there are enough labeled data; second, SVM is sensitive to label noise [19]. This inverse function form makes sure that when there is more data, the noise introduced by noisy prior model is small. The value 400 is picked to give enough weight to prior model when there are only 32 examples on Reuters dataset. We did not study the influence of the different function forms, but the performance of the WMSVM with prior knowledge seems to be robust in term of the coefficient value in the heuristic formula $400/m$ as shown in table 3.

Figures 3,4 report these experiments. They compare the performance among the prior model, standard SVM classifiers, and these WMSVM classifiers when the size of the true dataset is increasing. For OHSUMED, we report performance in micro-average F1 over Break Even Point (BEP), a commonly used measure in text categorization community [18]. For Reuters dataset, to stay comparable with that of [8], we report performance in macro-average F1 over BEP instead. It is clear that combining prior knowledge with training examples can dramatically improve the classification performance, particularly when the training dataset is small. The performance of WMSVM with the prior knowledge on Reuters is comparable to that of transductive SVM [8], but the training time is much less as only one iteration of SVM training is needed. Usually the performance of SVM increases when one adds more labeled examples. But if the newly added examples are all negative, it is possible that the performance of SVM actually decreases, as shown in Figure 4. Note that the influence of prior knowledge on the final performance is decreasing when the number of true labeled examples is increasing. This is due to the particular function form of parameter $\eta(400/m)$. But one can also understand this phenomenon by noting that the more labeled examples, drawn from an independently and identically distribution, the less the additional information one might have in prior knowledge.

η	800	400	200	100
WMSVM	0.671	0.681	0.691	0.680

Table 3: Macro-average F1 over different η values on top 10 most frequent categories from Reuters dataset, WMSVM on 32 true labeled training examples along with pseudo training dataset.

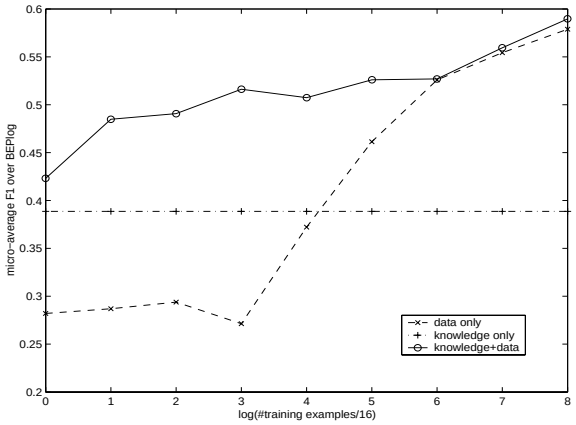

Figure 4: Comparison of macro-average Break-Even-Point using prior knowledge and data separately or together on the OHSUMED, 10 most frequent categories, measured as a log function of the number of training examples divided by 16, micro-average F1 over BEP.

7. CONCLUSION

For statistical learning methods like SVM, using human prior knowledge can in principle reduce the need for larger training dataset. Since weak predictors that estimate the conditional in-class probabilities can be derived from most human knowledge, the ability to incorporate prior knowledge through weak predictors thus has great practical implications. In this paper, we proposed a generalization of the standard SVM: Weighted Margin SVM, which can handle the imperfectly labeled dataset. SMO algorithm is extended to handle its training problem. We then introduced a two-step approach to incorporate fuzzy prior knowledge using the WMSVM. The empirical study of our approach is conducted through text classification experiments on standard

datasets. Preliminary results demonstrates its effectiveness in reducing the number of the labeled training examples needed. Furthermore, WMSVM is a fairly generic machine learning method and incorporating fuzzy prior knowledge is just one of its many possible applications. For example, WMSVM can be readily used in distributed learning with heterogeneous truthing. Further research directions include studies on the robustness of incorporating prior knowledge with respect to different quality of rough predication rules. More generally, how to combine the evidence from different sources and in different forms for effective modeling of data is an interesting future research direction.

8. ACKNOWLEDGMENTS

We want thank Dr. Zhixin Shi for his valuable comments. We also want to thank the anonymous reviewers for their feedback.

9. REFERENCES

[1] K. Bennett and A. Demiriz. Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems 11*, 1998.

[2] C. Chang and C. Lin. LIBSVM: a library for support vector machines (version 2.3), 2001.

[3] G. Fung and O. Mangasarian. Semi-supervised support vector machines for unlabeled data classification. *Optimization Methods and Software*, 15, 2001.

[4] G. Fung, O. L. Mangasarian, and J. Shavlik. Knowledge-based support vector machine classifiers. In *Data Mining Institute Technical Report 01-09*, Nov 2001.

[5] G. H. Golub and C. F. V. Loan. *Matrix Computation*. Johns Hopkins Univ Press, 1996.

[6] W. R. Hersh, C. Buckley, T. J. Leone, and D. H. Hickam. Ohsumed: An interactive retrieval evaluation and new large test collection for research, 1994.

[7] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[8] T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. 16th International Conf. on Machine Learning*, pages 200–209. Morgan Kaufmann, San Francisco, CA, 1999.

[9] T. Joachims. *Learning To Classify Text Using Support Vector Machines*. Kluwer Academic Publishers, Boston, 2002.

[10] S. Keerthi, S. Shevade, C. Bhattacharyya, and K. Murthy. Improvements to platt's smo algorithm for svm classifier design, 1999.

[11] W. Lam and C. Ho. Using a generalized instance set for automatic text categorization. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 81–89, Melbourne, AU, 1998. ACM Press, New York, US.

[12] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in kernel methods - support vector learning*. MIT Press, 1998.

[13] R. Schapire, M. Rochery, M. Rahim, and N. Gupta. Incorporating prior knowledge into boosting. In *Proceedings of the Nineteenth International Conference In Machine Learning*, 2002.

[14] B. Schölkopf, P. Simard, A. Smola, and V. Vapnik. Prior knowledge in support vector kernels. In B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in kernel methods - support vector learning*. MIT Press, 1998.

[15] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In P. Langley, editor, *Proceedings of ICML-00, 17th International Conference on Machine Learning*, pages 999–1006, Stanford, US, 2000. Morgan Kaufmann Publishers, San Francisco, US.

[16] V. N. Vapnik. *Statistical learning theory*. John Wiley & Sons, New York, NY, 1998.

[17] V. N. Vapnik. *The nature of statistical learning theory, 2nd Edition*. Springer Verlag, Heidelberg, DE, 1999.

[18] Y. Yang and X. Liu. A re-examination of text categorization methods. In M. A. Hearst, F. Gey, and R. Tong, editors, *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pages 42–49, Berkeley, US, 1999. ACM Press, New York, US.

[19] J. Zhang and Y. Yang. Robustness of regularized linear classification methods in text categorization. In *Proceedings of SIGIR-2003, 26st ACM International Conference on Research and Development in Information Retrieval*. ACM Press, 2003.