New ν-Support Vector Machines and their Sequential Minimal Optimization

Xiaoyun Wu XWU@CEDAR.BUFFALO.EDU
Rohini Srihari ROHINI@CEDAR.BUFFALO.EDU
Dept. of Computer Science and Engineering, State University of New York at Buffalo, Amherst, NY 14228

Abstract

Although the ν-Support Vector Machine, ν-SVM, (Schölkopf et al., 2000) has the advantage of using a single parameter ν to control both the number of support vectors and the fraction of margin errors, there are two issues that prevent it from being used in many real world applications. First, unlike the C-SVM that allows asymmetric misclassification cost, ν-SVM uses a symmetric misclassification cost. While lower error rate is promoted by this symmetric misclassification cost, it is not always the preferred measure in many applications. Second, the additional constraint from ν-SVM makes its training more difficult. Sequential Minimal Optimization (SMO) algorithms that are very easy to implement and scalable to very large problems do not exist in a good form for ν-SVM. In this paper, we proposed two new ν-SVM formulations. These formulations introduce means to control the misclassification cost ratio between false positives and false negative, while preserving the intuitive parameter ν. We also propose a SMO algorithm for the ν-SVM classification problem. Experiments show that our new ν-SVM formulation is effective in incorporating asymmetric misclassification cost, and the SMO algorithm for ν-SVM is comparable in speed to that for C-SVM.

1. Introduction

The performance of hard margin Support Vector Machine (SVM) (Vapnik, 1999) is vulnerable to any outliers in the training data. The C-style soft margin Support Vector Machine (C-SVM) was proposed by Cortes and Vapnik (Cortes & Vapnik, 1995) to address this robustness issue. A slack variable ξ_i that measures how much a training example fails to have a given functional margin from the decision hyperplane (w, b) is introduced, and $C\sum_{i=1}^{n}||\xi_i||_p$ is added to primal cost function to allow for the possibility of having a training error. This C-SVM formulation has since been proven to be successful in many practical pattern recognition problems including handwritten character recognition (Vapnik, 1999) and text classification (Joachims, 1998; Joachims, 1999; Lewis, 2001). The parameter C determines the trade-off between two conflicting goals: minimizing the training error and maximizing the margin. Unfortunately, C is a rather unintuitive parameter that we have no a priori way to select, therefore time consuming methods like cross-validation or leave-one-out are normally used.

A ν-SVM formulation has been proposed by Schölkopf et al (Schölkopf et al., 2000) to address this problem. In this formulation, instead of C, a parameter ν is used to control the trade-off between the larger margin and smaller training error. The parameter ν is an upper bound on the fraction of training margin errors and lower bound on the fraction of support vectors. With probability approaching 1, asymptotically, ν equals both the fraction of support vectors and fraction of training margin errors. A training margin error occurs when a training example fails to have the specified functional margin (i.e. its corresponding slack variable is not zero).

Although the ν-SVM has the advantage of having an intuitive parameter to tune, two problems prevent it from being applied to many practical problems. First, ν-SVM formulation uses a uniform misclassification cost. While uniform misclassification cost

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

results in a minimization of error rate, lower error rate is not always the most important goal in many cases. There are also applications where the uniform misclassification cost doesn't make sense, for example, spam filtering. This problem is first touched upon in (Chew et al., 2001) with introduction of separate misclassification cost C_+, C_- along with ν. We will show only one additional parameter is needed. Second, the dual problem of ν-SVM is more complicated than that of the C-SVM. Due to the additional constraint in its dual form, those large scale SVM training algorithms (namely decomposition method like SVM^{light} and SMO) cannot be used on ν-SVM training. The lack of efficient training algorithm on large data sets make it impractical to be applied to many real world problems.

The Sequential Minimal Optimization (SMO) algorithm was first proposed by Platt (Platt, 1998), and later enhanced by Keerthi (Keerthi et al., 1999; Shevade et al., 2000). SMO is a special case of the decomposition method and it optimize a minimal subset of just two samples at each iteration. Since for a working set of only two, the optimization problem can be solved analytically, there is no need to invoke a quadratic optimizer. The working set selection method and KKT condition testing method are very efficient. Because of this, and also the highly efficient working set selection and KKT condition testing method detailed in Platt&Keerthi's widely available pseudo code, SMO is the easiest training algorithm to implement. Its performance is comparable to that of other decomposition algorithms like SVM^{light} (Platt, 1998). Recently Chang et al. (Chang & Lin, 2001) proposed a SVM^{light} like decomposition algorithm for ν-SVM training. While theoretically decomposition method subsumes SMO, we feel it is easy to implement SMO based on Platt&Keerthi's description.

In this paper we present two new ν-SVM formulations that allow us to control the ratio of misclassification cost of false positives and false negatives. While (i) $\mu\nu$-SVM formulation introduces a parameter μ to the ratio of misclassification cost directly, (ii) $\nu_+\nu_-$-SVM does that by using separate ν parameters for positive and negative examples. The connections between our formulations and the original ν-SVM are provided. We also modified sequential minimal optimization (SMO) algorithm for $\mu\nu$-SVM training. Text classification is used as a running example to demonstrate that by allowing the asymmetric misclassification cost, different performance measures other than error rate, such as precision and recall, can be optimized. Experiments show our SMO for $\mu\nu$-SVM training is comparable in speed to SMO for C-SVM.

The rest of the paper is structured as following. In section 2, we introduce our new ν-SVM formulations, their connection to the original C-SVM/ν-SVM formulation, the dual problem and associated Karush-Kuhn-Tucker (KKT) conditions. The proposed SMO algorithm for ν-SVM will be detailed in section 3 and the computational comparison will be provided in section 4.

2. New ν-Support Vector Machines

2.1. New ν-SVM formulations

In the ν-SVM formulation, a uniform misclassification cost $\frac{1}{m}$ is used. In practice, this is not always desirable. First, there are many applications where symmetric misclassification doesn't make sense. For example, in spam filtering, one typically can tolerate spam being labeled as legitimate email to some degree, but not legitimate email being classified as spam. To model this, usually one associates a very high penalty to the case where email is labeled as spam in the cost function. Furthermore, the uniform misclassification cost minimizes the overall error rate. Applications can put their emphasis on different performance measurements, such as precision, recall and F1 measures typically used in text classification. When there is imbalance in the training set, which is the case when you try to convert a multi-class text categorization problem into multiple two-class text categorization problems, lower error rate may not guarantee preferred performance based on different measures. In order to work with these applications, we propose $\mu\nu$-SVM machine with an extra parameter μ to control the ratio of asymmetric misclassification cost.

Given a set of vectors $(x_1, ..., x_m)$, along with their corresponding labels $(y_1, ..., y_m)$ where $y_i \in \{+1, -1\}$, $\mu\nu$-SVM classifier defines a hyperplane (w, b) in mapped feature space that separates the training data by a margin ρ. The primal problem for $\mu\nu$-SVM formulation is defined by the following proposition.

Proposition 1 *Given a training sample set*

$$S = ((x_1, y_1), \cdots, (x_m, y_m))$$

where $y_i \in \{+1, -1\}$, the hyperplane (w, b) that *solves the following optimization problem*

$$\begin{aligned} \textit{minimize} \quad &: \quad \langle w \cdot w \rangle - \nu\rho + \sum_{i=1}^{m} c_{y_i}\xi_i \\ \textit{subject to} \quad &: \quad y_i(\langle w \cdot x_i \rangle + b) \geq \rho - \xi_i \\ & \quad\quad \xi_i \geq 0, i = 1, \cdots, m \end{aligned}$$

realizes the soft margin hyperplane with functional margin ρ defined by $\mu\nu$-SVM. Here $c_{+1} = \frac{\mu}{\mu m_+ + m_-}$ and $c_{-1} = \frac{1}{\mu m_+ + m_-}$, m_+, m_- denote the number of positive and negative examples. The μ is the ratio of misclassification cost between the false negatives and false positives.

There are several differences between our $\mu\nu$-SVM and ν-SVM formulation. First, instead of the uniform misclassification cost $\frac{1}{m}$, the asymmetric misclassification costs c_{+1}, c_{-1} are used, and the ratio between c_{+1}, c_{-1} is μ. The specific value of c_{+1}, c_{-1} are also due to the constraint $m_+c_{+1} + m_-c_{-1} = 1$. This constraint comes from the observation that ν should take its upper bound value 1.0 when all the support vectors take their upper bound values. Second, the $\rho > 0$ constraint disappeared. This is due to Crisp (Crisp & Burges, 1999), who shows that in original formulation, the $\rho > 0$ constraint is redundant. Discarding this constraint can actually lead to a simpler dual problem. As with ν-SVM formulation, parameter ν replaces the parameter C to control the trade-off between the larger margin and smaller training error. There is also an additional variable ρ to be optimized. The variable ρ is the functional margin of training set to hyperplane defined by (w, b). The parameter ν is used to control the importance of functional margin in the cost function: larger ν results in bigger ρ. More formally, the significance of ν is described in the following proposition.

Proposition 2 *Suppose we run $\mu\nu$-SVM on some data with the result that $\rho > 0$. Let $\nu_+ = \frac{\nu(\mu m_+ + m_-)}{\mu m_+}$, and $\nu_- = \frac{\nu(\mu m_+ + m_-)}{m_-}$, Then*
a. ν_+ (ν_-) is an upper bound on the fraction of positive (negative) margin errors. Margin error occurs where $\xi_i > 0$.
b. ν_+ (ν_-) is an lower bound on the fraction of positive (negative) support vectors.
c. With probability 1, asymptotically, ν_+ and ν_- equals both fractions.

The proof is similar to that of proposition 7.5 on ν property in (Schölkopf & Smola, 2002). Note that $\frac{\nu_+}{\nu_-} = \frac{m_-}{\mu m_+}$, this suggests that fixing the ratio of misclassification cost is equivalent to fixing ν_+/ν_-. Thus, there is an equivalent formulation that use ν_+,ν_- instead of μ, ν.

Proposition 3 *Given a training sample set*

$$S = ((x_1, y_1), \cdots, (x_m, y_m))$$

where $y_i \in \{+1, -1\}$, the hyperplane (w, b) that solves the following optimization problem

$$\begin{aligned} \textit{minimize} \quad &: \quad \langle w \cdot w \rangle - \nu_+\nu_-\rho + \sum_{i=1}^{m} c_{y_i}\xi_i \\ \textit{subject to} \quad &: \quad y_i(\langle w \cdot x_i \rangle + b) \geq \rho - \xi_i \\ & \quad\quad \xi_i \geq 0, i = 1, \cdots, m \end{aligned}$$

realizes the soft margin hyperplane with functional margin ρ defined by $\nu_+\nu_-$-SVM, where $c_{+1} = \frac{\nu_-}{m_+}$ and $c_{-1} = \frac{\nu_+}{m_-}$. Here m_+, m_- denote the number of positive and negative examples, and $\nu_+ \in (0, 1]$, $\nu_- \in (0, 1]$ are the parameters used to control the fraction of marginal training errors and fraction of support vectors for positive and negative training examples respectively.

2.2. Properties of these new ν-SVMs

Both of these two new formulations can be considered as a generalization of ν-SVM, as $\mu = 1$ reduces the $\mu\nu$-SVM into ν-SVM, $\nu_+m_+ = \nu_-m_-$ reduces the $\nu_+\nu_-$-SVM into ν-SVM. These new formulations also reveals some interesting characteristics of the solution to ν-SVM.

Proposition 4 *Suppose we run ν-SVM on some data with the result that $\rho > 0$. With probability 1, asymptotically, m_-/m_+ equals the ratio between fraction of training margin errors in positive and negative examples; it also equals the ratio between fraction of support vectors in positive and negative examples.*

Proof. Let ν-SVM as specialization of $\nu_+\nu_-$-SVM. Notice that $\nu_+ = \frac{\nu(m_+ + m_-)}{m_+}$, and $\nu_- = \frac{\nu(m_+ + m_-)}{m_-}$ when $\mu = 1$. This result follows naturally from Proposition 2.

This result further explains why ν-SVM might have trouble in dealing with unbalanced training data set. In a SVM classifier trained by ν-SVM, the fraction of training marginal errors in the category with less examples has a tendency to be bigger than that of the category with more examples. The same is true for the fraction of support vectors. This clearly indicates that more attention has been paid to the category with more examples, and the final decision hyperplane is more likely to make decision in favor of the category with more examples. This result also points out $\mu = \frac{m_-}{m_+}$ will result in a SVM classifier that pays equal attention to both categories regardless of the number of examples they have.

Like the ν-SVM, there is a valid range for parameter ν in $\mu\nu$-SVM.

Proposition 5 *Solution to $\mu\nu$-SVM with $\rho > 0$ exists if and only if $\nu \leq \nu_{max}$, where*

$$\nu_{max} = \min(\frac{2\mu m_+}{\mu m_+ + m_-}, \frac{2m_-}{\mu m_+ + m_-}).$$

With m_+, m_- denotes the number of positive and negative examples.

Proof. Notice $\nu_+ = \frac{\nu(\mu m_+ + m_-)}{\mu m_+}$, and $\nu_- = \frac{\nu(\mu m_+ + m_-)}{m_-}$, and also $\nu_+ \in (0, 1]$, $\nu_- \in (0, 1]$. The result is readily followed. The range of ν in the ν-SVM, $\nu \leq 2.0 \min(m_+, m_1)/m$), can be considered as a special case when $\mu = 1$, since we also have $m = m_+ + m_-$. Another special case occurs where $\mu = m_-/m_+$: the range then became $\nu \leq 1.0$. The comparison illustrates another potential problem of original ν-SVM formulation when it comes to unbalanced data set: the range of ν is so small that its ability to control these two different goals (minimizing the training error and maximizing the margin) are greatly reduced.

There is a connection between our $\mu\nu$-SVM formulation and C-SVM with asymmetric misclassification cost. This connection can be considered as a generalization of the connection between original ν-SVM formulation and C-SVM with uniform misclassification cost.

Proposition 6 *If training of $\mu\nu$-SVM leads to a solution with $\rho > 0$, then corresponding C-SVM, where $C_{+1} = \frac{\mu}{\rho(\mu m_+ + m_-)}$ and $C_{-1} = \frac{1}{\rho(\mu m_+ + m_-)}$, leads to the same decision function. Here the primal of C-SVM is defined as:*

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

Proof. After we get the solution to the primal problem of $\mu\nu$-SVM in proposition 1, we fix ρ to minimize the remaining variable α_i, b, ξ_i, the result doesn't change. Hence the solution solves the C-SVM problem with $C'_{+1} = \frac{\mu}{\mu m_+ + m_-}$ and $C'_{-1} = \frac{1}{\mu m_+ + m_-}$. Rescale the variables w, b, and ξ by a factor of $1/\rho$, the result follows.

While the last result points out the way to derive the equivalent $\mu\nu$-SVM from the corresponding C-SVM, this following proposition shows how to derive the equivalent C-SVM from the corresponding $\mu\nu$-SVM.

Proposition 7 *If the training of C-SVM with asymmetric misclassification cost C_{+1}, C_{-1} leads to a solution α_i, then corresponding $\mu\nu$-SVM with $\nu = \sum \alpha_i/(C_{+1}m_+ + C_{-1}m_-)$ and $\mu = C_{+1}/C_{-1}$ leads to same decision function.*

Proof. Let $w' = w\rho$, $b' = b\rho$, $\xi' = \xi\rho$, the primal of $\mu\nu$-SVM became:

$$\begin{aligned} \text{minimize} \quad &: \quad \langle w' \cdot w' \rangle - \nu/\rho + \sum_{i=1}^{m} c_{y_i} \xi'_i/\rho \\ \text{subject to} \quad &: \quad y_i(\langle w' \cdot x_i \rangle + b') \geq 1 - \xi'_i \end{aligned}$$

To make this equivalent to the C-SVM problem, we prefix ν, as let $\mu = C_{+1}/C_{-1}$, $1/\rho = C_{-1}m_- + C_{+1}m_+$. We thus have $C_{+1} = \frac{\mu}{\rho(\mu m_+ + m_-)}$ and $C_{-1} = \frac{1}{\rho(\mu m_+ + m_-)}$. With constraint $\sum_{i=1}^{m} \alpha_i = \frac{\nu}{\rho}$ (from next proposition), the result follows.

2.3. Dual problem and KKT conditions

The derivation of the ν-SVM is similar to that of C-SVM. The dual problem is defined as following.

Proposition 8 *Given a training sample set*

$$S = ((x_1, y_1), \cdots, (x_m, y_m))$$

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

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

then the weight vector $w^* = \sum_{i=1}^{m} y_i \alpha_i x_i$ realizes the maximal weighted soft margin hyperplane defined by $\mu\nu$-SVM. Where $c_{+1} = \frac{\mu}{\mu m_+ + m_-}$ and $c_{-1} = \frac{1}{\mu m_+ + m_-}$.

Compared with dual problem of C-SVM, there are two differences. First, linear term $\sum_{i=1}^{m}\alpha_i$ no longer appears in the cost function. Second, there is an additional constraint which makes training of ν-SVM more complicated. Despite these differences, strong connection between ν-SVM and C-SVM has been revealed in a number of studies (Schölkopf & Smola, 2002; Crisp & Burges, 1999; Chang & Lin, 2001).

The KKT conditions in this case are therefore as follows:

$$\alpha_i[y_i(\langle x_i \cdot w\rangle + b) - \rho + \xi_i] = 0, \qquad i = 1, \cdots, n$$
$$\xi_i(\alpha_i - c_{y_i}) = 0, \qquad i = 1, \cdots, n$$

This implies that samples with non-zero slack can only occur when $\alpha_i = c_{y_i}$. Samples for which $c_{y_i} > \alpha_i > 0$ have an effective weighted margin of $\rho/\|w\|$ from the hyperplane (w^*, b^*). The threshold b^* and margin parameter ρ can be computed as

$$b^* = -\frac{\max_{y_i=-1}(\langle w^* \cdot x_i\rangle + \min_{y_j=+1}\langle w^* \cdot x_j\rangle)}{2}$$
$$\rho = -\frac{\max_{y_i=-1}(\langle w^* \cdot x_i\rangle - \min_{y_j=+1}\langle w^* \cdot x_j\rangle)}{2}$$

3. Modification of the SMO for $\mu\nu$-SVM

SMO can be easily adapted to one class ν-SVM (Schölkopf & Smola, 2002) since there is only one equality constraint in its dual problem and one variable (functional margin) under optimization in its KKT condition testing. However there are two problems when one tries to use the SMO to train a two class $\mu\nu$-SVM. First, there is an additional equality constraint. Second, with both functional margin and bias subject to optimization, how to test KKT condition for each pattern efficiently becomes a problem. We will address these two issues in the following two subsections.

3.1. Observe two equality constraints

The dual problem of C-SVM has one linear equality constraint: $\sum_{i=1}^{m} y_i\alpha_i = 0$. This suggests that for a working set of size 2, if at each step, we have $y_1\alpha_1^{new} + y_2\alpha_2^{new} = y_1\alpha_1^{old} + y_2\alpha_2^{old}$, we will not violate the equality constraint. This is not true anymore for the ν-SVM when $y_1 \neq y_2$, the equality constraint $\sum_{i=0}^{m}\alpha_i = \nu$ will be violated. To reveal the structure of these two equality constraints, we rewrite them as following.

$$\sum_{y_i=+1}\alpha_i = \frac{\nu}{2}$$
$$\sum_{y_i=-1}\alpha_i = \frac{\nu}{2}$$

This different view of the equality constraints reveals that the Lagrange multipliers for positive examples should be updated with that of the positive examples, and the Lagrange multipliers for negative examples should be updated with that of the negative examples. The smallest set of patterns that can be optimized together is still 2.

3.2. KKT conditions Test

We turned the additional equality constraint into a constraint on the way the working sets are formed. Now we need to derive an efficient way to test KKT condition for each pattern. As Keerthi pointed out (Keerthi et al., 1999), a better KKT condition testing method not only helps in the pathological cases, but also speeds up the convergence for more general cases.

We start the discussion by introducing some notations. Let $G_i = \sum_{j=1}^{m}\alpha_j y_j K(x_i, x_j)$, $E_i = G_i - b + y_i$. The KKT condition for a pattern x_i can be reformulated by considering these different cases.

$$\alpha_i = 0 \;\Rightarrow\; y_i(G_i - b - y_i\rho) > 0 \qquad (1)$$
$$0 < \alpha_i < c_{y_i} \;\Rightarrow\; y_i(G_i - b - y_i\rho) = 0 \qquad (2)$$
$$\alpha_i = c_{y_i} \;\Rightarrow\; y_i(G_i - b - y_i\rho) < 0 \qquad (3)$$

Define $bpr = b+\rho$, and $bnr = b-\rho$, and define index sets based on their α values and target label:

$$I_0 \equiv \{i : 0 < \alpha_i < c_{y_i}\}$$
$$I_1 \equiv \{i : y_i = +1, \alpha_i = 0\}$$
$$I_2 \equiv \{i : y_i = -1, \alpha_i = c_{y_i}\}$$
$$I_3 \equiv \{i : y_i = +1, \alpha_i = 0\}$$
$$I_4 \equiv \{i : y_i = -1, \alpha_i = c_{y_i}\}$$

Since the examples from same class will be optimized together, the KKT conditions can be rewritten as

$$G_i \geq bnr, i \in I_0 \cup I_1$$
$$G_i \leq bnr, i \in I_0 \cup I_3$$

for $y_i = +1$, and

$$G_i \geq bpr, i \in I_0 \cup I_2$$
$$G_i \leq bpr, i \in I_0 \cup I_4$$

for $y_i = -1$. Define

$$\begin{aligned} bnr_{up} &= \min(G_i : i \in I_0 \cup I_1) \\ bnr_{low} &= \max(G_i : i \in I_0 \cup I_3) \end{aligned}$$

for $y_i = +1$, and

$$\begin{aligned} bpr_{up} &= \min(G_i : i \in I_0 \cup I_2) \\ bpr_{low} &= \max(G_i : i \in I_0 \cup I_4) \end{aligned}$$

for $y_i = -1$. The KKT condition implies $bpr_{up} \geq bpr_{low}$ and $bnr_{up} \geq bnr_{low}$. For each pattern, the KKT condition can now be tested without the knowledge of b and ρ. For example, for $y_i = +1$, the following can be used to test KKT condition.

$$\begin{aligned} G_i \geq bnr_{low} : i \in I_0 \cup I_1 \\ G_i \leq bnr_{up} : i \in I_0 \cup I_3 \end{aligned}$$

The similar results can be derived for the patterns where $y_i = -1$. Note that in the actual implementation, to ensure the numerical stability, a small tolerance parameter is introduced in each of these tests.

It can be shown that the analytical solution of working set of two is the same, despite that the cost function in the dual problem of ν-SVM is different from that of the C-SVM. At each optimization step, $|E_1 - E_2|$ can now be replaced by $|G_1 - G_2|$, thus the knowledge of b is not needed. Again, we can keep the G_i in a cache for these patterns that are not at boundary to speed up the computation.

3.3. Two Variations

In line with what Keerthi (Keerthi et al., 1999) has suggested, there are two ways of picking up the pair to optimize when the loop is over I_0 only.

Method 1. Loop through all the $i_2 \in I_0$. For each i_2, check the optimality and if violated, choose i_1 based on which index set it belongs to (This is essentially the same as Keerthi's method 1. The only difference now is that examples should be chosen from same class to ensure the equality constraint).

Method 2. Always work with the worst violating pairs for each class. More specifically, at each step, optimize one pair of positive examples and one pair of negative examples as long as there are violations in both positive and negative examples.

4. Experiments

Experiments are conducted in two separate parts to test the effectiveness of our new ν-SVM formulation in (i) dealing with an unbalanced data set and (ii) the efficiency of our SMO training algorithm.

4.1. Effectiveness of new ν-SVMs

For empirical evaluation of our new SVM formulations, we select text categorization as our running examples since most text categorization problems have unbalanced training data set and higher F1 measure instead of lower error rate are normally preferred. We used standard text classification data set: "ModApte" split of Reuters-21578 data set (Yang & Liu, 1999). The data set has a training set of 7,769 documents, and a test set of 3,019 documents, all in 90 categories. The effectiveness of C-SVM is proven by many studies (Yang & Liu, 1999; Joachims, 1998; Joachims, 1999; Lewis, 2001) to have state-of-the-art performance on the text categorization problems. Proposition 6 and proposition 7 further point out that $\mu\nu$-SVM is equivalent to C-SVM. Thus the main objective of our empirical study here is to show that our new formulations can provide some guidance on parameter selection so that at least some extensive parameter tuning procedures can be saved.

Although ν can be used to control the fraction of training errors, there are no known principled ways to predetermine ν for any given learning task. Since F1 measure emphasizes equally both recall and precision, from Proposition 2, we know $\mu = m_+/m_-$ would lead to good F1 measures since equal attention has been paid to both positive and negative examples. To test whether this is true, we fixed ν, ran experiments with two different μ values: 1.0 (corresponding to original ν-SVM), m_+/m_- (to optimizing F1 measure). Smaller ν is required for categories with less positive examples by original ν-SVM formulation, but it is also more likely to run into the over-fitting problem. In this experiment we choose 0.004. This may not be the optimal value for ν in terms of F1 measure, but we will have enough categories to work with using both μ settings, yet the over-fitting problem is not hurting the result. Since experiments show that for categories with larger number (more than 120 in this study) of positive examples, the performance differences between two μ values can be ignored, only results from 37 categories (with number of positive training examples between 16 and 120) are shown here.

Figure 1 plots the two F1 measures versus the category rank. Categories are ranked in the descending order based on the number of positive examples in the training set. It showed that asymmetric misclassification cost with $\mu = m_-/m_+$ leads to better F1 measure on most categories, and the improvement is bigger on the categories with less positive training examples. Actual F1 measure on selected categories

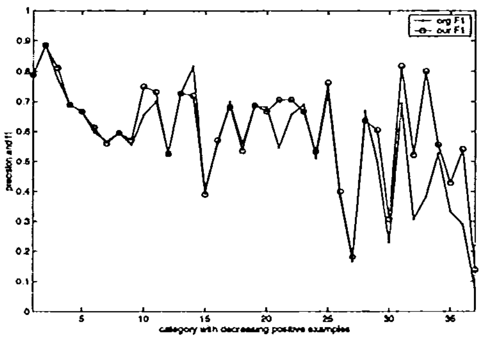

Figure 1. F1 measure versus category rank

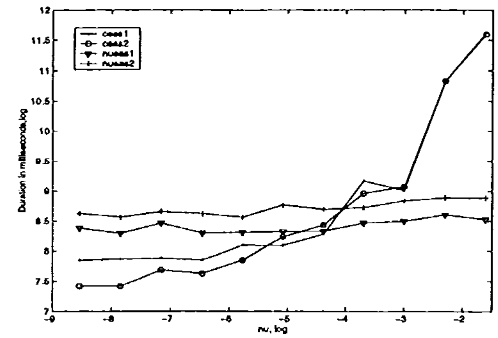

Figure 4. training time versus different ν

	rank	p/n num	org F1	our F1
coffee	1	111/7658	0.788732	0.788732
bop	6	75/7694	0.597403	0.613333
rubber	21	37/7732	0.656250	0.705882
orange	36	16/7753	0.289855	0.540541

Figure 2. F1 measures on selection categories

are shown in Figure 2. Further studies also support this observation.

4.2. Efficiencies of SMO for ν-SVM

Since previous studies have shown that SMO is one of fastest SVM training algorithms (Platt, 1998; Keerthi et al., 1999), we simply need to compare our SMO for ν-SVM (two variations: nueasl, nueas2) with SMO for C-SVM (again two variations: ceasl, ceas2). To do that, we implemented both the Keerthi improved version SMOs for C-SVM and our SMOs for ν-SVM. To make the result comparable, Proposition 6 and Proposition 7 are used to select parameters so that C-SVM and $\mu\nu$-SVM lead to the same decision function (thus, same support vectors). We studied the efficiency of our $\mu\nu$-SVM under different ν, and training set size. The data sets used in this experiments are randomly constructed from a larger USPS set. Note that the same tolerance value is used across the experiments and the training time is very much dependent on this tolerance value.

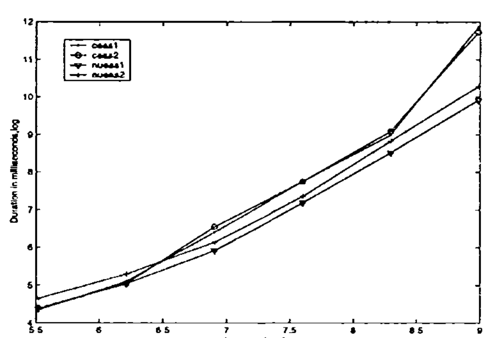

Figure 3. training time versus dataset size

Figure 3 plots training time of four SMOs implementations on six data sets, each doubled in size. It is a log-log plot of training time versus dataset size. It is clear our SMO methods are faster when the data set is large enough. This is due to the fact that SMO for ν-SVM can be considered as the sum of two separate SMO for C-SVM problem with half the size; this is also due to the empirical result that SMOs are scaling as $N^{1.9}$ (Platt, 1998). Furthermore, unlike the SMO for C-SVM, the second variation is always slower than the first variations in SMO for ν-SVM. The reason is that we have to stop optimization on the unbound αs as soon as violations disappear in either positive or negative examples. If we stop the optimization of the unbound αs only when there are no violations in both positive and negative examples, numerical experiments show that the algorithm may not converge.

Figure 4 plots the effect of different ν on the training time. Here the dataset is fixed, the training time is shown with 11 different ν values, each doubled from the previous. The experiments show again that our SMO for ν-SVM is comparable in performance to SMO for C-SVM, and variation 1 of SMO for ν-SVM outperforms variation 2 of SMO for ν-SVM. In the meantime, the training time of our SMO stays rather flat when its constraint increases from 0.000195 to 0.2; the training time of SMO for the corresponding C-SVM increased a lot when its constraint goes from

21.08 down to 0.017. This suggests that our SMO for ν-SVM is very robust in term of ν.

5. Conclusion

In this paper, we first pointed out the potential problems of the original ν-SVM formulation on the unbalanced training data set. We then proposed two new SVM formulations that allow asymmetric misclassification cost: $\mu\nu$-SVM controls ratio of misclassification cost directly, $\nu_+\nu_-$-SVM does that by using a separate ν (controls both the fraction of margin training error and fraction of support vectors) for positive and negative examples. These new formulations combine the strength from both the C-SVM and ν-SVM: they allow asymmetric misclassification cost while retaining the intuitive parameter ν. Experiments in text classification show that our new formulations not only have the capacity for better performance on unbalanced data sets, but also provide guidance on parameter selection so that better performance can be achieved without expensive parameter tuning process.

We also proposed a SMO algorithm for ν-SVM formulations. By pairing up patterns from the same class into working set of size 2, both equality constraints can be observed. By maintaining and updating four threshold/margin parameters, we are able to derive an efficient way of checking KKT condition for each pattern without the knowledge of b and ρ. Following Keerthi's suit, we proposed two variations on how the working set is picked when we focus on the patterns that are not at boundary. Numerical experiments show that our SMO algorithm for ν-SVM training is comparable to Keerthi's improved SMO algorithm for C-SVM. The introduction of these new ν-SVM formulations and the SMO algorithm for ν-SVM is expected to help researchers to adopt the ν-SVM formulations in different applications.

Acknowledgments

We would like to thank Dr. Chih-Jen Lin, Dr. Zhixin Shi, Dr. Ajay Shekhawat, and Munirathnam Srikanth for their valuable comments.

References

Chang, C.-C., & Lin, C.-J. (2001). Training ν-support vector classifiers: Theory and algorithms. *Neural Computation*, 2119–2147.

Chew, H., Bogner, R., & Lim, C. (2001). Dual nu-support vector machine with error rate and training size biasing. *International Conference on Acoustics, Speech and Signal Processing, ICASSP 2001.*

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*, 273–297.

Crisp, D. J., & Burges, C. J. (1999). A geometric interpretation of ν-svm classifiers.

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. *Proceedings of ECML-98, 10th European Conference on Machine Learning* (pp. 137–142). Chemnitz, DE: Springer Verlag, Heidelberg, DE.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. *Proc. 16th International Conf. on Machine Learning* (pp. 200–209). Morgan Kaufmann, San Francisco, CA.

Keerthi, S., Shevade, S., Bhattacharyya, C., & Murthy, K. (1999). Improvements to platt's smo algorithm for svm classifier design.

Lewis, D. (2001). Applying support vector machines to the trec-2001 batch filtering and routing tasks. *NIST Special Publication 500-250: The Tenth Text REtrieval Conference* (pp. 286–292).

Platt, J. (1998). Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods - support vector learning.* MIT Press.

Schölkopf, B., o Smola, Williamson, R., & Bartlett, P. (2000). New support vector algorithms.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: Support vector machines, regularization,optimization and beyond.* Cambridge, Massachusetts: The MIT Press.

Shevade, S. K., Keerthi, S. S., Bhattacharyya, C., & Murthy, K. R. K. (2000). Improvements to the SMO algorithm for SVM regression. *IEEE-NN, 11*, 1188.

Vapnik, V. N. (1999). *The nature of statistical learning theory, 2nd edition.* Heidelberg, DE: Springer Verlag.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. *22nd Annual International SIGIR* (pp. 42–49). Berkley.