Asymmetric Support Vector Machines: Low False-Positive Learning Under the User Tolerance

Shan-Hung Wu[†‡] Keng-Pei Lin[†] Chung-Min Chen[‡] Ming-Syan Chen[†]
[†]Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, ROC
[‡]Telcordia Applied Research Center, Taipei, Taiwan, ROC
{brandonwu, kplin}@arbor.ee.ntu.edu.tw, chungmin@research.telcordia.com,
mschen@cc.ee.ntu.edu.tw

ABSTRACT

Many practical applications of classification require the classifier to produce a very low false-positive rate. Although the Support Vector Machine (SVM) has been widely applied to these applications due to its superiority in handling high dimensional data, there are relatively little effort other than setting a threshold or changing the costs of slacks to ensure the low false-positive rate. In this paper, we propose the notion of Asymmetric Support Vector Machine (ASVM) that takes into account the false-positives and the user tolerance in its objective. Such a new objective formulation allows us to raise the confidence in predicting the positives, and therefore obtain a lower chance of false-positives. We study the effects of the parameters in ASVM objective and address some implementation issues related to the Sequential Minimal Optimization (SMO) to cope with large-scale data. An extensive simulation is conducted and shows that ASVM is able to yield either noticeable improvement in performance or reduction in training time as compared to the previous arts.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*; I.2 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Experimentation

Keywords

Support Vector Machine (SVM), Classification, Low False-Positive Learning

1. INTRODUCTION

In many real-world applications of classification, users are particularly sensitive to the wrong predictions of a certain class. For example, in spam filtering [1, 6, 16], users may overlook/delete important information if a good mail is misclassified to spam; in facial image recognition [31] and network intrusion detection [3], costly but wrong decisions may follow if a false match or alarm is fired; in computer-aided disease diagnosis [12, 33], patients may lose the golden period of treatment if their symptoms are wrongly classified as negative. A classifier must produce a very low false-positive (or negative) rate when applied to these applications.

Many efforts [1, 7, 10, 15, 19, 20, 21, 24, 25, 32] have been made upon different classifiers to reduce the false-positive rate. Although the Support Vector Machines (SVMs) have demonstrated high prediction accuracy in the literature [9, 14, 15, 30], there are relatively few studies [15, 19, 28] on further reducing the SVMs' false-positive rate. Two common techniques, the parameter tuning [11, 19] and thresholding [15, 28], are applied prior and posterior to the SVM algorithms respectively. The former adjusts the parameter (i.e., cost) of each slack variable in an SVM objective. This approach requires either time-consuming searches for the optimal combination of the costs [11] or domain-specific knowledge of the pattern contents (e.g., relations between different email categories in spam filtering [19]) based on which the learned classifier may not generalize well due to the heuristic nature in setting the costs. The latter establishes a threshold (larger than 0) based on the Receiver Operating Characteristic (ROC) curve of a testing data. Only those patterns with predicted scores higher than the threshold will be classified as positive. The false-positive rate can be lowered as the threshold increases, yet fewer patterns may be predicted as positive meanwhile. Such a technique suffers from an unwanted trade-off between minimizing the false-positive rate and maximizing the true-positive rate.

Note the objective of traditional SVMs is to maximize the margin between the positive and negative classes in order to obtain high classification performance such as accuracy or Area Under Curve (AUC). For applications sensitive to the false-positives, keeping the resultant false-positive rate under a maximal user tolerance is usually a concern prior to achieving high classification performance [32]. For example, users are unlikely to accept a spam filter capable of identifying 100% of spam but half of the spam predictions are actually good mails. There is a basic need for a new SVM that seeks high classification performance only when the false-positive rate meets the user tolerance.

In this paper we propose the Asymmetric Support Vector Machine (ASVM), a support vector learning algorithm that takes into account the false-positive rate and user tolerance in its objective formulation. ASVM is *asymmetric* in the sense that it maximizes the margin between the negative class and the *core* [5] (i.e., high confidence subset) of the positive class. Basically, the smaller the core (i.e., the higher the confidence), the less chance a false-positive may occur. Given a user tolerance, we are able to determine a proper size of the core that ensures satisfactory false-positive rate, and at the same time the class-margin is maximized to yield high

KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA.

classification performance. ASVM avoids the trade-off between the false- and true-positive rates in thresholding, and is applicable to any applications since no prior or domain-specific knowledge is required.

To the best of our knowledge, this is the first work that exploits the asymmetry in SVM's objective to control the false-positive rate. Following summarizes our contributions:

- We propose the notion of asymmetric support vector learning and formulate the ASVM objective. The asymmetry is realized by maximizing a *core-margin* in addition to the class-margin employed by traditional SVMs.
- We study the effects of the ASVM parameters in detail and observe their linkage to the empirical measure over the portion of outliers. This allows ASVM to incorporate with the prior knowledge if the fraction of noises or low confident patterns is known in advance in a dataset.
- We address some implementation issues of ASVM and propose a bi-training technique based on the Sequential Minimal Optimization (SMO) [18, 22, 28]. By this means ASVM can cope with large-scaled datasets.
- An extensive simulation is conducted based on both the synthetic and real-world datasets [2, 23]. Experimental results show that, as compared to the thresholding technique, ASVM is able to render about 6.4% improvement in AUC when a maximal user tolerance to the false-positive rate must be met, and become the best classifier in the low-false positive region along the ROC Convex Hull. On the other hand, as compared to the parameter tuning technique, ASVM is able to achieve a comparable performance but require merely an order less training time.

The rest of the paper is organized as follows. In Section 2, we review some related studies and explain the basics of SVMs. Section 3 introduces ASVM. We also look into the effect of each parameter in the ASVM objective. In section 4 we evaluate the performance of ASVM based on the simulation results. We also discuss some implementation and training issues to handle large-scale data. Section 5 concludes the paper.

2. PRELIMINARIES

In this section, we briefly review related studies, and give preliminaries of SVMs. We specify some terminologies and assumptions that will be used throughout the text.

2.1 Related Works

The naive bayes classifier [1, 24, 25] is probably the earliest method used in the low false-positive learning. Parameters of the probability model can be easily adjusted to associate the positive predictions with high confidence. Recent efforts on low false-positive learning include utility [8, 19], boosting [10], compression [7], cascaded classifiers [32], and ensemble [21]. Studies [8, 19] employ the utilities, sometimes called stratifications, to change the prior of a decision tree or costs of SVM slacks. The study [10] induces a decision tree that is able to give confidence-rated predictions by following the AdaBoost algorithm. Authors of [7] derive two compression models for the positive and negative classes respectively, and assign the label of a pattern to the class having higher compression rate. These compression models are adaptive so the false-positive rate may be controlled. Authors of the study [32] proposes a two-stage cascaded classifier. Patterns reported as positive in the first stage are further validated in the second to reduce the false-positive rate. The study [21] merges different classifiers (those submitted to TREC 2005 Spam Evaluation Track [13]) and combines their outputs using the log-odd average to achieve low false-positive rate.

In this paper we focus on the support vector learning. Following details the objective formulations of SVMs as they are relevant to our study.

2.2 Support Vector Machines

Given a sample $\mathbf{Z}_m = ((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_m, y_m))$ of m training instances drawn i.i.d. from $\mathcal{X} \times \{\pm 1\}$, where $\mathbf{x}_i \in \mathcal{X}$ denotes a pattern and $y_i \in \{\pm 1\}$ is a class label. Our goal for classification is to find a real value function f such that $\forall\,(\mathbf{x}, y) \in \mathcal{X} \times \{\pm 1\}$, $f(\mathbf{x}) \geq 0$, if $y = 1$; $f(\mathbf{x}) < 0$ otherwise. The value $f(\mathbf{x})$ is called the *decision value*.

The SVM Classifier. The Support Vector Machine (SVM) [9, 14, 30] searches a hyperplane $\{f(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$ that maximizes the margin between the two classes of training patterns. To separate the overlapped classes, $\mathbf{x}_i$ are usually mapped to a high dimensional Reproducing Kernel Hilbert Space (RKHS), $\mathcal{H}$, by a function Φ. Let $\{\langle \mathbf{w}, \Phi(\mathbf{x})\rangle + b : \Phi(\mathbf{x}) \in \mathcal{H}\}$, $\mathbf{w} \in \mathcal{H}$ and $b \in \mathbb{R}$, be a hyperplane corresponding to a linear function $f(\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x})\rangle + b$ in $\mathcal{H}$, the primal objective of SVM can be formulated as a quadratic optimization problem:

$$\arg\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i, \tag{1}$$
$$\text{subject to } y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0,$$

for all $i = 1, \cdots, m$, where ξ_i are slack variables and C is a constant denoting the cost of each slack. The above objective puts the positive training instances $(\mathbf{x}_i, 1)$ at one side of the margin $\{\langle \mathbf{w}, \Phi(\mathbf{x})\rangle + b) \geq 1 : \Phi(\mathbf{x}) \in \mathcal{H}\}$, and the negative ones $(\mathbf{x}_i, -1)$ at another side $\{\langle \mathbf{w}, \Phi(\mathbf{x})\rangle + b) \leq -1 : \Phi(\mathbf{x}) \in \mathcal{H}\}$. Instances $(\mathbf{x}_i, y_i)$ falling outside of their corresponding regions are called *outliers* and have positive penalties $\xi_i > 0$. The parameter C controls the trade-off between maximizing the margin (i.e., $2/\|\mathbf{w}\|$) and minimizing the training error (i.e., $\sum_{i=1}^{m}\xi_i$). Eq. (1) can be solved efficiently [18, 22, 28]. Obtaining $\mathbf{w}$ and b, one may predict the label of a testing pattern $\mathbf{x}'$ by using $\text{sgn}\,(f(\mathbf{x}'))$. Studies [4, 29, 30] show that the large margin can actually lead to better generalization performance in prediction. ■

One-Class SVM. There is another type of SVM [5, 26] that aims at distinguishing the regular patterns from outliers. Given a sample $\mathbf{X}_m = (\mathbf{x}_1, \cdots, \mathbf{x}_m)$ of m unlabeled patterns drawn i.i.d. from $\mathcal{X}$ with distribution D, the one-class SVM searches for the smallest ball that encloses the *support* of D. When data are mapped to an RKHS, finding the smallest ball is equivalent to searching a hyperplane that approaches the dataset as close as possible from the origin [26]. Let $\{\langle \mathbf{w}, \Phi(\mathbf{x})\rangle - \rho : \Phi(\mathbf{x}) \in \mathcal{H}\}$, $\rho \in \mathbb{R}$, be the hyperplane, the objective of one-class SVM is formulated as follows:

$$\arg\min_{\mathbf{w},\rho,\xi_i} \frac{1}{2}\|\mathbf{w}\|^2 - \rho + C\sum_{i=1}^{m}\xi_i, \tag{2}$$
$$\text{subject to } \langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle \geq \rho - \xi_i \text{ and } \xi_i \geq 0,$$

for all $i = 1, \cdots, m$. The above objective puts all instances $\mathbf{x}_i$ at the upper side of the hyperplane $\{\langle \mathbf{w}, \Phi(\mathbf{x})\rangle - \rho \geq 0 : \Phi(\mathbf{x}) \in \mathcal{H}\}$ and let the boundary $\langle \mathbf{w}, \Phi(\mathbf{x})\rangle - \rho = 0$ approach the elements of $\mathbf{X}_m$ by maximizing its margin from the origin (i.e., $\rho/\|\mathbf{w}\|$). Patterns $\mathbf{x}_i$ falling outside the region $\{\langle \mathbf{w}, \Phi(\mathbf{x})\rangle - \rho \geq 0 : \Phi(\mathbf{x}) \in \mathcal{H}\}$ are called outliers and have $\xi_i > 0$. The parameter C controls the trade-off between maximizing the margin (i.e., $\rho/\|\mathbf{w}\|$)

and minimizing the training error (i.e., $\sum_{i=1}^{m} \xi_i$). Solving Eq. (2), the function $\text{sgn}(\langle \mathbf{w}, \Phi(\mathbf{x}') \rangle - \rho)$ can be used to indicate whether a testing pattern $\mathbf{x}'$ belongs to the support or not. ■

Note that solving Eqs. (1) and (2) may involve calculating the dot product $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$ in an infinite-dimensional RKHS. Choosing a positive definite kernel k, by Mercer's theorem, one may efficiently obtain the above term using $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = k(\mathbf{x}_i, \mathbf{x}_j)$. In this paper, we restrict our discussion on the Gaussian Radial Basis Function (RBF) kernel, i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-q \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2)$, where q is a constant.

To reduce the false-positive rate of the SVM classifier, current solutions either set a threshold [15, 28] or differentiate the cost C of the slack variables [11, 19]. In thresholding [15, 28], a testing instance $\mathbf{x}'$ may be predicted as positive only if $\langle \mathbf{w}, \Phi(\mathbf{x}') \rangle + b \geq t$, where $t > 0$ is a threshold whose value is determined from the ROC curve. Clearly, the larger the value of t, the less chance a false-positive occurs in a prediction. However, fewer true-positives can be identified. The latter approach [11, 19] associates different costs C_i to different slacks ξ_i in Eq. (1). This approach is time consuming as it requires either human interaction [19] or extra searches [11] to obtain proper values of C_i.

3. ASVM

In this section, we introduce the Asymmetric Support Vector Machine (ASVM) and its rationale. We also show how ASVM can incorporate the user tolerance to achieve low false-positive learning[1].

3.1 An Asymmetric Formulation

Recall that in traditional SVM classifier, the margin are maximized between the positive and negative classes described by the training (noisy) instances. To lower the false-positive rate, we aim at searching for a better described positive class that is able to catch a higher confidence area amongst the positive training patterns. Note changing the value of C in Eq. (1) to identify more outliers from the positive patterns may not lead to a better description since by definition the outliers do not reflect the low confidence points in the underlying data distribution. One naive solution is to adopt two one-class SVMs, with different values of C in Eq. (2), to estimate proper borders of the two classes and let the decision boundary sit at the middle of the two balls. However, the balls are independent of each other. This approach does not take into account the interaction (e.g., overlap, margin) between the two classes, and the accuracy of predictions is expected to be low from the statistical learning theory [30] point of view.

We formulate the objective of ASVM as follows:

$$\arg \min_{\mathbf{w},\rho,\gamma,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 - \rho - \frac{\mu}{\tau}\gamma + \frac{1}{\tau m}\sum_{i=1}^{m} \xi_i, \tag{3}$$

$$\text{subject to } y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle - \rho) + \frac{1}{2}(y_i - 1)\gamma \geq -\xi_i,$$

$$\xi_i \geq 0, \text{ and } \gamma \geq 0,$$

for $i = 1, \cdots, m$, where μ and τ are constants. The concept of Eq. (3) is illustrated in Figure 1. Note we use the shorthand $\mathbf{x}$ for $\Phi(\mathbf{x})$. Consider two parallel hyperplanes $\{\langle \mathbf{w}, \Phi(\mathbf{x}) \rangle - \rho : \Phi(\mathbf{x}) \in \mathcal{H}\}$ and $\{\langle \mathbf{w}, \Phi(\mathbf{x}) \rangle - \rho + \gamma : \Phi(\mathbf{x}) \in \mathcal{H}\}$. The above objective puts the positive patterns at the upper side of the first plane $\{\langle \mathbf{w}, \Phi(\mathbf{x}) \rangle \geq \rho : \Phi(\mathbf{x}) \in \mathcal{H}\}$; and the negative ones at the lower side of the second $\{\langle \mathbf{w}, \Phi(\mathbf{x}) \rangle \leq \rho - \gamma : \Phi(\mathbf{x}) \in \mathcal{H}\}$. Instances falling outside their corresponding regions are called slacks and have positive penalties $\xi_i > 0$. We set $f(\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x}) \rangle - \rho + \frac{\gamma}{2}$, and predict the label of a testing instance $\mathbf{x}'$ by $\text{sgn}(f(\mathbf{x}'))$.

[1]Due to the space limitation, we focus ourselves on the two-class classification problem. The ASVM objective proposed in this article can be easily extended to the muti-class problem.

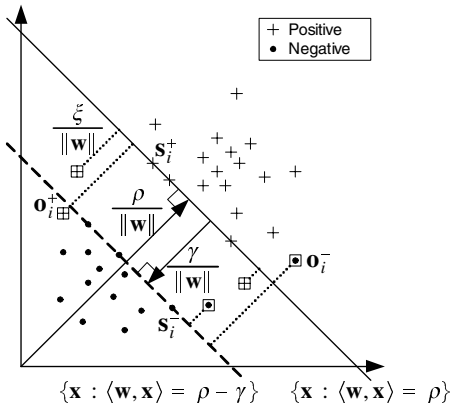


Figure 1: A logic view of ASVM in RKHS. Two margins, the core-margin ($\rho/\|\mathbf{w}\|$) and class-margin ($\gamma/\|\mathbf{w}\|$), are maximized simultaneously to allow classifying the negative class and the *core* of the positive class.

ASVM maximizes two margins, the *core-margin* (i.e., $\rho/\|\mathbf{w}\|$) and the traditional class-margin (i.e., $\gamma/\|\mathbf{w}\|$) as in SVM. The rational behind is that, by enlarging the core-margin, we are able to enclose the *core* [5] (i.e., high confidence description) of the positive class in a set $\{\Phi(\mathbf{x}) : \langle \mathbf{w}, \Phi(\mathbf{x}) \rangle \geq \rho\}$. At the same time, the class-margin is maximized between the negative class and this core to achieve high accuracy in prediction as well as its generalization. The false-positive rate is expected to be lowered when ρ increases. Note ASVM is orthogonal to most previous studies described in Section 2, and can be readily integrated with the techniques like thresholding [15, 28], utility/cost-tuning [8, 19], cascading [32], and ensemble [21].

We may transform Eq. (3) by using the Lagrangian into the following dual objective:

$$\arg\max_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j), \tag{4}$$

$$\text{subject to } \sum_{i=1}^{m} \alpha_i \geq 2\frac{\mu}{\tau} + 1, \sum_{i=1}^{m} \alpha_i y_i = 1,$$

$$\text{and } 0 \leq \alpha_i \leq \frac{1}{\tau m}.$$

The details can be found in Appendix. We will discuss how to solve this problem efficiently later.

Learning Under the User Tolerance. Consider two toy datasets shown in Figures 2(a) and (b). Figure 2(c) depicts the margin (with decision values ± 1) and the decision line $\{\Phi(\mathbf{x}) : \langle \mathbf{w}, \Phi(\mathbf{x}) \rangle + b = 0\}$ returned by the SVM classifier given parameters $C = 1$, $q = 0.5$. The parameters are found using the cross-validation [17]. We mark the slacks with squares. Figure 2(d) depicts an enclosing ball of the positive class returned by the one-class SVM with parameters $C = 0.25$, $q = 1$. The outputs of ASVM for these two datasets are shown in Figures 2(e) and (f) with parameters $\mu = 0.15$, $\tau \approx 0$, $q = 0.5$ and $\mu = 0.15$, $\tau = 0.0225$, $q = 1.5$ respectively. Comparing Figures 2(c) and (e), we can see that ASVM behaves similarly to the SVM classifier when τ is close to 0.

By increasing μ, we are able to obtain a larger margin, as depicted in Figure 2(g) ($\mu = 0.3$, $\tau \approx 0$, $q = 0.5$). The effect of μ is

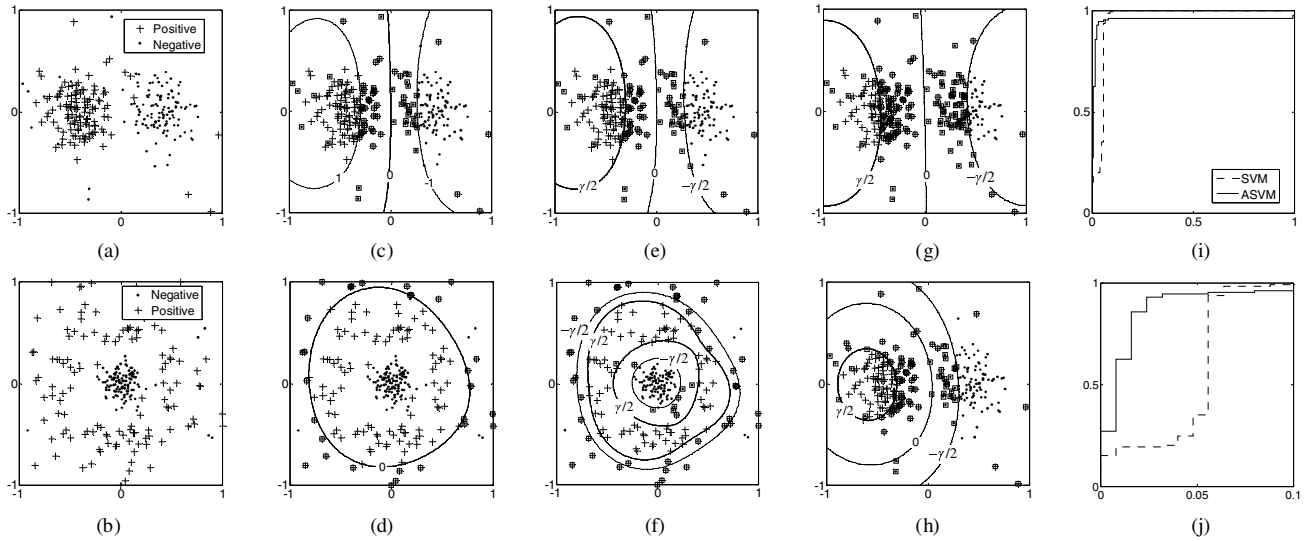


Figure 2: Toy examples. (a, b) Distributions of the the first and second datasets. (c) Decision boundary given by the SVM classifier. (d) Enclosing ball of the positive class returned by the one-class SVM. (e) Decision boundary given by the ASVM. (f) Enclosing balls returned by ASVM. (g) Increasing μ of ASVM results in a larger class-margin. (h) Obtaining a high confidence region of the positive class by increasing τ. (i) The ROCs achieved by the SVM output in (c) and the ASVM output in (h). (j) The areas under respective ROCs that meet a user tolerance 0.1 to the false-positive rate.

analogous to that of C in SVM. On the other hand, as illustrated in Figure 2(h), we are able to capture the dense region of the positive classes by increasing τ ($\mu = 0.15$, $\tau = 0.05$, $q = 0.5$) since the core-margin grows as τ increases. The dense region, unlike those captured by one-class SVM, are *antagonistic* to the negative class since by Eq. (3) it aims at excluding as many negative instances as possible. We may see this clearly by comparing Figures 2(d) and (f). Note we omit the decision line in Figure 2(f) for simplicity. The captured dense region may reasonably represent the high confidence area of the positive class due to its high density, purity (in class label), and long distance to the negative class.

ASVM is useful in the situations where a given a user tolerance t to the false-positive rate must be met. Figure 2(i) shows two typical ROC curves resulted by the SVM and ASVM classifiers in Figures 2(c) and (h) respectively. Both SVM and ASVM achieve 95% accuracy in prediction. The AUC given by ASVM is 0.95, which is slightly lower than that (0.96) achieved by SVM. However, benefiting from a better description of the positive class, ASVM can significantly reduce the chance that a false-positive occurs from an instance with high decision value. Denote t-AUC the area under the ROC curve in y-axis and t in x-axis. Suppose $t = 0.1$, Figure 2(j) depicts the performance of SVM and ASVM when the false-positive rate must be less then 0.1. In such a case, the 0.1-AUC given by ASVM is $0.86t$, which is about 56% higher than that ($0.55t$) given by SVM.

3.2 The Effects of Parameters

Although we have seen by Figure 2 the relations between the parameters, μ and τ, and the margins, the values of these parameters are still unintuitive to users. In this section, we show that the effects of μ and τ can actually be quantified in terms of the portion of outliers.

Let m^+ (resp. m^-) be the number of the positive (resp. negative) instances in $\mathbf{Z}_m$. Denote $\mathbf{s}_i^+$ (resp. $\mathbf{s}_i^-$) the positive (resp. negative) in-bound support vectors, i.e., instances $(\mathbf{x}_i, 1)$ (resp. $(\mathbf{x}_i, -1)$) having $0 < \alpha_i < \frac{1}{\tau m}$; and $\mathbf{o}_i^+$ (resp. $\mathbf{o}_i^-$) the positive (resp. negative) outliers, i.e., instances $(\mathbf{x}_i, 1)$ (resp. $(\mathbf{x}_i, -1)$) having $\alpha_i = \frac{1}{\tau m}$, as depicted in Figure 1. Let $\Pr^{emp}(\mathbf{s}_i^+) = \frac{1}{m}|\{\mathbf{s}_i^+\}|$ (resp. $\Pr^{emp}(\mathbf{s}_i^-)$) and $\Pr^{emp}(\mathbf{o}_i^+) = \frac{1}{m}|\{\mathbf{o}_i^+\}|$ (resp. $\Pr^{emp}(\mathbf{o}_i^-)$) be the portions of the positive (resp. negative) in-bound support vectors and the outliers amongst $\mathbf{Z}_m$ respectively.

THEOREM 3.1. *Assume $\rho > 0$ and $\gamma > 0$, then $\Pr^{emp}(\mathbf{o}_i^+) - \Pr^{emp}(\mathbf{o}_i^-)$ is upper-bounded by $\tau + \Pr^{emp}(\mathbf{s}_i^-)$.*

PROOF. *At KKT complementarity conditions, $\gamma > 0$ implies $\eta = 0$ (see Appendix). Therefore the term $\sum_{i=1}^{m} \alpha_i \geq 2\frac{\mu}{\tau} + 1$ in Eq. (4) becomes an equation. We have*

$$\begin{cases} \sum_{i=1}^{m^+} \alpha_i + \sum_{i=1}^{m^-} \alpha_i = 2\frac{\mu}{\tau} + 1 \\ \sum_{i=1}^{m^+} \alpha_i - \sum_{i=1}^{m^-} \alpha_i = 1 \end{cases}.$$

Summing the above two equations we have $\sum_{i=1}^{m^+} \alpha_i = \frac{\mu}{\tau} + 1$, $0 \leq \alpha_i \leq \frac{1}{\tau m}$. There exist at most $(\frac{\mu}{\tau} + 1)/(\frac{1}{\tau m})$ positive instances that have $\alpha_i = \frac{1}{\tau m}$. Since the outliers have $\alpha_i = \frac{1}{\tau m}$, we obtain

$$\Pr^{emp}(\mathbf{o}_i^+) \leq \frac{(\mu + \tau)m}{m} = \mu + \tau. \tag{5}$$

Now subtract the above two equations. We have $\sum_{i=1}^{m^-} \alpha_i = \frac{\mu}{\tau}$, $0 \leq \alpha_i \leq \frac{1}{\tau m}$. Since each α_i can contribute at most $\frac{1}{\tau m}$, there exist at least $(\frac{\mu}{\tau})/(\frac{1}{\tau m}) = \mu m$ negative instances that have $\alpha_i \geq 0$. This implies that $\Pr^{emp}(\mathbf{s}_i^-) + \Pr^{emp}(\mathbf{o}_i^-) \geq \frac{\mu m}{m} = \mu$, and therefore

$$\Pr^{emp}(\mathbf{o}_i^-) \geq \mu - \Pr^{emp}(\mathbf{s}_i^-). \tag{6}$$

Combining Eqs. (5) and (6), we obtain

$$\begin{aligned} \Pr^{emp}(\mathbf{o}_i^+) - \Pr^{emp}(\mathbf{o}_i^-) &\leq (\mu + \tau) - (\mu - \Pr^{emp}(\mathbf{s}_i^-)) \\ &= \tau + \Pr^{emp}(\mathbf{s}_i^-). \end{aligned}$$

■

THEOREM 3.2. *Assume $\rho > 0$ and $\gamma > 0$, then $\Pr^{emp}(\mathbf{o}_i^+) - \Pr^{emp}(\mathbf{o}_i^-)$ is lower-bounded by $\tau - \Pr^{emp}(\mathbf{s}_i^+)$.*

PROOF. *Consider $\sum_{i=1}^{m^+} \alpha_i = \frac{\mu}{\tau} + 1$, $0 \leq \alpha_i \leq \frac{1}{\nu m}$. Since each α_i can contribute at most $\frac{1}{\tau m}$, there exist at least $(\frac{\mu}{\tau} + 1) / (\frac{1}{\tau m}) = (\mu + \tau)m$ positive instances that have $\alpha_i \geq 0$. Hence, we obtain $\Pr^{emp}(\mathbf{s}_i^+) + \Pr^{emp}(\mathbf{o}_i^+) \geq \frac{(\mu+\tau)m}{m} = \mu + \tau$; that is,*

$$\Pr^{emp}(\mathbf{o}_i^+) \geq \mu + \tau - \Pr^{emp}(\mathbf{s}_i^+). \tag{7}$$

Now consider $\sum_{i=1}^{m^-} \alpha_i = \frac{\mu}{\tau}$, $0 \leq \alpha_i \leq \frac{1}{\tau m}$. There exist at most $(\frac{\mu}{\tau}) / (\frac{1}{\tau m}) = \mu m$ negative instances that have $\alpha_i = \frac{1}{\tau m}$. We have

$$\Pr^{emp}(\mathbf{o}_i^-) \leq \frac{\mu m}{m} = \mu. \tag{8}$$

Combining Eqs. (7) and (8), we obtain

$$\begin{aligned} \Pr^{emp}(\mathbf{o}_i^+) - \Pr^{emp}(\mathbf{o}_i^-) &\geq (\mu + \tau - \Pr^{emp}(\mathbf{s}_i^+)) - \mu \\ &= \tau - \Pr^{emp}(\mathbf{s}_i^+). \end{aligned}$$

■

THEOREM 3.3. *Assume $\rho > 0$ and $\gamma > 0$. Suppose the instances in $\mathbf{Z}_m$ are generated i.i.d. from a distribution D that is continuous with respect to $\mathbf{x}$. Suppose, moreover, the kernel is analytic and non-constant. The difference $\Pr^{emp}(\mathbf{o}_i^+) - \Pr^{emp}(\mathbf{o}_i^-)$ converges almost surely to τ, i.e., $\Pr(\lim_{m\to\infty}(\Pr^{emp}(\mathbf{o}_i^+) - \Pr^{emp}(\mathbf{o}_i^-)) = \tau) = 1$.*

PROOF. *With Theorems 3.1 and 3.2, this can be proofed intuitively by claiming that, when $m \to \infty$, both $\Pr^{emp}(\mathbf{s}_i^+) \to 0$ and $\Pr^{emp}(\mathbf{s}_i^-) \to 0$ [27].* ■

We can see that the parameter τ controls the difference between the outliers from the positive and negative classes. As a byproduct, we can see from Eqs. (5) and (7) that

$$(\mu + \tau) - \Pr^{emp}(\mathbf{s}_i^+) \leq \Pr^{emp}(\mathbf{o}_i^+) \leq (\mu + \tau) \tag{9}$$

and from Eqs. (6) and (8) that

$$\mu - \Pr^{emp}(\mathbf{s}_i^-) \leq \Pr^{emp}(\mathbf{o}_i^-) \leq \mu. \tag{10}$$

The parameter μ controls the basic portion of the outliers from each class. Note the effect of μ in ASVM is similar to that of the parameter ν in ν-SVM classifier [27]. Using the above conclusions ASVM may incorporate with the prior knowledge (in portion of the outliers) to obtain a more sophisticated high confidence area.

4. PERFORMANCE EVALUATION

In this section, we evaluate the performance of ASVM. We also study the scalability of ASVM and discuss some implementation issues to cope with large-scale data.

4.1 Metrics and Settings

We implement ASVM based on LIBSVM [11]. To evaluate the performance of ASVM, we consider several public real-world datasets obtained from the UCI machine learning repository [2] and IJCNN 2001 competition [23]. We control a 1:9 ratio between the positive and negative instances by either resampling (for two-class datasets) or merging the class labels (for multi-class datasets) [31]. Users under such a ratio are sensitive to the false-positives since any increment in the false-positive rate may seriously affect the positive predictions. In each dataset the training and testing instances are split according to a 5:1 ratio. We use 10-fold cross validation in each training process.

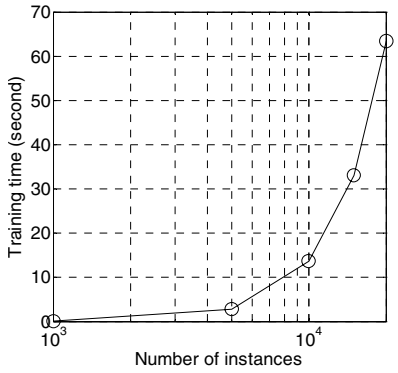


Figure 3: The scalability of ASVM based on the SMO implementation.

This paper focuses on low false-positive learning. In particular, we are interested in the performance of a classifier *provided that a user tolerance t, $0 \leq t \leq 1$, to the false-positive rate must be met*. We focus on the t-ROC space, i.e., an ROC space with the axis of false-positive rate ranging from 0 to t. We use the following metrics in our performance evaluation:

- **Slopes in t-ROC Space**: This metrics is useful to investigate the trade-off between different classifiers when the iso-performance line varies.
- **t-AUC**: This metrics demonstrates the *discriminability* of a classifier in t-ROC space. We let each classifier maximize this metrics in training time.

We compare ASVM with the ThresHolding (TH) [15, 28] and the Parameter Tuning (PT) [11, 19] techniques, which are both available in LIBSVM by default. Note that since we focus on a general purpose classifier, no prior knowledge, such as that used in [19], is assumed. In thresholding, the standard SVM classifier is used and has two parameters, C and q, as we have seen in Section 2.2, which need to be determined during the training time. We adopt a 2-dimensional grid search [17] for the optimal combination of these two parameters that maximizes t-AUC. In parameter tuning, we differentiate the parameter C of a standard SVM between the positive (C^+) and negative (C^-) classes, and employ a 3-dimensional grid search for the optimal combination of C^+, C^-, and q maximizing t-AUC. In ASVM, there are three parameters, μ, τ, and q, as we have seen in Sections 2.2 and 3. Rather than adopting a 3-dimensional grid search directly, we first fix a very small τ (to simulate the conventional SVM classifier) and apply a 2-dimensional grid search for the optimal combination of μ and q that maximizes t-AUC. After proper μ and q are obtained, we perform a linear search (i.e., 1-dimensional grid search) for τ maximizing the t-AUC further.

4.2 SMO Implementation

For better scalability, we reduce the ASVM dual to the Sequential Minimal Optimization (SMO) [22] problem. In order to match the SMO input, we need to rewrite the constraint $\sum_{i=1}^{m} \alpha_i \geq 2\frac{\mu}{\tau} + 1$ in Eq. (4) as $\sum_{i=1}^{m} \alpha_i = 2\frac{\mu}{\tau} + 1$. Doing so effectively relaxes the constraint $\gamma \geq 0$ in the ASVM primal (Eq. (3)) and therefore a special care is needed when selecting μ in the training time to prevent a

	ThresHolding (TH)			ASVM			Improvement		
Training target	1-AUC	0.1-AUC	0.05-AUC	1-AUC	0.1-AUC	0.05-AUC	%	%	%
Diabetes	**0.828618**	0.031508	–	0.828550	**0.040713**	–	-8.2e-5	29.2	–
Statlog German	**0.767854**	0.019167	–	0.763777	**0.019292**	–	-0.5	0.7	–
Breast Cancer	0.995638	0.095638	–	**0.995895**	**0.095895**	–	2.6e-2	0.3	–
Ionosphere	0.987302	0.087302	–	**0.996825**	**0.096825**	–	1.0	10.9	–
Australian	**0.948124**	**0.066397**	–	0.925197	0.065787	–	-2.4	-0.9	–
Covertype	**0.982942**	**0.083836**	0.0345622	0.982186	0.083080	**0.0348064**	-7.7e-2	-0.9	0.7
IJCNN	0.959185	0.078075	0.0349608	**0.976187**	**0.083115**	**0.0387323**	1.8	6.5	10.8

Table 1: Performance comparison between the ThresHolding (TH) and ASVM in terms of t-AUC, where $t = 1$, 0.1, and 0.05 are given in training time.

negative class-margin γ. One easy way is to check whether $\gamma < 0$ during each iteration of a grid search and skip the corresponding candidates. Another way is to train an auxiliary hyperplane with γ always equals to 0 in Eq. (3) first during each iteration of the grid search. We are able to estimate the *basic portion of zero* by calculating the portions of the negative instances falling across the auxiliary hyperplane. Following Eqs. (9) and (10), we can see that $\gamma \geq 0$ as long as

$$\mu \geq \textit{basic portion of zero.}$$

This approach, called *bi-training*, is particularly useful to those cases, such as on-line training, where the grid-search technique is infeasible. We adopt the former approach and omit the detailed discussions about the latter due to the space limitation. Figure 3 shows the scalability of ASVM. Currently, we are able to handle about 20 thousand instances within a minute.

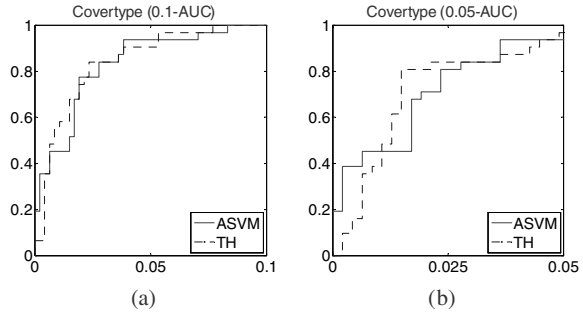
4.3 Comparison with Thresholding

In this section, we compare the testing results of ASVM with those of ThresHolding (TH). TH is based on traditional SVM classifier. As mentioned in Section 3, ASVM is also compatible to this technique and therefore we consider setting up different thresholds for ASVM's positive predictions as well. The resultant performance of both the classifiers can be easily arranged and shown in an ROC space, where each point on an ROC curve presents a trade-off between the true- and false-positive rates given a certain threshold (not necessarily larger than 0 in this case).

We use datasets including Pima Indian Diabetes, Statlog German, Wisconsin Breast Cancer, Ionosphere, Statlog Australian, Covertype, and IJCNN in our experiments. We consider $t = 1$ and 0.1 for each dataset in the training phase. For larger datasets such as Covertype and IJCNN, we consider $t = 0.05$ additionally since under such a configuration the training instances are still sufficient to apply the learned model to the testing data. Note that since the ratio between the positive and negative instances is 1:9, we differentiate the parameter C in TH between the positive (C^+) and negative (C^-) classes and set C^+:C^- = 9:1 to compensate for the skew data distribution[2].

Table 1 shows the maximal t-AUCs achieved by TH and ASVM respectively. As we can see, for Diabetes the 1-AUCs given by TH and ASVM are very close to each other. By comparing the 1-AUCs of the rest datasets, we can see that, generally, ASVM give similar performance as SVM in classification. When focusing on 0.1-AUCs, however, we observe that ASVM is able to give 33% improvement over TH. The other datasets based on which ASVM can make noticeable improvement include Ionosphere (10.9% for 0.1-AUC) and IJCNN (5.1% for 0.1-AUC, 3.8% for 0.05-AUC).

[2]This is suggested in LIBSVM [11].



(a) (b)

Figure 4: The ROC curves of TH and ASVM given $t = 0.1$ and 0.05 in training time.

We believe this is mainly because that ASVM successfully obtain a high confidence area of the positive class in these datasets. Overall, ASVM gives about 6.4% improvement in t-AUC when $t \leq 0.1$.

Notice that in the Statlog Australian dataset, the advantage of ASVM does not help a better performance. We believe this is because that the classes are separable in RKHS. Under such a case, SVM is good enough to make low false-positive predictions.

Next, we study the detailed performance of ASVM and TH within the 0.1- and 0.05-ROC space. Our observation shows that ASVM is usually the best classifier at the very first segment of the false-positive rate (starting from 0). This is true even for the Covertype dataset, despite the fact that ASVM does not achieve the highest 0.1-AUC in Table 1. Figure 4(a) illustrates the ROC curves returned by ASVM and TH using $t = 0.1$ in training time. As we can see, ASVM is the best classifier when the false-positive rate ranges from 0 to 0.019 and gives the sharpest range of slope, $[15.129, \infty]$, along the ROC Convex Hull. The true-positive rate is 0.774 at the point of false-positive rate 0.019. Figure 4(b) illustrates the ROC curves when $t = 0.05$ is used. Again, ASVM is the best classifier when the false-positive rate is above 0 and under 0.002. It also gives the sharpest slopes ranging from 32.780 to ∞ along the ROC Convex Hull. The true-positive rate is 0.387 at the point of false-positive rate 0.002. ASVM is useful in the situations that the cost of the false-positives is high (or, the slope of the iso-performance line is sharp).

4.4 Comparison with Parameter Tuning

In this section, we compare the testing results of ASVM with those of Parameter Tuning (PT). Although both PT and ASVM have three parameters (C^+, C^-, q and μ, τ, q respectively), they are trained in different way. In PT, the effects of C^+ and C^- are

	Param. Tuning (PT)			ASVM			Improvement		
Training target	1-AUC	0.1-AUC	0.05-AUC	1-AUC	0.1-AUC	0.05-AUC	%	%	%
Covertype	**0.984043**	**0.084180**	**0.0358381**	0.982186	0.083080	0.0348064	-0.2	-1.3	-2.9
IJCNN	**0.981917**	0.079808	0.0354446	0.976187	**0.083115**	**0.0387323**	-0.6	4.1	9.3

Table 2: Performance comparison between the Parameter Tuning (PT) and ASVM in terms of t-AUC, where $t = 1$, 0.1, and 0.05 are given in training time.

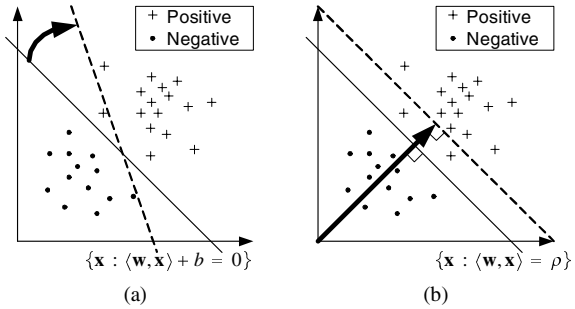


Figure 5: Decision planes in RKHS. (a) In PT, the movement of a decision plane is unpredictable when the values of C^+ and C^- are changed. (b) In ASVM, changing the value of τ effectively shifts the decision boundary toward the positive class.

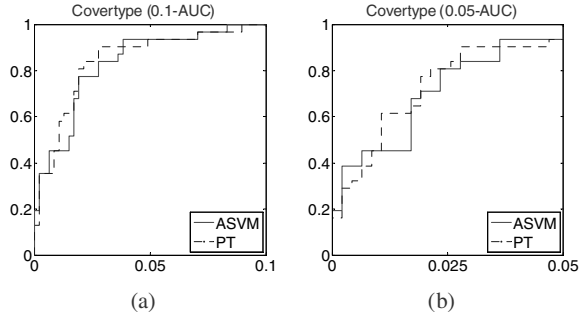


Figure 6: The ROC curves of PT and ASVM given $t = 0.1$ and 0.05 in training time.

correlated. Changing any value of C^+, C^-, and q may result in movement of a decision boundary as well as its margin, as shown in Figure 5(a). Under such a case, we need to search the entire 3-dimensional space for the best combination of C^+, C^-, and q. In ASVM, on the other hand, we can see from Figure 5(b) that given μ and q, increasing the value of τ effectively shifts the decision boundary toward the positive class. The class margin is enlarged, but its placement, which is determined by μ and q, is not affected by τ. Based on this observation we adopt a heuristic training method aiming at reducing the training times of a 3-dimensional grid search. As mentioned before, we first apply a 2-dimensional grid search for τ and q to determine a proper placement of the decision boundary when $\tau \approx 0$, and then increase τ to obtain a high confidence area of the positive class.

The maximal t-AUCs achieved by PT and ASVM are summarized in Table 2. Note we omit small datasets due to the space limitation. As we can see, the difference between the results of ASVM and PT is not significant, ranging between $\pm 3\%$.

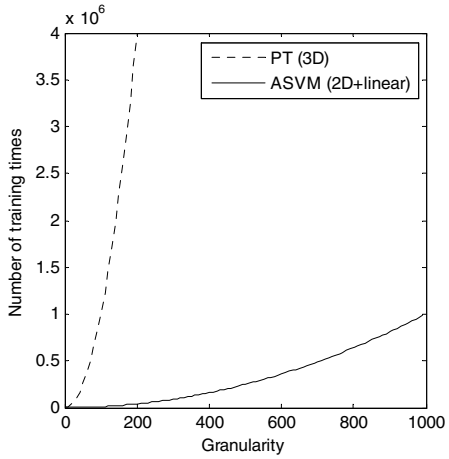


Figure 7: Number of iterations required to complete a grid serach.

To see the detailed performance of ASVM and PT within the 0.1- and 0.05-ROC space, let's consider again the Covertype dataset. Figure 6(a) illustrates the ROC curves returned by ASVM and PT using $t = 0.1$ in training time. As we can see, ASVM is the best classifier when the false-positive rate ranges from 0 to 0.002 and gives the sharpest range of slope, $[26.476, \infty]$, along the ROC Convex Hull. The true-positive rate is 0.355 at the point of false-positive rate 0.002. Figure 6(b) illustrates the ROC curves when $t = 0.05$ is used. In this case ASVM remains the best in the range $[0, 0.002]$ of the false-positive rate. It also gives the sharpest range of slope $[26.476, \infty]$ along the ROC Convex Hull. The true-positive rate is 0.387 at the point of false-positive rate 0.002. Generally, ASVM is able to give comparable performance against PT in terms of either t-AUC, $t \leq 0.1$, or slopes.

Next, we compare the number of training times required in the grid searches adopted by ASVM and PT respectively. The results are depicted in Figure 7 whose x-axis denotes the granularity, i.e., the number that a search range in each dimension is divided into. As we can see, ASVM requires an order less training times than PT. This is because we perform only a 2-dimensional search (for μ and q) with one extra linear search (for τ) rather than a 3-dimensional search as PT does. From the above discussions, ASVM is able to give comparable performance as compared with PT while significantly reducing the total training times.

4.5 Asymptotic Property of τ

Another advantage of ASVM is that it is able to give more insight into the dataset. In Section 3, we showed that there is an asymptotic relationship on the difference of the portion of the outliers between two classes. In order to give a more comprehensive view, we test the asymptotic property of τ in a synthetic dataset with 90 posi-

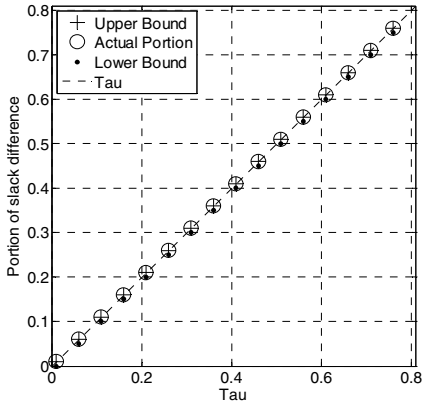


Figure 8: The asymptotic property of τ.

tive labeled and 10 negative labeled instances. Figure 8 shows the experimental results and compares the the difference derived theoretically with that obtained in the simulation under different values of τ. Note the dotted line along the diagonal depicts the values of τ.

As we can see, the actual portion of outliers lies within the theoretical upper and lower bounds. Actually, these three lines will converge to a single when the number of training data increases. From above, the relation between the difference of the portion of outliers and τ is justified.

5. CONCLUSIONS

We proposed ASVM, an Asymmetric Support Vector Machine that takes into account the false-positives and the user tolerance. ASVM maximizes the margin between the negative class and the core of the positive class. This allows us to raise the confidence in predicting the positives and obtain a lower false-positive rate. We quantitated the effects of μ and τ in terms of the portion of outliers. Experimental results showed that ASVM is able to either give 6.4% improvement in AUC and stay as the best classifier in the low-false positive region of the ROC Convex Hull as compared to the thresholding, or achieve a significant reduction in training time as compared to the parameter tuning.

6. REFERENCES

[1] I. Androutsopoulos, J. Koutsias, K. Chandrinos, and C. Spyropoulos. An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *Proc. of SIGIR*, 2000.

[2] A. Asuncion and D.J. Newman. *UCI Machine Learning Repository*, 2007.

[3] D. Barbara, N. Wu, and S. Jajodia. Detecting novel network intrusions using bayes estimators. In *Proc. of the 1st SIAM Conference on Data Mining (SDM)*, 2001.

[4] P. Bartlett and J. Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. In *Advances in Kernel Methods: Support Vector Learning*. MIT Press, 1998.

[5] A. Ben-Hur, D. Horn, H.T. Siegelmann, and V. Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2:125–137, 2001.

[6] P. Boykin and V. Roychowdhury. Leveraging social networks to fight spam. *IEEE Computer*, 2005.

[7] A. Bratko, G. Cormack, B. Filipic, T. Lynam, and B. Zupan. Spam filtering using statistical data compression models. *Journal of Machine Learning Research*, 7, 2006.

[8] L. Breiman. *Classification and Regression Trees*. Chapman & Hall, 1998.

[9] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[10] X. Carreras and L. Marquez. Boosting trees for anti-spam email fltering. In *Proc. of the 4th International Conference on Recent Advances in Natural Language Processing*, 2001.

[11] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. *Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm*, 2001.

[12] H.D. Cheng, X. Cai, X. Chen, L. Hu, and X. Lou. Computer-aided detection and classification of microcalcifications in mammograms: a survey. *Pattern Recognition*, 36(12):2967–2991, 2003.

[13] G. Cormack and T. Lynam. Overview of the trec 2005 spam evaluation track. In *Fourteenth Text REtrieval Conference (TREC-2005)*. NIST, 2005.

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

[15] H. Drucker, D. Wu, and V. Vapnik. Support vector machines for spam categorization. *IEEE Transactions on Neural networks*, 10(5), 1999.

[16] J. Goodman, G. Cormack, and D. Heckerman. Spam and the ongoing battle for the inbox. *Communications of the ACM*, 50(2):24–33, February 2007.

[17] C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification. *Technical report, http://www.csie.ntu.edu.tw/~cjlin/libsvm*, 2003.

[18] J. Kivinen, A. Smola, and R. Williamson. Online learning with kernels. *Advances in Neural Information Processing Systems. MIT Press*, 14:785–793, 2002.

[19] A. Kolcz and J. Alspector. SVM-based filtering of e-mail spam with content-specific misclassification costs. In *Proc. of TextDM*, 2001.

[20] H.-Y. Lam and D.-Y. Yeung. A learning approach to spam detection based on social networks. In *Proc. of the 4th Conference on Email and Anti-Spam (CEAS)*, 2007.

[21] T. Lynam, G. Cormack, and D. Cheriton. On-line spam filter fusion. In *Proc. of SIGIR*, pages 123–130, 2006.

[22] J. Platt. Sequenital minimal optimization: A fast algorithm for training support vector machines. In *Advances in Kernel Methods: Support Vector Learning*. MIT Press, 1998.

[23] D. Prokhorov. *IJCNN 2001 neural network competition*, 2001. Slide presentation in IJCNN'01, Ford Research Laboratory.

[24] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk e-mail. In *AAAI Technical Report WS-98-05*, 1998.

[25] K. Schneider. A comparison of event models for naive bayes anti-spam e-mail filtering. In *Proc. of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, 2003.

[26] B. Scholkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. C.Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 2001.

[27] B. Scholkopf and A. Smola. *Learning with Kernels:: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[28] D. Sculley and G. Wachman. Relaxed online support vector machines for spam filtering. In *Proc. of SIGIR*, 2007.

[29] J. Shawe-Taylor, P.L. Bartlett, R.C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.

[30] V. Vapnik. *Statistical Learning Theory*. Wiley, NY, 1998.

[31] P. Viola and M. Jones. Fast and robust classification using asymmetric adaboost and a detector cascade. In *Proc. of Neural Information Processing Systems (NIPS)*, 2002.

[32] W. Yih, J. Goodman, and G. Hulten. Learning at low false positive rates. In *Proc. of the 3rd Conference on Email and Anti-Spam (CEAS)*, 2006.

[33] B. Zheng, W. Qian, and L.P. Clarke. Digital mammography: mixed feature neural network with spectralentropy decision for detection of microcalcifications. *IEEE Transactions on Medical Imaging*, 15(5):589–597, 1996.

7. APPENDIX

7.1 Derivation of the ASVM Dual

To solve Eq. (3), we introduce a Lagrangian:

$$\begin{aligned} L &= \frac{1}{2}||\mathbf{w}||^2 - \rho + \frac{1}{\tau m}\sum_{i=1}^{m}\xi_i - \frac{\mu}{\tau}\gamma \\ &\quad - \sum_{i=1}^{m}\alpha_i\left(y_i\left(\langle\mathbf{w},\Phi(\mathbf{x}_i)\rangle - \rho\right) + \frac{1}{2}\gamma(y_i - 1) + \xi_i\right) \\ &\quad - \sum_{i=1}^{m}\beta_i\xi_i - \eta\gamma, \end{aligned} \tag{11}$$

where α_i, β_i, and η are Lagrange multipliers larger than or equal to 0. The Lagrangian L must be maximized with respect to α_i, β_i, and η, and minimized with respect to $\mathbf{w}$, ρ, γ, and ξ_i. At the Karush-Khun-Tucker (KKT) condition, we have

$$\begin{aligned} \frac{\partial L_P}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{m}\alpha_i y_i \Phi(\mathbf{x}_i) = 0 \\ &\Rightarrow \mathbf{w} = \sum_{i=1}^{m}\alpha_i y_i \Phi(\mathbf{x}_i), \end{aligned} \tag{12}$$

$$\frac{\partial L_P}{\partial \rho} = -1 + \sum_{i=1}^{m}\alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{m}\alpha_i y_i = 1, \tag{13}$$

$$\begin{aligned} \frac{\partial L_P}{\partial \gamma} &= -\frac{\mu}{\tau} - \frac{1}{2}\sum_{i=1}^{m}\alpha_i(y_i - 1) - \eta = 0 \\ &\Rightarrow \sum_{i=1}^{m}\alpha_i \geq 2\frac{\mu}{\tau} + 1, \end{aligned} \tag{14}$$

$$\frac{\partial L_P}{\partial \xi_i} = \frac{1}{\tau m} - \alpha_i - \beta_i = 0 \Rightarrow 0 \leq \alpha_i \leq \frac{1}{\tau m}. \tag{15}$$

Replacing the corresponding terms in Eq. (11) by those in Eqs. (12)-(15) and substituting the kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ for the dot product $\langle\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle$, we obtain the dual objective of ASVM.

Note we may also rewrite $f(\mathbf{x}) = \sum_{i=1}^{m}\alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) - \rho + \frac{\gamma}{2}$. The values of ρ and γ can be recovered using the KKT complementarity conditions. At optimum, we have

$$\alpha_i\left(y_i\left(\langle\mathbf{w},\Phi(\mathbf{x}_i)\rangle - \rho\right) + \frac{1}{2}\gamma(y_i - 1) + \xi_i\right) = 0, \tag{16}$$
$$\beta_i\xi_i = 0, \text{ and } \eta\gamma = 0,$$

$\forall 1 \leq i \leq m$. For each positive in-bound support vector $\mathbf{s}_i^+$, the second term at the left hand side of Eq. (16) must be zero. We have $\rho = \sum_{j=1}^{m}\alpha_j y_j k(\mathbf{x}_j, \mathbf{s}_i^+)$. Furthermore, for each $\mathbf{s}_i^-$, the equation $\gamma = \rho - \sum_{j=1}^{m}\alpha_j y_j k(\mathbf{x}_j, \mathbf{s}_i^-)$ holds.

7.2 Notation

$\mathcal{X}$	the pattern domain
$\mathbf{x}$	a pattern
y	a class label, $y \in \{\pm 1\}$
$\mathbf{Z}_m$	a sample of m training instances $((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_m, y_m))$
$\mathcal{Z}_m$	the domain of samples of size m
$\mathcal{H}$	Reproducing Kernel Hilbert Space (RKHS)
Φ	the feature map
k	a positive definite kernel
$\mathcal{F}$	a class of functions
f	a real value or $\{\pm 1\}$ function
$\mathcal{A}$	a class of events
A	an event
D	the distribution of $\mathcal{X} \times \{\pm 1\}$
$\|A\|$	the cardinality of a set (event) A
$\Pr\{A\}$	the probability of a set (event) A
fa	the false alarm rate $D\{(\mathbf{x}, -1) : f(\mathbf{x}) > \rho - \frac{\gamma}{2}\}$,
er	the misclassification rate $D\{(\mathbf{x}, y) : f(\mathbf{x}) \neq y\}$
$\mathbf{s}^+$ ($\mathbf{s}^-$)	positive (negative) in-bound support vectors
$\mathbf{o}^+$ ($\mathbf{o}^-$)	positive (negative) outliers
ρ	the core-margin
γ	the class-margin
ξ	the slack variable
α, β, η	Lagrange multipliers
μ, τ	ASVM parameters
q	the parameter of Gaussian RBF kernel
$\Pr^{emp}$	the empirical probability
t	the user tolerance