Solution Path for pin-SVM Classifiers with Positive and Negative τ Values

Xiaolin Huang, Lei Shi, and Johan A. K. Suykens

***Abstract*—Applying the pinball loss in a support vector machine (SVM) classifier results in pin-SVM. The pinball loss is characterized by a parameter τ. The τ value is related to the quantile distance considered in pin-SVM and different values are suitable for different problems. Therefore, tuning τ becomes an important issue. In this paper, we establish an algorithm to find the entire solution path for pin-SVM with different τ values. This algorithm is based on the fact that the optimal solution to pin-SVM is continuous and piecewise linear with respect to τ. Another contribution is that we show that the non-negative constraint on τ is not necessary, i.e., we can extend τ to negative values. First, in some applications, a negative τ may lead to better accuracy. Second, $\tau = -1$ corresponds to a simple solution, which links SVM and the classical kernel rule. Solution for $\tau = -1$ can be directly obtained and then be used as a starting point of the solution path. The proposed method efficiently traverses τ values through the solution path, and then achieves good performance by a suitable τ. Particularly, $\tau = 0$ corresponds to C-SVM, meaning that the traversal algorithm can output a result at least as good as C-SVM with respect to validation error.**

***Index Terms*—support vector machine, pinball loss, solution path, piecewise linear**

I. Introduction

The pinball loss is defined on $\mathbb{R}$ as

$$L_\tau(u) = \begin{cases} u, & u \geq 0, \\ -\tau u, & u < 0, \end{cases} \tag{1}$$

where $u \in \mathbb{R}$, τ is the absolute value of the slope on $u < 0$. It also can be written as a function of two variables. The two definitions are equal and we simply use (1) in this paper. The pinball loss is a generalization of the ℓ_1 loss ($\tau = 1$) and the hinge loss ($\tau = 0$). In the classical support vector machine (SVM, [1] [2]), one minimizes the sum of the regularization term and the hinge loss, which is called *C-SVM* and takes the following form:

$$\min_{w,b} \quad \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{m} L_{\tau=0}\left(1 - y_i(w^T\phi(x_i) + b)\right), \tag{2}$$

where $x_i \in \mathbb{R}^n, y_i \in \{-1, +1\}$ are the training data, ϕ is the feature map, and $C > 0$ is the trade-off parameter. C-SVM has been insightfully investigated, including its statistical properties, learning theory, and solving algorithms, see [3]–[10]. In classification problems, there are many possible loss functions, including squared hinge loss, logistic loss, least squares loss, and so on. The properties of these loss functions have been insightfully investigated by [4] [5] [11] and [12]. In this paper, we will discuss a variation of the hinge loss. Our discussion is mainly with ℓ_2-norm, but other regularization terms, like ℓ_1-norm [13] or elastic-net [14], are also of interest.

C-SVM is basically to maximize the margin by minimizing $\|w\|_2^2$. In C-SVM, the margin is related to the closest distance between two classes, since the hinge loss is minimized. Due to the fact that the distance is measured by the minimal distance, C-SVM is easily corrupted by noise, especially the feature noise around the decision boundary. Some de-noising methods have been discussed by [15] [16] [17] etc. The sensitivity to noise comes from the fact that the minimal distance is maximized. Thus to improve the classification performance for noise-polluted data, we maximize the quantile distance between two sets. The quantile value is closely related to the pinball loss $L_\tau(u)$ with a positive τ value, which has been well studied in the regression field; see, e.g., [18] [19] and [20]. From the link between the pinball loss and the quantile value, C-SVM (2) has been extended to the following support vector machine with the pinball loss (*pin-SVM*, [21])

$$\min_{w,b} \quad \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{m} L_\tau\left(1 - y_i(w^T\phi(x_i) + b)\right). \tag{3}$$

When a positive τ is used, the margin considered in pin-SVM corresponds to the quantile distance. Compared with the closest distance, the quantile distance is less sensitive to the noise on features. The hinge loss is a particular case of the pinball loss for $\tau = 0$. Hence, pin-SVM can be regarded as an extension to C-SVM. Introducing flexibility on τ can improve the classification performance of C-SVM. Different τ values are suitable for different data. This raises the question how to effectively tune τ.

Manuscript received 2014;

This work was supported: • EU: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC AdG A-DATADRIVE-B (290923). This paper reflects only the authors' views, the Union is not liable for any use that may be made of the contained information. • Research Council KUL: GOA/10/09 MaNet, CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants • Flemish Government: ○ FWO: projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); PhD/Postdoc grants ○ IWT: projects: SBO POM (100031); PhD/Postdoc grants ○ iMinds Medical Information Technologies SBO 2014 • Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017). L. Shi is also supported by the National Natural Science Foundation of China (11201079) and the Fundamental Research Funds for the Central Universities of China (20520133238, 20520131169). Johan Suykens is a professor at KU Leuven, Belgium.

X. Huang and J. A. K. Suykens are with the Department of Electrical Engineering ESAT-STADIUS, KU Leuven, B-3001 Leuven, Belgium. (e-mails: huangxl06@mails.tsinghua.edu.cn, johan.suykens@esat.kuleuven.be). L. Shi is with School of Mathematical Sciences, Fudan University, Shanghai, 200433, P.R. China. (e-mail: leishi@fudan.edu.cn)

In this paper, we will establish an algorithm which traverses all τ values which correspond to convex losses and selects a suitable one. Its basis is that the solution path of the dual problem of (3) is continuous piecewise linear with respect to τ. The continuity and piecewise linearity make it possible to search on the solution path via linear algebra operations. A similar technique has been considered in C-SVM (2) for tuning regularization parameter C, since its solution path w.r.t. C is also continuous and piecewise linear; see, e.g., [22] [23] and [24]. Another important application of piecewise linear solution path is to solve Lasso regularized optimization problem and its variation, such as the Dantzig selector. In those fields, the Forward Stagewise Linear Regression and the Least Angle Regression are both based on the piecewise linearity as discussed in [25] [26] and [27].

Besides the piecewise linearity, efficiently traversing requires a starting point which can be easily obtained. Recalling the definition of the pinball loss (1), one can find that when $\tau = -1$, $L_\tau(u)$ is a linear function, which follows that pin-SVM (2) becomes a non-constrained quadratic programming problem and can be easily solved. However, in previous researches, τ is required to be non-negative (in C-SVM, $\tau = 0$; and in [21], $\tau \geq 0$). In Fig.1, we plot the pinball loss for different τ values. Convexity of the pinball loss requires that $\tau \geq -1$. From this point of view, the non-negative condition on τ is indeed not necessary and pin-SVM with a negative τ is worth studying.

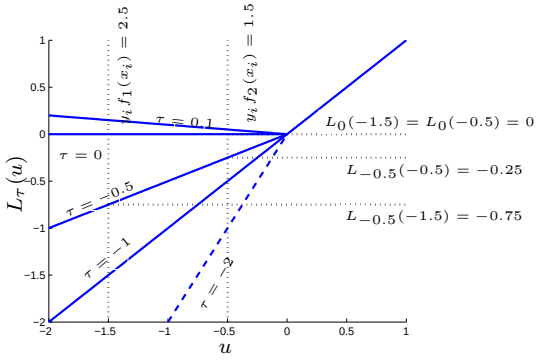


Fig. 1. Plots of the pinball loss for different τ values. The convex ones are displayed by solid lines. When $\tau < -1$, the pinball loss is non-convex as shown by the dashed line. Consider two functions $f_1(x)$ and $f_2(x)$. Assume that they have the same norm, then $y_i f(x_i)$ is related to the distance to $yf(x) = 0$: $y_i f_1(x_i) > y_i f_2(x_i)$ means this point farer from the decision boundary of f_1 than that of f_2. The hinge loss does not distinguish f_1 and f_2. While, sigmoid [28], log-likelihood, exponential [29], distance-weighted discrimination [30], and the pinball loss with a negative τ prefer f_1 to f_2.

Considering negative τ values in pin-SVM is not only because $\tau = -1$ corresponds to a simple solution, but also due to its statistical meaning. For a classification function $f(x) = w^T\phi(x) + b$ and a training point (x_i, y_i), the absolute value of $(y_i f(x_i) - 1)/\|w\|_2$ measures the distance of this point to the curves $\{x : y_i f(x_i) = 1\}$. When $y_i f(x_i) < 1$, the classification is incorrect or not strongly correct. In this case, we want to minimize the distance, i.e., penalty is given to $y_i f(x_i)$ and the penalty is minimized. When $y_i f(x_i) > 1$, traditionally, we do not care about its positions, then the hinge loss is applied in SVM formulation. If one also wants to draw information from the points which are correctly classified, gains can be given when $y_i f(x_i) > 1$. Maximizing the gains encourages a larger distance from $\{x : y_i f(x_i) = 1\}$. Altogether, we can minimize the distance to the curve $\{x : y_i f(x_i) = 1\}$ when $y_i f(x_i) < 1$ and maximize the distance when $y_i f(x_i) > 1$, resulting in the pinball loss defined as (1). The emphasis for $y_i f(x_i)$ less or larger than 1 could be different and the ratio is described by τ in $L_\tau(u)$. In subsection II-A, one can also observe that τ controls the upper bound of the dual variables in the dual problem of (3). If we put equal attention to all the training data, then $\tau = -1$ and it is closely related to the classical kernel rule [31] [32]. It also has been applied in one-bit compressive sensing, which could be regarded as a classification problem. In that classification task, there are only a few measurements and the pinball loss with $\tau = -1$ has became popular, see, e.g., [33] [34].

Before discussing pin-SVM with negative τ values and establishing a traversal algorithm, we first illustrate the performance of different τ values in Fig.2. (The experimental details will be given in subsection III-A.) Generally speaking, different problems need different τ values. In the view of classification accuracy, we can not in advance expect the suitable τ value, which is related to feature distribution, noise level, and problem size. This simple example also implies that pin-SVM with negative τ is worthy to study and an efficient algorithm to find a suitable τ is needed.

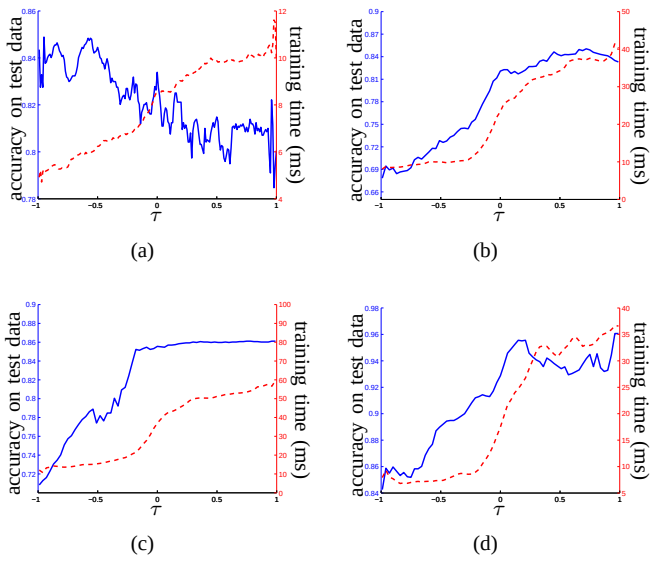


Fig. 2. The classification accuracy (blue solid lines) and the training time (red dashed lines, solved by sequential minimization optimization [6][35]) for different τ values. The data sets are downloaded from UCI Repository of Machine Learning Datasets [36] and include (a) Spect; (b) Monk1; (c) Monk2; (d) Monk3. The training time and τ value are positive correlated. The best test accuracy is achieved at different τ values for different sets.

The rest of this paper is organized as follows: in Section II, pin-SVM with a negative τ is investigated. Section III shows that the solution path of pin-SVM is continuous piecewise linear and then establishes an algorithm traversing the entire path. The proposed algorithm is evaluated by numerical experiments in Section IV. Section V ends the paper with conclusions.

II. Negative τ Values for Pinball Loss

A. Pin-SVM formulation

When the pinball loss is applied in classification, the corresponding support vector machine in the primal space is given by (3). The dual problem has been discussed in [21]. In this subsection, we will revisit the dual problem and investigate the role of τ. First, (3) can be formulated as the following constrained quadratic programming problem

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}w^T w + \sum_{i=1}^{m} C_i \xi_i \\ \text{s.t.} \quad & y_i \left[w^T \phi(x_i) + b\right] \geq 1 - \xi_i, i = 1, 2, \ldots, m, \\ & y_i \left[w^T \phi(x_i) + b\right] \leq 1 + \frac{1}{\tau}\xi_i, i = 1, 2, \ldots, m, \end{aligned} \tag{4}$$

where C_i could be different. A typical setting, which is simple and suitable for unbalanced training problems, is

$$\begin{aligned} C_i &= C_0, && \forall i : y_i = 1, \\ C_i &= \tfrac{\#j:y_j=-1}{\#j:y_j=1} C_0, && \forall i : y_i = -1, \end{aligned} \tag{5}$$

where $C_0 > 0$ is a constant defined by the user. We introduce the Lagrange multipliers $\alpha_i, \beta_i \geq 0$ corresponding to the constraints in (4). These dual variables meet the following complementary slackness condition,

$$\begin{aligned} &\alpha_i \left(1 - \xi_i - y_i \left[w^T \phi(x_i) + b\right]\right) = 0, i = 1, \ldots, m, \\ &\beta_i \left(y_i \left[w^T \phi(x_i) + b\right] - \tfrac{1}{\tau}\xi_i - 1\right) = 0, i = 1, \ldots, m. \end{aligned} \tag{6}$$

Then we get the dual problem below,

$$\begin{aligned} \min_{\alpha,\beta} \quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\alpha_i - \beta_i) y_i \mathcal{K}_{ij} y_j (\alpha_j - \beta_j) - \sum_{i=1}^{m}(\alpha_i - \beta_i) \\ \text{s.t.} \quad & \sum_{i=1}^{m} y_i (\alpha_i - \beta_i) = 0 \\ & \alpha_i + \frac{1}{\tau}\beta_i = C_i, i = 1, 2, \ldots, m, \\ & \alpha_i \geq 0, \beta_i \geq 0, i = 1, 2, \ldots, m. \end{aligned}$$

Introduce $\lambda_i = \alpha_i - \beta_i$ and eliminate the equality constraint $\alpha_i + \frac{1}{\tau}\beta_i = C_i$. The dual problem of pin-SVM is formulated as

$$\begin{aligned} \min_{\lambda} \quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\lambda_i y_i \mathcal{K}_{ij} y_j \lambda_j - \sum_{i=1}^{m}\lambda_i \\ \text{s.t.} \quad & \sum_{i=1}^{m} y_i \lambda_i = 0, \\ & -\tau C_i \leq \lambda_i \leq C_i, i = 1, 2, \ldots, m, \end{aligned} \tag{7}$$

where $\mathcal{K}$ corresponds to a positive definite kernel with $\mathcal{K}_{ij} = \mathcal{K}(x_i, x_j) = \phi(x_i)^T \phi(x_j)$.

Solving (7) results in the optimal solution. Then the obtained function can be represented as

$$f^{\lambda,b}(x) = \sum_{i=1}^{m} y_i \lambda_i \mathcal{K}(x, x_i) + b,$$

where b is computed according to the complementary slackness condition (6), i.e., $y_i f^{\lambda,b}(x_i) = 1, \forall i : \alpha_i \neq 0 \ \& \ \beta_i \neq 0$. Since $\lambda_i = \alpha_i - \beta_i$ and $\alpha_i \beta_i = 0$, we can calculate the bias term by

$$\sum_{j=1}^{m} y_j \lambda_j \mathcal{K}(x_i, x_j) + b = y_i, \forall i : -\tau C_i < \lambda_i < C_i.$$

In the primal space, τ describes the slope of the pinball loss $L_\tau(u)$, as displayed in Fig.1. In the dual problem (7), the objective is a quadratic function independent of τ and the feasible set is $-\tau C_i \leq \lambda_i \leq C_i$. The upper bound is controlled by C_i. After C_i is given, we can further tune the lower bound by τ. For many problems, tuning the upper bound can improve the classification accuracy. Similarly, one can expect that the performance also relies on the lower bound, as displayed by Fig.2.

B. Pinball loss with negative τ

In classification problems, the hinge loss, i.e., pin-SVM with $\tau = 0$, has been well studied; see, e.g., [4] [5]. One important study on loss functions used in classification problems has been given by [12]. A typical classification loss L has the following properties:

1) $L(u)$ is Lipschitz with a constant;
2) $L(u)$ is convex;
3) $\frac{\partial L(u)}{\partial u}|_{u=1} > 0$;
4) $L(u)$ is non-negative.

It is not hard to verify that the pinball loss with $\tau \geq 0$, including the hinge loss, satisfies these properties, which follows that when $\tau \geq 0$, $L_\tau(u)$ enjoys many nice properties for classification, such as classification-calibration and Bayes consistency. The corresponding learning rates can be analyzed as well. In this paper, we further discuss the pinball loss with negative τ values. When $\tau \geq -1$, the pinball loss is still a convex function and properties 1)–3) holds, then we have:

Theorem 1: $L_\tau(u)$ is calibrated if $\tau \geq -1$.

This is a direct corollary of Theorem 2 of [12]. However, many existing analysis for loss functions cannot be extended to the pinball loss with negative τ values, since $L_\tau(u)$ with $-1 \leq \tau < 0$ may take negative value and is not lower bounded. If there is no regularization term in (3), it is meaningless to consider negative τ values in the pinball loss. However, in practice, we always pursue the discriminant function in a bounded function space and there is a regularization term to guarantee a good generalization capability. In that case, the pinball loss with negative τ values becomes meaningful, as discussed before from both the primal space and dual space. Generally, we need to analyze loss functions in a bounded function space. This is different from existing results on loss functions, which are usually obtained free of approximation error caused by the size of function space. Analyzing loss functions together with the function space could be an interesting topic, not only for the pinball loss but also for other loss functions which are not lower bounded, such as the one used in [33] for one-bit compressive sensing.

C. Particular cases $\tau = 0$ and $\tau = -1$

Among all the possible values, $\tau = 0$ is a particular case, which corresponds to C-SVM. In pin-SVM, the dual variables can be categorized into three types: *lower bounded support vectors* ($\lambda_i = -\tau C_i$), *free support vectors* ($-\tau C_i < \lambda_i < C_i$), and *upper bounded support vectors* ($\lambda_i = C_i$). When $\tau = 0$, the lower bounded support vectors become zero. This brings sparseness, which is meaningful for reducing the storage space for support vectors, but is not necessary from the viewpoint of accuracy.

Another interesting choice is $\tau = -1$, for which the optimal dual variables can be obtained directly. One can verify that when C_i are set as (5), the bias term has no effect on the objective value of the primal problem (3). We simply set b equal to zero. Then the function corresponding to pin-SVM with $\tau = -1$ is

$$f(x) = \sum_{i:y_i=+1} C_i \mathcal{K}(x, x_i) - \sum_{i:y_i=-1} C_i \mathcal{K}(x, x_i). \quad (8)$$

If a linear kernel $\mathcal{K}(x, x_i) = x^T x_i$ is used, the decision rule becomes

$$\mathrm{sgn}(f(x)) = \begin{cases} +1, \text{ if } x^T \bar{x}_+ > x^T \bar{x}_-, \\ -1, \text{ if } x^T \bar{x}_+ < x^T \bar{x}_-, \end{cases}$$

where $\bar{x}_+$ and $\bar{x}_-$ are the mean of $\{x_i,\ i : y_i = +1\}$ and $\{x_i,\ i : y_i = -1\}$, respectively. In other words, we use the angles between x and the centers of the two classes to determine the label. When a nonlinear kernel is used, pin-SVM with $\tau = -1$ is to classify the data according to the angle in the feature space. Another interesting observation is that (8) gives the classical kernel methods, given by [31]. The discussions therein provided insightful understanding for pin-SVM with $\tau = -1$.

III. Finding Solutions for $\tau \geq -1$

A. Solution path

In C-SVM, τ is fixed to be zero, which has been extended to $\tau \geq 0$ in [21]. This paper further shows that negative τ values with $\tau \geq -1$ are worth considering as well. As a simple example, we consider four data sets "Spect", "Monk1", "Monk2", and "Monk3" (downloaded from the UCI Repository of Machine Learning Datasets, [36]). The radial basis function (RBF) kernel

$$\mathcal{K}(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right)$$

is used. We tune the kernel parameter σ and the regularization parameter C_0 based on C-SVM. Then different τ values are tested and the classification accuracy is plotted in Fig.2. Notice that in this experiment, we solve pin-SVM by sequential minimization optimization techniques.

The basic finding from the results is that we need different τ values for different problems, which requires an efficient algorithm to find suitable τ values. The basis of this algorithm is the continuity and piecewise linearity of the solution path of pin-SVM (7), which can be easily verified from the observation that τ determines the boundary of the feasible set of (7). Denote the optimal dual variables of (7) with a given τ as $\lambda(\tau)$, the optimal bias term as $b(\tau)$. Then $\lambda(\tau)$ and $b(\tau)$ are continuous piecewise linear functions w.r.t. τ. To give an illustration, we set $C_0 = 5, \sigma = 1$ and plot the solution path of several dual variables for dataset "Spect" in Fig.3.

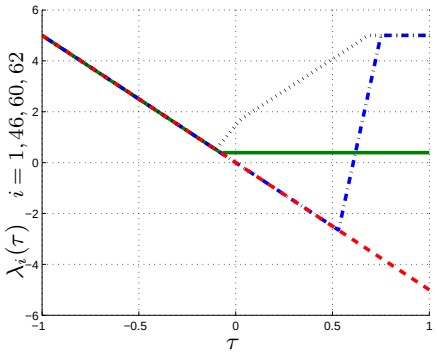


Fig. 3. Typical solution paths of several dual variables for "Spect" ($C_0 = 5, \sigma = 1$): (a) $\lambda_1(\tau)$ is shown by green solid line; (b) $\lambda_{46}(\tau)$ by black dotted line; (c) $\lambda_{60}(\tau)$ by red dashed line; (d) $\lambda_{62}(\tau)$ by blue dash-dotted line.

Fig.3 shows typical solution paths. $\lambda_{60}(\tau)$ (red dashed line) is a linear function with respect to τ, which is the simplest case and means that the corresponding point is always a lower bounded support vector. Except of this case, other solution paths are polylines. The first changing point corresponds to the λ value, from which a lower bounded support vector becomes a free support vector. For the free support vectors, its value could keep unchange, like $\lambda_1(\tau)$ shown by green solid line. The value also may increase until $\lambda_i = C_i$, from then the corresponding point becomes an upper bounded support vector, as illustrated by $\lambda_{46}(\tau)$ (black dotted line) and $\lambda_{62}(\tau)$ (blue dash-dotted line).

Generally, $\lambda(\tau)$ and $b(\tau)$ are piecewise linear to τ, therefore, simple linear algebra operations help us searching the solution path to effectively find the solutions for different τ values. Searching on a piecewise linear path, we have to: i) quickly find a starting point; ii) determine the slope; iii) detect the changing point. The details will be discussed in the next subsection.

B. Traversal algorithm

For given λ and b, define the following three sets

$$\begin{aligned} \mathcal{E}(\lambda, b) &= \left\{i : y_i f^{\lambda,b}(x_i) = 1, -\tau C_i \leq \lambda_i \leq C_i\right\}, \\ \mathcal{L}(\lambda, b) &= \left\{i : y_i f^{\lambda,b}(x_i) \leq 1, \lambda_i = C_i\right\}, \\ \mathcal{U}(\lambda, b) &= \left\{i : y_i f^{\lambda,b}(x_i) \geq 1, \lambda_i = -\tau C_i\right\}, \end{aligned}$$

where

$$f^{\lambda,b}(x) = \sum_{i=1}^{m} \alpha_i \mathcal{K}(x_i, x) + b.$$

According to the KKT condition for pin-SVM (7), λ and b are the optimal solutions if and only if the following two conditions are met:

1) $\sum_{i=1}^{m} y_i \lambda_i = 0$,

2) there is a partition of the index set E, L, U, such that

$$E \subseteq \mathcal{E}(\lambda, b), L \subseteq \mathcal{L}(\lambda, b), U \subseteq \mathcal{U}(\lambda, b).$$

Notice that when the intersection set between $\mathcal{E}(\lambda, b)$ and $\mathcal{L}(\lambda, b)$, or $\mathcal{U}(\lambda, b)$ is not empty, there could be multiple partitions satisfying the above conditions.

Suppose $\lambda(\tau_0)$ and $b(\tau_0)$ are optimal to pin-SVM with τ_0. Then there is a partition E, L, U satisfying:

$$\begin{array}{l} \sum_{i=1}^{m} y_i \lambda_i(\tau_0) = 0 \\ E \subseteq \mathcal{E}(\lambda(\tau_0), b(\tau_0)), \\ L \subseteq \mathcal{L}(\lambda(\tau_0), b(\tau_0)), \\ U \subseteq \mathcal{U}(\lambda(\tau_0), b(\tau_0)). \end{array}$$

Consider an increase $\Delta\tau \geq 0$. Since $\lambda(\tau)$ and $b(\tau)$ are continuous piecewise linear, when $\Delta\tau$ is small enough, $\lambda(\tau_0 + \Delta\tau)$ and $b(\tau_0 + \Delta\tau)$ can be written as

$$\begin{array}{rcl} \lambda_i(\tau_0 + \Delta\tau) & = & \lambda_i(\tau_0) + a_i \Delta\tau, \\ b(\tau_0 + \Delta\tau) & = & b(\tau_0) + a_0 \Delta\tau. \end{array}$$

Correspondingly, $y_i f^{\lambda(\tau_0+\Delta\tau), b(\tau_0+\Delta\tau)}(x_i)$ is also a linear function, i.e.,

$$\begin{array}{rl} & y_i f^{\lambda(\tau_0+\Delta\tau), b(\tau_0+\Delta\tau)}(x_i) - y_i f^{\lambda(\tau_0), b(\tau_0)}(x_i) \\ = & y_i \left(\sum_{j=1}^{m} y_j a_j \mathcal{K}_{ij} + a_0 \right) \Delta\tau. \end{array}$$

Suitable $a_i, i = 0, 1, \ldots, m$ should satisfy:

1) for $i \in E$, the function value keeps unchanged;
2) for $i \in L$, the dual variable keeps unchanged;
3) for $i \in U$, the dual variable equals to $-\tau C_i$;
4) for $i \in E$, the dual variable is in $[-\tau C_i, C_i]$;
5) for $i \in L$, the function value is not larger than 1;
6) for $i \in U$, the function value is not smaller than 1.

The first three conditions provide linear equations:

$$\left\{ \begin{array}{l} \sum_{j=1}^{m} y_j a_j \mathcal{K}_{ij} + a_0 = 0, \quad \forall i \in E, \\ a_i = 0, \quad \forall i \in L, \\ a_i = -1, \quad \forall i \in U, \\ \sum_{i=1}^{m} y_i a_i = 0, \end{array} \right. \tag{9}$$

Since E, L, and U is a partition of the index set, the above system involves $m+1$ equations and $m+1$ variables, which can determine a_i (in degenerated cases, there are multiple solutions and we simply set the undetermined a_i to be zero).

After calculating a_i, we need further to find the step length $\Delta\tau$, which should keep the last three conditions valid. In other words, $\Delta\tau$ should satisfy:

$$-(\tau + \Delta\tau) C_i \leq a_i \Delta\tau + \lambda_i(\tau_0) \leq C_i, \forall i \in E, \tag{10}$$

$$y_i \left(\sum_{j=1}^{m} y_j a_j \mathcal{K}_{ij} + a_0 \right) \Delta\tau \leq 1 - y_i f^{\lambda(\tau_0), b(\tau_0)}(x_i), \quad \forall i \in L, \tag{11}$$

$$y_i \left(\sum_{j=1}^{m} y_j a_j \mathcal{K}_{ij} + a_0 \right) \Delta\tau \geq 1 - y_i f^{\lambda(\tau_0), b(\tau_0)}(x_i), \quad \forall i \in U. \tag{12}$$

Simple calculation can find the maximal step length, denoted by $\overline{\Delta\tau}$. Then all the optimal solutions $\lambda(\tau), b(\tau)$ for $\tau \in \left[\tau_0, \tau_0 + \overline{\Delta\tau}\right]$ are found. And we repeat the above process by setting $\tau_0 = \tau_0 + \overline{\Delta\tau}$.

After illustrating the key update procedure, we give the algorithm in detail:

Initialization

The starting point is $\tau_0 = -1$, which is the smallest possible value for τ. Moreover, its optimal dual variables can be directly obtained, i.e., $\lambda_i(-1) = C_i$. In this case, any partition meets the requirement on the dual variables. $b(-1)$ is selected such that there exists a partition E, L, U and the corresponding $a_i, i = 0, 1, \ldots, m$ satisfy:

$$\left\{ \begin{array}{l} \sum_{i=1}^{m} y_i a_i = 0, \\ a_i = 0, \quad \forall i \in L, \\ a_i = -1, \quad \forall i \in U. \end{array} \right. \tag{13}$$

There could be multiple solutions meeting the above condition. In this paper, we use the one with the least absolute value as $b(\tau)|_{\tau=-1}$. Along with determining $\lambda(\tau)|_{\tau=-1}, b(\tau)|_{\tau=-1}$, the corresponding E, L, and U are found.

Updating

After obtaining the partition E, L, U for τ_0, we can update λ and b following the discussion in the last subsection. First, the linear equations (9) are solved to get $a_i, i = 0, 1, \ldots, m$. In (9), $a_i, i \in L \bigcup U$ are directly given and calculating $a_i, i \in E$ involves an inverse problem. In general situation, the number of elements in E is not large and hence (9) can be solved efficiently.

When $a_i, i = 0, 1, \ldots, m$ are found, we can solve linear equations (10)–(11) to find $\overline{\Delta\tau}$, the maximal value of $\Delta\tau$. Then the optimal solutions between $[\tau_0, \tau_0 + \overline{\Delta\tau}]$ are obtained:

$$\begin{array}{rcll} \lambda_i(\tau_0 + \Delta\tau) & = & \lambda_i(\tau_0) + a_i \Delta\tau, & \forall \Delta\tau \in [0, \overline{\Delta\tau}], \\ b(\tau_0 + \Delta\tau) & = & b(\tau_0) + a_0 \Delta\tau, & \forall \Delta\tau \in [0, \overline{\Delta\tau}]. \end{array}$$

After calculating the optimal solution between τ_0 and $\tau_0 + \overline{\Delta\tau}$, we move to $\tau_0 + \overline{\Delta\tau}$. Correspondingly, the partition is updated. Suppose i_0 is the index which determines $\overline{\Delta\tau}$. There are four situations. If $i_0 \in E$ and $a_{i_0} > 0$, then $\lambda_{i_0}(\tau_0 + \overline{\Delta\tau}) = C_{i_0}$ and we will put i_0 into L, i.e., the partition is updated to be $E \setminus \{i_0\}, L \bigcup \{i_0\}, U$. The other three situations are

- if $i_0 \in E, a_{i_0} < 0$, the partition is $E \setminus \{i_0\}, L, U \bigcup \{i_0\}$;
- if $i_0 \in L$, the partition is $E \bigcup \{i_0\}, L \setminus \{i_0\}, U$;
- if $i_0 \in U$, the partition is $E \bigcup \{i_0\}, L, U \setminus \{i_0\}$.

Termination

With the update processing, τ_0 is increased until two events happen. The first one is that $\tau_0 > 1$. As analyzed before, the reasonable τ value is less than one and hence we do not need to calculate $\lambda(\tau), b(\tau)$ for $\tau > 1$. Another possibility is that in update processing, any $\Delta\tau > 0$ satisfies (10)–(12). Then $\overline{\Delta\tau} = \infty$ and we terminate the algorithm since the solutions for all τ are obtained.

C. Discussion about different τ values

In [21], the role of τ has been discussed from a statistical analysis viewpoint. Now, with the help of the traversal algorithm, we can calculate the solutions for all reasonable τ values and observe the corresponding performance. Generally,

a positive τ encourages the results to have a small within-class scatter. Since scatter lacks of invariance for scaling, we need normalization for pre-processing. In this paper, we simply scale each feature to have the same range. Minimizing within-class scatter is suitable to deal with data coming from a centralized distribution. A typical example is that features of each class are drawn from Gaussian distributions. On the contrary, when this property is not true, e.g., when the features come from a uniform distribution, or a mixture of several Gaussian distributions, minimizing the within-class scatter is not reasonable and a negative τ may lead to a better result. Now we have established a traversal algorithm and then can numerically investigate the relationship between the problem structure and the suitable τ value. From the relationship, we may learn useful information via the selected τ.

For visualization, we consider a 2-dimensional problem where the features come from Gaussian distributions: data x_i with $y_i = 1$ are drawn from $\mathcal{N}(\mu_1, \Sigma_1)$ and data x_i with $y_i = -1$ are from $\mathcal{N}(\mu_2, \Sigma_2)$, where

$$\mu_1 = \begin{bmatrix} 0.5 \\ -3 \end{bmatrix}, \ \mu_1 = \begin{bmatrix} -0.5 \\ 3 \end{bmatrix}, \ \Sigma_1 = \Sigma_2 = \begin{bmatrix} 0.2 & 0 \\ 0 & 3 \end{bmatrix}.$$

It is not hard to verify that the corresponding Bayes classifier is $f_c(x) = 2.5x(1) + x(2)$, displayed in Fig.4 by the dashed red lines. We artificially add noise. Their labels are selected from $\{-1, +1\}$ with equal probability and the features come from a Gaussian distribution, of which the mean is $[0, 0]^T$ and the covariance matrix is

$$\begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}.$$

This noise has no effect on the Bayes classifier but it will affect the training results of SVMs. Applying the traversal algorithm, we obtain the classifiers corresponding to different τ values and then display the classifiers from different τ values by the blue solid lines. In Fig.4(a) there are 200 training data for each class from the considered distribution and 10 noisy data points are added. One can find that a positive τ is suitable for this case, since the features in each class is clustered around its center. From another point of view, if a large τ is selected by the traversal algorithm, we can expect that the original distribution is centralized. In Fig.4(b), we reduce the number of points in class $+1$ and test the unbalanced situation. Here the number of points in class $+1$ is one tenth of that in class -1. In Fig.4(b), the varying range for different τ values is significantly larger than that in Fig.4(a).

Next, we add more noisy data into the training set. In Fig. 5(a) we show the case that 20 noisy data points are added to each class and the classifiers obtained from different τ values. The corresponding probability density functions (p.d.f.) of $y_i f(x_i)/\max_i\{y_i f(x_i)\}$ are displayed in Fig.5(b). Compared with the previous experiment, we need a larger τ which draws support from the data structure when the noise is more heavy. Pin-SVM with a positive τ is related to quantile distance, which is more stable to feature noise. Thus, a significant improvement from $\tau = 0$ generally implies that the data contain heavy noise.

We may also meet mixture distributions in applications. For example, in Fig.6, points in one class come from two

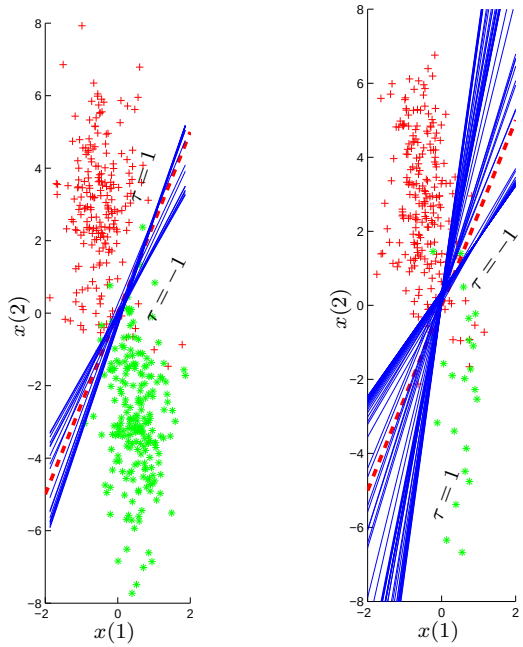


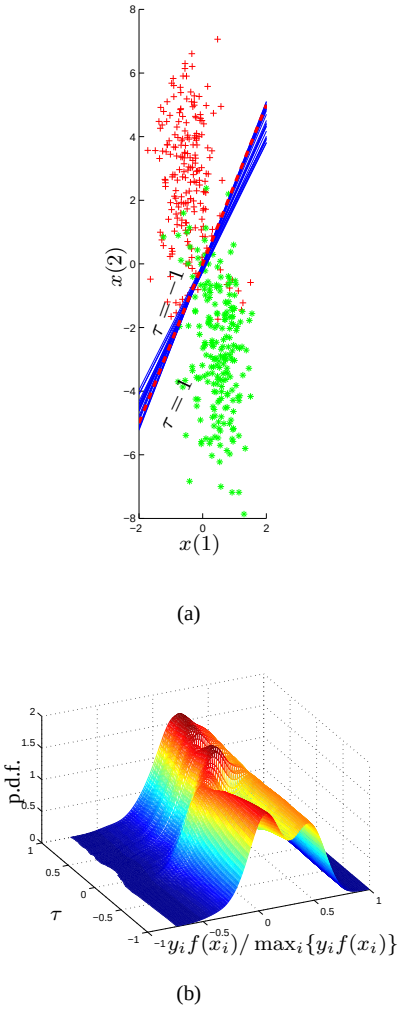
Fig. 4. Sampling points in class $+1$ are shown by green stars and points in class -1 are shown by red crosses. The Bayes classifier is given by red dashed lines and blue solid lines display the classifiers obtained by the pin-SVM with different τ values. (a) 10 noisy data points are added; (b) unbalanced case, where the number of the training points in class $+1$ is only one tenth of those in class -1.

Gaussian distributions, then a large τ value, which pursues a small within-class scatter, is not reasonable. In this case, a negative τ leads to a more accurate result. In applications, we choose the suitable τ value via cross-validation and there could be gap between the selected one and the best τ. Though selecting τ via cross-validation is not optimal, it still provides useful hints for understanding the feature distribution.

The above analysis is valid for nonlinear classifiers as well but the discussion will be in the feature space. As an example, we consider the data set "Monk1" from UCI dataset and the pin-SVM with a RBF kernel ($C_0 = 1, \delta = 1$). Via the traversal algorithm, the classification function $f(x)$ for different τ values are obtained and the p.d.f. of $y_i f(x_i)/\max_i\{y_i f(x_i)\}$ are shown in Fig.7(a). With an increasing τ, the scatter of $y_i f(x_i)/\max_i\{y_i f(x_i)\}$ becomes smaller. This trend can also be observed from Fig.7(b), which shows the p.d.f. for $\tau = -0.5, 1, 1.5$. From both the two figures, one can observe that a positive τ value is suitable, meaning that the data points in each class are centralized in the feature space, e.g., they may come from a Gaussian distribution. If a negative τ value achieves an accurate result, it generally indicates that there are sub-classes, like the distributions considered in Fig.6.

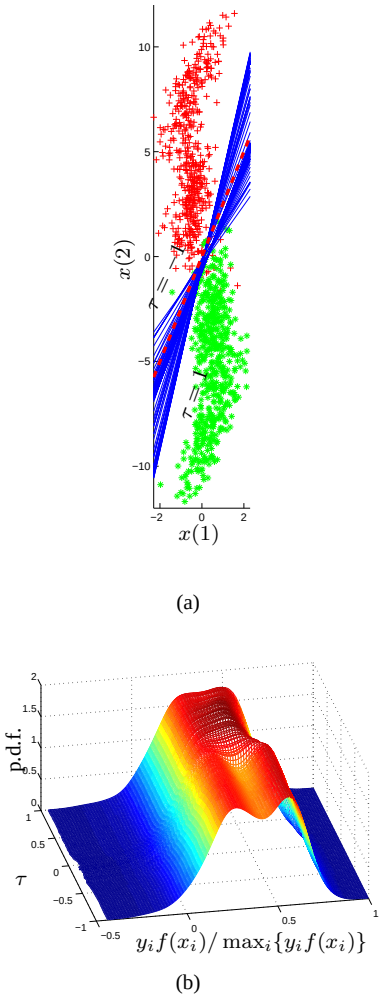
IV. Numerical Experiments

In the above sections, we extended the parameter τ into negative values and then established an algorithm traversing the entire solution path. With the help of this algorithm we



(a) (b)

Fig. 5. The meaning of plots in (a) is as the same as that in Fig. 4 but 20 noisy data points are added; in (b) the probability density functions of $y_i f(x_i)/\max_i\{y_i f(x_i)\}$ for different τ are displayed.



(a) (b)

Fig. 6. The meaning of plots in (a) is as the same as that in Fig. 4. In this figure, the training data of each class come from two Gaussian distributions; (b) displays the probability density functions of $y_i f(x_i)/\max_i\{y_i f(x_i)\}$ for different τ.

can test different τ values ($-1 \leq \tau \leq 1$) in a short time and choose the most suitable one. In this section, we will illustrate the performance of the proposed method on real-life data sets. The involved data are downloaded from the UCI Dataset [36] and LIBSVM data sets [9]. For some problems, there are training and test data provided. Otherwise, we randomly select m observations to train the classifier and use the remaining for test. All the experiments are done in Matlab 2013a in Core i5-1.80 GHz, 4.0GB RAM.

In our experiments, the RBF kernel is used and C_i is set according to (5). The regularization coefficient C_0 and the bandwidth in the RBF kernel σ are tuned by 10-fold cross-validation. When the number of training data is less than 10000, the traversal algorithm is applied. For each pair of σ and C_0, the traversal algorithm outputs the solution of (3) for all $-1 \leq \tau \leq 1$. In validation process, we consider $\tau = -1, -0.99, \ldots, 0, 0.01, \ldots, 0.99$. Then the parameters with the least total validation error are picked out. The candidate values for C_0 is $\{0.1, 0.5, 1, 2, 5, 10\}$ and that for σ is $\{0.01, 0.1, 1, 10\}$. We repeat the training and test process 10 times, then report the average classification accuracy on the test sets and the average time in Table I. For comparison, we also apply boosted tree method [40] [41]. The number of ensemble learning cycles is set such that the boosted tree has similar computational time as the SVMs. The average classification accuracy and the computational time are also reported in Table I.

One efficient algorithm to solve pin-SVM (7) with a given τ is the sequential minimization optimization algorithm (SMO), which is designed for C-SVM by [6][37][38][39] and has been modified for pin-SVM by [35]. Roughly speaking, we need about 200 times of C-SVM to select the suitable τ from $\{-1, -0.99, \ldots, 0.99\}$ by SMO. From the result reported in Table I, the ratio between computation time of solving C-SVM by SMO and that of the traversal algorithm is far less than 200, showing the efficiency of the proposed traversal algorithm. Generally, the computational time of the traversal algorithm is acceptable and selecting a suitable τ (particularly, for some applications, the best τ is indeed negative) can improve the accuracy, when the problem size is not large.

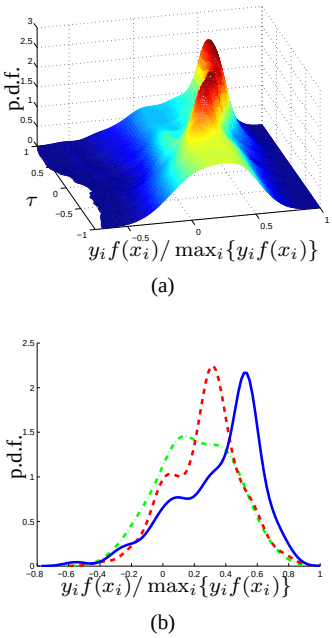


Fig. 7. The probability density functions of $y_i f(x_i)/\max_i\{y_i f(x_i)\}$ for Monk1. (a) p.d.f. for different τ; (b) p.d.f. for $\tau = -0.5$ (green dot-dashed line), $\tau = 0$ (red dashed line), and $\tau = 0.5$ (blue solid line).

As explained before, C-SVM is a special case of pin-SVM with $\tau = 0$. Thus, considering different τ values could improve the performance from C-SVM almost surely in the view of validation error. But when the data size increases, the performance varying for different τ values becomes less significant. In other words, for a large range of τ, the classification performance are almost the same. Moreover, the computational efficiency of the traversal algorithm decreases with the data scale increasing, since the main factor of computing time for the traversal algorithm is the number of segments in the solution path. Therefore, when there are plenty of training data, we suggest $\tau = 0$, i.e., C-SVM. If one wants to further improve the accuracy, we can consider several small values, e.g., $\tau = -0.02, -0.01, 0, 0.01, 0.02$, and use SMO to solve the corresponding pin-SVM, which is the strategy used in our experiments when the number of training data $m \geq 10000$.

To further observe the role of different τ values, we plot p.d.f. of $y_i f(x_i)/\max_i\{y_i f(x_i)\}$ for four real data sets in Fig.8. Curves in different color correspond to the results of pin-SVM with different τ values: the RGB vector is set to be $[0.5 - \tau/2, 0, 0.5 + \tau/2]$. Generally, the result for different τ values may vary a lot, especially when the data size is small, e.g., in Fig.8(a) and Fig.8(b), there are only 124 and 169 training data, respectively. While, in Fig.8(d), which corresponds to data set "RNA", the p.d.f. for $\tau \geq 0$ are hard to distinguish. This phenomenon also can be observed in Table I, where the improvement achieved by tuning τ becomes less significant when the problem size increases. Roughly speaking, if there not enough training data, drawing information from correctly classified points is helpful. In fact, dealing with small training set is the purpose of other loss functions with non-zero value on correctly classified points, such as the distance weighted discriminant [30] and robust one-bit compressive sensing [33]. As reported in Table I, for small-scale problems, suitably selecting a non-zero τ can improve the classification accuracy, hence, it is worthy to use the traversal algorithm, which takes a longer time but gives a better result.

TABLE I
TEST ACCURACY, THE SUITABLE τ VALUES SELECTED BY CROSS VALIDATION, AND COMPUTATION TIME

Data	m	boosted tree	C-SVM	pin-SVM	τ
Spect	80	80.00	82.55	85.53	-0.44
		0.116 s	0.008 s	0.063 s	
Monk1	124	73.75	82.18	84.17	0.37
		0.211 s	0.094 s	0.241 s	
Monk2	169	77.53	84.54	85.12	0.03
		0.380 s	0.029 s	0.390 s	
Monk3	122	91.80	92.96	95.44	0.28
		0.405 s	0.028 s	0.216 s	
Haber.	150	72.68	73.32	73.40	0.02
		0.382 s	0.023 s	0.293 s	
Statlog	150	81.31	82.31	83.00	-0.11
		0.387 s	0.034 s	0.303 s	
Iono.	200	87.33	93.22	93.75	0.11
		40.73 s	0.042 s	0.567 s	
Pima	300	74.61	73.20	74.00	0.23
		0.140 s	0.048 s	0.767 s	
Breast	500	96.87	96.78	97.30	-0.13
		2.037 s	0.072 s	2.675 s	
Trans.	500	76.19	74.45	76.97	-0.44
		1.590 s	0.144 s	0.925 s	
Splice	500	89.73	86.51	86.77	0.10
		1.743 s	0.213 s	4.163 s	
Guide1	1000	96.42	96.44	96.88	0.06
		1.731 s	0.783 s	7.350 s	
Spamb.	3000	94.03	96.70	96.78	0.03
		3.205 s	0.939 s	33.56 s	
RNA	10000	95.12	96.16	96.18	0.02
		36.10 s	13.18 s	63.16 s	
Magic	15000	85.25	87.25	87.25	-0.02
		50.10 s	21.21 s	107.3 s	
IJCNN1	20000	92.89	97.13	97.13	0.00
		46.10 s	25.24 s	121.1 s	

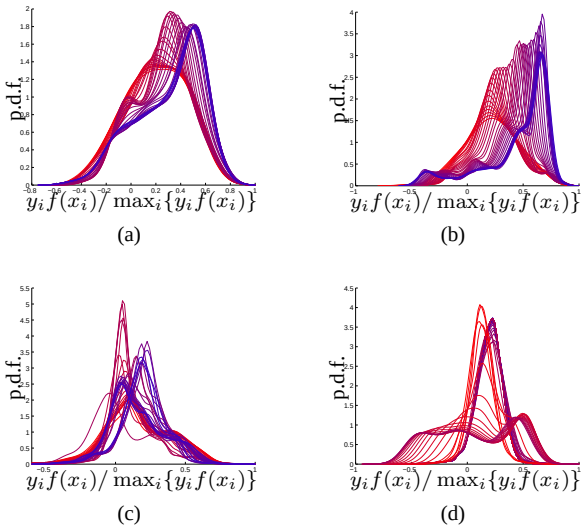


Fig. 8. The probability density functions of $y_i f(x_i)/\max_i\{y_i f(x_i)\}$ for different τ values. When τ is smaller, the color has more red components and for a larger τ, the corresponding color has more blue components. (a) Monk1; (b) Monk2; (c) Spamb.; (d) RNA.

V. Conclusion

Support vector machines with the pinball loss have a tuning parameter τ, which is the slope of the loss function. τ also controls the feasible set in the dual space. Traditionally, τ is fixed to be zero (corresponding to C-SVM) or a positive τ. However, the nonnegativity condition on τ is not necessary and only $\tau \geq -1$ is needed to keep the problem convex. In this paper, we extended pin-SVM to negative τ values, which encourages the correctly classifier points going away from the decision boundary via giving gains according to the distance. The meaning of negative τ values are explained from both primal and dual space. One interesting observation is that the optimal variables for $\tau = -1$ are easily calculated and that loss is closely linked with classical kernel rule.

Extending pin-SVM to $\tau \geq -1$ requires an efficient method for tuning τ. In this paper, we established an algorithm to traverse the entire path from $\tau = -1$, which is based on the fact that the solution path of pin-SVM is piecewise linear w.r.t. τ. In numerical experiments, the traversal algorithm shows the effectiveness and can improve the classification accuracy, especially for small-scale problems.

The theoretical understanding for the pinball loss with negative τ is a difficult but interesting topic. The main difficulty is that we cannot analyze the loss functions which may take negative values independently from the function space. In fact, investigating its property in a bounded function space is meaningful for non-negative loss functions as well, because in practice a loss function is always minimized together with a regularization term.

Acknowledgment

The authors are grateful to anonymous reviewers for their most insightful comments.

References

[1] C. Cortes and V. Vapnik, "Support-vector networks", *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[2] V. Vapnik, *Statistical Learning Theory*. Wiley, New York, 1998.

[3] B. Schölkopf and A.J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, Cambridge, 2002.

[4] F. Cucker and D.-X. Zhou, *Learning Theory: An Approximation Theory Viewpoint*, Cambridge University Press, 2007.

[5] I. Steinwart and A. Christmann, *Support Vector Machines*, Springer, 2008.

[6] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods – Support Vector Learning*. MIT Press, Cambridge, 1999, pp. 185–208.

[7] K.-W. Chang, C.-J. Hsieh, and C.-J. Lin, "Coordinate descent method for large-scale l2-loss linear support vector machines", *The Journal of Machine Learning Research*, vol. 9, pp. 1369–1398, 2008.

[8] K. De Brabanter, P. Karsmakers, F. Ojeda, C. Alzate, J. De Brabanter, K. Pelckmans, B. De Moor, J. Vandewalle, and J.A.K. Suykens, "LS-SVMlab Toolbox User's Guide version 1.8," Internal Report 10-146, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2010.

[9] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, arical 27, 2011.

[10] G.-X. Yuan, C.-J. Ho, and C.-J. Lin, "Recent advances of large-scale linear classification", *Proceedings of the IEEE*, vol. 100, no. 9, pp. 2584–2603, 2012.

[11] G. Lugosi and N. Vayatis, "On the Bayes-risk consistency of regularized boosting methods", *Annals of Statistics*, vol. 32, no. 1, pp. 30–55, 2004.

[12] P. Bartlett, M. Jordan, and J.D. McAuliffe, "Convexity, classification, and risk bounds", *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 138–156, 2006.

[13] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines," *Advances in Neural Information Processing Systems*, vol. 16, pp. 49–56, 2004.

[14] H. Zou and T. Hastie, and R. Tibshirani, "Regression shrinkage and selection via the elastic net, with applications to microarrays," *Journal of the Royal Statistical Society: Series B*, vol. 67, pp. 301–320, 2003.

[15] Q. Song, W. Hu, and W. Xie, "Robust support vector machine with bullet hole image classification", *IEEE Transactions on System, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 32, no. 4, pp. 440–448, 2002.

[16] G. Lanckriet, L. Ghaoui, C. Bhattacharyya, and M. Jordan, "A robust minimax approach to clssification", *Journal of Machine Learning Research*, vol. 3, pp. 555–582, 2003.

[17] P. Shivaswamy, C. Bhattacharyya, and A. Smola, "Second order cone programming approaches for handling missing and uncertain data", *Journal of Machine Learning Research*, vol. 7, pp. 1283–1314, 2006.

[18] R. Koenker, *Quantile Regression*. Cambridge University Press, 2005.

[19] I. Steinwart and A. Christmann, *How SVMs can estimate quantile and the median*. in *Advances in Neural Information Processing Systems*, 2008, pp. 585–592.

[20] I. Steinwart and A. Christmann, "Estimating conditional quantiles with the help of the pinball loss," *Bernoulli*, vol. 17, no. 1, pp. 211–225, 2011.

[21] X. Huang, L. Shi, and J.A.K. Suykens, "Support vector machine classifier with pinball loss," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36. no. 5, pp. 984–997, 2014.

[22] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu, "The entire regularization path for the support vector machine," *Journal of Machine Learning Research*, vol. 5, pp. 1391–1415, 2004.

[23] S. Rosset and J. Zhu, "Piecewise linear regularized solution paths," *The Annals of Statistics*, vol. 35, no. 3, pp. 1012–1030, 2007.

[24] C. Ong, S. Shao, and J. Yang, "An improved algorithm for the solution of the regularization path of support vector machine," *IEEE Transactions on Neural Networks*, vol. 21, no. 3, pp. 451–462, 2010.

[25] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[26] M. Park and T. Hastie, "L1-regularization path algorithm for generalized linear models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 4, pp. 659–677, 2007.

[27] T. Hesterberg, N. Choi, L. Meier, and F. Chris, "Least angle and l_1 penalized regression: A review," *Statistics Surveys*, vol. 2, pp. 61–93, 2008.

[28] L. Mason, J. Baxter, P. Bartlett, and M. Frean, *Boosting algorithms as gradient descent in function space*. in *Advances in Neural Information Processing Systems*, vol., 12, pp. 512–518, 2000.

[29] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)", *The Annals of Statistics*, vol. 28. no. 2, pp. 337–407, 2000.

[30] J.S. Marron, M.J. Todd, and J. Ahn, "Distance-weighted discrimination", *Journal of the American Statistical Association*, vol. 102. no. 480, pp. 1267–1271, 2007.

[31] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, Springer-Verlag, New York, 1996.

[32] C. Abraham, G. Biau, and B. Cadre, "On the kernel rule for function classification", *Annals of the Institute of Statistical Mathematics*, vol. 58, no. 3, pp. 619–633, 2006.

[33] Y. Plan and R. Vershynin, "Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach", *IEEE Transactions on Information Theory*, vol. 59, no. 1, pp. 482–494, 2013.

[34] A. Ai, A. Lapanowski, Y. Plan, and R. Vershynin, "One-bit compressed sensing with non-Gaussian measurements", *Linear Algebra and its Applications*, vol. 441, pp. 222–239, 2014.

[35] X. Huang, L. Shi, and J.A.K. Suykens, "Sequential minimial optimization for SVM with pinball loss", *Neurocomputing*, vol. 149, pp. 1596–1603, 2015

[36] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[37] R.-E. Fan, P.-H. Chen, and C.-J. Lin, "Working set selection using second order information for training support vector machines," *The Journal of Machine Learning Research*, vol. 6, pp. 1889–1918, 2005.

[38] L. Bottou and C.-J. Lin, "Support vector machine solvers," in *Large Scale Kernel Machines*, MIT Press, Cambridge, 2007, pp. 301–320.

[39] J. Lopez and J. Dorronsoro, “Simple proof of convergence of the SMO algorithm for different SVM variants,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1142–1147, 2012.
[40] Y. Freund and R. Schapire, “A desicion-theoretic generalization of on-line learning and an application to boosting,” in proceedings of *Computational Learning Theory: Second European Conference*, pp. 23–37, 1995.
[41] D. Coppersmith, S. June, and J. Hosking, “Partitioning nominal attributes in decision trees,” *Data Mining and Knowledge Discovery*, vol. 3, no. 2, pp. 197–217, 1999.