Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Learning SVM Classifiers with Indefinite Kernels

Suicheng Gu and **Yuhong Guo**
Department of Computer and Information Sciences
Temple University
Philadelphia, PA 19122, USA
yuhong@temple.edu

Abstract

Recently, training support vector machines with indefinite kernels has attracted great attention in the machine learning community. In this paper, we tackle this problem by formulating a joint optimization model over SVM classifications and kernel principal component analysis. We first reformulate the kernel principal component analysis as a general kernel transformation framework, and then incorporate it into the SVM classification to formulate a joint optimization model. The proposed model has the advantage of making consistent kernel transformations over training and test samples. It can be used for both binary classification and multi-class classification problems. Our experimental results on both synthetic data sets and real world data sets show the proposed model can significantly outperform related approaches.

Introduction

Support vector machines (SVMs) with kernels have attracted a lot attention due to their good generalization performance. The kernel function in a standard SVM produces a similarity kernel matrix over samples, which is required to be positive semi-definite. This positive semi-definite property of the kernel matrix ensures the SVMs can be efficiently solved using convex quadratic programming. However, in many applications the underlying similarity functions do not produce positive semi-definite kernels (Chen et al. 2009). For example, the sigmoid kernels with various values of the hyper-parameters (Lin and Lin 2003), the hyperbolic tangent kernels (Smola, Ovari, and Williamson 2000), and the kernels produced by protein sequence similarity measures derived from Smith- Waterman and BLAST scores (Saigo et al. 2004) are all indefinite kernels. Training SVMs with indefinite kernels poses a challenging optimization problem since convex solutions for standard SVMs are not valid in this learning scenario.

Learning with indefinite kernels has been addressed by many researchers in various ways in the literature. One most simple and popular way to address the problem is to identify a corresponding positive semi-definite kernel matrix by modifying the spectrum of the indefinite kernel matrix (Wu, Chang, and Zhang 2005). Several simple representative spectrum modification methods have been proposed in the literature, including "*clip*" (or "*denoise*") which neglects the negative eigenvalues (Graepel et al. 1999; Pekalska, Paclik, and Duin 2001), "*flip*" which flips the sign of the negative eigenvalues (Graepel et al. 1999), and "*shift*" which shifts all the eigenvalues by a positive constant (Roth et al. 2003). More sophisticated approaches simultaneously derive a positive semi-definite kernel matrix from the given indefinite kernel matrix and train a SVM classifier within unified optimization frameworks (Chen and Ye 2008; Chen, Gupta, and Recht 2009; Luss and d'Aspremont 2007). A few other works use indefinite similarity matrices as kernels directly by formulating variant optimization problems from standard SVMs. In (Lin and Lin 2003), a SMO-type method is proposed to find stationary points for the nonconvex dual formulation of SVMs with nonpositive semi-definite sigmoid kernels. This method, however, is based on the assumption that there is a corresponding reproducing kernel Hilbert space to ensure valid SVM formulations. The work in (Ong et al. 2004) interprets learning with an indefinite kernel as minimizing the distance between two convex hulls in a pseudo-Euclidean space. In (Pekalska and Haasdonk 2008), the authors extended the kernel linear and quadratic discriminants to indefinite kernels. The approach in (Guo and Schuurmans 2009) minimizes the sensitivity of the classifier to perturbations of the training labels, which yields an upper bound of classical SVMs.

In this paper, we propose a novel joint optimization model over SVM classifications and kernel principal component analysis to address the problem of learning with indefinite kernels. We first reformulate the kernel principal component analysis (KPCA) into a general kernel transformation framework which can incorporate the spectrum modification methods. Next we incorporate this framework into the SVM classification to formulate a joint max-min optimization model. Training SVMs with indefinite kernels can then be conducted by solving the joint optimization problem using an efficient iterative algorithm. Different from many related approaches, our proposed model has the advantage of making consistent transformations over training and test samples. The experimental results on both synthetic data sets and real world data sets demonstrated the proposed model can significantly outperform the spectrum modification methods, the robust SVMs and the kernel Fisher's discriminant on indefinite kernels (IKFD).

Related Work

The dual formulation of standard SVMs is a linear constrained quadratic programming, which provides a natural form to address nonlinear classification using kernels

$$\begin{aligned} \max_{\alpha} \quad & \alpha^\top e - \frac{1}{2}\alpha^\top Y K_0 Y \alpha \qquad (1) \\ \text{s.t.} \quad & \alpha^\top \text{diag}(Y) = 0, \;\; 0 \le \alpha \le C \end{aligned}$$

where Y is a diagonal matrix of the labels, and K_0 is a kernel matrix. The positive semi-definite property of K_0 ensures the problem (1) to be a convex optimization problem and thus a global optimal solution can be solved efficiently. However, when K_0 is indefinite, one loses the underlying theoretical support for the kernel methods and the optimization problem (1) is no longer convex. For the nonconvex optimization problem (1) with indefinite kernels, with a simple modification, a sequential minimal optimization (SMO) algorithm can still converge to a stationary point, but not necessarily a global maximum (Lin and Lin 2003).

Instead of solving the quadratic optimization problem (1) with indefinite kernels directly, many approaches are focused on deriving a surrogate positive semi-definite kernel matrix K from the indefinite kernel K_0. A simple and popular way to obtain such a surrogate kernel matrix is to modify the spectrum of K_0 using methods such as *clip*, *flip*, and *shift* (Wu, Chang, and Zhang 2005). Let $K_0 = U \Lambda U^\top$, where $\Lambda = diag(\lambda_1, \ldots, \lambda_N)$ is the diagonal matrix of the eigenvalues, and U is the orthogonal matrix of corresponding eigenvectors. The *clip* method produces an approximate positive semi-definite kernel K_{clip} by clipping all negative eigenvalues to zero,

$$K_{clip} = U \text{diag}(\max(\lambda_1, 0), \cdots, \max(\lambda_N, 0)) U^\top. \qquad (2)$$

The *flip* method flips the sign of negative eigenvalues of K_0 to form a positive semi-definite kernel matrix K_{flip}, such that

$$K_{flip} = U \text{diag}(|\lambda_1|, \cdots, |\lambda_N|) U^\top. \qquad (3)$$

The *shift* method obtains the positive semi-definite kernel matrix K_{shift} by shifting the whole spectrum of K_0 with the minimum required amount η, such that

$$K_{shift} = U \text{diag}(\lambda_1 + \eta, \ldots, \lambda_N + \eta) U^\top. \qquad (4)$$

These spectrum modification methods are straightforward and simple to use. However, some information valuable for the classification model might be lost by simply modifying the spectrum of input kernels. Therefore, approaches that simultaneously train the classification model and learn the approximated positive semi-definite kernel matrix have been developed (Chen and Ye 2008; Chen, Gupta, and Recht 2009; Luss and d'Aspremont 2007). In (Luss and d'Aspremont 2007) a robust SVM with indefinite kernels was proposed, which treats the indefinite kernel as a noisy observation of the true positive semi-definite kernel and solves the following convex optimization problem

$$\begin{aligned} \max_{\alpha} \min_{K} \quad & \alpha^\top e - \frac{1}{2}\alpha^\top Y K Y \alpha + \rho \|K - K_0\|_F^2 \qquad (5) \\ \text{s.t.} \quad & \alpha^\top \text{diag}(Y) = 0; \quad 0 \le \alpha \le C; \quad K \succeq 0 \end{aligned}$$

where a positive semi-definite kernel K is introduced to approximate the original K_0, and ρ controls the magnitude of the penalty on the distance between K and K_0. An analysis about the indefinite SVM of (5) was conducted in (Ying, Campbelly, and Girolami 2009), which shows the objective function is smoothed by the penalty term. In (Chen and Ye 2008), Chen and Ye reformulated (5) into a semi-infinite quadratically constrained linear program formulation, which can be solved iteratively to find a global optimal solution. They further employed an additional pruning strategy to improve the efficiency of the algorithm.

Many approaches mentioned above treat training and test samples in an inconsistent way. That is, the training is conducted on the proxy positive semi-definite kernel matrix K, but the predictions on test samples are still conducted using the original unmodified similarities. This is an obvious drawback that could degrade the performance of the produced classification model. Wu et al. (Wu, Chang, and Zhang 2005) addressed this problem for the case of spectrum modifications by recomputing the spectrum modification on the matrix that augments K_0 with similarities on test samples. Chen et al. (Chen, Gupta, and Recht 2009) addressed the problem of learning SVMs with indefinite kernels using the primal form of Eq.(5) while further restricting K to be a spectrum modification of K_0. They then obtained the consistent treatment of training and test samples by solving a positive semi-definite minimization problem over the distance between augmented K_0 and K matrices. The model we propose in this paper however can address this inconsistency problem in a more principled way without solving additional optimization problems.

Kernel Principal Component Analysis

In this section, we present the kernel principal component analysis (KPCA) as a kernel transformation method and then demonstrate its connection to spectrum modification methods. Let $X = \{\mathrm{x}_i\}_{i=1}^N$ denote the training samples, where $\mathrm{x}_i \in \mathbb{R}^n$. To employ kernel techniques, a mapping function, $\phi : \mathbb{R}^n \to \mathbb{R}^f$, can be deployed to map the data to a typically high dimensional space. The training samples in this mapped space can be represented as $\Phi = [\phi(x_1), \ldots, \phi(x_N)]$ and the standard kernel matrix can be viewed as the inner product of the sample matrix in the high dimensional space, $K_0 = \Phi^\top \Phi$.

KPCA

The kernel principal component analysis (Schölkopf, Smola, and Muller 1999) can be solved by minimizing the distance between the high dimensional data matrix and the reconstructed data matrix

$$\min_{W} \; \left\| \Phi - W W^\top \Phi \right\|_F^2, \quad \text{s.t. } W^\top W = I_d \qquad (6)$$

where W, a $f \times d$ matrix, can be viewed as a transformation matrix that transforms the data samples to a lower d-dimensional subspace $Z = W^\top \Phi$; $\|\cdot\|_F$ denotes the Frobenius norm; and I_d denotes a $d \times d$ identity matrix. This minimization problem is equivalent to

$$\max_W \; tr(W^\top \Phi\Phi^\top W), \quad \text{s.t. } W^\top W = I_d \tag{7}$$

which has a closed form solution $W = U_d$, where U_d is the top d eigenvectors of $\Phi\Phi^\top$. Moreover we have $\Phi\Phi^\top W \Lambda_d^{-1} = W$, where Λ_d is a $d \times d$ diagonal matrix with its diagonal values as the top d eigenvalues of $\Phi\Phi^\top$. Here we assumed the top d eigenvalues are not zeros. Let $V = \Phi^\top W \Lambda_d^{-1}$, then we have $W = \Phi V$ and (7) can be reformulated into

$$\max_V \; tr(V^\top K_0 K_0 V), \quad \text{s.t. } V^\top K_0 V = I_d. \tag{8}$$

After solving the optimization problem above for the V matrix, the transformation matrix, W, and the low dimensional map of the training samples, Z, can be obtained consequently. Then the transformed kernel matrix for the training samples in the low dimensional space can be produced

$$K_v = Z^\top Z = \Phi^\top W W^\top \Phi = K_0 V V^T K_0. \tag{9}$$

Although the standard kernel principal component analysis assumes the kernel matrix K_0 to be positive semi-definite, the optimization problem (8) we derived above can be generalized to the case of indefinite kernels if V is guaranteed to be a real valued matrix by selecting a proper d value. Even when K_0 is an indefinite kernel matrix, K_v is still guaranteed to be positive semi-definite for real valued V. Thus the equation (9) provides a principle strategy to transform an indefinite kernel matrix K_0 to a positive semi-definite matrix K_v with a proper selected V. Moreover, given a new sample x, it can be transformed by $W^\top \phi(x) = V^\top \Phi^\top \phi(x) = V^\top \mathrm{k}_0$, where k_0 denotes the original similarity vector between the new sample x and training samples. The transformed similarity vector between the new sample x and the training samples is $k_v = K_0 V V^T k_0$. By using this transformation strategy, we can easily transform the test samples and the training samples in a *consistent* way.

Connections to Spectrum Modifications

The kernel transformation strategy we developed above is a general framework. By selecting different V matrix, various kernel transformations can be produced. We now show that the spectrum modification methods reviewed in the previous section can be equivalently reexpressed as kernel transformations in the form of Eq.(9) with proper V matrices.

Assume $K_0 = U \Lambda U^\top$, where U is an orthogonal matrix and Λ is a diagonal matrix of real eigenvalues, that is, $\Lambda = diag(\lambda_1, \ldots, \lambda_N)$. The *clip* spectrum modification method can be reexpressed as

$$K_{clip} = K_0 V_{clip} V_{clip}^\top K_0 \tag{10}$$

for a constructed V_{clip} matrix

$$V_{clip} = U|\Lambda|^{-\frac{1}{2}} \mathrm{diag}\left(I_{\{\lambda_1 > 0\}}, \ldots, I_{\{\lambda_N > 0\}}\right) \tag{11}$$

where $|\Lambda| = \mathrm{diag}(|\lambda_1|, \ldots, |\lambda_N|)$, and $I_{\{\cdot\}}$ is an indicator function. The *flip* method can be reexpressed as

$$\begin{aligned} K_{flip} &= K_0 V_{flip} V_{flip}^\top K_0 & (12) \\ \text{for} \quad V_{flip} &= U|\Lambda|^{-\frac{1}{2}}. & (13) \end{aligned}$$

Similarly, the *shift* method is reexpressed as

$$\begin{aligned} K_{shift} &= K_0 V_{shift} V_{shift}^\top K_0 & (14) \\ \text{for} \quad V_{shift} &= U|\Lambda|^{-1}(\Lambda + \eta I)^{\frac{1}{2}}. & (15) \end{aligned}$$

Training SVMs with Indefinite Kernels

In this section, we address the problem of training SVMs with indefinite kernels by developing a joint optimization model over SVM classifications and KPCA. Our model simultaneously trains a SVM classifier and identifies a proper transformation V matrix. We present this model for binary classifications first and then extend it to address multi-class classification problems. An iterative optimization algorithm is developed to solve the joint optimizations.

Binary classifications

We first extend the standard two-class SVMs to formulate a joint optimization problem of SVMs and the kernel principal component analysis

$$\begin{aligned} \min_{\mathrm{W}, w, b, \xi} \quad & \frac{1}{2}\mathrm{w}^\top \mathrm{w} + C \sum_i \xi_i + \rho \left\| \Phi - W W^\top \Phi \right\|_F^2 & (16) \\ \text{s.t.} \quad & y_i(\mathrm{w}^\top W^\top \Phi(:, i) + b) \geq 1 - \xi_i, \;\; \xi_i \geq 0, \; \forall i; \\ & W^\top W = I_d; \end{aligned}$$

where $y_i \in \{+1, -1\}$ is the label of the ith training sample, $\Phi(:, i)$ is the ith column of the general feature matrix representation Φ, C is the standard tradeoff parameter in SVMs, and ρ is a parameter to control the tradeoff between the SVM objective and the reconstruction error of KPCA. Previous approaches in (Chen and Ye 2008; Chen, Gupta, and Recht 2009; Luss and d'Aspremont 2007) use the distance between the proxy kernel K and the original K_0 as a regularizer for SVMs. The joint optimization model proposed here can be similarly interpreted as employing the distance between the proxy and original feature vectors as a regularizer. However, for the problem of learning with indefinite kernels, the feature vectors are not real valued vectors and they are actually only available implicitly through kernel matrices. Therefore, we need to reformulate the optimization problem in terms of kernels.

By exploiting the derivation results in the previous section, we propose to replace the distance regularizer in (16) with a kernel transformation regularizer (8) to obtain an alternative joint optimization in terms of the input kernel

$$\begin{aligned} \min_{\mathrm{w}, b, \xi, V} \quad & \frac{1}{2}\mathrm{w}^\top \mathrm{w} + C \sum_i \xi_i - \rho \, tr(V^\top K_0 K_0 V) & (17) \\ \text{s.t.} \quad & y_i(\mathrm{w}^\top V^\top K_0(:, i) + b) \geq 1 - \xi_i, \;\; \xi_i \geq 0, \; \forall i; \\ & V^\top K_0 V = I_d; \;\; K_0 V V^\top K_0 \succeq 0. \end{aligned}$$

When V is constrained to be a real valued matrix, the constraint $K_0VV^TK_0 \succeq 0$ can be dropped. We will assume V has real values from now on. More conveniently, following the dual formulation of standard SVMs, we consider the regularized dual SVM formulation and focus on the following optimization problem

$$\begin{aligned} \max_{\alpha} \min_{V} \quad & \alpha^\top e - \frac{1}{2}\alpha^\top Y K_0 V V^T K_0 Y \alpha \qquad (18) \\ & -\rho\, tr(V^\top K_0 K_0 V) \\ \text{s.t.} \quad & \alpha^\top \text{diag}(Y) = 0;\ \ 0 \leq \alpha \leq C; \\ & V^\top K_0 V = I_d; \end{aligned}$$

where $Y = \text{diag}(y_1, \ldots, y_N)$ is a diagonal matrix.

Multi-class Classifications

Multi-class classification problems can be solved by training multiple binary SVMs (Hsu and Lin 2002). In this paper, we deploy the 1-vs-1 strategy for multi-class classification. A simple deployment of this strategy requires training $k(k-1)/2$ binary classifiers, each for a pair of classes, for a k-class problem. This means that an optimization problem (18) has to be solved for each pair of classes and a different proxy kernels K_{ab} will be learned for each pair of classes $\{a, b\}$, by learning a different V_{ab}. However, with different proxy kernels K_{ab} for each pair of classes, the consistent transformation of samples for the overall multi-class classification cannot be maintained. To ensure a data sample has a consistent representation in all binary classification problems, we construct a framework to use a single target (proxy) kernel matrix K_v for all binary classifications by introducing a set of sub-kernel transformation matrix $\{D_{ab}\}_{1\leq a<b\leq k}$ and address all the $k(k-1)/2$ binary classifications in one joint optimization.

Assume the training set has N samples and each class a has N_a samples. We first consider a given pair of classes a and b. Let $N_{ab} = N_a + N_b$ be the sample size of the class set $\{a, b\}$, $L^{ab} = [\ell_1^{ab}, \ldots, \ell_{N_{ab}}^{ab}]$ denote a $1 \times N_{ab}$ vector whose jth entry is the index value for the jth sample of the class set $\{a, b\}$ in the original training set, and K_{ab} denote the proxy kernel matrix of the samples in these two classes. Thus the proxy kernel K_v of all training samples is a $N \times N$ matrix, and the K_{ab}, a $N_{ab} \times N_{ab}$ matrix, is its sub-matrix. We now construct an indicator matrix $D_{ab} \in \{0, 1\}^{N \times N_{ab}}$ as below to build a connection between these two kernel matrices

$$D_{ab}(i, j) = \left\{ \begin{array}{ll} 1, & \text{if } \ell_j^{ab} = i \\ 0, & \text{otherwise.} \end{array} \right. .$$

Given D_{ab}, the kernel matrix K_{ab} of class a and b can be computed as

$$K_{ab} = K_v(L^{ab}, L^{ab}) = D_{ab}^\top K_v D_{ab}. \qquad (19)$$

Thus D_{ab} can be viewed as a sub-kernel transformation matrix. Note that $K_v = K_0VV^\top K_0$. Then we can combine the $k(k-1)/2$ classifications in a joint optimization problem

$$\begin{aligned} \max_{\alpha} \min_{V} \quad & -\rho\, tr(V^\top K_0 K_0 V) + \sum_{1\leq a<b\leq k} \Big(\alpha_{ab}^\top e - \\ & \frac{1}{2}\alpha_{ab}^\top Y_{ab} D_{ab}^T K_0 V V^T K_0 D_{ab} Y_{ab} \alpha_{ab}\Big) \qquad (20) \\ \text{s.t.} \quad & \alpha_{ab}^\top \text{diag}(Y_{ab}) = 0,\ \ \forall 1 \leq a < b \leq k; \\ & 0 \leq \alpha_{ab} \leq C,\ \ \forall 1 \leq a < b \leq k; \\ & V^\top K_0 V = I_d \end{aligned}$$

where α denotes a collection of $\{\alpha_{ab}\}_{1\leq a<b\leq k}$, and Y_{ab} is a diagonal matrix whose diagonal entries are the binary labels for the binary classification problem over classes $\{a, b\}$. When $k = 2$, the binary classification problem in (18) can be recovered from (20).

An Iterative Algorithm

The objective of the outer maximization problem in (20) is a pointwise minimum of a family of concave quadratic functions of α, and hence is a concave function of α. Thus (20) is a concave maximization problem over α subject to linear constraints (Boyd and Vandenberghe 2004). In this section, we develop an iterative algorithm to solve the optimization problem (20). In each iteration of the algorithm, V and α are alternatively optimized. When V is fixed, we can divide the maximization problem into $k(k-1)/2$ standard binary SVMs and optimize each α_{ab} independently. When $\{\alpha_{ab}\}_{1\leq a<b\leq k}$ are fixed, V can be computed by solving the following optimization problem

$$\max_{V}\ \ \text{tr}(V^\top K_0 M K_0 V) \quad \text{s.t. } V^\top K_0 V = I_d \qquad (21)$$

where

$$M = \frac{1}{2}\sum_{1\leq a<b\leq k} (D_{ab} Y_{ab} \alpha_{ab} \alpha_{ab}^\top Y_{ab} D_{ab}^\top) + \rho I_N. \qquad (22)$$

The above problem can be solved via the following general eigenvalue problem,

$$K_0 M K_0 \mathrm{v} = \lambda K_0 \mathrm{v}. \qquad (23)$$

Note that for positive ρ values, M is guaranteed to be positive definite. Thus we will solve the following eigenvalue problem instead

$$\begin{aligned} M K_0 M K_0 \mathrm{v} &= \lambda M K_0 \mathrm{v}, \qquad (24) \\ M K_0 \mathrm{u} &= \lambda \mathrm{u}, \qquad (25) \end{aligned}$$

for $\mathrm{u} = MK_0\mathrm{v}$. Moreover, we assume K_0 is invertible [1]. Let $U = [\mathrm{u}_1, \ldots, \mathrm{u}_d]$ be the top d largest eigenvectors of MK_0, then $V = K_0^{-1}M^{-1}U$. Finally the optimal solution of (21) can be recovered by setting $V^* = [\mathrm{v}_1^*, \ldots, \mathrm{v}_d^*]$, where $\mathrm{v}_i^* = \mathrm{v}_i/\sqrt{\mathrm{v}_i^\top K_0 \mathrm{v}_i}$. Here the renormalization is necessary to ensure the orthogonal constraints in (21) for indefinite K_0.

Determining feasible d values. To ensure each v_i be real values, we should select d to guarantee that each u_i satisfies $\mathrm{u}_i^\top K_0 \mathrm{u}_i > 0$. To determine d, we have the following lemma

[1] It is easy to remove the zero eigenvalues of K_0 by simply adding a tiny positive/negative diagonal matrix ϵI_N without changing the distribution of K_0's eigenvalues.

Lemma 1 *For each eigenpair, (λ_i, u_i), of MK_0, if $\lambda_i > 0$, then we have $u_i^\top K_0 u_i > 0$.*

Proof: Since $MK_0\mathrm{u}_i = \lambda_i \mathrm{u}_i$, we have

$$\mathrm{u}_i^\top K_0 M K_0 \mathrm{u}_i = \lambda_i \mathrm{u}_i^\top K_0 \mathrm{u}_i.$$

$$\text{Then} \qquad \mathrm{u}_i^\top K_0 \mathrm{u}_i = (\mathrm{u}_i^\top K_0 M K_0 \mathrm{u}_i)/\lambda_i.$$

Since both K_0MK_0 and K_0 are symmetric matrices, u_i has real values. Moreover K_0MK_0 is positive semi-definite according to (22). Therefore $\frac{\mathrm{u}_i^\top K_0 M K_0 \mathrm{u}_i}{\lambda_i} > 0$ □.

According to Lemma 1, the top d eigenvectors $\{\mathrm{v}_i^*\}_{1 \le i \le d}$ have real values, if $d \le d_0$, where d_0 is the number of positive eigenvalues of MK_0. As we discussed before, M is guaranteed to be positive definite for positive ρ values, and we assume K_0 is invertible. It is easy to show MK_0 and $M^{\frac{1}{2}} K_0 M^{\frac{1}{2}}$ have the same eigenvalues by using a similar transformation from (23) to (24). According to the Sylvester law of inertia (Golub and Loan 1996), $M^{\frac{1}{2}} K_0 M^{\frac{1}{2}}$ and K_0 have the same inertia, and thus have the same number of positive eigenvalues. Therefore the value d_0 can be determined directly from K_0.

Experiments

We conducted experiments on both synthetic data sets and real world data sets to compare the proposed method, denoted as SVM-CA, with a few spectrum modification methods (*clip*, *flip*, and *shift*), the robust SVM (Luss and d'Aspremont 2007), and the kernel Fisher's discriminant on indefinite kernels (IKFD) (Pekalska and Haasdonk 2008). We used the robust SVM code found on the authors' website[2]. In the experiments below, the regularization parameter ρ for SVM-CA, robust SVM and IFKD, the parameter C in SVMs, the reduced dimensionality d in SVM-CA were all selected by 10-fold cross-validations from the following candidate sets, $\rho, C \in \{0.01, 0.1, 1, 10, 100\}$, and $d \in \{2, 3, 5, 8, 13, 21, 34, 55\}$.

Experiments on Synthetic Data Sets

We constructed four 3-class 2-dimensional data sets, each with 300 samples. For each data set, the three classes, each with 100 samples, are generated using three Gaussian distributions with the covariance matrix $\Lambda = \mathrm{diag}(\sigma^2, \sigma^2)$ and mean vectors $\mu_1 = (-3, 3)$, $\mu_2 = (3, -3)$ and $(3\sqrt{3}, 3\sqrt{3})$, respectively. We generate the similarity kernel matrix by adding additive white Gaussian noise to the linear kernel matrix, $K_0(i, j) = \mathrm{x}_i^T \mathrm{x}_j + z_{ij}$, where $z_{ij} \sim N(0, \eta)$. With the Gaussian noise, the kernel K_0 is not positive semi-definite.

By considering different σ^2 and η values, synthetic data sets with different properties can be generated. We considered two values for σ^2, $\sigma^2 = 2$ and $\sigma^2 = 4$, and two different η values, $\eta = 20$ and $\eta = 100$. With larger σ^2 value, the generated data is harder to be separable. With larger η, the kernel matrix K_0 can be more indefinite. With different pairs of (σ^2, η), we obtained four synthetic data sets. The characteristics of the data sets are given in Table 1, where λ_{min} and λ_{max} are the smallest and largest eigenvalues of each synthetic indefinite kernel matrix K_0, respectively, $\sum \lambda_i^+$ and $\sum \lambda_j^-$ are the sums of the positive and negative eigenvalues of K_0, respectively.

We run experiments on the four synthetic data sets comparing the SVM-CA to the other five approaches. Our experimental results in terms of classification error rates are reported in Table 1. These results are averaged over 50 runs using 80% of the data as training set and the remainder as test set. It is apparent that the values of σ^2 and η determine the hardness of the classification problems, and thus affect the performance of these approaches. When either σ^2 or η gets larger, the error rate for each approach increases. When η is small, the spectrum modification methods, especially the spectrum clip, yield good performance. When η is large, which means the kernel K_0 is far away from being positive semi-definite, the spectrum modifications are inefficient to capture the information provided by the indefinite kernels and thus produce inferior results. Among the three spectrum modification approaches, the clip method obtains the lowest error rates on all the four data sets. The robust SVM is highly related to the spectrum clip, and it yields similar results as the clip method. Both IKFD and SVM-CA approaches obtain much better results than the other four approaches. They produced good results even on the data sets with large η and large σ^2. Overall, the proposed SVM-CA produced the best results comparing to all the other approaches.

Experiments on Real World Data Sets

We then conducted experiments on several real world data sets used for learning with indefinite kernels, including a few data sets used in (Chen et al. 2009), i.e., *yeast, amazon, aural sonar, voting, patrol* and *protein*, and a data set collected in (Pekalska and Haasdonk 2008), i.e., *catcortex*. These data sets are represented by similarity (or dissimilarity) matrices produced using different similarity measures. For example, a sequence-alignment similarity measure is used for the *protein* data set, the Smith-Waterman E-value is used to measure the similarity between two protein sequences for the *yeast* data set, etc. We also used the *glass* data set obtained from the UCI machine learning repository (Newman et al. 1998), for which we used a sigmoid kernel to compute an indefinite kernel matrix K_0. These data sets together represent a diverse set of indefinite similarities. We assumed symmetric similarity kernel matrix K_0 in our proposed model. When the original matrix K_0 given in the data is not symmetric, we reset it as $K_0 = (K_0^\top + K_0)$. When the original matrix K_0 in the data represents dissimilarity, we just reset it as $K_0 = m - K_0$, where m is the largest entry of the original matrix K_0. There are six binary (two-class) and four multi-class data sets in total. We computed the eigenvalue information of the kernel matrix K_0 for each data set as well. The indefiniteness measure $|\frac{\sum \lambda_i^-}{\sum \lambda_j^+}|$ obtained for each data set is given as follows: (Yeast5v7: 0.56), (Yeast5v12: 0.56), (Yeast7v12: 0.57), (Amazon: 0.01), (Aural Sonar: 0.26), (Voting: 0.00), (Protein: 0.25), (Glass: 0.00), (Patrol: 0.36) and (Catcortex: 0.10). Here the value 0.00 denotes a positive value smaller than 0.005.

[2] http://www.tau.ac.il/~rluss/

Table 1: Characteristics of the four synthetic data sets and the average classification errors (%) of the six comparison methods.

Data	σ^2	η	λ_{min}	$\lvert\frac{\lambda_{min}}{\lambda_{max}}\rvert$	$\lvert\frac{\sum\lambda_i^-}{\sum\lambda_j^+}\rvert$	Clip	Flip	Shift	Robust SVM	IKFD	SVM-CA
Synth 1	2	20	-143	.02	.47	1.50	2.00	15.83	1.53	1.20	0.72
Synth 2	2	100	-693	.11	.82	9.67	11.00	22.33	9.05	2.43	1.83
Synth 3	4	20	-140	.02	.44	4.00	4.83	21.50	4.11	1.69	1.17
Synth 4	4	100	-702	.11	.80	16.17	16.67	38.17	15.24	4.70	3.50

Table 2: Comparison results in terms of classification error rates (%) on binary classification data sets. The means and standard deviations of the error rates over 50 random repeats are reported.

Dataset	Yeast5v7	Yeast5v12	Yeast7v12	Amazon	Aural Sonar	Voting
Clip+SVM	40.0±1.1	20.0±1.3	25.5±1.2	10.3±0.9	11.2±0.8	3.0±0.3
Flip+SVM	46.0±0.6	17.8±1.2	22.0±1.0	11.0±0.9	16.8±0.9	3.2±0.3
Shift+SVM	35.0±0.5	42.8±1.5	46.7±1.9	16.0±0.8	17.3±0.9	5.8±0.5
IKFD	34.2±1.0	17.5±1.0	14.0±1.0	15.6±0.9	**8.4±0.6**	5.7±0.3
Robust SVM	29.0±1.0	18.0±1.0	15.0±0.9	**8.8±0.8**	11.0±0.9	3.3±0.3
SVM-CA	**25.0±0.9**	**10.7±0.8**	**10.5±0.8**	9.5±0.9	8.6±0.6	**2.7±0.3**

We compared our proposed SVM-CA to the other five approaches on both the six binary data sets and the four multi-class data sets. The experimental results are reported in Table 2 and Table 3 respectively. These results are produced over 50 runs using randomly selected 90% of the data samples as training set and the remainder as test set. Both the average classification error rates and their standard deviations are reported. Among the three spectrum modification algorithms, the spectrum *clip* obtained the lowest error rates on five of the ten data sets, while spectrum *flip* and *shift* obtained the lowest error rates on four and one data sets, respectively. The robust SVM slightly outperformed the spectrum modifications on eight data sets. The IKFD outperformed the spectrum modifications on five data sets. Our proposed SVM-CA clearly outperformed all the other approaches and achieved the lowest classification error rates on four of the total six binary data sets and all the four multi-class data sets. On data sets such as Protein, Patrol and Catcortex, where the $|\frac{\sum\lambda_i^-}{\sum\lambda_j^+}|$ values are large, the improvements achieved by the proposed approach over the other SVM training methods are largely significant. These results on both synthetic and real world data sets demonstrated the effectiveness of the proposed joint optimization model.

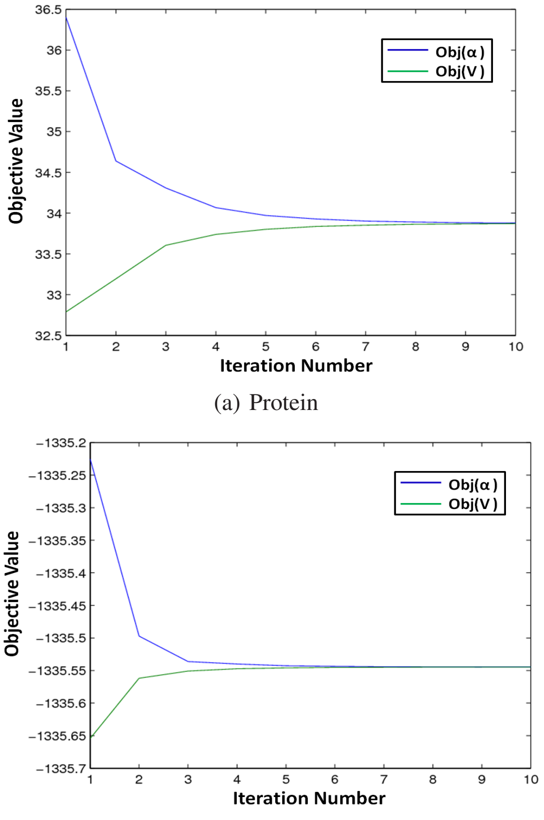
Convergence Experiments. We also conducted experiments to study the convergence property of the proposed iterative algorithm in Section 4.3. The experiments are conducted on two data sets *protein* and *catcortex*. In each experiment, we plot the objective values of the SVM-CA formulation in (20) after each update of V and α. The plots are shown in Figure 1. We can see that the objective values of the maximization and minimization sub-problems gradually converges within 10 iterations on the two data sets. This suggests the iterative algorithm we proposed can effectively solve the target convex optimization.

Table 3: Comparison results in terms of classification error rates (%) on multi-class classification data sets. The means and standard deviations of the error rates over 50 random repeats are reported.

Dataset	Protein	Glass	Patrol	Catcortex
Clip+SVM	6.3±0.7	41.1±1.2	48.6±1.5	10.5±2.0
Flip+SVM	4.0±0.7	39.4±1.1	44.8±1.4	13.5±2.3
Shift+SVM	5.5±0.7	38.3±0.9	51.4±1.5	49.0±4.0
IKFD	8.2±0.9	43.3±1.1	25.7±1.8	12.5±1.9
Robust SVM	16.4±1.1	39.1±1.0	31.3±1.4	9.4±1.7
SVM-CA	**2.5±0.5**	**37.3±0.8**	**12.4±0.8**	**4.5±1.4**

Conclusion

In this paper, we investigated the problem of training SVMs with indefinite kernels. We first reformulated the kernel principal component analysis (KPCA) to a kernel transformation model and demonstrated its connections to spectrum modification methods with indefinite kernels. We then presented a novel joint optimization model over SVM classifications and principal component analysis to conduct SVM training with indefinite kernels assisted by kernel component analysis. The proposed model can be used for both binary classifications and multi-class classifications. An efficient iterative algorithm was proposed to solve the proposed joint optimization problem. Moreover, the proposed approach can make consistent transformations over training and test samples. Our experimental results on both synthetic data sets and real world data sets demonstrated the proposed approach can significantly outperform the spectrum modification methods, the robust SVMs and the kernel Fisher's discriminant on indefinite kernels (IKFD).



(a) Protein

(b) Catcortex

Figure 1: Convergence of SVM-CA on the Protein and Catcortex data sets. The **Obj**(α) and **Obj**(V) denote the objective values after updating V and α, respectively, at each iteration.

References

Boyd, S., and Vandenberghe, L. 2004. *Convex Optimization*. Cambridge University Press.

Chen, J., and Ye, J. 2008. Training SVM with indefinite kernels. In *Proceedings of International conference on Machine Learning (ICML)*.

Chen, Y.; Garcia, E.; Gupta, M.; Rahimi, A.; and Cazzanti, L. 2009. Similarity-based classification: Concepts and algorithms. *Journal of Machine Learning Research* 10:747–776.

Chen, Y.; Gupta, M.; and Recht, B. 2009. Learning kernels from indefinite similarities. In *Proceedings of International conference on Machine Learning (ICML)*.

Golub, G., and Loan, C. V. 1996. *Matrix Computations*. Johns Hopkins University Press.

Graepel, T.; Herbrich, R.; Bollmann-Sdorra, P.; and Obermayer, K. 1999. Classification on pairwise proximity data. In *Advances in Neural Information Processing Systems (NIPS)*.

Guo, Y., and Schuurmans, D. 2009. A reformulation of support vector machines for general confidence functions. In *Proceedings of Asian Conference on Machine Learning*.

Hsu, C., and Lin, C. 2002. A comparison of methods for multi-class support vector machines. *IEEE transact. on Neural Networks* 13(2):415–425.

Lin, H., and Lin, C. 2003. A study on sigmoid kernel for SVM and the training of non-PSD kernels by SMO-type methods. Technical report.

Luss, R., and d'Aspremont, A. 2007. Support vector machine classification with indefinite kernels. In *Advances in Neural Information Processing Systems (NIPS)*.

Newman, D.; Hettich, S.; Blake, C.; and Merz, C. 1998. UCI repository of machine learning datasets.

Ong, C.; Mary, X.; Canu, S.; and Smola, A. 2004. Learning with non-positive kernels. In *Proceedings of International conference on Machine Learning (ICML)*.

Pekalska, E., and Haasdonk, B. 2008. Kernel discriminant analysis for positive definite and indefinite kernels. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 31(6):1017–1032.

Pekalska, E.; Paclik, P.; and Duin, R. 2001. A generalized kernel approach to dissimilarity-based classification. *Journal of Machine Learning Research* 2:175–211.

Roth, V.; Laub, J.; Kawanabe, M.; and Buhmann, J. 2003. Optimal cluster preserving embedding of nonmetric proximity data. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 25(12):1540–1551.

Saigo, H.; Vert, J.; Ueda, N.; and Akutsu, T. 2004. Protein homology detection using string alignment kernels. *Bioinformatics* 20, Issue 11:1682–1689.

Schölkopf, B.; Smola, A.; and Muller, K. 1999. Kernel principal component analysis. In *Advances in Kernel Methods-Support Vector Learning*, 327–352.

Smola, A.; Ovari, Z.; and Williamson, R. C. 2000. Regularization with dot-product kernels. In *Advances in Neural Information Processing Systems (NIPS)*.

Wu, G.; Chang, E.; and Zhang, Z. 2005. An analysis of transformation on non-positive semidefinite similarity matrix for kernel machines. In *Proceedings of International conference on Machine Learning (ICML)*.

Ying, Y.; Campbelly, C.; and Girolami, M. 2009. Analysis of SVM with indefinite kernels. In *Advances in Neural Information Processing Systems (NIPS)*.