Support Vector Machine Classification with Indefinite Kernels

Ronny Luss
ORFE, Princeton University
Princeton, NJ 08544
rluss@princeton.edu

Alexandre d'Aspremont
ORFE, Princeton University
Princeton, NJ 08544
aspremon@princeton.edu

Abstract

In this paper, we propose a method for support vector machine classification using indefinite kernels. Instead of directly minimizing or stabilizing a nonconvex loss function, our method simultaneously finds the support vectors and a proxy kernel matrix used in computing the loss. This can be interpreted as a robust classification problem where the indefinite kernel matrix is treated as a noisy observation of the true positive semidefinite kernel. Our formulation keeps the problem convex and relatively large problems can be solved efficiently using the analytic center cutting plane method. We compare the performance of our technique with other methods on several data sets.

1   Introduction

Here, we present an algorithm for support vector machine (SVM) classification using indefinite kernels. Our interest in indefinite kernels is motivated by several observations. First, certain similarity measures take advantage of application-specific structure in the data and often display excellent empirical classification performance. Unlike popular kernels used in support vector machine classification, these similarity matrices are often indefinite and so do not necessarily correspond to a reproducing kernel Hilbert space (see [1] for a discussion).

An application of classification with indefinite kernels to image classification using Earth Mover's Distance was discussed in [2]. Similarity measures for protein sequences such as the Smith-Waterman and BLAST scores are indefinite yet have provided hints for constructing useful positive semidefinite kernels such as those decribed in [3] or have been transformed into positive semidefinite kernels (see [4] for example). Here instead, our objective is to directly use these indefinite similarity measures for classification.

Our work also closely follows recent results on kernel learning (see [5] or [6]), where the kernel matrix is learned as a linear combination of given kernels, and the resulting kernel is explicitly constrained to be positive semidefinite (the authors of [7] have adapted the SMO algorithm to solve the case where the kernel is written as a positively weighted combination of other kernels). In our case however, we never *explicitly* optimize the kernel matrix because this part of the problem can be solved explicitly, which means that the complexity of our method is substantially lower than that of classical kernel learning methods and closer in spirit to the algorithm used in [8], who formulate the multiple kernel learning problem of [7] as a semi-infinite linear program and solve it with a column generation technique similar to the analytic center cutting plane method we use here.

Finally, it is sometimes impossible to prove that some kernels satisfy Mercer's condition or the numerical complexity of evaluating the exact positive semidefinite kernel is too high and a proxy (and not necessarily positive semidefinite) kernel has to be used instead (see [9] for example). In both cases, our method allows us to bypass these limitations.

1

1.1 Current results

Several methods have been proposed for dealing with indefinite kernels in SVMs. A first direction embeds data in a pseudo-Euclidean (pE) space: [10] for example, formulates the classification problem with an indefinite kernel as that of minimizing the distance between convex hulls formed from the two categories of data embedded in the pE space. The nonseparable case is handled in the same manner using reduced convex hulls (see [11] for a discussion of SVM geometric interpretations).

Another direction applies direct spectral transformations to indefinite kernels: flipping the negative eigenvalues or shifting the kernel's eigenvalues and reconstructing the kernel with the original eigenvectors in order to produce a positive semidefinite kernel (see [12] and [2]).

Yet another option is to reformulate either the maximum margin problem or its dual in order to use the indefinite kernel in a convex optimization problem (see [13]). An equivalent formulation of SVM with the same objective but where the kernel appears in the constraints can be modified to a convex problem by eliminating the kernel from the objective. Directly solving the nonconvex problem sometimes gives good results as well (see [14] and [10]).

1.2 Contribution

Here, instead of directly transforming the indefinite kernel, we simultaneously learn the support vector weights and a proxy positive semidefinite kernel matrix, while penalizing the distance between this proxy kernel and the original, indefinite one. Our main result is that the kernel learning part of that problem can be solved explicitly, meaning that the classification problem with indefinite kernels can simply be formulated as a perturbation of the positive semidefinite case.

Our formulation can also be interpreted as a worst-case robust classification problem with uncertainty on the kernel matrix. In that sense, indefinite similarity matrices are seen as noisy observations of an unknown positive semidefinite kernel. From a complexity standpoint, while the original SVM classification problem with indefinite kernel is nonconvex, the robustification we detail here is a convex problem, and hence can be solved efficiently with guaranteed complexity bounds.

The paper is organized as follows. In Section 2 we formulate our main classification problem and detail its interpretation as a robust SVM. In Section 3 we describe an algorithm for solving this problem. Finally, in Section 4, we test the numerical performance of these methods on various applications.

2 SVM with indefinite kernels

Here, we introduce our robustification of the SVM classification problem with indefinite kernels.

2.1 Robust classification

Let $K \in \mathbf{S}^n$ be a given kernel matrix and $y \in \mathbf{R}^n$ be the vector of labels, with $Y = \mathbf{diag}(y)$ the matrix with diagonal y, where $\mathbf{S}^n$ is the set of symmetric matrices of size n and $\mathbf{R}^n$ is the set of n-vectors of real numbers. We can write the dual of the SVM classification problem with hinge loss and quadratic penalty as:

$$\begin{array}{ll} \text{maximize} & \alpha^T e - \mathbf{Tr}(K(Y\alpha)(Y\alpha)^T)/2 \\ \text{subject to} & \alpha^T y = 0 \\ & 0 \le \alpha \le C \end{array} \qquad (1)$$

in the variable $\alpha \in \mathbf{R}^n$ and where e is an n-vector of ones. When K is positive semidefinite, this problem is a convex quadratic program. Suppose now that we are given an indefinite kernel matrix $K_0 \in \mathbf{S}^n$. We formulate a robust version of problem (1) by restricting K to be a positive semidefinite kernel matrix in some given neighborhood of the original (indefinite) kernel matrix K_0:

$$\max_{\{\alpha^T y=0,\ 0\le\alpha\le C\}} \quad \min_{\{K\succeq 0,\ \|K-K_0\|_F^2 \le \beta\}} \quad \alpha^T e - \frac{1}{2}\mathbf{Tr}(K(Y\alpha)(Y\alpha)^T) \qquad (2)$$

in the variables $K \in \mathbf{S}^n$ and $\alpha \in \mathbf{R}^n$, where the parameter $\beta > 0$ controls the distance between the original matrix K_0 and the proxy kernel K. This can be interpreted as a worst-case robust

2

classification problem with bounded uncertainty on the kernel matrix K. The above problem is infeasible for some values of β so we replace here the hard constraint on K by a penalty on the distance between the proxy positive semidefinite kernel and the given indefinite matrix. The problem we solve is now:

$$\max_{\{\alpha^T y=0, 0\leq\alpha\leq C\}} \min_{\{K\succeq 0\}} \alpha^T e - \frac{1}{2}\mathbf{Tr}(K(Y\alpha)(Y\alpha)^T) + \rho\|K - K_0\|_F^2 \qquad (3)$$

in the variables $K \in \mathbf{S}^n$ and $\alpha \in \mathbf{R}^n$, where the parameter $\rho > 0$ controls the magnitude of the penalty on the distance between K and K_0. The inner minimization problem is a convex conic program on K. Also, as the pointwise minimum of a family of concave quadratic functions of α, the solution to the inner problem is a concave function of α, and hence the outer optimization problem is also convex (see [15] for further details). Thus, (3) is a concave maximization problem subject to linear constraints and is therefore a convex problem in α.

Our key result here is that the inner kernel learning optimization problem can be solved in closed form. For a fixed α, the inner minimization problem is equivalent to the following problem:

$$\begin{aligned} \text{minimize} \quad & \|K - (K_0 + \frac{1}{4\rho}(Y\alpha)(Y\alpha)^T)\|_F^2 \\ \text{subject to} \quad & K \succeq 0 \end{aligned}$$

in the variable $K \in \mathbf{S}^n$. This is the projection of the $K_0 + (1/4\rho)(Y\alpha)(Y\alpha)^T$ on the cone of positive semidefinite matrices. The optimal solution to this problem is then given by:

$$K^* = (K_0 + (1/4\rho)(Y\alpha)(Y\alpha)^T)_+ \qquad (4)$$

where X_+ is the positive part of the matrix X, i.e. $X_+ = \sum_i \max(0, \lambda_i)x_i x_i^T$ where λ_i and x_i are the ith eigenvalue and eigenvector of the matrix X. Plugging this solution into (3), we get:

$$\max_{\{\alpha^T y=0, \ 0\leq\alpha\leq C\}} \alpha^T e - \frac{1}{2}\mathbf{Tr}(K^*(Y\alpha)(Y\alpha)^T) + \rho\|K^* - K_0\|_F^2$$

in the variable $\alpha \in \mathbf{R}^n$, where $(Y\alpha)(Y\alpha)^T$ is the rank one matrix with coefficients $y_i\alpha_i\alpha_j y_j$, $i, j = 1, \ldots, n$. We can rewrite this as an eigenvalue optimization problem by using the eigenvalue representation of X_+. Letting the eigenvalue decomposition of $K_0 + (1/4\rho)(Y\alpha)(Y\alpha)^T$ be VDV^T, we get $K^* = VD_+V^T$ and, with v_i the ith column of V, we can write:

$$\begin{aligned} \mathbf{Tr}(K^*(Y\alpha)(Y\alpha)^T) &= (Y\alpha)^T V D_+ V^T (Y\alpha) \\ &= \sum_{i=1}^n \max\left(0, \lambda_i\left(K_0 + \frac{1}{4\rho}(Y\alpha)(Y\alpha)^T\right)\right)(\alpha^T Y v_i)^2 \end{aligned}$$

where $\lambda_i(X)$ is the ith eigenvalue of the quantity X. Using the same technique, we can also rewrite the term $\|K^* - K_0\|_F^2$ using this eigenvalue decomposition. Our original optimization problem (3) finally becomes:

$$\begin{aligned} \text{maximize} \quad & \alpha^T e - \frac{1}{2}\sum_i \max(0, \lambda_i(K_0 + (Y\alpha)(Y\alpha)^T/4\rho))(\alpha^T Y v_i)^2 \\ & +\rho\sum_i \left(\max(0, \lambda_i(K_0 + (Y\alpha)(Y\alpha)^T/4\rho))\right)^2 \\ & -2\rho\sum_i \mathbf{Tr}((v_i v_i^T)K_0)\max(0, \lambda_i(K_0 + (Y\alpha)(Y\alpha)^T/4\rho)) + \rho\,\mathbf{Tr}(K_0 K_0) \\ \text{subject to} \quad & \alpha^T y = 0, 0 \leq \alpha \leq C \end{aligned} \qquad (5)$$

in the variable $\alpha \in \mathbf{R}^n$.

2.2 Dual problem

Because problem (3) is convex with at least one compact feasible set, we can formulate the dual problem to (5) by simply switching the max and the min. The inner maximization is a quadratic program in α, and hence has a quadratic program as its dual. We then get the dual by plugging this inner dual quadratic program into the outer minimization, to get the following problem:

$$\begin{aligned} \text{minimize} \quad & \mathbf{Tr}(K^{-1}(Y(e - \lambda + \mu + y\nu))(Y(e - \lambda + \mu + y\nu))^T)/2 + C\mu^T e + \rho\|K - K_0\|_F^2 \\ \text{subject to} \quad & K \succeq 0, \lambda, \mu \geq 0 \end{aligned}$$

$$(6)$$

3

in the variables $K \in \mathbf{S}^n$, $\lambda, \mu \in \mathbf{R}^n$ and $\nu \in \mathbf{R}$. This dual problem is a quadratic program in the variables λ and μ which correspond to the primal constraints $0 \le \alpha \le C$ and ν which is the dual variable for the constraint $\alpha^T y = 0$. As we have seen earlier, any feasible solution to the primal problem produces a corresponding kernel in (4), and plugging this kernel into the dual problem in (6) allows us to calculate a dual feasible point by solving a quadratic program which gives a dual objective value, i.e. an upper bound on the optimum of (5). This bound can then be used to compute a duality gap and track convergence.

2.3 Interpretation

We noted that our problem can be viewed as a worst-case robust classification problem with uncertainty on the kernel matrix. Our explicit solution of the optimal worst-case kernel given in (4) is the projection of a penalized rank-one update to the indefinite kernel on the cone of positive semidefinite matrices. As ρ tends to infinity, the rank-one update has less effect and in the limit, the optimal kernel is the kernel given by zeroing out the negative eigenvalues of the indefinite kernel. This means that if the indefinite kernel contains a very small amount of noise, the best positive semidefinite kernel to use with SVM in our framework is the positive part of the indefinite kernel.

This limit as ρ tends to infinity also motivates a heuristic for the transformation of the kernel on the testing set. Since the negative eigenvalues of the training kernel are thresholded to zero in the limit, the same transformation should occur for the test kernel. Hence, we update the entries of the full kernel corresponding to training instances by the rank-one update resulting from the optimal solution to (3) and threshold the negative eigenvalues of the full kernel matrix to zero. We then use the test kernel values from the resulting positive semidefinite matrix.

3 Algorithms

We now detail two algorithms that can be used to solve Problem (5). The optimization problem is the maximization of a nondifferentiable concave function subject to convex constraints. An optimal point always exists since the feasibility set is bounded and nonempty. For numerical stability, in both algorithms, we quadratically smooth our objective to calculate a gradient instead. We first describe a simple projected gradient method which has numerically cheap iterations but has no convergence bound. We then show how to apply the much more efficient analytic center cutting plane method whose iterations are slightly more complex but which converges linearly.

Smoothing   Our objective contains terms of the form $\max\{0, f(x)\}$ for some function $f(x)$, which are not differentiable (described in the section below). These functions are easily smoothed out by a regularization technique (see [16] for example). We replace them by a continuously differentiable $\frac{\epsilon}{2}$-approximation as follows:

$$\varphi_\epsilon(f(x)) = \max_{0 \le u \le 1} \left(u f(x) - \frac{\epsilon}{2} u^2 \right).$$

and the gradient is given by $\nabla \varphi_\epsilon(f(x)) = u^*(x) \nabla f(x)$ where $u^*(x) = \operatorname{argmax} \varphi_\epsilon(f(x))$.

Gradient   Calculating the gradient of our objective requires a full eigenvalue decomposition to compute the gradient of each eigenvalue. Given a matrix $X(\alpha)$, the derivative of the ith eigenvalue with respect to α is given by:

$$\frac{\partial \lambda_i(X(\alpha))}{\partial \alpha} = v_i^T \frac{\partial X(\alpha)}{\partial \alpha} v_i \tag{7}$$

where v_i is the ith eigenvector of $X(\alpha)$. We can then combine this expression with the smooth approximation above to get the gradient.

We note that eigenvalues of symmetric matrices are not differentiable when some of them have multiplicities greater than one (see [17] for a discussion). In practice however, most tested kernels were of full rank with distinct eigenvalues so we ignore this issue here. One may also consider projected subgradient methods, which are much slower, or use subgradients for analytic center cutting plane methods (which does not affect complexity).

4

3.1 Projected gradient method

The projected gradient method takes a steepest descent, then projects the new point back onto the feasible region (see [18] for example). In order to use these methods the objective function must be differentiable and the method is only efficient if the projection step is numerically cheap. We choose an initial point $\alpha_0 \in \mathbf{R}^n$ and the algorithm proceeds as follows:

Projected gradient method

1. Compute $\alpha_{i+1} = \alpha_i + t\nabla f(\alpha_i)$.
2. Set $\alpha_{i+1} = p_A(\alpha_{i+1})$.
3. If gap $\leq \epsilon$ stop, otherwise go back to step 1.

The complexity of each iteration breaks down as follows.

Step 1. This requires an eigenvalue decomposition and costs $O(n^3)$. We note that a line search would be costly because it would require multiple eigenvalue decompositions to recalculate the objective multiple times.

Step 2. This is a projection onto the region $A = \{\alpha^T y = 0,\ 0 \leq \alpha \leq C\}$ and can be solved explicitly by sorting the vector of entries, with cost $O(n \log n)$.

Stopping Criterion. We can compute a duality gap using the results of §2.2: let $K_i = (K_0 + (Y\alpha_i)(Y\alpha_i)^T/4\rho)_+$ (the kernel at iteration i), then solving problem (1) which is just an SVM with a convex kernel K_i produces an upper bound on (5), and hence a bound on the suboptimality of the current solution.

Complexity. The number of iterations required by this method to reach a target precision of ϵ is typically in $O(1/\epsilon^2)$.

3.2 Analytic center cutting plane method

The analytic center cutting plane method (ACCPM) reduces the feasible region on each iteration using a new *cut* of the feasible region computed by evaluating a subgradient of the objective function at the analytic center of the current set, until the volume of the reduced region converges to the target precision. This method does not require differentiability. We set $\mathcal{A}_0 = \{\alpha^T y = 0, 0 \leq \alpha \leq C\}$ which we can write as $\{A_0 \leq b_0\}$ to be our first localization set for the optimal solution. The method then works as follows (see [18] for a more complete reference on cutting plane methods):

Analytic center cutting plane method

1. Compute α_i as the analytic center of $\mathcal{A}_i$ by solving:

$$\alpha_{i+1} = \underset{y \in \mathbf{R}^n}{\operatorname{argmin}}\ -\sum_{i=1}^{m} \log(b_i - a_i^T y)$$

   where a_i^T represents the ith row of coefficients from the left-hand side of $\{A_0 \leq b_0\}$.
2. Compute $\nabla f(x)$ at the center α_{i+1} and update the (polyhedral) localization set:

$$\mathcal{A}_{i+1} = \mathcal{A}_i \cup \{\nabla f(\alpha_{i+1})(\alpha - \alpha_{i+1}) \geq 0\}$$

3. If gap $\leq \epsilon$ stop, otherwise go back to step 1.

The complexity of each iteration breaks down as follows.

Step 1. This step computes the analytic center of a polyhedron and can be solved in $O(n^3)$ operations using interior point methods for example.

5

Step 2. This simply updates the polyhedral description.

Stopping Criterion. An upper bound is computed by maximizing a first order Taylor approximation of $f(\alpha)$ at α_i over all points in an ellipsoid that covers $\mathcal{A}_i$, which can be done explicitly.

Complexity. ACCPM is provably convergent in $O(n \log(1/\epsilon)^2)$ iterations when using cut elimination which keeps the complexity of the localization set bounded. Other schemes are available with slightly different complexities: an $O(n^2/\epsilon^2)$ is achieved in [19] using (cheaper) approximate centers for example.

4 Experiments

In this section we compare the generalization performance of our technique to other methods of applying SVM classification given an indefinite similarity measure. We also test SVM classification performance on positive semidefinite kernels using the LIBSVM library. We finish with experiments showing convergence of our algorithms. Our algorithms were implemented in Matlab.

4.1 Generalization

We compare our method for SVM classification with indefinite kernels to several of the kernel pre-processing techniques discussed earlier. The first three techniques perform spectral transformations on the indefinite kernel. The first, denoted *denoise*, thresholds the negative eigenvalues to zero. The second transformation, called *flip*, takes the absolute value of all eigenvalues. The last transformation, *shift*, adds a constant to each eigenvalue making them all positive. See [12] for further details. We finally also compare with using SVM on the original indefinite kernel (SVM converges but the solution is only a stationary point and is not guaranteed to be optimal).

We experiment on data from the USPS handwritten digits database (described in [20]) using the indefinite Simpson score (SS) to compare two digits and on two data sets from the UCI repository (see [21]) using the indefinite Epanechnikov (EP) kernel. The data is randomly divided into training and testing data. We apply 5-fold cross validation and use an accuracy measure (described below) to determine the optimal parameters C and ρ. We then train a model with the full training set and optimal parameters and test on the independent test set.

Table 1: Statistics for various data sets.

Data Set	# Train	# Test	λ_{min}	λ_{max}
USPS-3-5-SS	767	773	-34.76	453.58
USPS-4-6-SS	829	857	-37.30	413.17
Diabetes-EP	614	154	-0.27	18.17
Liver-EP	276	69	-1.38e-15	3.74

Table 1 provides statistics including the minimum and maximum eigenvalues of the training kernels. The main observation is that the USPS data uses highly indefinite kernels while the UCI data use kernels that are nearly positive semidefinite. Table 2 displays performance by comparing accuracy and recall. Accuracy is defined as the percentage of total instances predicted correctly. Recall is the percentage of true positives that were correctly predicted positive.

Our method is referred to as Indefinite SVM. We see that our method performs favorably among the USPS data. Both measures of performance are quite high for all methods. Our method does not perform as well on the UCI data sets but is still favorable on one of the measures in each experiment. Notice though that recall is not good in the liver data set overall which could be the result of overfitting one of the classification categories. The liver data set uses a kernel that is almost positive semidefinite - this is an example where the input is almost a true kernel and Indefinite SVM finds one slightly better. We postulate that our method will perform better in situations where the similarity measure is highly indefinite as in the USPS dataset, while measures that are almost positive semidefinite maybe be seen as having a small amount of noise.

6

Table 2: Performance Measures for various data sets.

Data Set	Measure	Denoise	Flip	Shift	SVM	Indefinite SVM
USPS-3-5-SS	Accuracy	95.47	95.73	90.43	74.90	**96.25**
	Recall	94.50	95.45	92.11	72.73	**96.65**
USPS-4-6-SS	Accuracy	97.78	**97.90**	94.28	90.08	**97.90**
	Recall	98.42	98.65	93.68	88.49	**98.87**
Diabetes-EP	Accuracy	**75.32**	74.68	68.83	**75.32**	68.83
	Recall	90.00	90.00	92.00	90.00	**95.00**
Liver-EP	Accuracy	63.77	63.77	57.97	63.77	**65.22**
	Recall	22.58	22.58	**25.81**	22.58	22.58

4.2 Algorithm Convergence

We ran our two algorithms on data sets created by randomly perturbing the four USPS data sets used above. The average results with one standard deviation above and below the mean are displayed in Figure 1 with the duality gap in log scale (note that the codes were not stopped here and that the target gap improvement is usually much smaller than 10^{-8}). As expected, ACCPM converges much faster (in fact linearly) to a higher precision while each iteration requires solving a linear program of size n. The gradient projection method converges faster in the beginning but stalls at a higher precision, however each iteration only requires sorting the current point.
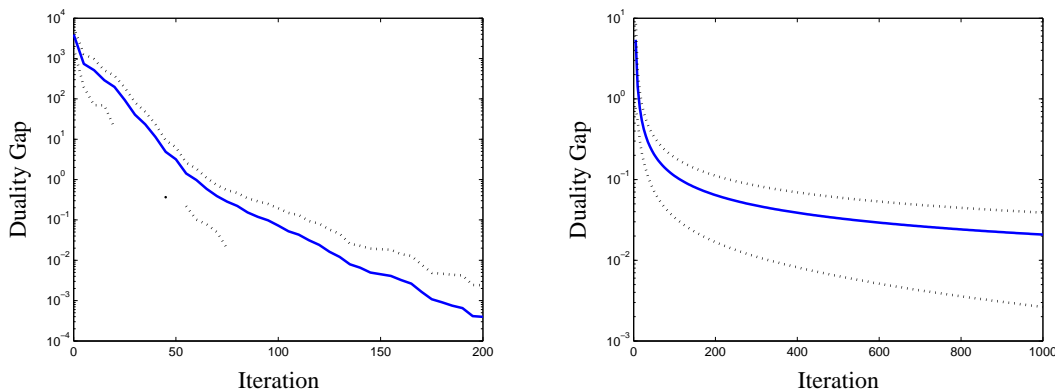
Figure 1: Convergence plots for ACCPM (left) & projected gradient method (right) on randomly perturbed USPS data sets (average gap versus iteration number, dashed lines at plus and minus one standard deviation).

5 Conclusion

We have proposed a technique for incorporating indefinite kernels into the SVM framework without any explicit transformations. We have shown that if we view the indefinite kernel as a noisy instance of a true kernel, we can learn an explicit solution for the optimal kernel with a tractable convex optimization problem. We give two convergent algorithms for solving this problem on relatively large data sets. Our initial experiments show that our method can at least fare comparably with other methods handling indefinite kernels in the SVM framework but provides a much clearer interpretation for these heuristics.

7

References

[1] C. S. Ong, X. Mary, S. Canu, and A. J. Smola. Learning with non-positive kernels. *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[2] A. Zamolotskikh and P. Cunningham. An assessment of alternative strategies for constructing emd-based kernel functions for use in an svm for image classification. *Technical Report UCD-CSI-2007-3*, 2004.

[3] H. Saigo, J. P. Vert, N. Ueda, and T. Akutsu. Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11):1682–1689, 2004.

[4] G. R. G. Lanckriet, N. Cristianini, M. I. Jordan, and W. S. Noble. Kernel-based integration of genomic data using semidefinite programming. 2003. citeseer.ist.psu.edu/648978.html.

[5] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[6] C. S. Ong, A. J. Smola, and R. C. Williamson. Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, 6:1043–1071, 2005.

[7] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[8] S. Sonnenberg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

[9] Marco Cuturi. Permanents, transport polytopes and positive definite kernels on histograms. *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, 2007.

[10] B. Haasdonk. Feature space interpretation of svms with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4), 2005.

[11] K. P. Bennet and E. J. Bredensteiner. Duality and geometry in svm classifiers. *Proceedings of the 17th International conference on Machine Learning*, pages 57–64, 2000.

[12] G. Wu, E. Y. Chang, and Z. Zhang. An analysis of transformation on non-positive semidefinite similarity matrix for kernel machines. *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

[13] H.-T. Lin and C.-J. Lin. A study on sigmoid kernel for svm and the training of non-psd kernels by smo-type methods. 2003.

[14] A. Woźnica, A. Kalousis, and M. Hilario. Distances and (indefinite) kernels for set of objects. *Proceedings of the 6th International Conference on Data Mining*, pages 1151–1156, 2006.

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

[16] C. Gigola and S. Gomez. A regularization method for solving the finite convex min-max problem. *SIAM Journal on Numerical Analysis*, 27(6):1621–1634, 1990.

[17] M. Overton. Large-scale optimization of eigenvalues. *SIAM Journal on Optimization*, 2(1):88–120, 1992.

[18] D. Bertsekas. *Nonlinear Programming, 2nd Edition*. Athena Scientific, 1999.

[19] J.-L. Goffin and J.-P. Vial. Convex nondifferentiable optimization: A survey focused on the analytic center cutting plane method. *Optimization Methods and Software*, 17(5):805–867, 2002.

[20] J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5), 1994.

[21] A. Asuncion and D.J. Newman. *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences, 2007. http://www.ics.uci.edu/~mlearn/MLRepository.html.

8