Multiple Kernel Learning, Conic Duality, and the SMO Algorithm

Francis R. Bach & Gert R. G. Lanckriet {FBACH,GERT}@CS.BERKELEY.EDU
Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720, USA

Michael I. Jordan JORDAN@CS.BERKELEY.EDU
Computer Science Division and Department of Statistics, University of California, Berkeley, CA 94720, USA

Abstract

While classical kernel-based classifiers are based on a single kernel, in practice it is often desirable to base classifiers on combinations of multiple kernels. Lanckriet et al. (2004) considered conic combinations of kernel matrices for the support vector machine (SVM), and showed that the optimization of the coefficients of such a combination reduces to a convex optimization problem known as a quadratically-constrained quadratic program (QCQP). Unfortunately, current convex optimization toolboxes can solve this problem only for a small number of kernels and a small number of data points; moreover, the sequential minimal optimization (SMO) techniques that are essential in large-scale implementations of the SVM cannot be applied because the cost function is non-differentiable. We propose a novel dual formulation of the QCQP as a second-order cone programming problem, and show how to exploit the technique of Moreau-Yosida regularization to yield a formulation to which SMO techniques can be applied. We present experimental results that show that our SMO-based algorithm is significantly more efficient than the general-purpose interior point methods available in current optimization toolboxes.

1. Introduction

One of the major reasons for the rise to prominence of the support vector machine (SVM) is its ability to cast nonlinear classification as a convex optimization problem, in particular a quadratic program (QP). Convexity implies that the solution is unique and brings a suite of standard numerical software to bear in finding the solution. Convexity alone, however, does not imply that the available algorithms scale well to problems of interest. Indeed, off-the-shelf algorithms do not suffice in large-scale applications of the SVM, and a second major reason for the rise to prominence of the SVM is the development of special-purpose algorithms for solving the QP (Platt, 1998; Joachims, 1998; Keerthi et al., 2001).

Recent developments in the literature on the SVM and other kernel methods have emphasized the need to consider multiple kernels, or parameterizations of kernels, and not a single fixed kernel. This provides needed flexibility and also reflects the fact that practical learning problems often involve multiple, heterogeneous data sources. While this so-called "multiple kernel learning" problem can in principle be solved via cross-validation, several recent papers have focused on more efficient methods for kernel learning (Chapelle et al., 2002; Grandvalet & Canu, 2003; Lanckriet et al., 2004; Ong et al., 2003). In this paper we focus on the framework proposed by Lanckriet et al. (2004), which involves joint optimization of the coefficients in a conic combination of kernel matrices and the coefficients of a discriminative classifier. In the SVM setting, this problem turns out to again be a convex optimization problem—a quadratically-constrained quadratic program (QCQP). This problem is more challenging than a QP, but it can also be solved in principle by general-purpose optimization toolboxes such as Mosek (Andersen & Andersen, 2000). Again, however, this existing algorithmic solution suffices only for small problems (small numbers of kernels and data points), and improved algorithmic solutions akin to sequential minimization optimization (SMO) are needed.

While the multiple kernel learning problem is convex, it is also non-smooth—it can be cast as the minimization of a non-differentiable function subject to linear

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004.

constraints (see Section 3.1). Unfortunately, as is well known in the non-smooth optimization literature, this means that simple local descent algorithms such as SMO may fail to converge or may converge to incorrect values (Bertsekas, 1995). Indeed, in preliminary attempts to solve the QCQP using SMO we ran into exactly these convergence problems.

One class of solutions to non-smooth optimization problems involves constructing a smooth approximate problem out of a non-smooth problem. In particular, *Moreau-Yosida (MY) regularization* is an effective general solution methodology that is based on inf-convolution (Lemarechal & Sagastizabal, 1997). It can be viewed in terms of the dual problem as simply adding a quadratic regularization term to the dual objective function. Unfortunately, in our setting, this creates a new difficulty—we lose the sparsity that makes the SVM amenable to SMO optimization. In particular, the QCQP formulation of Lanckriet et al. (2004) does not lead to an MY-regularized problem that can be solved efficiently by SMO techniques.

In this paper we show how these problems can be resolved by considering a novel dual formulation of the QCQP as a second-order cone programming (SOCP) problem. This new formulation is of interest on its own merit, because of various connections to existing algorithms. In particular, it is closely related to the classical maximum margin formulation of the SVM, differing only by the choice of the norm of the inverse margin. Moreover, the KKT conditions arising in the new formulation not only lead to support vectors as in the classical SVM, but also to a dual notion of "support kernels"—those kernels that are active in the conic combination. We thus refer to the new formulation as the *support kernel machine (SKM)*.

As we will show, the conic dual problem defining the SKM is exactly the multiple kernel learning problem of Lanckriet et al. (2004).[1] Moreover, given this new formulation, we can design a Moreau-Yosida regularization which preserves the sparse SVM structure, and therefore we can apply SMO techniques.

Making this circle of ideas precise requires a number of tools from convex analysis. In particular, Section 3 defines appropriate approximate optimality conditions for the SKM in terms of subdifferentials and approximate subdifferentials. These conditions are then used in Section 4 in the design of an MY regularization for the SKM and an SMO-based algorithm. We present the results of numerical experiments with the new method in Section 5.

[1] It is worth noting that this dual problem cannot be obtained directly as the Lagrangian dual of the QCQP problem—Lagrangian duals of QCQPs are semidefinite programming problems.

2. Learning the kernel matrix

In this section, we (1) begin with a brief review of the multiple kernel learning problem of Lanckriet et al. (2004), (2) introduce the support kernel machine (SKM), and (3) show that the dual of the SKM is equivalent to the multiple kernel learning primal.

2.1. Multiple kernel learning problem

In the multiple kernel learning problem, we assume that we are given n data points (x_i, y_i), where $x_i \in \mathcal{X}$ for some input space $\mathcal{X}$, and where $y_i \in \{-1, 1\}$. We also assume that we are given m matrices $K_j \in \mathbb{R}^{n\times n}$, which are assumed to be symmetric positive semidefinite (and might or might not be obtained from evaluating a kernel function on the data $\{x_i\}$). We consider the problem of learning the best linear combination $\sum_{j=1}^m \eta_j K_j$ of the kernels K_j with nonnegative coefficients $\eta_j \geqslant 0$ and with a trace constraint $\operatorname{tr}\sum_{j=1}^m \eta_j K_j = \sum_{j=1}^m \eta_j \operatorname{tr} K_j = c$, where $c > 0$ is fixed. Lanckriet et al. (2004) show that this setup yields the following optimization problem:

$$
\begin{array}{lll}
 & \min & \zeta - 2e^\top \alpha \\
(L) & \text{w.r.t.} & \zeta \in \mathbb{R}, \alpha \in \mathbb{R}^n \\
 & \text{s.t.} & 0 \leqslant \alpha \leqslant C,\ \alpha^\top y = 0 \\
 & & \alpha^\top D(y) K_j D(y) \alpha \leqslant \frac{\operatorname{tr} K_j}{c}\zeta, j \in \{1, \ldots, m\},
\end{array}
$$

where $D(y)$ is the diagonal matrix with diagonal y, $e \in \mathbb{R}^n$ the vector of all ones, and C a positive constant. The coefficients η_j are recovered as Lagrange multipliers for the constraints $\alpha^\top D(y) K_j D(y) \alpha \leqslant \frac{\operatorname{tr} K_j}{c}\zeta$.

2.2. Support kernel machine

We now introduce a novel classification algorithm that we refer to as the "support kernel machine" (SKM). It will be motivated as a block-based variant of the SVM and related margin-based classification algorithms. But our underlying motivation is the fact that the dual of the SKM is exactly the problem (L). We establish this equivalence in the following section.

2.2.1. Linear classification

In this section we let $\mathcal{X} = \mathbb{R}^k$. We also assume we are given a decomposition of $\mathbb{R}^k$ as a product of m blocks: $\mathbb{R}^k = \mathbb{R}^{k_1} \times \cdots \times \mathbb{R}^{k_m}$, so that each data point x_i can be decomposed into m block components, i.e. $x_i = (x_{1i}, \ldots, x_{mi})$, where each x_{ji} is in general a vector.

The goal is to find a linear classifier of the form

$y = \text{sign}(w^\top x + b)$ where w has the same block decomposition $w = (w_1, \ldots, w_m) \in \mathbb{R}^{k_1+\cdots+k_m}$. In the spirit of the soft margin SVM, we achieve this by minimizing a linear combination of the inverse of the margin and the training error. Various norms can be used to combine the two terms, and indeed many different algorithms have been explored for various combinations of ℓ_1-norms and ℓ_2-norms. In this paper, our goal is to encourage the sparsity of the vector w at the level of blocks; in particular, we want most of its (multivariate) components w_i to be zero. A natural way to achieve this is to penalize the ℓ_1-norm of w. Since w is defined by blocks, we minimize the square of a weighted block ℓ_1-norm, $(\sum_{j=1}^m d_j ||w_j||_2)^2$, where within every block, an ℓ_2-norm is used. Note that a standard ℓ_2-based SVM is obtained if we minimize the square of a block ℓ_2-norm, $\sum_{j=1}^m ||w_j||_2^2$, which corresponds to $||w||_2^2$, i.e., ignoring the block structure. On the other hand, if $m = k$ and $d_j = 1$, we minimize the square of the ℓ_1-norm of w, which is very similar to the LP-SVM proposed by Bradley and Mangasarian (1998). The primal problem for the SKM is thus:

$$
\begin{array}{rll}
 & \min & \frac{1}{2}(\sum_{j=1}^m d_j ||w_j||_2)^2 + C \sum_{i=1}^n \xi_i \\
(P) & \text{w.r.t.} & w \in \mathbb{R}^{k_1} \times \cdots \times \mathbb{R}^{k_m},\ \xi \in \mathbb{R}^n_+,\ b \in \mathbb{R} \\
 & \text{s.t.} & y_i(\sum_j w_j^\top x_{ji} + b) \geqslant 1 - \xi_i, \forall i \in \{1, \ldots, n\}.
\end{array}
$$

2.2.2. Conic duality and optimality conditions

For a given optimization problem there are many ways of deriving a dual problem. In our particular case, we treat problem (P) as a second-order cone program (SOCP) (Lobo et al., 1998), which yields the following dual (see Appendix A for the derivation):

$$
\begin{array}{rll}
 & \min & \frac{1}{2}\gamma^2 - \alpha^\top e \\
(D) & \text{w.r.t.} & \gamma \in \mathbb{R}, \alpha \in \mathbb{R}^n \\
 & \text{s.t.} & 0 \leqslant \alpha \leqslant C,\ \alpha^\top y = 0 \\
 & & ||\sum_i \alpha_i y_i x_{ji}||_2 \leqslant d_j \gamma,\ \forall j \in \{1, \ldots, m\}.
\end{array}
$$

In addition, the Karush-Kuhn-Tucker (KKT) optimality conditions give the following complementary slackness equations:

$$
\begin{array}{ll}
(a) & \alpha_i(y_i(\sum_j w_j^\top x_{ji} + b) - 1 + \xi_i) = 0, \quad \forall i \\
(b) & (C - \alpha_i)\xi_i = 0, \quad \forall i \\
(c) & \binom{w_j}{||w_j||_2}^\top \binom{-\sum_i \alpha_i y_i x_{ji}}{d_j \gamma} = 0, \quad \forall j \\
(d) & \gamma(\sum d_j t_j - \gamma) = 0.
\end{array}
$$

Equations (a) and (b) are the same as in the classical SVM, where they define the notion of a "support vector." That is, at the optimum, we can divide the data points into three disjoint sets: $I_0 = \{i, \alpha_i = 0\}$, $I_M = \{i, \alpha_i \in (0, C)\}$, and $I_C = \{i, \alpha_i = C\}$, such that points belonging to I_0 are correctly classified points not on the margin and such that $\xi_i = 0$; points in I_M are correctly classified points on the margin such that $\xi_i = 0$ and $y_i(\sum_j w_j^\top x_{ji} + b) = 1$, and points in I_C are points on the "wrong" side of the margin for which $\xi_i \geqslant 0$ (incorrectly classified if $\xi_i \geqslant 1$) and $y_i(\sum_j w_j^\top x_{ji} + b) = 1 - \xi_i$. The points whose indices i are in I_M or I_C are the support vectors.

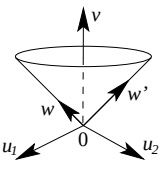

Figure 1. Orthogonality of elements of the second-order cone $\mathcal{K}_2 = \{w = (u, v), u \in \mathbb{R}^2, v \in \mathbb{R}, ||u||_2 \leqslant v\}$: two elements w, w' of $\mathcal{K}_2$ are orthogonal and nonzero if and only if they belong to the boundary and are anti-proportional.

While the KKT conditions (a) and (b) refer to the index i over data points, the KKT conditions (c) and (d) refer to the index j over components of the input vector. These conditions thus imply a form of sparsity not over data points but over "input dimensions." Indeed, two non-zero elements (u, v) and (u', v') of a second-order cone $\mathcal{K}_d = \{(u, v) \in \mathbb{R}^d \times \mathbb{R},\ ||u||_2 \leqslant v\}$ are orthogonal if and only if they both belong to the boundary, and they are "anti-proportional" (Lobo et al., 1998); that is, $\exists \eta > 0$ such that $||u||_2 = v,\ ||u'||_2 = v',\ (u, v) = \eta(-u', v')$ (see Figure 1).

Thus, if $\gamma > 0$, we have:

- if $||\sum_i \alpha_i y_i x_{ji}||_2 < d_j \gamma$, then $w_j = 0$,
- if $||\sum_i \alpha_i y_i x_{ji}||_2 = d_j \gamma$, then $\exists \eta_j > 0$, such that $w_j = \eta_j \sum_i \alpha_i y_i x_{ji}$, $||w_j||_2 = \eta_j d_j \gamma$.

Sparsity thus emerges from the optimization problem. Let $\mathcal{J}$ denote the set of active dimensions, i.e., $\mathcal{J}(\alpha, \gamma) = \{j : ||\sum_i \alpha_i y_i x_{ji}||_2 = d_j \gamma\}$. We can rewrite the optimality conditions as

$$\forall j,\ w_j = \eta_j \sum_i \alpha_i y_i x_{ji},\ \text{with } \eta_j = 0 \text{ if } j \notin \mathcal{J}.$$

Equation (d) implies that $\gamma = \sum_j d_j ||w_j||_2 = \sum_j d_j(\eta_j d_j \gamma)$, which in turn implies $\sum_{j \in \mathcal{J}} d_j^2 \eta_j = 1$.

2.2.3. Kernelization

We now remove the assumption that $\mathcal{X}$ is a Euclidean space, and consider embeddings of the data points x_i in a Euclidean space via a mapping $\phi : \mathcal{X} \to \mathbb{R}^f$. In correspondence with our block-based formulation of

the classification problem, we assume that $\phi(x)$ has m distinct block components $\phi(x) = (\phi_1(x), \ldots, \phi_m(x))$. Following the usual recipe for kernel methods, we assume that this embedding is performed implicitly, by specifying the inner product in $\mathbb{R}^f$ using a *kernel function*, which in this case is the sum of individual kernel functions on each block component:

$$\begin{aligned} k(x_i, x_j) &= \phi(x_i)^\top \phi(x_j) = \textstyle\sum_{s=1}^m \phi_s(x_i)^\top \phi_s(x_j) \\ &= \textstyle\sum_{s=1}^m k_s(x_i, x_j). \end{aligned}$$

We now "kernelize" the problem (P) using this kernel function. In particular, we consider the dual of (P) and substitute the kernel function for the inner products in (D):

$$\begin{aligned} & \min \ \tfrac{1}{2}\gamma^2 - e^\top \alpha \\ (D_K) \ & \text{w.r.t.} \ \ \gamma \in \mathbb{R}, \alpha \in \mathbb{R}^n \\ & \text{s.t.} \ \ 0 \leqslant \alpha \leqslant C, \ \alpha^\top y = 0 \\ & \qquad (\alpha^\top D(y) K_j D(y) \alpha)^{1/2} \leqslant \gamma d_j, \forall j, \end{aligned}$$

where K_j is the j-th Gram matrix of the points $\{x_i\}$ corresponding to the j-th kernel.

The sparsity that emerges via the KKT conditions (c) and (d) now refers to the kernels K_j, and we refer to the kernels with nonzero η_j as "support kernels." The resulting classifier has the same form as the SVM classifier, but is based on the kernel matrix combination $K = \sum_j \eta_j K_j$, which is a sparse combination of "support kernels."

2.3. Equivalence of the two formulations

By simply taking $d_j = \sqrt{\frac{\operatorname{tr} K_j}{c}}$, we see that problem (D_K) and (L) are indeed equivalent—thus the dual of the SKM is the multiple kernel learning primal. Care must be taken here though—the weights η_j are defined for (L) as Lagrange multipliers and for (D_K) through the anti-proportionality of orthogonal elements of a second-order cone, and a priori they might not coincide: although (D_K) and (L) are equivalent, their dual problems have different formulations. It is straightforward, however, to write the KKT optimality conditions for (α, η) for both problems and verify that they are indeed equivalent. One direct consequence is that for an optimal pair (α, η), α is an optimal solution of the SVM with kernel matrix $\sum_j \eta_j K_j$.

3. Optimality conditions

In this section, we formulate our problem (in either of its two equivalent forms) as the minimization of a non-differentiable convex function subject to linear constraints. Exact and approximate optimality conditions are then readily derived using subdifferentials. In later sections we will show how these conditions lead to an MY-regularized algorithmic formulation that will be amenable to SMO techniques.

3.1. Max-function formulation

A rearrangement of the problem (D_K) yields an equivalent formulation in which the quadratic constraints are moved into the objective function:

$$\begin{aligned} & \min \ \max_j \left\{ \tfrac{1}{2d_j^2} \alpha^\top D(y) K_j D(y) \alpha - \alpha^\top e \right\} \\ (S) \ & \text{w.r.t.} \ \ \alpha \in \mathbb{R}^n \\ & \text{s.t.} \ \ 0 \leqslant \alpha \leqslant C, \ \alpha^\top y = 0. \end{aligned}$$

We let $J_j(\alpha)$ denote $\frac{1}{2d_j^2} \alpha^\top D(y) K_j D(y) \alpha - \alpha^\top e$ and $J(\alpha) = \max_j J_j(\alpha)$. Problem (S) is the minimization of the non-differentiable convex function $J(\alpha)$ subject to linear constraints. Let $\mathcal{J}(\alpha)$ be the set of active kernels, i.e., the set of indices j such that $J_j(\alpha) = J(\alpha)$. We let $F_j(\alpha) \in \mathbb{R}^n$ denote the gradient of J_j, that is, $F_j = \frac{\partial J_j}{\partial \alpha} = \frac{1}{d_j^2} D(y) K_j D(y) \alpha - e$.

3.2. Optimality conditions and subdifferential

Given any function $J(\alpha)$, the *subdifferential* of J at α $\partial J(\alpha)$ is defined as (Bertsekas, 1995):

$$\partial J(\alpha) = \{ g \in \mathbb{R}^n, \ \forall \alpha', J(\alpha') \geqslant J(\alpha) + g^\top (\alpha' - \alpha) \}.$$

Elements of the subdifferential $\partial J(\alpha)$ are called *subgradients*. When J is convex and differentiable at α, then the subdifferential is a singleton and reduces to the gradient. The notion of subdifferential is especially useful for characterizing optimality conditions of non-smooth problems (Bertsekas, 1995).

The function $J(\alpha)$ defined in the earlier section is a pointwise maximum of convex differentiable functions, and using subgradient calculus we can easily see that the subdifferential $\partial J(\alpha)$ of J at α is equal to the convex hull of the gradients F_j of J_j for the active kernels. That is:

$$\partial J(\alpha) = \text{convex hull} \{ F_j(\alpha), j \in \mathcal{J}(\alpha) \}.$$

The Lagrangian for (S) is equal to $\mathcal{L}(\alpha) = J(\alpha) - \delta^\top \alpha + \xi^\top (\alpha - Ce) + b \alpha^\top y$, where $b \in \mathbb{R}$, $\xi, \delta \in \mathbb{R}^n_+$, and the global minimum of $\mathcal{L}(\alpha, \delta, \xi, b)$ with respect to α is characterized by the equation

$$0 \in \partial \mathcal{L}(\alpha) \Leftrightarrow \delta - \xi - by \in \partial J(\alpha).$$

The optimality conditions are thus the following: $\alpha, (b, \delta, \xi)$ is a pair of optimal primal/dual variables

if and only if:

$$(OPT_0)\qquad \begin{array}{l} \delta - \xi - by \in \partial J(\alpha) \\ \forall i,\ \delta_i \alpha_i = 0,\ \xi_i(C-\alpha_i) = 0 \\ \alpha^\top y = 0,\ 0 \leqslant \alpha \leqslant C. \end{array}$$

As before, we define $I_M(\alpha) = \{i,\ 0 < \alpha_i < C\}$, $I_0(\alpha) = \{i,\ \alpha_i = 0\}$, $I_C(\alpha) = \{i,\ \alpha_i = C\}$. We also define, following (Keerthi et al., 2001), $I_{0+} = I_0 \cap \{i, y_i = 1\}$ and $I_{0-} = I_0 \cap \{i, y_i = -1\}$, $I_{C+} = I_C \cap \{i, y_i = 1\}$, $I_{C-} = I_C \cap \{i, y_i = -1\}$. We can then rewrite the optimality conditions as

$$(OPT_1)\qquad \begin{array}{l} \nu - be = D(y) \sum_{j \in \mathcal{J}(\alpha)} d_j^2 \eta_j F_j(\alpha) \\ \eta \geqslant 0, \sum_j d_j^2 \eta_j = 1 \\ \forall i \in I_M \cup I_{0+} \cup I_{C-},\ \nu_i \geqslant 0 \\ \forall i \in I_M \cup I_{0+} \cup I_{C-},\ \nu_i \leqslant 0. \end{array}$$

3.3. Approximate optimality conditions

Exact optimality conditions such as (OPT_0) or (OPT_1) are generally not suitable for numerical optimization. In non-smooth optimization theory, one instead formulates optimality criteria in terms of the ε-subdifferential, which is defined as

$$\partial_\varepsilon J(\alpha) = \{g \in \mathbb{R}^n, \forall \alpha', J(\alpha') \geqslant J(\alpha) - \varepsilon + g^\top(\alpha' - \alpha)\}.$$

When $J(\alpha) = \max_j J_j(\alpha)$, then the ε-subdifferential contains (potentially strictly) the convex hull of the gradients $F_j(\alpha)$, for all ε-active functions, i.e., for all j such that $\max_i J_i(\alpha) - \varepsilon \leqslant J_j(\alpha)$. We let $\mathcal{J}_\varepsilon(\alpha)$ denote the set of all such kernels. So, we have $\mathcal{C}_\varepsilon(\alpha) = \text{convex hull}\{F_j(\alpha), j \in \mathcal{J}_\varepsilon(\alpha)\} \subseteq \partial_\varepsilon J(\alpha)$.

Our stopping criterion, referred to as $(\varepsilon_1, \varepsilon_2)$-optimality, requires that the ε_1-subdifferential is within ε_2 of zero, and that the usual KKT conditions are met. That is, we stop whenever there exist ν, b, g such that

$$(OPT_2)\qquad \begin{array}{l} g \in \partial_{\varepsilon_1} J(\alpha) \\ \forall i \in I_M \cup I_{0+} \cup I_{C-},\ \nu_i \geqslant 0 \\ \forall i \in I_M \cup I_{0+} \cup I_{C-},\ \nu_i \leqslant 0 \\ ||\nu - be - D(y)g||_\infty \leqslant \varepsilon_2. \end{array}$$

Note that for one kernel, i.e., when the SKM reduces to the SVM, this corresponds to the approximate KKT conditions usually employed for the standard SVM (Platt, 1998; Keerthi et al., 2001; Joachims, 1998). For a given α, checking optimality is hard, since even computing $\partial_{\varepsilon_1} J(\alpha)$ is hard in closed form. However, a sufficient condition for optimality can be obtained by using the inner approximation $\mathcal{C}_{\varepsilon_1}(\alpha)$ of this ε_1-subdifferential, i.e., the convex hull of gradients of ε_1-active kernels. Checking this sufficient condition is a linear programming (LP) existence problem, i.e., find η such that:

$$(OPT_3)\qquad \begin{array}{l} \eta \geqslant 0, \eta_j = 0 \text{ if } j \notin \mathcal{J}_{\varepsilon_1}(\alpha),\ \sum_j d_j^2 \eta_j = 1 \\ \max_{i \in I_M \cup I_{0-} \cup I_{C+}} \{(K(\eta)D(y)\alpha)_i - y_i\} \\ \qquad \leqslant \min_{i \in I_M \cup I_{0+} \cup I_{C-}} \{(K(\eta)D(y)\alpha)_i - y_i\} + 2\varepsilon_2, \end{array}$$

where $K(\eta) = \sum_{j \in \mathcal{J}_{\varepsilon_1}(\alpha)} \eta_j K_j$. Given α, we can determine whether it is $(\varepsilon_1, \varepsilon_2)$-optimal by solving the potentially large LP (OPT_3). If in addition to having α, we know a potential candidate for η, then a *sufficient* condition for optimality is that this η verifies (OPT_3), which doesn't require solving the LP. Indeed, the iterative algorithm that we present in Section 4 outputs a pair (α, η) and only these sufficient optimality conditions need to be checked.

3.4. Improving sparsity

Once we have an approximate solution, i.e., values α and η that satisfy (OPT_3), we can ask whether η can be made sparser. Indeed, if some of the kernels are close to identical, then some of the η's can potentially be removed—for a general SVM, the optimal α is not unique if data points coincide, and for a general SKM, the optimal α and η are not unique if data points or kernels coincide. When searching for the minimum ℓ_0-norm η which satisfies the constraints (OPT_3), we can thus consider a simple heuristic approach where we loop through all the nonzero η_j and check whether each such component can be removed. That is, for all $j \in \mathcal{J}_{\varepsilon_1}(\alpha)$, we force η_j to zero and solve the LP. If it is feasible, then the j-th kernel can be removed.

4. Regularized support kernel machine

The function $J(\alpha)$ is convex but not differentiable. It is well known that in this situation, steepest descent and coordinate descent methods do not necessarily converge to the global optimum (Bertsekas, 1995). SMO unfortunately falls into this class of methods. Therefore, in order to develop an SMO-like algorithm for the SKM, we make use of Moreau-Yosida regularization. In our specific case, this simply involves adding a second regularization term to the objective function of the SKM, as follows:

$$(R)\qquad \begin{array}{ll} \min & \frac{1}{2}(\sum_j d_j ||w_j||_2)^2 + \frac{1}{2}\sum_j a_j^2 ||w_j||_2^2 + C\sum_i \xi_i \\ \text{w.r.t.} & w \in \mathbb{R}^{k_1} \times \cdots \times \mathbb{R}^{k_m}, \xi \in \mathbb{R}_+^n, b \in \mathbb{R} \\ \text{s.t.} & y_i(\sum_j w_j^\top x_{ji} + b) \geqslant 1 - \xi_i,\ \forall i \in \{1, \ldots, n\}, \end{array}$$

where (a_j) are the MY-regularization parameters.

4.1. Dual problem

The conic dual is readily computed as:

$$\begin{aligned} \min \ & \frac{1}{2}\gamma^2 + \frac{1}{2}\sum_j \frac{(\mu_j - \gamma d_j)^2}{a_j^2} - \sum_i \alpha_i \\ \text{w.r.t.} \ & \gamma \in \mathbb{R}_+, \mu \in \mathbb{R}^m, \alpha \in \mathbb{R}^n \\ \text{s.t.} \ & 0 \leqslant \alpha_i \leqslant C, \ \alpha^\top y = 0 \\ & ||\sum_i \alpha_i y_i x_{ji}||_2 \leqslant \mu_j, \ \forall j. \end{aligned}$$

If we define the function $G(\alpha)$ as

$$G(\alpha) = \min_{\gamma\in\mathbb{R}_+,\mu\in\mathbb{R}^m}\{\tfrac{1}{2}\gamma^2 + \tfrac{1}{2}\sum_j \tfrac{(\mu_j-\gamma d_j)^2}{a_j^2} - \sum_i \alpha_i, \\ ||\sum_i \alpha_i y_i x_{ji}||_2 \leqslant \mu_j, \forall j\},$$

then the dual problem is equivalent to minimizing $G(\alpha)$ subject to $0 \leqslant \alpha \leqslant C$ and $\alpha^\top y = 0$. We prove in Appendix B that $G(\alpha)$ is differentiable and we show how to compute $G(\alpha)$ and its derivative in time $O(m \log m)$.

4.2. Solving the MY-regularized SKM using SMO

Since the objective function $G(\alpha)$ is differentiable, we can now safely envisage an SMO-like approach, which consists in a sequence of local optimizations over only two components of α. Since the ε-optimality conditions for the MY-regularized SKM are exactly the same as for the SVM, but with a different objective function (Platt, 1998; Keerthi et al., 2001):

$$\begin{aligned} (OPT_4) \quad & \max_{i\in I_M\cup I_{0-}\cup I_{C+}} \{y_i \nabla G(\alpha)_i\} \\ & \leqslant \min_{i\in I_M\cup I_{0+}\cup I_{C-}} \{y_i \nabla G(\alpha)_i\} + 2\varepsilon, \end{aligned}$$

choosing the pair of indices can be done in a manner similar to that proposed for the SVM, by using the fast heuristics of Platt (1998) and Keerthi et al. (2001). In addition, caching and shrinking techniques (Joachims, 1998) that prevent redundant computations of kernel matrix values can also be employed.

A difference between our setting and the SVM setting is the line search, which cannot be performed in closed form for the MY-regularized SKM. However, since each line search is the minimization of a convex function, we can use efficient one-dimensional root finding, such as Brent's method (Brent, 1973).

4.3. Theoretical bounds

In order to be able to check efficiently the approximate optimality condition (OPT_3) of Section 3.3, we need estimates for α and η from the solution of the MY-regularized SKM obtained by SMO. It turns out that the KKT conditions for the MY-regularized SKM also lead to support kernels, i.e., there is a sparse non-negative weight vector η such that α is a solution of the SVM with the kernel $K = \sum_j \eta_j K_j$. However, in the regularized case, those weights η can be obtained directly from α as a byproduct of the computation of $G(\alpha)$ and its derivative. Those weights $\eta(\alpha)$ do not satisfy $\sum_j d_j^2 \eta_j = 1$, but can be used to define weights $\tilde{\eta}(\alpha)$ that do (we give expressions for $\eta(\alpha)$ and $\tilde{\eta}(\alpha)$ in Appendix B).

Let ε_1, ε_2 be the two tolerances for the approximate optimality conditions for the SKM. In this section, we show that if (a_j) are small enough, then an $\varepsilon_2/2$-optimal solution of the MY-regularized SKM α, together with $\tilde{\eta}(\alpha)$, is an $(\varepsilon_1, \varepsilon_2)$-optimal solution of the SKM, and an a priori bound on (a_j) is obtained that does not depend on the solution α.

Theorem 1 *Let $0 < \varepsilon < 1$. Let $y \in \{-1, 1\}^n$ and K_j, $j = 1, \ldots, m$ be m positive semidefinite kernel matrices. Let d_j and a_j, $j = 1, \ldots, m$, be $2m$ strictly positive numbers. If α is an ε-optimal solution of the MY-regularized SKM, then $(\alpha, \tilde{\eta}(\alpha))$ is an $(\varepsilon_1, \varepsilon_2)$-optimal solution of the SKM, with*

$$\varepsilon_1 = nC \max_j \frac{a_j^2}{d_j^2}(2 + \max_j \frac{a_j^2}{d_j^2}) \text{ and } \varepsilon_2 = \varepsilon + C \max_j \frac{a_j^2 M_j}{d_j^4},$$

where $M_j = \max_u \sum_v |(K_j)_{uv}|$.

Corollary 1 *With the same assumptions and*

$$||a||_\infty^2 \leqslant \min\Big\{ \min_j d_j^2 \frac{\frac{\varepsilon_1}{nC}}{1 + (1 + \frac{\varepsilon_1}{nC})^{1/2}}, \frac{\varepsilon_2/2}{\max_j \frac{M_j C}{d_j^4}} \Big\},$$

if α is an $\varepsilon_2/2$-optimal solution of the MY-regularized SKM, then $(\alpha, \tilde{\eta}(\alpha))$ is an $(\varepsilon_1, \varepsilon_2)$-optimal solution of the SKM.

4.4. A minimization algorithm

We solve the SKM by solving the MY-regularized SKM with decreasing values of the regularization parameters (a_j). In our simulations, the kernel matrices are all normalized, i.e., have unit diagonal, so we can choose all d_j equal. We use $a_j(\kappa) = \kappa$ and $d_j(\kappa) = (1 - \kappa)$, where κ is a constant in $[0, 1]$. When $\kappa = 1$, the MY-regularized SKM is exactly the SVM based on the sum of the kernels, while when $\kappa = 0$, it is the non-MY-regularized SKM.

The algorithm is as follows: given the data and precision parameters $\varepsilon_1, \varepsilon_2$, we start with $\kappa = 1$, which

solves the SVM up to precision $\varepsilon_2/2$ using standard SMO, and then update κ to $\mu\kappa$ (where $\mu < 1$) and solve the MY-regularized SKM with constant κ using the adjusted SMO up to precision $\varepsilon_2/2$, and so on. At the end of every SMO optimization, we can extract weights $\tilde{\eta}_j(\alpha)$ from the α solution, as shown in Appendix B, and check the $(\varepsilon_1, \varepsilon_2)$-optimality conditions (OPT_3) of the original problem (without solving the LP). Once they are satisfied, the algorithm stops.

Since each SMO optimization is performed on a differentiable function with Lipschitz gradient and SMO is equivalent to steepest descent for the ℓ_1-norm (Joachims, 1998), classical optimization results show that each of those SMO optimizations is finitely convergent (Bertsekas, 1995). Corollary 1 directly implies there are only a finite number of such optimizations; thus, the overall algorithm is finitely convergent.

Additional speed-ups can be easily achieved here. For example, if for successive values of κ, some kernels have a zero weight, we might as well remove them from the algorithm and check after convergence if they can be safely kept out. In simulations, we use the following values for the free parameters: $\mu = 0.5$, $\varepsilon_1/n = 0.0005$, $\varepsilon_2 = 0.0001$, where the value for ε_1/n corresponds to the average value this quantity attains when solving the QCQP (L) directly using Mosek.

5. Simulations

We compare the algorithm presented in Section 4.4 with solving the QCQP (L) using Mosek for two datasets, ionosphere and breast cancer, from the UCI repository, and nested subsets of the adult dataset from Platt (1998). The basis kernels are Gaussian kernels on random subsets of features, with varying widths. We vary the number of kernels m for fixed number of data points n, and vice versa. We report running time results (Athlon MP 2000+ processor, 2.5G RAM) in Figure 2. Empirically, we obtain an average scaling of $m^{1.1}$ and $n^{1.4}$ for the SMO-based approach and $m^{1.6}$ and $n^{4.1}$ for Mosek. Thus the algorithm presented in this paper appears to provide a significant improvement over Mosek in computational complexity, both in terms of the number of kernels and the number of data points.

6. Conclusion

We have presented an algorithm for efficient learning of kernels for the support vector machine. Our algorithm is based on applying sequential minimization techniques to a smoothed version of a convex non-smooth optimization problem. The good scaling with respect to the number of data points makes it possible to learn kernels for large scale problems, while the good scaling with respect to the number of basis kernels opens up the possibility of application to large-scale feature selection, in which the algorithm selects kernels that define non-linear mappings on subsets of input features.

Ionosphere, $n = 351$		
m	SMO	Mosek
6	2	4
12	3	8
24	54	20
48	56	51
96	88	162
192	166	548

Breast cancer, $n = 683$		
m	SMO	Mosek
3	11	11
6	20	17
12	54	45
24	141	120
48	149	492
96	267	764

Adult, $n = 1605$		
m	SMO	Mosek
3	20	92
6	23	205
12	36	1313
24	119	*
48	618	*
96	957	*

Adult, $m = 4$		
n	SMO	Mosek
450	17	4
750	29	17
1100	44	52
1605	72	114
2265	121	5455
3185	202	8625
4781	410	*
6212	670	*

Figure 2. Running times in seconds for Mosek and SMO. (Top) Ionosphere and breast cancer data, with fixed number of data points n and varying number of kernels m. (Bottom) Adult dataset: (left) with fixed n and varying m, (right) with fixed m and varying n ($*$ means Mosek ran out of memory).

Appendix A. Dual of the SKM

The primal problem (P) can be put in the following equivalent form, where $\mathcal{K}_k = \{(u, v) \in \mathbb{R}^{k+1}, ||u||_2 \leqslant v\}$ is the second-order cone of order k (we now omit the summation intervals, with the convention that index i goes from 1 to n and index j goes from 1 to m):

$$\begin{aligned} \min \;& \tfrac{1}{2}u^2 + C\textstyle\sum_i \xi_i \\ \text{w.r.t.} \;& u \in \mathbb{R}, t \in \mathbb{R}^m, b \in \mathbb{R}, \xi \in \mathbb{R}^n_+, (w_j, t_j) \in \mathcal{K}_{k_j}, \; \forall j \\ \text{s.t.} \;& y_i(\textstyle\sum_j w_j^\top x_{ji} + b) \geqslant 1 - \xi_i, \; \forall i \\ & \textstyle\sum_j d_j t_j \leqslant u. \end{aligned}$$

The cone $\mathcal{K}_k$ is self-dual, so the conic Lagrangian corresponding to the problem is

$$\begin{aligned} \mathcal{L} \;=\; & \tfrac{1}{2}u^2 + C\textstyle\sum_i \xi_i - \sum_i \alpha_i (y_i(\sum_j w_j^\top x_{ji} + b) - 1 + \xi_i) \\ & - \textstyle\sum_i \beta_i \xi_i + \gamma(\sum_j d_j t_j - u) - \sum_j (\lambda_j^\top w_j + \mu_j t_j), \end{aligned}$$

with $\alpha_i \in \mathbb{R}_+$, $\beta_i \in \mathbb{R}_+$, $\gamma \in \mathbb{R}_+$, $(\lambda_j, \mu_j) \in \mathcal{K}_{k_j}$.

After computing derivatives with respect to the primal variables and setting them to zero, we readily get the dual function: $g(\alpha, \beta, \gamma, \lambda, \mu) = -\frac{1}{2}\gamma^2 + \sum_i \alpha_i$ defined on the domain defined by $\alpha_i \geqslant 0, \beta_i \geqslant 0, \gamma \geqslant 0, ||\lambda_j||_2 \leqslant \mu_j, d_j\gamma - \mu_j = 0, \sum_i \alpha_i y_i x_{ji} + \lambda_j = 0, \sum_i \alpha_i y_i = 0, C - \alpha_i - \beta_i = 0$. After elimination of dummy variables, we obtain problem (D).

Appendix B. Computation of $G(\alpha)$

Let $\gamma_j(\alpha) = \frac{1}{d_j}||\sum_i \alpha_i y_i x_{ji}||_2$. We can first maximize over each μ_i; a short calculation reveals:

$$\min_{\mu_j \geqslant ||\sum_i \alpha_i y_i x_{ji}||_2} (\mu_j - \gamma\delta_j)^2 = d_j^2 \max(0, \gamma_j - \gamma)^2,$$

which implies that

$$G(\alpha) = \min_\gamma \{\tfrac{1}{2}\gamma^2 + \tfrac{1}{2}\sum_j \tfrac{d_j^2}{a_j^2}\psi(\gamma_j^2, \gamma) - \sum_i \alpha_i\},$$

where $\psi(x, y) = \max(0, \sqrt{x} - y)^2$. The function ψ is continuously differentiable, with partial derivatives equal to $\left(\frac{\partial\psi}{\partial x}, \frac{\partial\psi}{\partial y}\right) = (1 - y/\sqrt{x}, 2y - 2\sqrt{x})$ if $y \leqslant \sqrt{x}$, and zero otherwise. Also, for given x, it is a piecewise quadratic function of y. We thus need to minimize a piecewise quadratic differentiable strictly convex function of γ, which can be done easily by inspecting all points of non-differentiability, which requires sorting the sequence (γ_j). The complexity of such an algorithm is $O(m \log m)$.

Because of strict convexity the minimum with respect to γ is unique and denoted $\gamma(\alpha)$. In addition, this uniqueness implies that $G(\alpha)$ is differentiable and that its derivative is equal to:

$$\begin{aligned}\nabla G(\alpha) &= \tfrac{1}{2}\sum_j \tfrac{d_j^2}{a_j^2}\tfrac{\partial\psi}{\partial x}(\gamma_j^2(\alpha), \gamma(\alpha))\nabla\gamma_j^2(\alpha) - e \\ &= \sum_{j \in \mathcal{J}(\alpha)} \tfrac{1}{a_j^2}\left(1 - \tfrac{\gamma(\alpha)}{\gamma_j(\alpha)}\right) D(y)K_j D(y)\alpha - e.\end{aligned}$$

We define $\eta_j(\alpha) = \frac{1}{a_j^2}\left(1 - \frac{\gamma(\alpha)}{\gamma_j(\alpha)}\right)$ if $\gamma_j(\alpha) \geqslant \gamma(\alpha)$, and zero otherwise. We also define $\tilde{\eta}_j(\alpha) = \eta_j(\alpha)/(1 - a_j^2\eta_j(\alpha))$. Using the optimality conditions for $\gamma(\alpha)$, it is easy to prove that $\sum_j d_j^2\tilde{\eta}_j(\alpha) = 1$. The weights $\tilde{\eta}_j(\alpha)$ provide an estimate of the weights for the SKM, and can be used to check optimality. Corollary 1 shows that if (a_j) is small enough, then if α is approximately optimal for the MY-regularized SKM, the pair $(\alpha, \tilde{\eta}(\alpha))$ is approximately optimal for the SKM.

Acknowledgements

We wish to acknowledge support from a grant from Intel Corporation, and a graduate fellowship to Francis Bach from Microsoft Research.

References

Andersen, E. D., & Andersen, K. D. (2000). The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. *High Perf. Optimization* (pp. 197–232).

Bertsekas, D. (1995). *Nonlinear programming*. Nashua, NH: Athena Scientific.

Bradley, P. S., & Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. *International Conference on Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Brent, R. P. (1973). *Algorithms for minimization without derivatives*. Englewood Cliffs, NJ: Prentice-Hall.

Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*, *46*, 131–159.

Grandvalet, Y., & Canu, S. (2003). Adaptive scaling for feature selection in SVMs. *Neural Information Processing Systems*. Cambridge, MA: MIT Press.

Joachims, T. (1998). Making large-scale support vector machine learning practical. *Advances in Kernel Methods: Support Vector Machines*. Cambridge, MA: MIT Press.

Keerthi, S. S., Shevade, S. K., Bhattacharyya, C., & Murthy, K. R. K. (2001). Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, *13*, 637–649.

Lanckriet, G. R. G., Cristianini, N., Ghaoui, L. E., Bartlett, P., & Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming. *J. Machine Learning Research*, *5*, 27–72.

Lemarechal, C., & Sagastizabal, C. (1997). Practical aspects of the Moreau-Yosida regularization: Theoretical preliminaries. *SIAM J. Optim.*, *7*, 867–895.

Lobo, M. S., Vandenberghe, L., Boyd, S., & Lébret, H. (1998). Applications of second-order cone programming. *Lin. Alg. and its Applications*, *284*, 193–228.

Ong, S., Smola, A. J., & Williamson, R. C. (2003). Hyperkernels. *Neural Information Processing Systems*. Cambridge, MA: MIT Press.

Platt, J. (1998). Fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods: Support Vector Learning*. Cambridge, MA: MIT Press.