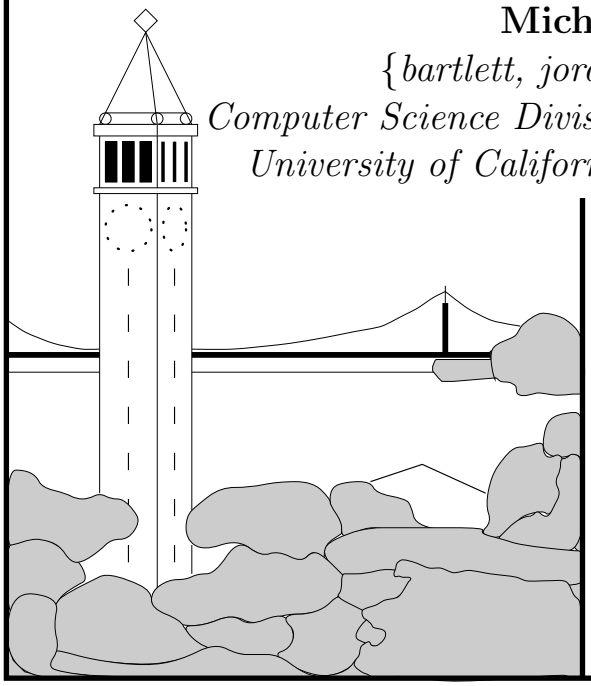
Learning the Kernel Matrix with Semi-Definite Programming

Gert R.G. Lanckriet
gert@cs.berkeley.edu
Department of Electrical Engineering and Computer Science
University of California, Berkeley, CA 94720, USA

Nello Cristianini
nello@wald.ucdavis.edu
Department of Statistics
University of California, Davis, CA 95616, USA

Laurent El Ghaoui
elghaoui@eecs.berkeley.edu
Department of Electrical Engineering and Computer Science
University of California, Berkeley, CA 94720, USA

Peter Bartlett
Michael I. Jordan
{bartlett, jordan}@stat.berkeley.edu
Computer Science Division and Department of Statistics
University of California, Berkeley, CA 94720, USA



Report No. UCB/CSD-02-1206
October, 2002

Computer Science Division (EECS)
University of California
Berkeley, California 94720

Learning the Kernel Matrix with Semi-Definite Programming

Gert R.G. Lanckriet
gert@cs.berkeley.edu
Department of Electrical Engineering and Computer Science
University of California, Berkeley, CA 94720, USA

Nello Cristianini
nello@wald.ucdavis.edu
Department of Statistics
University of California, Davis, CA 95616, USA

Laurent El Ghaoui
elghaoui@eecs.berkeley.edu
Department of Electrical Engineering and Computer Science
University of California, Berkeley, CA 94720, USA

Peter Bartlett
Michael I. Jordan
{bartlett, jordan}@stat.berkeley.edu
Computer Science Division and Department of Statistics
University of California, Berkeley, CA 94720, USA

October, 2002

Abstract

Kernel-based learning algorithms work by embedding the data into a Euclidean space, and then searching for linear relations among the embedded data points. The embedding is performed implicitly, by specifying the inner products between each pair of points in the embedding space. This information is contained in the so-called kernel matrix, a symmetric and positive definite matrix that encodes the relative positions of all points. Specifying this matrix amounts to specifying the geometry of the embedding space and inducing a notion of similarity in the input space—classical model selection problems in machine learning. In this paper we show how the kernel matrix can be learned from data via semi-definite programming (SDP) techniques. When applied to a kernel matrix associated with both training and test data this gives a powerful transductive algorithm—using the labelled part of the data one can learn an embedding also for the unlabelled part. The similarity between test points is inferred from training points and their labels. Importantly, these learning problems are convex, so we obtain a method for learning both the model class and the function without local minima. Furthermore, this approach leads directly to a convex method to learn the 2-norm soft margin parameter in support vector machines, solving another important open problem. Finally, the novel approach presented in the paper is supported by positive empirical results.

Keywords: kernel methods, learning kernels, transduction, model selection, support vector machines, convex optimization, semi-definite programming

1 Introduction

Recent advances in kernel-based learning algorithms have brought the field of machine learning closer to the desirable goal of autonomy—the goal of providing learning systems that require as little intervention as possible on the part of a human user. In particular, kernel-based algorithms are generally formulated in terms of convex optimization problems, which have a single global optimum and thus do not require heuristic choices of learning rates, starting configurations or other free parameters. There are, of course, statistical model selection problems to be faced within the kernel approach; in particular, the choice of the kernel and the corresponding feature space are central choices that must generally be made by a human user. While this provides opportunities for prior knowledge to be brought to bear, it can also be difficult in practice to find prior justification for the use of one kernel instead of another. It would be desirable to explore model selection methods that allow kernels to be chosen in a more automatic way based on data.

It is important to observe that we do not necessarily need to choose a kernel *function*—the representation of a finite training data set via a kernel function is entirely specified by a finite-dimensional *kernel matrix* (also known as a *Gram matrix*) that contains as its entries the inner products (in an appropriate feature space) between pairs of data points. Note also that it is possible to show that any symmetric positive definite matrix is a valid Gram matrix, in the sense that it specifies the values of some inner product. This suggests viewing the model selection problem in terms of Gram matrices rather than kernel functions.

In this paper we focus on *transduction*—the problem of completing the labelling of a partially labelled dataset. In other words, we are required to make predictions only at a finite set of points, which are specified a priori. Thus, instead of learning a function, we need only learn a set of labels. There are many practical problems in which this formulation is natural—an example is the prediction of gene function, where the genes of interest are specified a priori, but the function of many of these genes is unknown.

We will address this problem by learning a kernel matrix corresponding to the entire dataset, a matrix that optimizes a certain cost function that depends on the available labels. In other words, we use the available labels to learn a good embedding, and we apply it to both the labelled and the unlabelled data. The resulting kernel matrix can then be used in combination with any of a number of existing learning algorithms that use kernels. One example that we discuss in detail is the support vector machine (SVM), where our methods yield a new transduction method for SVMs that scales polynomially with the number of test points. Furthermore, this approach will offer us a method to optimize the 2-norm soft margin parameter for these SVM learning algorithms, solving another important open problem.

All this can be done in full generality by using techniques from semi-definite programming (SDP), a branch of convex optimization that deals with the optimization of convex functions over the convex cone of positive semi-definite matrices, or convex subsets thereof. Any convex set of kernel matrices is a set of this kind. Furthermore, it turns out that many natural cost functions, motivated by error bounds, are convex in the kernel matrix.

In Section 2, we recall the main ideas from kernel-based learning algorithms, and introduce a variety of criteria that we use to assess the suitability of a kernel matrix: the hard margin,

the 1-norm and 2-norm soft margin, and the alignment. Section 3 reviews the basic definitions and results of semi-definite programming. Section 4 considers the optimization of the various criteria over a set of kernel matrices. For a set of linear combinations of fixed kernel matrices, these optimization problems reduce to SDP. If the linear coefficients are constrained to be positive, they can be simplified even further. If the linear combination contains the unity matrix, this can be proven to provide us with a convex method to optimize the 2-norm soft margin parameter in support vector machines. Section 5 presents error bounds that motivate one of our cost functions. Empirical results are reported in Section 6.

Notation

For a square, symmetric matrix X, $X \succeq 0$ means that X is positive semi-definite, while $X \succ 0$ means that X is positive definite. For a vector v, the notations $v \geq 0$, $v > 0$ are understood componentwise.

2 Kernel Methods

Kernel-based learning algorithms (see, for example, Cristianini and Shawe-Taylor, 2000, Schölkopf and Smola, 2002) work by embedding the data into a Hilbert space, and searching for linear relations in such a space. The embedding is performed implicitly, by specifying the inner product between each pair of points rather than by giving their coordinates explicitly. This approach has several advantages, the most important deriving from the fact that the inner product in the embedding space can often be computed much more easily than the coordinates of the points themselves.

Given an input set $\mathcal{X}$, and an embedding space $\mathcal{F}$, we consider a map $\Phi : \mathcal{X} \to \mathcal{F}$. Given two points $x_i \in \mathcal{X}$ and $x_j \in \mathcal{X}$, the function that returns the inner product between their images in the space $\mathcal{F}$ is known as the *kernel function*.

Definition 1 *A* kernel *is a function k, such that $k(x, z) = \langle \Phi(x), \Phi(z) \rangle$ for all $x, z \in \mathcal{X}$, where Φ is a mapping from $\mathcal{X}$ to an (inner product) feature space $\mathcal{F}$. A* kernel matrix *is a square matrix $K \in \Re^{n \times n}$ such that $K_{ij} = k(x_i, x_j)$ for some $x_1, \ldots, x_n \in \mathcal{X}$ and some kernel function k.*

The kernel matrix is also known as the Gram matrix. It is a symmetric, positive semi-definite matrix, and since it specifies the inner products between all pairs of points $\{x_i\}_{i=1}^n$, it completely determines the relative positions of those points in the embedding space.

Since in this paper we will consider a *finite* input set $\mathcal{X}$, we can characterize kernel functions and matrices in the following simple way.

Proposition 2 *Every positive semi-definite and symmetric matrix is a kernel matrix. Conversely, every kernel matrix is symmetric and positive semi-definite.*

Notice that, if we have a kernel matrix, we do not need to know the kernel function, nor the implicitly defined map Φ, nor the coordinates of the points $\Phi(x_i)$. We do not even need $\mathcal{X}$ to be a vector space; in fact in this paper it will be a generic finite set. We are guaranteed that the data are implicitly mapped to some Hilbert space by simply checking that the kernel matrix is symmetric and positive semi-definite.

The solutions sought by kernel-based algorithms such as the support vector machine (SVM) are linear functions in the feature space:

$$f(\mathbf{x}) = \mathbf{w}^T \Phi(\mathbf{x}),$$

for some weight vector $\mathbf{w} \in \mathcal{F}$. The kernel can be exploited whenever the weight vector can be expressed as a linear combination of the training points, $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \Phi(\mathbf{x}_i)$, implying that we can express f as

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}).$$

An important issue in applications is that of choosing a kernel k for a given learning task; intuitively, we wish to choose a kernel that induces the "right" metric in the space.

2.1 Criteria used in kernel methods

Kernel methods choose a function that is linear in the feature space by optimizing some criterion over the sample. This section describes several such criteria. (See, for example, Cristianini and Shawe-Taylor, 2000, Schölkopf and Smola, 2002). All of these criteria can be considered as measures of separation of the labelled data. We first consider the *hard margin* optimization problem.

Definition 3 Hard Margin *Given a labelled sample* $S_l = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$*, the hyperplane* $(\mathbf{w}, b)$ *that solves the optimization problem*

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \langle \mathbf{w}, \mathbf{w} \rangle \\ \textit{subject to} \quad & y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b) \geq 1, \quad i = 1, \ldots, n, \end{aligned} \tag{1}$$

if it exists, realizes the maximal margin classifier with geometric margin $\gamma = 1/\|\mathbf{w}_*\|_2$*, where* $\mathbf{w}_*$ *is that* $\mathbf{w}$ *that optimizes (1).*

Geometrically, γ corresponds to the distance between the convex hulls (the smallest convex sets that contain the data in each class) of the two classes (Bennett and Bredensteiner, 2000).

By transforming (1) into its corresponding dual problem, the solution is given by

$$\begin{aligned} w(K) &= 1/\gamma^2 \\ &= \langle \mathbf{w}_*, \mathbf{w}_* \rangle \\ &= \max_{\alpha} \; 2\alpha^T e - \alpha^T G(K) \alpha \;:\; \alpha \geq 0, \;\; \alpha^T y = 0, \end{aligned} \tag{2}$$

where e is the n-vector of ones, $\alpha \in \mathbb{R}^n$, $G(K)$ is defined by $G_{ij}(K) = [K]_{ij} y_i y_j = k(\mathbf{x}_i, \mathbf{x}_j) y_i y_j$, and $\alpha \geq 0$ means $\alpha_i \geq 0, \; i = 1, \ldots, n$.

The hard margin solution exists only when the labelled sample is linearly separable in feature space. For a non-linearly-separable labelled sample S_l, we can define the *soft margin.* We consider the 1-norm and 2-norm soft margins.

Definition 4 1-Norm Soft Margin *Given a labelled sample* $S_l = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$*, the hyperplane* $(\mathbf{w}, b)$ *that solves the optimization problem*

$$\begin{array}{ll} \min\limits_{\mathbf{w},b,\xi} & \langle \mathbf{w}, \mathbf{w} \rangle + C \sum\limits_{i=1}^{n} \xi_i \\ \textit{subject to} & y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n \\ & \xi_i \geq 0, \quad i = 1, \ldots, n \end{array} \tag{3}$$

realizes the 1-norm soft margin classifier with geometric margin $\gamma = 1/\|\mathbf{w}_*\|_2$ *, where* $\mathbf{w}_*$ *is that* $\mathbf{w}$ *that optimizes (3). This margin is also called the 1-norm soft margin.*

As for the hard margin, by considering the corresponding dual problem, we can express the solution of (3) as follows:

$$\begin{aligned} w_{S1}(K) &= \langle \mathbf{w}_*, \mathbf{w}_* \rangle + C \sum_{i=1}^{n} \xi_{i,*} \\ &= \max_{\alpha} \; 2\alpha^T e - \alpha^T G(K) \alpha \;:\; C \geq \alpha \geq 0, \;\; \alpha^T y = 0. \end{aligned} \tag{4}$$

Definition 5 2-Norm Soft Margin *Given a labelled sample* $S_l = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$*, the hyperplane* $(\mathbf{w}, b)$ *that solves the optimization problem*

$$\begin{array}{ll} \min\limits_{\mathbf{w},b,\xi} & \langle \mathbf{w}, \mathbf{w} \rangle + C \sum\limits_{i=1}^{n} \xi_i^2 \\ \textit{subject to} & y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n \\ & \xi_i \geq 0, \quad i = 1, \ldots, n \end{array} \tag{5}$$

realizes the 2-norm soft margin classifier with geometric margin $\gamma = 1/\|\mathbf{w}_*\|_2$ *, where* $\mathbf{w}_*$ *is that* $\mathbf{w}$ *that optimizes (5). This margin is also called the 2-norm soft margin.*

Again, by considering the corresponding dual problem, the solution of (5) can be expressed as

$$\begin{aligned} w_{S2}(K) &= \langle \mathbf{w}_*, \mathbf{w}_* \rangle + C \sum_{i=1}^{n} \xi_{i,*}^2 \\ &= \max_{\alpha} \; 2\alpha^T e - \alpha^T \left(G(K) + \frac{1}{C} I_n \right) \alpha \;:\; \alpha \geq 0, \;\; \alpha^T y = 0. \end{aligned} \tag{6}$$

With a fixed kernel, all of these criteria give upper bounds on misclassification probability (see, for example, Chapter 4 of Cristianini and Shawe-Taylor, 2000). Solving these optimization problems for a single kernel matrix is therefore a way of optimizing an upper bound on error probability.

In this paper, we allow the kernel matrix to be chosen from a class of kernel matrices. Previous error bounds are not applicable in this case. However, as we will see in Section 5, the margin γ can be used to bound the performance of support vector machines for transduction, with a linearly parameterized class of kernels.

We do not discuss further the merit of these different cost functions, deferring to the current literature on classification, where these cost functions are widely used with fixed kernels. Our goal

is to show these cost functions can be optimized—with respect to the kernel matrix—in an SDP setting.

Finally, we define the *alignment* of two kernel matrices (Cristianini et al., 2002). Given an (unlabelled) sample $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, we use the following (Frobenius) inner product between Gram matrices, $\langle K_1, K_2 \rangle_F = \text{trace}(K_1^T K_2) = \sum_{i,j=1}^n k_1(\mathbf{x}_i, \mathbf{x}_j) k_2(\mathbf{x}_i, \mathbf{x}_j)$.

Definition 6 Alignment *The* (empirical) alignment *of a kernel k_1 with a kernel k_2 with respect to the sample S is the quantity*

$$\hat{A}(S, k_1, k_2) = \frac{\langle K_1, K_2 \rangle_F}{\sqrt{\langle K_1, K_1 \rangle_F \langle K_2, K_2 \rangle_F}},$$

where K_i is the kernel matrix for the sample S using kernel k_i.

This can also be viewed as the cosine of the angle between two bi-dimensional vectors K_1 and K_2, representing the Gram matrices. If we consider $K_2 = yy^T$, where y is the vector of $\{\pm 1\}$ labels for the sample, then

$$\hat{A}(S, K, yy^T) = \frac{\langle K, yy^T \rangle_F}{\sqrt{\langle K, K \rangle_F \langle yy^T, yy^T \rangle_F}} = \frac{\langle K, yy^T \rangle_F}{n\sqrt{\langle K, K \rangle_F}}, \tag{7}$$

since $\langle yy^T, yy^T \rangle_F = n^2$.

3 Semi-Definite Programming (SDP)

In this section we review the basic definition of semi-definite programming as well as some important concepts and key results. Details and proofs can be found in Boyd and Vandenberghe (2001).

Semi-definite programming (Nesterov and Nemirovsky, 1994, Vandenberghe and Boyd, 1996, Boyd and Vandenberghe, 2001) deals with the optimization of convex functions over the convex cone[1] of symmetric, positive semi-definite matrices

$$\mathcal{P} = \left\{ X \in \mathbb{R}^{p \times p} \mid X = X^T, X \succeq 0 \right\},$$

or affine subsets of this cone. Given Proposition 2, $\mathcal{P}$ can be viewed as a search space for possible kernel matrices. This consideration leads to the key problem addressed in this paper—we wish to specify a convex cost function that will enable us to learn the optimal kernel matrix within $\mathcal{P}$ using semi-definite programming.

3.1 Definition of Semi-Definite Programming

A *linear matrix inequality,* abbreviated LMI, is a constraint of the form

$$F(x) := F_0 + x_1 F_1 + \ldots + x_q F_q \preceq 0.$$

Here, x is the vector of decision variables, and $F_0, \ldots, F_q$ are given symmetric $p \times p$ matrices. The notation $F(x) \preceq 0$ means that the symmetric matrix F is negative semi-definite. Note that

[1] $S \subseteq \mathbb{R}^d$ is a convex cone if $x, y \in S, \lambda, \mu \geq 0 \Rightarrow \lambda x + \mu y \in S$.

such a constraint is in general a *nonlinear* constraint; the term "linear" in the name LMI merely emphasizes that F is affine in x. Perhaps the most important feature of an LMI constraint is its convexity: the set of x that satisfy the LMI is a convex set.

An LMI constraint can be seen as an *infinite* set of scalar, affine constraints. Indeed, for a given x, $F(x) \preceq 0$ if and only if $z^T F(x) z \leq 0$ for every z; every constraint indexed by z is an affine inequality, in the ordinary sense. Alternatively, using a standard result from linear algebra, we may state the constraint as

$$\forall Z \in \mathcal{P} \ : \ \mathrm{trace}(F(x)Z) \leq 0. \tag{8}$$

A semi-definite program (SDP) is an optimization problem with a linear objective, and linear matrix inequality and affine equality constraints.

Definition 7 *A semi-definite program is a problem of the form*

$$\begin{array}{rl} \min\limits_{x} & c^T x \\ \textit{subject to} & F^j(x) = F_0^j + x_1 F_1^j + \ldots + x_q F_q^j \preceq 0, \quad j = 1, \ldots, L \\ & Ax = b, \end{array} \tag{9}$$

where $x \in \mathbb{R}^q$ is the vector of decision variables, $c \in \mathbb{R}^q$ is the objective vector,and matrices $F_i^j = (F_i^j)^T \in \mathbb{R}^{p \times p}$ are given.

By convexity of its LMI constraints, SDPs are convex optimization problems. The usefulness of the SDP formalism stems from two important facts. First, despite the seemingly very specialized form of SDPs, they arise in a host of applications; second, there exist "interior-point" algorithms to globally solve SDPs that have extremely good theoretical and practical computational efficiency (Vandenberghe and Boyd, 1996).

One very useful tool to reduce a problem to an SDP is the so-called Schur Complement Lemma, which will be invoked later in this paper.

Lemma 8 (Schur Complement Lemma) *Consider the partitioned symmetric matrix*

$$X = X^T = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix},$$

where A, C are square and symmetric. When $\det(A) \neq 0$, we define the Schur complement of A in X by the matrix $S = C - B^T A^{-1} B$. The Schur Complement Lemma states that if $A \succ 0$, then $X \succeq 0$ if and only if $S \succeq 0$.

To illustrate how this Lemma can be used to cast a nonlinear convex optimization problem as an SDP, consider the following result:

Lemma 9 *The quadratically constrained quadratic program (QCQP)*

$$\begin{array}{rl} \min\limits_{\mathbf{x}} & f_0(\mathbf{x}) \\ \textit{subject to} & f_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, M, \end{array} \tag{10}$$

with $f_i(\mathbf{x}) \triangleq (A_i\mathbf{x}+\mathbf{b}_i)^T(A_i\mathbf{x}+\mathbf{b}_i) - \mathbf{c}_i^T\mathbf{x} - d_i$, *is equivalent to the semi-definite programming problem:*

$$\begin{aligned} \min_{\mathbf{x},t} \quad & t \\ \textit{subject to} \quad & \begin{pmatrix} I & A_0\mathbf{x}+\mathbf{b_0} \\ (A_0\mathbf{x}+\mathbf{b_0})^T & \mathbf{c_0}^T\mathbf{x}+d_0+t \end{pmatrix} \succeq 0, \\ & \begin{pmatrix} I & A_i\mathbf{x}+\mathbf{b}_i \\ (A_i\mathbf{x}+\mathbf{b}_i)^T & \mathbf{c}_i^T\mathbf{x}+d_i \end{pmatrix} \succeq 0, \quad i=1,\ldots,M. \end{aligned} \tag{11}$$

This can be seen by rewriting the QCQP (10) as:

$$\begin{aligned} \min_{\mathbf{x},t} \quad & t \\ \text{subject to} \quad & t - f_0(\mathbf{x}) \geq 0, \\ & -f_i(\mathbf{x}) \geq 0, \quad i=1,\ldots,M. \end{aligned}$$

The convex quadratic inequality $t - f_0(\mathbf{x}) = (t + \mathbf{c_0}^T\mathbf{x} + d_0) - (A_0\mathbf{x}+\mathbf{b_0})^T I^{-1}(A_0\mathbf{x}+\mathbf{b_0}) \geq 0$ is now equivalent to the following LMI, using the Schur Complement Lemma 8:

$$\begin{pmatrix} I & A_0\mathbf{x}+\mathbf{b_0} \\ (A_0\mathbf{x}+\mathbf{b_0})^T & \mathbf{c_0}^T\mathbf{x}+d_0+t \end{pmatrix} \succeq 0.$$

Similar steps for the other quadratic inequality constraints finally yields (11), an SDP in standard form (9), equivalent to (10). This shows that a QCQP can be cast as an SDP. Of course, in practice a QCQP should not be solved using general-purpose SDP solvers, since the particular structure of the problem at hand can be efficiently exploited. The above does show that QCQPs, and in particular, linear programming problems, belong to the SDP family.

3.2 Duality

An important principle in optimization—perhaps even the most important principle—is that of *duality*. To illustrate duality in the case of an SDP, we will first review basic concepts in duality theory and then show how they can be extended to semi-definite programming. In particular, duality will give insights into optimality conditions for the semi-definite program.

Consider an optimization problem with n variables and m scalar constraints

$$\begin{aligned} \min_{x} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0, \quad i=1,\ldots,m, \end{aligned} \tag{12}$$

where $x \in \mathbb{R}^n$. In the context of duality, problem (12) is called the *primal problem*; we denote its optimal value p^*. For now, we do not assume convexity.

Definition 10 Lagrangian *The* Lagrangian $\mathcal{L} : \mathbb{R}^{n+m} \to \mathbb{R}$ *corresponding to the minimization problem (12) is defined as*

$$\mathcal{L}(x,\lambda) = f_0(x) + \lambda_1 f_1(x) + \ldots + \lambda_m f_m(x).$$

The $\lambda_i \in \mathbb{R},\ i=1,\ldots,m$ *are called* Lagrange multipliers *or* dual variables.

One can now notice that

$$h(x) = \max_{\lambda \geq 0} \mathcal{L}(x, \lambda) = \begin{cases} f_0(x) & \text{if } f_i(x) \leq 0,\ i = 1, \ldots, m \\ +\infty & \text{otherwise} \end{cases} \tag{13}$$

So, the function $h(x)$ coincides with the objective $f_0(x)$ in regions where the constraints $f_i(x) \leq 0,\ i = 1, \ldots, m$ are satisfied and $h(x) = +\infty$ in infeasible regions. In other words, h acts as a "barrier" of the feasible set of the primal problem. Thus we can as well use $h(x)$ as objective function and rewrite the original primal problem (12) as an *unconstrained* optimization problem:

$$p^* = \min_x \max_{\lambda \geq 0} \mathcal{L}(x, \lambda). \tag{14}$$

The notion of weak duality amounts to exchanging the "min" and "max" operators in the above formulation, resulting in a lower bound on the optimal value of the primal problem. Strong duality refers to the case when this exchange can be done without altering the value of the result: the lower bound is actually equal to the optimal value p^*. While weak duality always hold, even if the primal problem is not convex, strong duality may not hold. However, for a large class of generic convex problems, strong duality holds.

Lemma 11 Weak duality *Even if the original problem (14) is not convex, we can exchange the* **max** *and the* **min** *and get a lower bound on p^*:*

$$d^* = \max_{\lambda \geq 0} \min_x \mathcal{L}(x, \lambda) \leq \min_x \max_{\lambda \geq 0} \mathcal{L}(x, \lambda) = p^*.$$

The objective function of the maximization problem is now called the (Lagrange) dual function.

Definition 12 (Lagrange) dual function *The* (Lagrange) dual function $g : \mathbb{R}^m \to \mathbb{R}$ *is defined as*

$$\begin{aligned} g(\lambda) &= \min_x \mathcal{L}(x, \lambda) \\ &= \min_x\ f_0(x) + \lambda_1 f_1(x) + \ldots + \lambda_m f_m(x). \end{aligned} \tag{15}$$

Furthermore $g(\lambda)$ is concave, even if the $f_i(x)$ are not convex.

The concavity can easily be seen by considering first that for a given x, $\mathcal{L}(x, \lambda)$ is an affine function of λ and hence is a concave function. Since $g(\lambda)$ is the pointwise minimum of such concave functions, it is concave.

Definition 13 Lagrange dual problem *The* Lagrange dual problem *is defined as*

$$d^* = \max_{\lambda \geq 0} g(\lambda).$$

Since $g(\lambda)$ is concave, this will always be a convex optimization problem, even if the primal is not. By *weak duality*, we always have $d^* \leq p^*$, even for non-convex problems. The value $p^* - d^*$ is called the duality gap. For **convex** problems, we usually (although not always) have *strong duality* at the optimum, i.e.,

$$d^* = p^*$$

which is also referred to as a *zero duality gap*. For convex problems, a sufficient condition for zero duality gap is provided by *Slater's condition*:

Lemma 14 Slater's condition *If the primal problem (12) is convex and is* **strictly feasible**, *i.e.,* $\exists\ x_0\ :\ f_i(x_0) < 0,\ i = 1, \ldots, m,$ *then*

$$p^* = d^*.$$

3.3 SDP Duality and Optimality Conditions

Consider for simplicity the case of an SDP with a single LMI constraint, and no affine equalities:

$$p^* = \min_x\ c^T x \text{ subject to } F(x) = F_0 + x_1 F_1 + \ldots x_q F_q \preceq 0. \tag{16}$$

The general case of multiple LMI constraints and affine equalities can be handled by elimination of the latter and using block-diagonal matrices to represent the former as a single LMI.

The classical Lagrange duality theory outlined in the previous section does not directly apply here, since we are not dealing with finitely many constraints in scalar form; as noted earlier, the LMI constraint involves an infinite number of such constraints, of the form (8). One way to handle such constraints is to introduce a Lagrangian of the form

$$\mathcal{L}(x, Z) = c^T x + \text{trace}(ZF(x)),$$

where the dual variable Z is now a symmetric matrix, of same size as $F(x)$. We can check that such a Lagrange function fulfills the same role assigned to the function defined in Definition 10 for the case with scalar constraints. Indeed, if we define $h(x) = \max_{Z \succeq 0} \mathcal{L}(x, Z)$ then

$$h(x) = \max_{Z \succeq 0} \mathcal{L}(x, Z) = \begin{cases} c^T x & \text{if } F(x) \preceq 0, \\ +\infty & \text{otherwise.} \end{cases} \tag{17}$$

Thus, $h(x)$ is a barrier for the primal SDP (16), that is, it coincides with the objective of (16) on its feasible set, and is infinite otherwise. Notice that to the LMI constraint we now associate a multiplier *matrix*, which will be constrained to the positive semi-definite cone.

In the above, we made use of the fact that, for a given symmetric matrix F,

$$\phi(F) := \sup_{Z \succeq 0} \text{trace}(ZF)$$

is $+\infty$ if F has a positive eigenvalue, and zero if F is negative semi-definite. This property is obvious for diagonal matrices, since in that case the variable Z can be constrained to be diagonal without loss of generality. The general case follows from the fact that if F has the eigenvalue decomposition $F = U \Lambda U^T$, where Λ is a diagonal matrix containing the eigenvalues of F, and U is orthogonal, then $\text{trace}(ZF) = \text{trace}(Z'\Lambda)$, where $Z' = U^T Z U$ spans the positive semi-definite cone whenever Z does.

Using the above Lagrangian, one can cast the original problem (16) as an unconstrained optimization problem:

$$p^* = \min_x \max_{Z \succeq 0} \mathcal{L}(x, Z).$$

By weak duality, we obtain a lower bound on p^* by exchanging the min and max:

$$d^* = \max_{Z \succeq 0} \min_x \mathcal{L}(x, Z) \leq \min_x \max_{Z \succeq 0} \mathcal{L}(x, Z) = p^*.$$

The inner minimization problem is easily solved analytically, due to the special structure of the SDP. We obtain a closed form for the (Lagrange) dual function:

$$\begin{aligned} g(Z) = \min_x \mathcal{L}(x, Z) &= \min_x c^T x + \text{trace}(ZF_0) + \sum_{i=1}^{q} x_i \text{ trace}(ZF_i) \\ &= \begin{cases} \text{trace}(ZF_0) & \text{if } c_i = -\text{trace}(ZF_i),\ i = 1, \ldots, q \\ -\infty & \text{otherwise.} \end{cases} \end{aligned}$$

The dual problem can be explicitly stated as follows:

$$d^* = \max_{Z \succeq 0} \min_x \mathcal{L}(x, Z) = \max_Z \text{trace}(ZF_0) \ \text{ subject to } Z \succeq 0,\ c_i = -\text{trace}(ZF_i),\ i = 1, \ldots, q. \quad (18)$$

We observe that the above problem is an SDP, with a single LMI constraint and q affine equalities in the matrix dual variable Z.

While weak duality always holds, strong duality may not, even for SDPs. Not surprisingly, a Slater-type condition ensures strong duality. Precisely, if the primal SDP (16) is strictly feasible, that is, there exist a x_0 such that $F(x_0) \prec 0$, then $p^* = d^*$. If, in addition, the dual problem is also strictly feasible, meaning that there exist $Z \succ 0$ such that $c_i = \text{trace}(ZF_i)$, $i = 1, \ldots, q$, then both primal and dual optimal values are attained by some optimal pair (x^*, Z^*). In that case, we can characterize such optimal pairs as follows. In view of the equality constraints of the dual problem, the duality gap can be expressed as

$$\begin{aligned} p^* - d^* &= c^T x^* - \text{trace}(Z^* F_0) \\ &= -\text{trace}(Z^* F(x^*)). \end{aligned}$$

A zero duality gap is equivalent to $\text{trace}(Z^* F(x^*)) = 0$, which in turn is equivalent to $Z^* F(x^*) = O$, where O denotes the zero matrix, since the product of a positive semi-definite and a negative semi-definite matrix has zero trace if and only if it is zero.

To summarize, consider the SDP (16) and its Lagrange dual (18). If either problem is strictly feasible, then they share the same optimal value. If both problems are strictly feasible, then the optimal values of both problems are attained and coincide. In this case, a primal-dual pair (x^*, Z^*) is optimal if and only if

$$\begin{aligned} &F(x^*) \preceq 0, \\ &Z^* \succeq 0, \\ &c_i = -\text{trace}(Z^* F_i),\ \ i = 1, \ldots, q, \\ &Z^* F(x^*) = O. \end{aligned}$$

The above conditions represent the generalization to the SDP case, of the Karush-Kuhn-Tucker (KKT) conditions of linear programming. The first three sets of conditions express that x^* and Z^* are feasible for their respective problems; the last condition expresses a "complimentarity" condition that generalizes to the SDP case, the complementarity condition of linear programming.

For a pair of strictly feasible primal-dual SDPs, solving the primal minimization problem is equivalent to maximizing the dual problem and both can thus be considered simultaneously. Algorithms indeed make use of this relationship and use the duality gap as a stopping criterion. A general-purpose program such as SeDuMi (Sturm, 1999) handles those problems efficiently. This

code uses interior-point methods for SDP (Nesterov and Nemirovsky, 1994); these methods have a worst-case complexity of $O(q^2p^{2.5})$ for the general problem (16). In practice, problem structure can be exploited for great computational savings: e.g., when $F(x) \in \mathbb{R}^{p\times p}$ consists of L diagonal blocks of size p_i, $i = 1, \ldots, L$, these methods have a worst-case complexity of $O(q^2(\sum_{i=1}^{L} p_i^2)p^{0.5})$ (Vandenberghe and Boyd, 1996).

4 Algorithms for learning kernels

We work in a transduction setting, where some of the data (the training set) are labelled, and the remainder (the test set) are unlabelled, and the aim is to predict the labels of the test data. In this setting, optimizing the kernel corresponds to choosing a kernel matrix. This matrix has the form

$$K = \begin{pmatrix} K_{tr} & K_{tr,t} \\ K_{tr,t}^T & K_t \end{pmatrix}, \tag{19}$$

where $K_{ij} = \langle \Phi(x_i), \Phi(x_j) \rangle$, $i, j = 1, \ldots, n_{tr}, n_{tr}+1, \ldots, n_{tr}+n_t$ with n_{tr} and n_t the number of labelled (training) and unlabelled (test) data points respectively. By optimizing a cost function over the "training-data block" K_{tr}, we want to learn the optimal mixed block $K_{tr,t}$ and the optimal "test-data block" K_t.

This implies that training and test-data blocks must somehow be entangled: tuning training-data entries in K (to optimize their embedding) should imply that test-data entries are automatically tuned in some way as well. This can be achieved by constraining the search space of possible kernel matrices: we control the capacity of the search space of possible kernel matrices in order to prevent overfitting and achieve good generalization on test data.

We first consider a general optimization problem in which the kernel matrix K is restricted to a convex subset $\mathcal{K}$ of $\mathcal{P}$, the positive semi-definite cone. We then consider two specific examples. The first is the set of positive semi-definite matrices that can be expressed as a linear combination,

$$K = \sum_{i=1}^{m} \mu_i K_i, \tag{20}$$

of kernel matrices from the set $\{K_1, \ldots, K_m\}$. In this case, the set $\mathcal{K}$ is the intersection of a low-dimensional linear subspace with the positive semi-definite cone $\mathcal{P}$. Geometrically this can be viewed as computing all embeddings (for every K_i), in disjoint feature spaces, and then weighting these. The set $\{K_1, \ldots, K_m\}$ could be a set of initial "guesses" of the kernel matrix, e.g., linear, Gaussian or polynomial kernels with different kernel parameter values. Instead of fine-tuning the kernel parameter for a given kernel using cross-validation, one can now evaluate the given kernel for a range of kernel parameters and then optimize the weights in the linear combination of the obtained kernel matrices. Alternatively, the K_i could be chosen as the rank-one matrices $K_i = v_i v_i^T$, with v_i a subset of the eigenvectors of K_0, an initial kernel matrix, or with v_i some other set of orthogonal vectors. A practically important form is the case in which a diverse set of possibly good Gram matrices K_i (similarity measures/representations) has been constructed, e.g., using heterogenous data sources. The challenge is to combine these measures into one optimal similarity measure (embedding), to be used for learning.

The second example of a restricted set $\mathcal{K}$ of kernels is the set of positive semi-definite matrices that can be expressed as a linear combination,

$$K = \sum_{i=1}^{m} \mu_i K_i,$$

of kernel matrices from the set $\{K_1, \ldots, K_m\}$, but with the parameters μ_i constrained to be non-negative. This is a subset of the set defined in (20) above, and so it further constrains the class of functions that can be represented. It has two advantages: we shall see that the corresponding optimization problem has significantly reduced computational complexity, and it is more convenient for studying the statistical properties of a class of kernel matrices.

As we will see in Section 5, we can estimate the performance of support vector machines for transduction using properties of the class $\mathcal{K}$: for a thresholded version of $f(\mathbf{x})$, the proportion of errors on the test data is, with probability $1 - \delta$, bounded by

$$\frac{1}{n} \sum_{i=1}^{n} \phi(Y_i f(X_i)) + \frac{1}{\sqrt{n}} \left(4 + \sqrt{2 \log(1/\delta)} + \sqrt{\frac{C(\mathcal{K})}{n\gamma^2}} \right), \tag{21}$$

where ϕ is the 1-norm margin cost function, γ the margin on the training set and $C(\mathcal{K})$ is a certain measure of the complexity of the kernel class $\mathcal{K}$, which grows linearly with the trace of the kernel matrices in $\mathcal{K}$.

4.1 Hard margin

In this section, we derive the main optimization result of the paper: maximizing the margin of a hard margin SVM with respect to the kernel matrix can be realized in a semi-definite programming framework. The soft margin case is an extension of this basic result and will be discussed in a later section.

Inspired by (21), let us try to find the kernel matrix K in some convex subset $\mathcal{K}$ of positive semi-definite matrices for which the corresponding embedding shows maximal margin on the training data, keeping the trace of K constant:

$$\min_{K \in \mathcal{K}} w(K_{tr}) \qquad \text{s.t. } \operatorname{trace}(K) = c. \tag{22}$$

We first notice a fundamental property of the margin, a property that is crucial for the remainder of the paper.

Proposition 15 *The quantity*

$$w(K) = \max_{\alpha} \; 2\alpha^T e - \alpha^T G(K) \alpha \;\; : \;\; \alpha \geq 0, \;\; \alpha^T y = 0,$$

is convex in K.

This is easily seen by considering first that $2\alpha^T e - \alpha^T G(K)\alpha$ is an affine function of K, and hence is a convex function as well. Secondly, we notice that $w(K)$ is the pointwise maximum of such convex functions and is thus convex.

Problem (22) is now a convex optimization problem. The following theorem shows that, for a suitable choice of the set $\mathcal{K}$, this problem can be cast as an SDP.

Theorem 16 *Given a linearly separable labelled sample $S_l = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ with corresponding set of labels $y \in \mathbb{R}^n$, the kernel matrix $K \in \mathcal{K}$ that optimizes (22) can be found by solving the following problem:*

$$\begin{aligned} \min_{K,t,\lambda,\nu} \quad & t \\ \textit{subject to} \quad & \textit{trace}(K) = c, \\ & K \in \mathcal{K}, \\ & \begin{pmatrix} G(K_{tr}) & e+\nu+\lambda y \\ (e+\nu+\lambda y)^T & t \end{pmatrix} \succeq 0, \\ & \nu \geq 0. \end{aligned} \tag{23}$$

Proof We begin by substituting $w(K_{tr})$—the squared inverse margin as defined in (2)—into (22), which yields:

$$\min_{K \in \mathcal{K}} \max_{\alpha} \; 2\alpha^T e - \alpha^T G(K_{tr})\alpha \;:\; \alpha \geq 0, \;\; \alpha^T y = 0, \;\; \text{trace}(K) = c, \tag{24}$$

with c a constant. Assume that $K_{tr} \succ 0$, hence $G(K_{tr}) \succ 0$ (the following can be extended to the general semi-definite case). From Proposition 15, we know that $w(K_{tr})$ is convex in K. Given the convex constraints in (24), the optimization problem is thus certainly convex in K. We write this as:

$$\begin{aligned} \min_{K \in \mathcal{K}, t} \; t \;:\quad & t \geq \max_{\alpha} \; 2\alpha^T e - \alpha^T G(K_{tr})\alpha, \\ & \alpha \geq 0, \;\; \alpha^T y = 0, \;\; \text{trace}(K) = c. \end{aligned} \tag{25}$$

We now express the constraint $t \geq \max_\alpha \; 2\alpha^T e - \alpha^T G(K_{tr})\alpha$ as an LMI using duality. In particular, duality will allow us to drop the minimization and the Schur complement lemma then yields an LMI.

Define the Lagrangian of the maximization problem (2) by

$$\mathcal{L}(\alpha, \nu, \lambda) = 2\alpha^T e - \alpha^T G(K_{tr})\alpha + 2\nu^T \alpha + 2\lambda y^T \alpha,$$

where $\lambda \in \mathbb{R}, \nu \in \mathbb{R}^n$. By duality, we have

$$w(K_{tr}) = \max_{\alpha} \min_{\nu \geq 0, \lambda} \; \mathcal{L}(\alpha, \nu, \lambda) = \min_{\nu \geq 0, \lambda} \max_{\alpha} \; \mathcal{L}(\alpha, \nu, \lambda),$$

where $\nu \geq 0 \Leftrightarrow \nu_i \geq 0$ for $i = 1, \ldots, n$. Since $G(K_{tr}) \succ 0$, at the optimum, we have

$$\alpha = G(K_{tr})^{-1}(e + \nu + \lambda y),$$

and we can form the dual problem

$$w(K_{tr}) = \min_{\nu, \lambda} \; (e+\nu+\lambda y)^T G(K_{tr})^{-1}(e+\nu+\lambda y) \;:\; \nu \geq 0.$$

This implies that for any $t > 0$, the constraint $w(K_{tr}) \leq t$ holds if and only if there exist $\nu \geq 0$ and λ such that

$$(e + \nu + \lambda y)^T G(K_{tr})^{-1}(e + \nu + \lambda y) \leq t,$$

or, equivalently (using the Schur complement lemma), such that

$$\begin{pmatrix} G(K_{tr}) & e + \nu + \lambda y \\ (e + \nu + \lambda y)^T & t \end{pmatrix} \succeq 0$$

holds. Taking this into account, (25) can be expressed as:

$$\begin{aligned} \min_{K,t,\lambda,\nu} \quad & t \\ \text{subject to} \quad & \text{trace}(K) = c, \\ & K \in \mathcal{K}, \\ & \begin{pmatrix} G(K_{tr}) & e + \nu + \lambda y \\ (e + \nu + \lambda y)^T & t \end{pmatrix} \succeq 0, \\ & \nu \geq 0, \end{aligned}$$

which yields (23). Notice that $\nu \geq 0 \Leftrightarrow \text{diag}(\nu) \succeq 0$, and thus an LMI. ■

Notice that if $\mathcal{K} = \{K \succeq 0\}$, this optimization problem is an SDP in the standard form (9). Of course, in that case there is no constraint to ensure that a large margin on the training data will give a large margin on the test data. Indeed, it is easy to see that the criterion would be optimized with a test matrix $K_t = 0$.

Consider the constraint $\mathcal{K} = \text{span}\{K_1, \ldots, K_m\} \cap \{K \succeq 0\}$. We obtain the following convex optimization problem:

$$\begin{aligned} \min_{K} \quad & w(K_{tr}) \\ \text{subject to} \quad & \text{trace}(K) = c, \\ & K \succeq 0, \\ & K = \sum_{i=1}^{m} \mu_i K_i, \end{aligned} \tag{26}$$

which can be written in the standard form of a semi-definite program, in a manner analogous to (23):

$$\begin{aligned} \min_{\mu_i,t,\lambda,\nu} \quad & t \\ \text{subject to} \quad & \text{trace}\left(\sum_{i=1}^{m} \mu_i K_i\right) = c, \\ & \sum_{i=1}^{m} \mu_i K_i \succeq 0, \\ & \begin{pmatrix} G(\sum_{i=1}^{m} \mu_i K_{i,tr}) & e + \nu + \lambda y \\ (e + \nu + \lambda y)^T & t \end{pmatrix} \succeq 0, \\ & \nu \geq 0. \end{aligned} \tag{27}$$

Notice that the SDP approach is consistent with the bound in (21). The margin is optimized over the labelled data (via the use of $K_{i,tr}$), while the positive semi-definiteness and the trace constraint are imposed for the entire kernel matrix K (via the use of K_i). This leads to a general method for learning the kernel matrix with semi-definite programming, when using a margin criterion for hard margin SVMs. Applying the complexity results mentioned in Section 3.3 leads to a worst-case complexity $O(n^{4.5})$ when using interior-point methods to solve this particular SDP.

Furthermore, this gives a new transduction method for hard margin SVMs. Whereas Vapnik's original method for transduction scales exponentially in the number of test samples, the new SDP method has polynomial time complexity.

Remark. For the specific case in which the K_i are rank-one matrices $K_i = v_i v_i^T$, with v_i orthonormal (e.g., the normalized eigenvectors of an initial kernel matrix K_0), the semi-definite program reduces to a quadratically constrained quadratic program (QCQP) (see Appendix A):

$$\begin{aligned} \max_{\alpha,t} \quad & 2\alpha^T e - ct \\ \text{subject to} \quad & t \geq (\breve{v}_i^T \alpha)^2, i = 1, \ldots, m \\ & \alpha^T y = 0, \\ & \alpha \geq 0, \end{aligned} \tag{28}$$

with $\breve{v}_i = \text{diag}(y)\ v_i(1 : n_{tr})$. This QCQP problem is a special form of SDP (Boyd and Vandenberghe, 2001) which can be solved efficiently with programs such as SeDuMi (Sturm, 1999) or Mosek (Andersen and Andersen, 2000). These codes use interior-point methods for QCQP (Nesterov and Nemirovsky, 1994) which yield a worst-case complexity of $O(mn_{tr}^2 + n_{tr}^3)$. This implies a major improvement compared to the worst-case complexity of a general SDP. Furthermore, the codes simultaneously solve the above problem and its dual form. They thus return optimal values for the dual variables as well—this allows us to obtain the optimal weights μ_i, for $i = 1, \ldots, m$.

4.2 Hard margin with kernel matrices that are positive linear combinations

To learn a kernel matrix from this linear class $\mathcal{K}$, one has to solve a convex optimization problem, more precisely a semi-definite programming problem. General-purpose programs such as SeDuMi (Sturm, 1999) use interior-point methods to solve SDP problems (Nesterov and Nemirovsky, 1994); they are polynomial time, but have a worst-case complexity $O(n^{4.5})$ in this particular case.

Consider a further restriction on the set of kernel matrices, where the matrices are restricted to positive linear combinations of kernel matrices $\{K_1, \ldots, K_m\} \cap \{K \succeq 0\}$:

$$K = \sum_{i=1}^{m} \mu_i K_i, \qquad \mu \geq 0.$$

Assuming positive weights yields a smaller set of kernel matrices, because the weights need not be positive for K to be positive semi-definite, even if the components K_i are positive semi-definite. Moreover, the restriction has beneficial computational effects: (1) the general SDP reduces to a QCQP, which has the significantly lower complexity of $O(mn_{tr}^2 + n_{tr}^3)$; (2) the constraint can result in improved numerical stability—it prevents the algorithm from using large weights with opposite sign that cancel. Finally, we shall see in Section 5 that the constraint also yields better estimates of the generalization performance of these algorithms.

Solving the original learning problem (26) subject to the extra constraint $\mu \geq 0$ yields:

$$\begin{aligned} \min_{K} \max_{\alpha \,:\, \alpha \geq 0, \alpha^T y = 0} \quad & 2\alpha^T e - \alpha^T G(K_{tr})\alpha \\ \text{subject to} \quad & \text{trace}(K) = c, \\ & K \succeq 0, \\ & K = \sum_{i=1}^{m} \mu_i K_i, \\ & \mu \geq 0, \end{aligned}$$

when $w(K_{tr})$ is expressed using (2). We can omit the second constraint, because this is implied by the last two constraints, if $K_i \succeq 0$. If we let $\text{trace}(K_i) = r_i$, where $r \in \mathbb{R}^m$, the problem reduces to:

$$\begin{aligned} \min_{\mu} \max_{\alpha \,:\, \alpha \geq 0, \alpha^T y = 0} \quad & 2\alpha^T e - \alpha^T G(\sum_{i=1}^{m} \mu_i K_{i,tr})\alpha \\ \text{subject to} \quad & \mu^T r = c, \\ & \mu \geq 0, \end{aligned}$$

where $K_{i,tr} = K_i(1 : n_{tr}, 1 : n_{tr})$. We can write this as:

$$\begin{aligned} & \min_{\mu \,:\, \mu \geq 0, \mu^T r = c} \max_{\alpha \,:\, \alpha \geq 0, \alpha^T y = 0} 2\alpha^T e - \alpha^T \text{diag}(y)(\sum_{i=1}^{m} \mu_i K_{i,tr})\text{diag}(y)\alpha \\ &= \min_{\mu \,:\, \mu \geq 0, \mu^T r = c} \max_{\alpha \,:\, \alpha \geq 0, \alpha^T y = 0} 2\alpha^T e - \sum_{i=1}^{m} \mu_i \alpha^T \text{diag}(y) K_{i,tr} \text{diag}(y)\alpha \\ &= \min_{\mu \,:\, \mu \geq 0, \mu^T r = c} \max_{\alpha \,:\, \alpha \geq 0, \alpha^T y = 0} 2\alpha^T e - \sum_{i=1}^{m} \mu_i \alpha^T G(K_{i,tr})\alpha \\ &= \max_{\alpha \,:\, \alpha \geq 0, \alpha^T y = 0} \min_{\mu \,:\, \mu \geq 0, \mu^T r = c} 2\alpha^T e - \sum_{i=1}^{m} \mu_i \alpha^T G(K_{i,tr})\alpha, \end{aligned}$$

with $G(K_{i,tr}) = \text{diag}(y)K_{i,tr}\text{diag}(y)$. The interchange of the order of the minimization and the maximization is justified by standard results in convex optimization (see, e.g., Boyd and Vandenberghe, 2001) because the objective is convex in μ (it is linear in μ) and concave in α, because the minimization problem is strictly feasible in μ, and the maximization problem is strictly feasible in α (we can skip the case for all elements of y having the same sign, because we cannot even consider a margin in such a case). We thus obtain:

$$\begin{aligned} & \max_{\alpha \,:\, \alpha \geq 0, \alpha^T y = 0} \min_{\mu \,:\, \mu \geq 0, \mu^T r = c} 2\alpha^T e - \sum_{i=1}^{m} \mu_i \alpha^T G(K_{i,tr})\alpha \\ &= \max_{\alpha \,:\, \alpha \geq 0, \alpha^T y = 0} \left[2\alpha^T e - \max_{\mu \,:\, \mu \geq 0, \mu^T r = c} \left(\sum_{i=1}^{m} \mu_i \alpha^T G(K_{i,tr})\alpha \right) \right] \\ &= \max_{\alpha \,:\, \alpha \geq 0, \alpha^T y = 0} \left[2\alpha^T e - \max_{i} \left(\frac{c}{r_i} \alpha^T G(K_{i,tr})\alpha \right) \right]. \end{aligned}$$

Finally, this can be reformulated as follows:

$$\begin{aligned} \max_{\alpha,t} \quad & 2\alpha^T e - ct \\ \text{subject to} \quad & t \geq \frac{1}{r_i}\alpha^T G(K_{i,tr})\alpha, \quad i = 1, \ldots, m \\ & \alpha^T y = 0, \\ & \alpha \geq 0. \end{aligned} \tag{29}$$

This problem is a convex optimization problem, more precisely a QCQP (Boyd and Vandenberghe, 2001). Note once again that such problems can be solved with worst-case complexity of $O(mn_{tr}^2 + n_{tr}^3)$. Note also that the optimal weights $\mu_i, \quad i = 1, \ldots, m$, can be recovered from the primal-dual solution found by standard software such as SeDuMi (Sturm, 1999).

4.3 1-Norm Soft margin

For the case of non-linearly separable data, we can consider the 1-norm soft margin cost function in (3). Training the SVM for a given kernel involves minimizing this quantity with respect to $\mathbf{w}, b, \xi$, which yields the optimal value (4): obviously this minimum is a function of the particular choice of K, which is expressed explicitly in (4) as a dual problem. Let us now optimize this quantity with respect to the kernel matrix K, i.e., let us try to find the kernel matrix $K \in \mathcal{K}$ for which the corresponding embedding shows minimal $w_{S1}(K_{tr})$, keeping the trace of K constant:

$$\min_{K \in \mathcal{K}} w_{S1}(K_{tr}) \qquad \text{s.t. trace}(K) = c. \tag{30}$$

This is again a convex optimization problem.

Theorem 17 *Given a labelled sample $S_l = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ with corresponding set of labels $y \in \mathbb{R}^n$, the kernel matrix $K \in \mathcal{K}$ that optimizes (30) can be found by solving the following convex optimization problem:*

$$\begin{aligned} \min_{K,t,\lambda,\nu,\delta} \quad & t \\ \textit{subject to} \quad & \textit{trace}(K) = c, \\ & K \in \mathcal{K}, \\ & \begin{pmatrix} G(K_{tr}) & e + \nu - \delta + \lambda y \\ (e + \nu - \delta + \lambda y)^T & t - 2C\delta^T e \end{pmatrix} \succeq 0, \\ & \nu \geq 0 \\ & \delta \geq 0. \end{aligned} \tag{31}$$

The 1-norm soft margin case follows as an easy extension of the main result in the previous section. A detailed proof is given in Appendix B.

Again, if $\mathcal{K} = \{K \succeq 0\}$, this is an SDP. Adding the additional constraint (20) that K is a linear combination of fixed kernel matrices leads to the following SDP:

$$\begin{aligned} \min_{\mu_i, t, \lambda, \nu, \delta} \quad & t \\ \text{subject to} \quad & \text{trace}\left(\sum_{i=1}^{m} \mu_i K_i\right) = c, \\ & \sum_{i=1}^{m} \mu_i K_i \succeq 0, \\ & \begin{pmatrix} G(\sum_{i=1}^{m} \mu_i K_{i,tr}) & e + \nu - \delta + \lambda y \\ (e + \nu - \delta + \lambda y)^T & t - 2C\delta^T e \end{pmatrix} \succeq 0, \\ & \nu, \delta \geq 0. \end{aligned} \tag{32}$$

Remark. For the specific case in which the K_i are rank-one matrices $K_i = v_i v_i^T$, with v_i orthonormal (e.g., the normalized eigenvectors of an initial kernel matrix K_0), the SDP reduces to a QCQP in a manner analogous to the hard margin case:

$$\begin{aligned} \max_{\alpha, t} \quad & 2\alpha^T e - ct \\ \text{subject to} \quad & t \geq (\breve{v}_i^T \alpha)^2, i = 1, \ldots, m \\ & \alpha^T y = 0, \\ & C \geq \alpha \geq 0, \end{aligned} \tag{33}$$

with $\breve{v}_i = \text{diag}(y)\ v_i(1 : n_{tr})$.

Solving the original learning problem subject to the extra constraint $\mu \geq 0$ yields, after a similar derivation:

$$\begin{aligned} \max_{\alpha, t} \quad & 2\alpha^T e - ct \\ \text{subject to} \quad & t \geq \frac{1}{r_i} \alpha^T G(K_{i,tr}) \alpha, \quad i = 1, \ldots, m \\ & \alpha^T y = 0, \\ & C \geq \alpha \geq 0. \end{aligned} \tag{34}$$

4.4 2-Norm Soft Margin

For the case of non-linearly separable data, we can also consider the 2-norm soft margin cost function (5). Again, training for a given kernel will minimize this quantity with respect to $\mathbf{w}, b, \xi$ and the minimum is a function of the particular choice of K, as expressed in (6) in dual form. Let us now optimize this quantity with respect to the kernel matrix K:

$$\min_{K \in \mathcal{K}} w_{S2}(K_{tr}) \qquad \text{s.t. trace}(K) = c. \tag{35}$$

This is again a convex optimization problem, and can be restated as follows.

Theorem 18 *Given a labelled sample $S_l = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ with corresponding set of labels $y \in \mathbb{R}^n$, the kernel matrix $K \in \mathcal{K}$ that optimizes (35) can be found by solving the following optimization problem.*

$$\begin{aligned} \min_{K,t,\lambda,\nu} \quad & t \\ \textit{subject to} \quad & \textit{trace}(K) = c, \\ & K \in \mathcal{K}, \\ & \begin{pmatrix} G(K_{tr}) + \frac{1}{C} I_{n_{tr}} & e + \nu + \lambda y \\ (e + \nu + \lambda y)^T & t \end{pmatrix} \succeq 0, \\ & \nu \geq 0. \end{aligned} \tag{36}$$

Proof After substitution of $w_{S2}(K_{tr})$ as defined in (6), (35) becomes:

$$\min_{K \succeq 0} \max_{\alpha} 2\alpha^T e - \alpha^T \left(G(K_{tr}) + \frac{1}{C} I_{n_{tr}} \right) \alpha \ : \ \alpha \geq 0, \ \ \alpha^T y = 0, \ \ \text{trace}(K) = c \tag{37}$$

with c a constant.

We note that $w_{S2}(K_{tr})$ is convex in K (it is the pointwise maximum of affine functions of K). Given the convex constraints in (37), the optimization problem is thus certainly convex in K. The theorem is a direct extension of the main result in Section 4.1. This can be seen by comparing (24) and (37). Replacing the matrix $G(K_{tr})$ in (23) by $G(K_{tr}) + \frac{1}{C} I_{n_{tr}}$ directly yields (36). ■

Again, if $\mathcal{K} = \{K \succeq 0\}$, this is an SDP. Moreover, constraining K to be a linear combination of fixed kernel matrices, we obtain:

$$\begin{aligned} \min_{\mu_i,t,\lambda,\nu} \quad & t \\ \text{subject to} \quad & \text{trace} \left(\sum_{i=1}^{m} \mu_i K_i \right) = c, \\ & \sum_{i=1}^{m} \mu_i K_i \succeq 0, \\ & \begin{pmatrix} G(\sum_{i=1}^{m} \mu_i K_{i,tr}) + \frac{1}{C} I_{n_{tr}} & e + \nu + \lambda y \\ (e + \nu + \lambda y)^T & t \end{pmatrix} \succeq 0, \\ & \nu \geq 0. \end{aligned} \tag{38}$$

Also, when the K_i are rank-one matrices, $K_i = v_i v_i^T$, with v_i orthonormal, we obtain a QCQP (see Appendix C):

$$\begin{aligned} \max_{\alpha,t} \quad & 2\alpha^T e - \frac{1}{C} \alpha^T \alpha - ct \\ \text{subject to} \quad & t \geq (\breve{v}_i^T \alpha)^2, i = 1, \ldots, m \\ & \alpha^T y = 0, \\ & \alpha \geq 0, \end{aligned} \tag{39}$$

and, finally, imposing the constraint $\mu \geq 0$ yields:

$$\begin{aligned} \max_{\alpha,t} \quad & 2\alpha^T e - \frac{1}{C}\alpha^T\alpha - ct \\ \text{subject to} \quad & t \geq \frac{1}{r_i}\alpha^T G(K_{i,tr})\alpha, \quad i = 1,\ldots,m \\ & \alpha^T y = 0, \\ & \alpha \geq 0, \end{aligned} \tag{40}$$

following a similar derivation as before.

4.5 Learning the 2-Norm Soft Margin Parameter $\tau = \frac{1}{C}$

This section shows how the 2-norm soft margin parameter of SVMs can be learned using SDP or QCQP. More details can be found in De Bie et al. (2002).

In the previous section, we tried to find the kernel matrix $K \in \mathcal{K}$ for which the corresponding embedding yields minimal $w_{S2}(K_{tr})$, keeping the trace of K constant. Similarly, we can simultaneously and automatically tune the parameter $\tau = 1/C$ such that the quantity $w_{S2}(K_{tr}, \tau)$ is minimized, as is proposed in De Bie et al. (2002). First of all, consider the dual formulation (6) and notice that $w_{S2}(K_{tr}, \tau)$ is convex in $\tau = 1/C$ (being the pointwise maximum of affine and thus convex functions in τ). Secondly, since $\tau \to \infty$ leads to $w_{S2}(K_{tr}, \tau) \to 0$, we impose the constraint $\text{trace}\,(K + \tau I_n) = c$. This results in the following convex optimization problem:

$$\min_{K \in \mathcal{K}, \tau \geq 0} w_{S2}(K_{tr}, \tau) \qquad \text{s.t. } \text{trace}\,(K + \tau I_n) = c. \tag{41}$$

According to Theorem 18, this can be restated as follows:

$$\begin{aligned} \min_{K,t,\lambda,\nu,\tau} \quad & t \\ \text{subject to} \quad & \text{trace}\,(K + \tau I_n) = c, \\ & K \in \mathcal{K}, \\ & \begin{pmatrix} G(K_{tr}) + \tau I_{n_{tr}} & e + \nu + \lambda y \\ (e + \nu + \lambda y)^T & t \end{pmatrix} \succeq 0, \\ & \nu, \tau \geq 0. \end{aligned} \tag{42}$$

Again, if $\mathcal{K} = \{K \succeq 0\}$, this is an SDP. Imposing the additional constraint that K is a linear function of fixed kernel matrices, we obtain the SDP:

$$\begin{aligned} \min_{\mu_i,t,\lambda,\nu,\tau} \quad & t \\ \text{subject to} \quad & \text{trace}\left(\sum_{i=1}^m \mu_i K_i + \tau I_n\right) = c, \\ & \sum_{i=1}^m \mu_i K_i \succeq 0, \\ & \begin{pmatrix} G(\sum_{i=1}^m \mu_i K_{i,tr}) + \tau I_{n_{tr}} & e + \nu + \lambda y \\ (e + \nu + \lambda y)^T & t \end{pmatrix} \succeq 0, \\ & \nu, \tau \geq 0, \end{aligned} \tag{43}$$

and imposing the additional constraint that the K_i are rank-one matrices, we obtain a QCQP:

$$\begin{aligned} \max_{\alpha,t} \quad & 2\alpha^T e - ct \\ \text{subject to} \quad & t \geq (\breve{v}_i^T \alpha)^2, i = 1, \ldots, m \\ & t \geq \frac{1}{n}\alpha^T \alpha \\ & \alpha^T y = 0, \\ & \alpha \geq 0, \end{aligned} \tag{44}$$

with $\breve{v}_i = \text{diag}(y)\ \bar{v}_i = \text{diag}(y)\ v_i(1 : n_{tr})$. Finally, imposing the constraint that $\mu \geq 0$ yields the following:

$$\begin{aligned} \max_{\alpha,t} \quad & 2\alpha^T e - ct \\ \text{subject to} \quad & t \geq \frac{1}{r_i}\alpha^T G(K_{i,tr})\alpha, \quad i = 1, \ldots, m \\ & t \geq \frac{1}{n}\alpha^T \alpha \\ & \alpha^T y = 0, \\ & \alpha \geq 0, \end{aligned} \tag{45}$$

$$\tag{46}$$

which, as before, is a QCQP.

Solving (45) corresponds to learning the kernel matrix as a positive linear combination of kernel matrices according to a 2-norm soft margin criterion and simultaneously learning the 2-norm soft margin parameter $\tau = 1/C$. Comparing (45) with (29), we can see that this reduces to learning an augmented kernel matrix K' as a positive linear combination of kernel matrices and the identity matrix, $K' = K + \tau I_n = \sum_{i=1}^m \mu_i K_i + \tau I_n$, using a hard margin criterion. However, there is an important difference: when evaluating the resulting classifier, the actual kernel matrix K is used, instead of the augmented K' (see, for example, Shawe-Taylor and Cristianini, 1999).

For $m = 1$, we notice that (43) directly reduces to (45) if $K_1 \succeq 0$. This corresponds to automatically tuning the parameter $\tau = 1/C$ for a 2-norm soft margin SVM with kernel matrix K_1. So, even when not learning the kernel matrix, this approach can be used to learn the 2-norm soft margin parameter $\tau = 1/C$ automatically.

4.6 Alignment

In this section, we consider the problem of optimizing the alignment between a set of labels and a kernel matrix from some class $\mathcal{K}$ of positive semi-definite kernel matrices. We show that, if $\mathcal{K}$ is a class of linear combinations of fixed kernel matrices, this problem can be cast as an SDP. This result generalizes the approach presented in Cristianini et al. (2002).

Theorem 19 *The kernel matrix $K \in \mathcal{K}$ which is maximally aligned with the set of labels $y \in \mathbb{R}^{n_{tr}}$ can be found by solving the following optimization problem:*

$$\begin{aligned} \max_{A,K} \quad & \langle K_{tr}, yy^T \rangle_F \\ \textit{subject to} \quad & \textit{trace}(A) \leq 1 \\ & \begin{pmatrix} A & K^T \\ K & I_n \end{pmatrix} \succeq 0 \\ & K \in \mathcal{K}, \end{aligned} \tag{47}$$

where I_n is the identity matrix of dimension n.

Proof We want to find the kernel matrix K which is maximally aligned with the set of labels y:

$$\begin{aligned} \max_{K} \quad & \hat{A}(S, K_{tr}, yy^T) \\ \text{subject to} \quad & K \in \mathcal{K},\ \text{trace}(K) = 1. \end{aligned}$$

This is equivalent to the following optimization problem:

$$\begin{aligned} \max_{K} \quad & \langle K_{tr}, yy^T \rangle_F \\ \text{subject to} \quad & \langle K, K \rangle_F = 1 \\ & K \in \mathcal{K},\ \text{trace}(K) = 1. \end{aligned} \tag{48}$$

To express this in the standard form (9) of a semi-definite program, we need to express the quadratic equality constraint $\langle K, K \rangle_F = 1$ as an LMI. First, notice that (48) is equivalent to

$$\begin{aligned} \max_{K} \quad & \langle K_{tr}, yy^T \rangle_F \\ \text{subject to} \quad & \langle K, K \rangle_F \leq 1 \\ & K \in \mathcal{K}. \end{aligned} \tag{49}$$

Indeed, we are maximizing an objective which is linear in the entries of K, so at the optimum $K = K^*$, the constraint $\langle K, K \rangle_F = \text{trace}(K^T K) \leq 1$ is achieved: $\langle K^*, K^* \rangle_F = 1$. The quadratic inequality constraint in (49) is now equivalent to

$$\exists A \ : \ K^T K \preceq A \quad \text{and} \quad \text{trace}(A) \leq 1.$$

Indeed, $A - K^T K \succeq 0$ implies $\text{trace}(A - K^T K) = \text{trace}(A) - \text{trace}(K^T K) \geq 0$ because of linearity of the trace. Using the Schur complement lemma, we can express $A - K^T K \succeq 0$ as an LMI:

$$A - K^T K \succeq 0 \Leftrightarrow \begin{pmatrix} A & K^T \\ K & I_n \end{pmatrix} \succeq 0.$$

We can thus rewrite the optimization problem (48) as:

$$\begin{aligned} \max_{A,K} \quad & \langle K_{tr}, yy^T \rangle_F \\ \text{subject to} \quad & \text{trace}(A) \leq 1 \\ & \begin{pmatrix} A & K^T \\ K & I_n \end{pmatrix} \succeq 0 \\ & K \in \mathcal{K}. \end{aligned}$$

which corresponds to (47). ■

Notice that, when $\mathcal{K}$ is the set of all positive semi-definite matrices, this is an SDP (an inequality constraint corresponds to a one-dimensional LMI; consider the entries of the matrices A and K as the unknowns x_i). In that case, one solution of (47) is found by simply selecting $K_{tr} = \frac{c}{n} yy^T$, for which the alignment (7) is equal to one and thus maximized.

Adding the additional constraint (20) that K is a linear combination of fixed kernel matrices leads to

$$\begin{aligned} \max_{K} \quad & \left\langle K_{tr}, yy^T \right\rangle_F \\ \text{subject to} \quad & \langle K, K \rangle_F \leq 1, \\ & K \succeq 0, \\ & K = \sum_{i=1}^{m} \mu_i K_i, \end{aligned} \tag{50}$$

which can be written in the standard form of a semi-definite program, in a similar way as for (47):

$$\begin{aligned} \max_{A,\mu_i} \quad & \left\langle \sum_{i=1}^{m} \mu_i K_{i,tr}, yy^T \right\rangle_F \\ \text{subject to} \quad & \operatorname{trace}(A) \leq 1, \\ & \begin{pmatrix} A & \sum_{i=1}^{m} \mu_i K_i^T \\ \sum_{i=1}^{m} \mu_i K_i & I_n \end{pmatrix} \succeq 0, \\ & \sum_{i=1}^{m} \mu_i K_i \succeq 0. \end{aligned} \tag{51}$$

Remark. For the specific case where the K_i are rank-one matrices $K_i = v_i v_i^T$, with v_i orthonormal (e.g., the normalized eigenvectors of an initial kernel matrix K_0), the semi-definite program reduces to a QCQP (see Appendix D):

$$\begin{aligned} \max_{\mu_i} \quad & \sum_{i=1}^{m} \mu_i (\bar{v}_i^T y)^2 \\ \text{subject to} \quad & \sum_{i=1}^{m} \mu_i^2 \leq 1 \\ & \mu_i \geq 0, i = 1, \ldots, m \end{aligned} \tag{52}$$

with $\bar{v}_i = v_i(1 : n_{tr})$. This corresponds exactly to the QCQP obtained as an illustration in Cristianini et al. (2002), which is thus entirely captured by the general SDP result obtained in this section.

Solving the original learning problem (50) subject to the extra constraint $\mu \geq 0$ yields:

$$\begin{aligned} \max_{K} \quad & \left\langle K_{tr}, yy^T \right\rangle_F \\ \text{subject to} \quad & \langle K, K \rangle_F \leq 1, \\ & K \succeq 0, \\ & K = \sum_{i=1}^{m} \mu_i K_i, \\ & \mu \geq 0. \end{aligned}$$

We can omit the second constraint, because this is implied by the last two constraints, if $K_i \succeq 0$. This reduces to:

$$\begin{aligned} \max_{\mu} \quad & \left\langle \sum_{i=1}^{m} \mu_i K_{i,tr}, yy^T \right\rangle_F \\ \text{subject to} \quad & \left\langle \sum_{i=1}^{m} \mu_i K_i, \sum_{j=1}^{m} \mu_j K_j \right\rangle_F \leq 1, \\ & \mu \succeq 0, \end{aligned}$$

where $K_{i,tr} = K_i(1 : n_{tr}, 1 : n_{tr})$. Expanding this further yields:

$$\begin{aligned} \left\langle \sum_{i=1}^{m} \mu_i K_{i,tr}, yy^T \right\rangle_F &= \sum_{i=1}^{m} \mu_i \left\langle K_{i,tr}, yy^T \right\rangle_F \\ &= \mu^T q, \qquad (53) \\ \left\langle \sum_{i=1}^{m} \mu_i K_i, \sum_{j=1}^{m} \mu_j K_j \right\rangle_F &= \sum_{i,j=1}^{m} \mu_i \mu_j \left\langle K_i, K_j \right\rangle_F \\ &= \mu^T S \mu \qquad (54) \end{aligned}$$

with $q_i = \left\langle K_{i,tr}, yy^T \right\rangle_F = \text{trace}(K_{i,tr} yy^T) = \text{trace}(y^T K_{i,tr} y) = y^T K_{i,tr} y$ and $S_{ij} = \langle K_i, K_j \rangle_F$, where $q \in \mathbb{R}^m, S \in \mathbb{R}^{m \times m}$. We used the fact that $\text{trace}(ABC) = \text{trace}(BCA)$ (if the products are well-defined). We obtain the following learning problem:

$$\begin{aligned} \max_{\mu} \quad & \mu^T q \\ \text{subject to} \quad & \mu^T S \mu \leq 1, \\ & \mu \geq 0, \end{aligned}$$

which is a QCQP.

4.7 Induction

In previous sections we have considered the transduction setting, where it is assumed that the covariate vectors for both training (labelled) and test (unlabelled) data are known beforehand. While this setting captures many realistic learning problems, it is also of interest to consider

possible extensions of our approach to the fuller setting of induction, in which the covariates are known beforehand only for the training data.

Consider the following situation. We learn the kernel matrix as a positive linear combination of normalized kernel matrices K_i. Those K_i are obtained through the evaluation of a kernel function or through a known procedure (e.g., a string matching kernel), granting $K_i \succeq 0$. So, $K = \sum_{i=1}^{m} \mu_i K_i \succeq 0$. Normalization is done by replacing $K_i(k,l)$ by $K_i(k,l)/\sqrt{K_i(k,k) \cdot K_i(l,l)}$. In this case, the extension to an induction setting is elegant and simple.

Let n_{tr} be the number of training data points (all labelled). Consider the transduction problem for those n_{tr} data points and 1 unknown test point, e.g., for a hard margin SVM. The optimal weights $\mu_i^*, \quad i = 1, \ldots, m$ are learned by solving (29):

$$\begin{aligned} \max_{\alpha,t} \quad & 2\alpha^T e - ct \\ \text{subject to} \quad & t \geq \frac{1}{n_{tr}+1} \alpha^T G(K_{i,tr}) \alpha, \quad i = 1, \ldots, m \\ & \alpha^T y = 0, \\ & \alpha \geq 0. \end{aligned} \tag{55}$$

Even without knowing the test point and the entries of the K_i's related to it (column and row $n_{tr}+1$), we know that $K(n_{tr}+1, n_{tr}+1) = 1$ because of the normalization. So, $\text{trace}(K_i) = n_{tr}+1$. This allows solving for the optimal weights $\mu_i^*, \quad i = 1, \ldots, m$ and the optimal SVM parameters $\alpha_j^*, \quad j = 1, \ldots, n_{tr}$ and b^*, without knowing the test point. When a test point becomes available, we complete the K_i's by computing their $(n_{tr}+1)$-th column and row (evaluate the kernel function or follow the procedure and normalize). Combining those K_i with weights μ_i^* yields the final kernel matrix K, which can then be used to label the test point:

$$y = \text{sign}\left(\sum_{i=1}^{m} \sum_{j=1}^{n_{tr}} \mu_i^* \alpha_i K_i(x_j, x)\right).$$

Remark: The optimal weights are independent of the number of unknown test points that are considered in this setting. Consider the transduction problem (55) for l unknown test points instead of one unknown test point:

$$\begin{aligned} \max_{\tilde{\alpha},\tilde{t}} \quad & 2\tilde{\alpha}^T e - c\tilde{t} \\ \text{subject to} \quad & \tilde{t} \geq \frac{1}{n_{tr}+l} \tilde{\alpha}^T G(K_{i,tr}) \tilde{\alpha}, \quad i = 1, \ldots, m \\ & \tilde{\alpha}^T y = 0, \\ & \tilde{\alpha} \geq 0. \end{aligned} \tag{56}$$

One can see that solving (56) is equivalent to solving (55) where the optimal values relate as $\tilde{\alpha}^* = \frac{n_{tr}+l}{n_{tr}+1}\alpha^*$ and $\tilde{t}^* = \frac{n_{tr}+l}{n_{tr}+1}t^*$ and where the optimal weights $\mu_i^*, \quad i = 1, \ldots, m$ are the same.

Tackling the induction problem in full generality remains a challenge for future work. Obviously, one could consider the transduction case with zero test points, yielding the induction case. If the weights μ_i are constrained to be nonnegative and furthermore the matrices K_i are guaranteed to be positive semi-definite, the weights can be reused at new test points. To deal with induction in a general SDP setting, one could solve a transduction problem for each new test point. For every

test point, this leads to solving an SDP of dimension $n_{tr}+1$, which is computationally expensive. Clearly there is a need to explore recursive solutions to the SDP problem that allow the solution of the SDP of dimension n_{tr} to be used in the solution of an SDP of dimension $n_{tr}+1$. Such solutions would of course also have immediate applications to on-line learning problems.

5 Error Bounds for Transduction

In the problem of transduction, we have access to the unlabelled test data, as well as the labelled training data, and the aim is to optimize accuracy in predicting the test data. We assume that the data are fixed, and that the order is chosen randomly, yielding a random partition into training and test sets. For convenience, we suppose here that the training and test sets have the same size.

Fix a sequence S of $2n$ pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{2n}, y_{2n})$ from $\mathcal{X} \times \mathcal{Y}$. Let $\pi : \{1, \ldots, 2n\} \to \{1, \ldots, 2n\}$ be a random permutation, chosen uniformly, and let $(X_i, Y_i) = (\mathbf{x}_{\pi(i)}, y_{\pi(i)})$. The first half of this randomly ordered sequence is the training data, and the second half is the test data. For a function $f : \mathcal{X} \to \Re$, we write the proportion of errors on the test data of a thresholded version of f as

$$\mathrm{er}(f) = \frac{1}{n} |\{n+1 \leq i \leq 2n : Y_i f(X_i) \leq 0\}|.$$

The following theorem shows that the error of a kernel classifier on the test data can be bounded in terms of the average of a certain cost function of the training data margins, as well as properties of the kernel matrix. For $\gamma > 0$, define the margin cost function $\phi_\gamma : \Re \to \Re^+$ as

$$\phi_\gamma(a) = \begin{cases} 1 & \text{if } a \leq 0, \\ 1 - a/\gamma & 0 < a \leq \gamma, \\ 0 & a > \gamma. \end{cases}$$

Notice that the 1-norm soft margin cost function is a convex upper bound on this. We consider kernel classifiers obtained by thresholding kernel expansions of the form

$$f(x) = \langle \mathbf{w}, \mathbf{x} \rangle = \sum_{i=1}^{2n} \alpha_i k(\mathbf{x}_i, \mathbf{x}), \tag{57}$$

where $\mathbf{w} = \sum_{i=1}^{2n} \alpha_i \Phi(\mathbf{x}_i)$ is chosen with bounded norm,

$$\|\mathbf{w}\|^2 = \sum_{i,j=1}^{2n} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) = \alpha' K \alpha \leq 1, \tag{58}$$

where K is the $2n \times 2n$ kernel matrix with $K_{ij} = k(X_i, X_j)$. With this constraint, the value of the margin $yf(\mathbf{x})$ is the distance in feature space between $\Phi(\mathbf{x})$ and the decision boundary. Notice that $\mathbf{w}^h$, the $\mathbf{w}$ here, corresponds to the normalized version of $\mathbf{w}^t$, the optimal $\mathbf{w}$ in (3): $\mathbf{w}^h = \mathbf{w}^t / ||\mathbf{w}^t||_2 = \gamma \mathbf{w}^t$. Assuming $b = 0$ for simplicity (though without loss of generality), one can see

$$\xi_i = \begin{cases} 0 & = \phi_\gamma(y_i f(\mathbf{x}_i)) & \text{if } y_i f(\mathbf{x}_i) = y_i \langle \mathbf{w}^h, \Phi(\mathbf{x}_i) \rangle = \gamma y_i \langle \mathbf{w}^t, \Phi(\mathbf{x}_i) \rangle > \gamma, \\ 1 - y_i \langle \mathbf{w}^t, \Phi(\mathbf{x}_i) \rangle & = 1 - y_i \langle \mathbf{w}^h, \Phi(\mathbf{x}_i) \rangle / \gamma & \\ & = \phi_\gamma(y_i f(\mathbf{x}_i)) & 0 < y_i f(\mathbf{x}_i) \leq \gamma, \\ 1 - y_i \langle \mathbf{w}^t, \Phi(\mathbf{x}_i) \rangle & \geq \phi_\gamma(y_i f(\mathbf{x}_i)) & y_i f(\mathbf{x}_i) > \gamma. \end{cases}$$

Hence, γ as defined here is consistent with γ as defined in Section 2.1.

Let $F_{\mathcal{K}}$ denote the class of functions on S of the form (57) satisfying (58), for some $K \in \mathcal{K}$,

$$F_{\mathcal{K}} = \left\{ x_j \mapsto \sum_{i=1}^{2n} \alpha_i K_{ij} : K \in \mathcal{K},\, \alpha' K \alpha \leq 1 \right\},$$

where $\mathcal{K}$ is a set of positive semi-definite $2n \times 2n$ matrices.

We are also interested in the class of kernel expansions obtained from certain linear combinations of a fixed set $\{K_1, \ldots, K_m\}$ of kernel matrices. Consider the class $F_{\mathcal{K}_B}$, with

$$\mathcal{K}_B = \left\{ \sum_{j=1}^{m} \mu_j K_j : K \succeq 0,\, \mu_j \in \Re,\, \mathrm{trace}(K) \leq B \right\},$$

and the class $F_{\mathcal{K}_B^+}$, with

$$\mathcal{K}_B^+ = \left\{ \sum_{j=1}^{m} \mu_j K_j : K \succeq 0,\, \mu_j \geq 0,\, \mathrm{trace}(K) \leq B \right\},$$

Theorem 20 *Let $\phi : \Re \to \Re^+$ satisfy $\phi \geq \phi_\gamma$. With probability at least $1-\delta$ over the data (X_i, Y_i) chosen as above, every function $f \in F_{\mathcal{K}}$ has* $\mathrm{er}(f)$ *no more than*

$$\frac{1}{n} \sum_{i=1}^{n} \phi(Y_i f(X_i)) + \frac{1}{\sqrt{n}} \left(4 + \sqrt{2 \log(1/\delta)} + \sqrt{\frac{C(\mathcal{K})}{n\gamma^2}} \right),$$

where

$$C(\mathcal{K}) = \mathbf{E} \max_{K \in \mathcal{K}} \sigma' K \sigma,$$

with the expectation over σ chosen uniformly from $\{\pm 1\}^{2n}$.

Furthermore,

$$C(\mathcal{K}_B) = B\mathbf{E} \max_{K \in \mathcal{K}} \sigma' \frac{K}{trace(K)} \sigma,$$

and this is always no more than Bn, and

$$C(\mathcal{K}_B^+) \leq B \min \left(m, n \max_j \frac{\lambda_j}{trace(K_j)} \right),$$

where λ_j is the largest eigenvalue of K_j.

Notice that the test error is bounded by a sum of the average over the training data of a margin cost function plus a complexity penalty term that depends on the ratio between the trace of the kernel matrix and the squared margin parameter, γ^2. The kernel matrix here is the full matrix, combining both test and training data.

The bound on the complexity $C(\mathcal{K}_B^+)$ of the kernel class $\mathcal{K}_B^+$ is easier to check than the bound on $C(\mathcal{K}_B)$. The first term in the minimum shows that the set of positive linear combinations of a small set of kernel matrices is not very complex. The second term shows that, even if the set is large, as long as the largest eigenvalue does not dominate the sum of the eigenvalues (the trace), the set of positive linear combinations is not too complex. The proof of the theorem is in Appendix E.

6 Empirical results

We present results for hard margin and soft margin support vector machines. We use a kernel matrix $K = \sum_{i=1}^{3} \mu_i K_i$, where the K_i's are initial "guesses" of the kernel matrix. We use a polynomial kernel function $k_1(\mathbf{x}_1, \mathbf{x}_2) = (1 + \mathbf{x}_1^T \mathbf{x}_2)^d$ for K_1, a Gaussian kernel function $k_2(\mathbf{x}_1, \mathbf{x}_2) = \exp(-0.5(\mathbf{x}_1 - \mathbf{x}_2)^T(\mathbf{x}_1 - \mathbf{x}_2)/\sigma)$ for K_2 and a linear kernel function $k_3(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1^T \mathbf{x}_2$ for K_3. Afterwards, all K_i are normalized. After evaluating the initial kernel matrices $\{K_i\}_{i=1}^3$, the weights $\{\mu_i\}_{i=1}^3$ are optimized according to a hard margin, a 1-norm soft margin and a 2-norm soft margin criterion, respectively; the semi-definite programs (27), (32) and (38) are solved using the general-purpose optimization software SeDuMi (Sturm, 1999), leading to optimal weights $\{\mu_i^*\}_{i=1}^3$. Next, the weights $\{\mu_i\}_{i=1}^3$ are constrained to be non-negative and optimized according to the same criteria: the second order cone programs (29), (34) and (40) are solved using the general-purpose optimization software Mosek (Andersen and Andersen, 2000), leading to optimal weights $\{\mu_{i,+}^*\}_{i=1}^3$. For positive weights, we also report results where the 2-norm soft margin hyperparameter C is automatically learned according to (45).

Empirical results on standard benchmark datasets are summarized in Tables 1, 2 and 3. The Wisconsin breast cancer dataset contained 16 incomplete examples which were not used. The breast cancer, ionosphere and sonar data were obtained from the UCI repository. The heart data were obtained from STATLOG and normalized. Data for the twonorm problem data were generated as specified by Breiman (1998). Each dataset was randomly partitioned into 80% training and 20% test sets. The reported results are the averages over 30 random partitions. The kernel parameters for K_1 and K_2 are given in Tables 1, 2 and 3 by d and σ respectively. For each of the kernel matrices, an SVM is trained using the training block K_{tr} and tested using the mixed block $K_{tr,t}$ as defined in (19). The margin γ (for a hard margin criterion) respectively optimal soft margin cost functions w_{S1}^* and w_{S2}^* (for soft margin criteria) are reported for the initial kernel matrices K_i, as well as for the optimal $\sum_i \mu_i^* K_i$ and $\sum_i \mu_{i,+}^* K_i$. Furthermore, the average test set accuracy (TSA), the average value for C and the average weights over the 30 partitions are listed. For comparison, the performance of the best soft margin SVM with a Gaussian kernel is reported—the soft margin hyperparameter C and the kernel parameter σ for the Gaussian kernel were tuned using cross-validation over 30 random partitions of the training set.

Note that not every K_i gives rise to a linearly separable embedding of the training data, in which case no hard margin classifier can be found (indicated with a dash). The matrices $\sum_i \mu_i^* K_i$ and $\sum_i \mu_{i,+}^* K_i$ however, always allow the training of a hard margin SVM and its margin is indeed larger than the margin for each of the different components K_i - this is consistent with the SDP/QCQP optimization. For the soft margin criteria, the optimal value of the cost function for $\sum_i \mu_i^* K_i$ and $\sum_i \mu_{i,+}^* K_i$ is smaller than its value for the individual K_i - again consistent with the SDP/QCQP optimizations. Notice that constraining the weights μ_i to be positive results in slightly smaller margins and larger cost functions, as expected.

Furthermore, the number of test set errors is usually smaller for $\sum_i \mu_i^* K_i$ and $\sum_i \mu_{i,+}^* K_i$ than for each of the different components K_i. This supports the use of the error bound (21) and the criteria proposed in Section 4. Also notice that $\sum_i \mu_{i,+}^* K_i$ does often almost as well as $\sum_i \mu_i^* K_i$, and sometimes even better: we can thus improve the computational complexity substantially without a significant loss of performance. The performance of $\sum_i \mu_i^* K_i$ and $\sum_i \mu_{i,+}^* K_i$ is comparable with the best soft margin SVM with RBF kernel. However, the RBF SVM needs additional tuning of the kernel parameter using cross-validation, while the kernel learning approach doesn't. Moreover, when using the 2-norm soft margin SVM with auto-tuned hyperparameter C, we no longer need

to do cross-validation for C. This leads to an even smaller optimal cost function w^*_{S2} (compared to the case SM2, with $C = 1$) and performs well on the test set, while it offers the advantage of automatically adjusting C.

One might wonder why there is a difference between the SDP and the QCQP approach for the twonorm data, since both seem to find positive weights μ_i. However, one shouldn't forget that the values in Table 3 are averages over 30 randomizations—for some randomizations the SDP has actually found negative weights, although the averages are positive.

As a further example illustrating the flexibility of the SDP framework, consider the following setup. Let $\{K_i\}_{i=1}^5$ be Gaussian kernels with $\sigma = 0.01, 0.1, 1, 10, 100$ respectively. Combining those optimally with $\mu_i \geq 0$ for a 2-norm soft margin SVM, with auto-tuning of C, yields the results in Table 4—averages over 30 randomizations in 80% training and 20% test sets. The test set accuracies obtained for $\sum_i \mu^*_{i,+} K_i$ are competitive with those for the best soft margin SVM with an RBF kernel, tuned using cross-validation. The average weights show that some kernels are selected and others are not. Effectively we obtain a data-based choice of smoothing parameter without recourse to cross-validation.

In Cristianini et al. (2001) empirical results are given for optimization of the alignment using a kernel matrix $K = \sum_{i=1}^N \mu_i v_i v_i^T$. The results show that optimizing the alignment indeed improves the generalization power of Parzen window classifiers. As explained in Section 4.6, it turns out that in this particular case, the SDP in (51) reduces to exactly the quadratic program that is obtained in Cristianini et al. (2001) and thus those results also provide support for the general framework presented in the current paper.

7 Discussion

In this paper we have presented a new method for learning a kernel matrix from data. Our approach makes use of semi-definite programming (SDP) ideas. It is motivated by the fact that every symmetric, positive definite matrix can be viewed as a kernel matrix (corresponding to a certain embedding of a finite set of data), and the fact that SDP deals with the optimization of convex cost functions over the convex cone of positive semi-definite matrices (or convex subsets of this cone). Thus convex optimization and machine learning concerns merge to provide a powerful methodology for learning the kernel matrix with SDP.

We have focused on the transductive setting, where the labelled data are used to learn an embedding, which is then applied to the unlabelled part of the data. Based on a new generalization bound for transduction, we have shown how to impose convex constraints that effectively control the capacity of the search space of possible kernels and yield an efficient learning procedure that can be implemented by SDP. Furthermore, this approach leads directly to a convex method to learn the 2-norm soft margin parameter in support vector machines, solving another important open problem. Promising empirical results on standard benchmark datasets are reported; these results underline the fact that the new approach provides a principled way to combine multiple kernels to yield a classifier that may perform better than any individual kernel.

There are several challenges that need to be met in future research on SDP-based learning algorithms. First, it is clearly of interest to explore other convex quality measures for a kernel matrix, which may be appropriate for other learning algorithms. For example, in the setting of Gaussian processes, the relative entropy between the zero-mean Gaussian process prior P with covariance kernel K and the corresponding Gaussian process approximation Q to the true intractable posterior

process depends on K as

$$D[P||Q] = \frac{1}{2}\log\det K + \frac{1}{2}\text{trace}\left(y^T K y\right) + d,$$

where the constant d is independent of K. One can verify that $D[P||Q]$ is convex with respect to $R = K^{-1}$ (see e.g., Vandenberghe et al., 1998). Minimizing this measure with respect to R, and thus K, is motivated from PAC-Bayesian generalization error bounds for Gaussian processes (see e.g., Seeger, 2002) and can be achieved by solving a so-called *maximum-determinant problem* (Vandenberghe et al., 1998)—an even more general framework that contains semi-definite programming as a special case.

Secondly, the investigation of other parameterizations of the kernel matrix is an important topic for further study. While the linear combination of kernels that we have studied here is likely to be useful in many practical problems—capturing a notion of combining Gram matrix "experts"—it is also worth considering other parameterizations as well. Any such parameterizations have to respect the constraint that the quality measure for the kernel matrix is convex with respect to the parameters of the proposed parameterization. One class of examples arises via the positive definite matrix completion problem (Vandenberghe et al., 1998). Here we are given a symmetric kernel matrix K that has some entries which are fixed. The remaining entries—the parameters in this case—are to be chosen such that the resulting matrix is positive definite, while simultaneously a certain cost function is optimized, e.g., $\text{trace}(SK) + \log\det K^{-1}$ where S is a given matrix. This specific case reduces to solving a maximum-determinant problem which is convex in the unknown entries of K, the parameters of the proposed parametrization.

A third important area for future research consists in finding faster implementations of semi-definite programming. As in the case of quadratic programming (Platt, 1999), it seems likely that special purpose methods can be developed to exploit the exchangeable nature of the learning problem in classification and result in more efficient algorithms.

A Proof of result (28)

For the case $K_i = v_i v_i^T$, with v_i orthonormal, the original learning problem (26) becomes

$$\begin{array}{rl} \min\limits_{K} & w(K_{tr}) \\ \text{subject to} & \text{trace}(K) = c, \\ & K \succeq 0, \\ & K = \sum\limits_{i=1}^{m} \mu_i v_i v_i^T, \end{array}$$

or, according to (2),

$$\begin{array}{rl} \min\limits_{K} \max\limits_{\alpha \,:\, \alpha \geq 0, \alpha^T y = 0} & 2\alpha^T e - \alpha^T G(K_{tr}) \alpha \\ \text{subject to} & \text{trace}(K) = c, \\ & K \succeq 0, \\ & K = \sum\limits_{i=1}^{m} \mu_i v_i v_i^T. \end{array} \tag{59}$$

Recall that

$$G(K_{tr}) = \text{diag}(y) K_{tr} \text{diag}(y),$$

with

$$\begin{aligned} K_{tr} &= K(1:n_{tr}, 1:n_{tr}) \\ &= \sum_{i=1}^{m} \mu_i K_i(1:n_{tr}, 1:n_{tr}) \\ &= \sum_{i=1}^{m} \mu_i v_i(1:n_{tr}) v_i(1:n_{tr})^T \\ &= \sum_{i=1}^{m} \mu_i \bar{v}_i \bar{v}_i^T, \end{aligned}$$

where $\bar{v}_i = v_i(1:n_{tr})$. Furthermore, because the v_i are orthonormal, the μ_i in $K = \sum_{i=1}^{m} \mu_i v_i v_i^T$ are the eigenvalues of K. This implies

$$\text{trace}(K) = \sum_{i=1}^{m} \mu_i = e^T \mu$$

and

$$K \succeq 0 \Leftrightarrow \mu \geq 0 \Leftrightarrow \mu_i \geq 0, \quad i = 1, \ldots, m.$$

We can thus write (59) as:

$$\begin{aligned}
&\min_{\mu\,:\,\mu\geq 0, e^T\mu=c}\ \max_{\alpha\,:\,\alpha\geq 0,\alpha^T y=0} 2\alpha^T e-\alpha^T \mathrm{diag}(y)(\sum_{i=1}^{m}\mu_i\bar{v}_i\bar{v}_i^T)\mathrm{diag}(y)\alpha\\
&=\min_{\mu\,:\,\mu\geq 0, e^T\mu=c}\ \max_{\alpha\,:\,\alpha\geq 0,\alpha^T y=0} 2\alpha^T e-\sum_{i=1}^{m}\mu_i\alpha^T \mathrm{diag}(y)\bar{v}_i\bar{v}_i^T\mathrm{diag}(y)\alpha\\
&=\min_{\mu\,:\,\mu\geq 0, e^T\mu=c}\ \max_{\alpha\,:\,\alpha\geq 0,\alpha^T y=0} 2\alpha^T e-\sum_{i=1}^{m}\mu_i\alpha^T \breve{v}_i\breve{v}_i^T\alpha\\
&=\min_{\mu\,:\,\mu\geq 0, e^T\mu=c}\ \max_{\alpha\,:\,\alpha\geq 0,\alpha^T y=0} 2\alpha^T e-\sum_{i=1}^{m}\mu_i(\breve{v}_i^T\alpha)^2\\
&=\max_{\alpha\,:\,\alpha\geq 0,\alpha^T y=0}\ \min_{\mu\,:\,\mu\geq 0, e^T\mu=c} 2\alpha^T e-\sum_{i=1}^{m}\mu_i(\breve{v}_i^T\alpha)^2
\end{aligned}$$

with $\breve{v}_i = \mathrm{diag}(y)\ \bar{v}_i$.

We interchanged the order of the minimization and the maximization. Standard results in convex optimization (see e.g., Boyd and Vandenberghe, 2001) imply that we are allowed to do this and still obtain the same optimal value, because the objective is convex in μ (it is linear in μ) and concave in α, because the minimization problem is strictly feasible in μ, and the maximization problem as well in α (we can skip the case for all elements of y having the same sign, because we cannot even consider a margin in such a case). We further obtain:

$$\begin{aligned}
&\max_{\alpha\,:\,\alpha\geq 0,\alpha^T y=0}\ \min_{\mu\,:\,\mu\geq 0, e^T\mu=c} 2\alpha^T e-\sum_{i=1}^{m}\mu_i(\breve{v}_i^T\alpha)^2\\
&\quad=\max_{\alpha\,:\,\alpha\geq 0,\alpha^T y=0}\left[2\alpha^T e-\max_{\mu\,:\,\mu\geq 0, e^T\mu=c}\left(\sum_{i=1}^{m}\mu_i(\breve{v}_i^T\alpha)^2\right)\right]\\
&\quad=\max_{\alpha\,:\,\alpha\geq 0,\alpha^T y=0}\left[2\alpha^T e-\max_i\left(c(\breve{v}_i^T\alpha)^2\right)\right].
\end{aligned}$$

This can be reformulated as follows:

$$\begin{aligned}
\max_{\alpha,t}\quad & 2\alpha^T e-ct\\
\text{subject to}\quad & t\geq(\breve{v}_i^T\alpha)^2, i=1,\ldots,m\\
& \alpha^T y=0,\\
& \alpha\geq 0,
\end{aligned}\tag{60}$$

which gives the result (28).

B Proof of Theorem 17

After substitution of $w_{S1}(K_{tr})$ as defined in (4), (30) becomes:

$$\min_{K\in\mathcal{K}}\ \max_{\alpha}\ 2\alpha^T e-\alpha^T G(K_{tr})\alpha\ :\ C\geq\alpha\geq 0,\ \ \alpha^T y=0,\ \ \mathrm{trace}(K)=c,\tag{61}$$

with c a constant.

Assume that $K_{tr} \succ 0$, hence $G(K_{tr}) \succ 0$ (the following can be extended to the general case). We note that $w_{S1}(K_{tr})$ is convex in K (it is the pointwise maximum of affine functions of K). Given the convex constraints in (61), the optimization problem is thus certainly convex in K. We write this as:

$$\begin{aligned} \min_{K \in \mathcal{K}, t} \; t \; : \qquad & t \geq \max_{\alpha} \; 2\alpha^T e - \alpha^T G(K_{tr}) \alpha, \\ & C \geq \alpha \geq 0, \;\; \alpha^T y = 0, \;\; \operatorname{trace}(K) = c. \end{aligned} \tag{62}$$

We will now express $t \geq \max_\alpha \; 2\alpha^T e - \alpha^T G(K_{tr})\alpha$ as an LMI. This can be done in exactly the same way as we did before for the hard margin: we express the constraint using the dual minimization problem. This will allow us to drop the minimization and use the Schur complement lemma to obtain an LMI.

Define the Lagrangian of the maximization problem (4) by

$$\mathcal{L}(\alpha, \nu, \lambda, \delta) = 2\alpha^T e - \alpha^T G(K_{tr}) \alpha + 2\nu^T \alpha + 2\lambda y^T \alpha + 2\delta^T (Ce - \alpha),$$

where $\lambda \in \mathbb{R}$ and $\nu, \delta \in \mathbb{R}^n$. By duality, we have

$$w_{S1}(K_{tr}) = \max_{\alpha} \; \min_{\nu \geq 0, \delta \geq 0, \lambda} \; \mathcal{L}(\alpha, \nu, \lambda, \delta) = \min_{\nu \geq 0, \delta \geq 0, \lambda} \; \max_{\alpha} \; \mathcal{L}(\alpha, \nu, \lambda, \delta),$$

where $\nu \geq 0 \Leftrightarrow \nu_i \geq 0$ for $i = 1, \ldots, n$, similarly for δ. Since $G(K_{tr}) \succ 0$, at the optimum, we have

$$\alpha = G(K_{tr})^{-1}(e + \nu - \delta + \lambda y),$$

and can form the dual problem

$$w_{S1}(K_{tr}) = \min_{\nu, \, \delta, \, \lambda} \; (e + \nu - \delta + \lambda y)^T G(K_{tr})^{-1} (e + \nu - \delta + \lambda y) + 2C\delta^T e \;\; : \;\; \nu \geq 0, \;\; \delta \geq 0.$$

We obtain that for any $t > 0$, the constraint $w_{S1}(K_{tr}) \leq t$ is true if and only if there exist $\nu \geq 0$, $\delta > 0$ and λ such that

$$(e + \nu - \delta + \lambda y)^T G(K_{tr})^{-1} (e + \nu - \delta + \lambda y) + 2C\delta^T e \leq t,$$

or, equivalently (using the Schur complement lemma), such that

$$\begin{pmatrix} G(K_{tr}) & e + \nu - \delta + \lambda y \\ (e + \nu - \delta + \lambda y)^T & t - 2C\delta^T e \end{pmatrix} \succeq 0$$

holds. Taking this into account, (62) can be expressed as:

$$\begin{aligned} \min_{K, t, \lambda, \nu, \delta} \quad & t \\ \text{subject to} \quad & \operatorname{trace}(K) = c, \\ & K \in \mathcal{K}, \\ & \begin{pmatrix} G(K_{tr}) & e + \nu - \delta + \lambda y \\ (e + \nu - \delta + \lambda y)^T & t - 2C\delta^T e \end{pmatrix} \succeq 0, \\ & \nu \geq 0 \\ & \delta \geq 0, \end{aligned} \tag{63}$$

which yields (31). Notice that $\nu \geq 0 \Leftrightarrow \operatorname{diag}(\nu) \succeq 0$, and thus an LMI; similarly for $\delta \geq 0$. $\square$

C Proof of result (39)

For the case $K_i = v_i v_i^T$, with v_i orthonormal, the learning problem becomes

$$\begin{aligned} \min_{K} \quad & w_{S2}(K_{tr}) \\ \text{subject to} \quad & \text{trace}(K) = c, \\ & K \succeq 0, \\ & K = \sum_{i=1}^{m} \mu_i v_i v_i^T, \end{aligned}$$

or, according to (6),

$$\begin{aligned} \min_{K} \max_{\alpha \,:\, \alpha \geq 0, \alpha^T y = 0} \quad & 2\alpha^T e - \alpha^T \left(G(K_{tr}) + \frac{1}{C} I_{n_{tr}} \right) \alpha \\ \text{subject to} \quad & \text{trace}(K) = c, \\ & K \succeq 0, \\ & K = \sum_{i=1}^{m} \mu_i v_i v_i^T, \end{aligned} \tag{64}$$

which is equivalent to

$$\begin{aligned} \min_{K} \max_{\alpha \,:\, \alpha \geq 0, \alpha^T y = 0} \quad & 2\alpha^T e - \frac{1}{C} \alpha^T \alpha - \alpha^T G(K_{tr}) \alpha \\ \text{subject to} \quad & \text{trace}(K) = c, \\ & K \succeq 0, \\ & K = \sum_{i=1}^{m} \mu_i v_i v_i^T. \end{aligned} \tag{65}$$

Comparing this to (59), we see that both formulations are identical except for the terms in the objective function that only depend on α (and not on the kernel matrix, i.e., not on the weights μ_i for $K = \sum_{i=1}^{m} \mu_i v_i v_i^T$): these are $2\alpha^T e - \frac{1}{C}\alpha^T \alpha$ in (65) instead of only $2\alpha^T e$ in (59). Those terms are conserved through the entire further derivation of the hard margin case, as can easily be checked in Appendix A. For this reason, the result for the 2-norm soft margin case can be obtained by replacing $2\alpha^T e$ with $2\alpha^T e - \frac{1}{C}\alpha^T \alpha$ in the result (60) for the hard margin case. This yields

$$\begin{aligned} \max_{\alpha, t} \quad & 2\alpha^T e - \frac{1}{C} \alpha^T \alpha - ct \\ \text{subject to} \quad & t \geq (\breve{v}_i^T \alpha)^2, i = 1, \ldots, m \\ & \alpha^T y = 0, \\ & \alpha \geq 0, \end{aligned}$$

with $\breve{v}_i = \text{diag}(y)\ \bar{v}_i = \text{diag}(y)\ v_i(1 : n_{tr})$. This is exactly (39).

D Proof of result (52)

For the case $K_i = v_i v_i^T$, with v_i orthonormal, the original learning problem (50) becomes

$$\begin{aligned} \max_K \quad & \left\langle K_{tr}, yy^T \right\rangle_F \\ \text{subject to} \quad & \langle K, K \rangle_F \leq 1, \\ & K \succeq 0, \\ & K = \sum_{i=1}^{m} \mu_i v_i v_i^T. \end{aligned} \tag{66}$$

Working this further out gives:

$$\begin{aligned} \left\langle K_{tr}, yy^T \right\rangle_F &= \text{trace}(K(1:n_{tr}, 1:n_{tr}) yy^T) \\ &= \text{trace}((\sum_{i=1}^{m} \mu_i v_i(1:n_{tr}) v_i(1:n_{tr})^T) yy^T) \\ &= \sum_{i=1}^{m} \mu_i \text{trace}(\bar{v}_i \bar{v}_i^T yy^T) \\ &= \sum_{i=1}^{m} \mu_i (\bar{v}_i^T y)^2, \end{aligned} \tag{67}$$

$$\begin{aligned} \langle K, K \rangle_F &= \text{trace}(K^T K) \\ &= \text{trace}(KK) \\ &= \text{trace}((\sum_{i=1}^{m} \mu_i v_i v_i^T)(\sum_{j=1}^{m} \mu_j v_j v_j^T)) \\ &= \text{trace}(\sum_{i,j=1}^{m} \mu_i \mu_j v_i v_i^T v_j v_j^T) \\ &= \text{trace}(\sum_{i=1}^{m} \mu_i^2 v_i v_i^T) \\ &= \sum_{i=1}^{m} \mu_i^2 \text{trace}(v_i v_i^T) \\ &= \sum_{i=1}^{m} \mu_i^2 \text{trace}(v_i^T v_i) \\ &= \sum_{i=1}^{m} \mu_i^2 \end{aligned} \tag{68}$$

with $\bar{v}_i = v_i(1:n_{tr})$. We used the fact that $\text{trace}(ABC) = \text{trace}(BCA)$ (if the products are well-defined) and that the vectors $v_i, i = 1, \ldots, n$ are orthonormal: $v_i^T v_j = \delta_{ij}$. Furthermore, because the v_i are orthogonal, the μ_i in $K = \sum_{i=1}^{m} \mu_i v_i v_i^T$ are the eigenvalues of K. This implies

$$K \succeq 0 \Leftrightarrow \mu \geq 0 \Leftrightarrow \mu_i \geq 0, \quad i = 1, \ldots, m. \tag{69}$$

Using (67), (68) and (69) in (66), we obtain the following optimization problem:

$$\begin{aligned} \max_{\mu_i} \quad & \sum_{i=1}^{m} \mu_i (\bar{v}_i^T y)^2 \\ \text{subject to} \quad & \sum_{i=1}^{m} \mu_i^2 \leq 1 \\ & \mu_i \geq 0, \quad i = 1, \ldots, m, \end{aligned}$$

which yields the result (52).

E Proof of Theorem 20

For a function $g : \mathcal{X} \times \mathcal{Y} \to \Re$, define

$$\begin{aligned} \hat{\mathbf{E}}_1 g(X, Y) &= \frac{1}{n} \sum_{i=1}^{n} g(X_i, Y_i), \\ \hat{\mathbf{E}}_2 g(X, Y) &= \frac{1}{n} \sum_{i=1}^{n} g(X_{n+i}, Y_{n+i}). \end{aligned}$$

The proof of the first part involves the following five steps:

Step 1. For any class F of real functions defined on $\mathcal{X}$,

$$\sup_{f \in F} \mathrm{er}(f) - \hat{\mathbf{E}}_1 \phi_\gamma(Yf(X)) \leq \sup_{f \in F} \hat{\mathbf{E}}_2 \phi_\gamma(Yf(X)) - \hat{\mathbf{E}}_1 \phi_\gamma(Yf(X)).$$

To see this, notice that $\mathrm{er}(f)$ is the average over the test set of the indicator function of $Yf(X) \leq 0$, and that $\phi_\gamma(Yf(X))$ bounds this function.

Step 2. For any class G of $[0, 1]$-valued functions,

$$\Pr\left(\sup_{g \in G} \hat{\mathbf{E}}_2 g - \hat{\mathbf{E}}_1 g \geq \mathbf{E}\left(\sup_{g \in G} \hat{\mathbf{E}}_2 g - \hat{\mathbf{E}}_1 g \right) + \epsilon \right) \leq \exp\left(\frac{-\epsilon^2 n}{4} \right),$$

where the expectation is over the random permutation. This follows from McDiarmid's inequality. To see this, we need to define the random permutation π using a set of $2n$ independent random variables. To this end, choose $\pi_1, \ldots, \pi_{2n}$ uniformly at random from the interval $[0, 1]$. These are almost surely distinct. For $j = 1, \ldots, 2n$, define $\pi(j) = |\{i : \pi_i \leq \pi_j\}|$, that is, $\pi(j)$ is the position of π_j when the random variables are ordered by size. It is easy to see that, for any g, $\hat{\mathbf{E}}_2 g - \hat{\mathbf{E}}_1 g$ changes by no more than $2/n$ when one of the π_i changes. McDiarmid's inequality implies the result.

Step 3. For any class G of $[0, 1]$-valued functions,

$$\mathbf{E}\left(\sup_{g \in G} \hat{\mathbf{E}}_2 g - \hat{\mathbf{E}}_1 g \right) \leq \hat{R}_{2n}(G) + \frac{4}{\sqrt{n}},$$

where $\hat{R}_{2n}(G) = \mathbf{E} \sup_{g \in G} \frac{1}{n} \sum_{i=1}^{2n} \sigma_i g(X_i)$, and the expectation is over the independent, uniform, $\{\pm 1\}$-valued random variables $\sigma_1, \ldots, \sigma_{2n}$. This result is essentially Lemma 3 of Bartlett and

Mendelson (2001); that lemma contained a similar bound for i.i.d. X_i, but the same argument holds for fixed X_i, randomly permuted.

Step 4. If the class F of real-valued functions defined on $\mathcal{X}$ is closed under negations, $\hat{R}_{2n}(\phi_\gamma \circ F) \leq \frac{1}{\gamma}\hat{R}_{2n}(F)$, where each $f \in F$ defines a $g \in \phi_\gamma \circ F$ by $g(x,y) = \phi_\gamma(yf(x))$. This bound is the contraction lemma in Ledoux and Talagrand (1991).

Step 5. For the class $F_\mathcal{K}$ of kernel expansions, notice (as in the proof of Lemma 26 of Bartlett and Mendelson (2001)) that

$$\begin{aligned}
\hat{R}_{2n}(F_\mathcal{K}) &= \frac{1}{n}\mathbf{E}\max_{f\in F_\mathcal{K}}\sum_{i=1}^{2n}\sigma_i f(x_i) \\
&= \frac{1}{n}\mathbf{E}\max_{K\in\mathcal{K}}\max_{\|w\|\leq 1}\langle w, \sum_{i=1}^{2n}\sigma_i\Phi(X_i)\rangle \\
&= \frac{1}{n}\mathbf{E}\max_{K\in\mathcal{K}}\left\|\sum_{i=1}^{2n}\sigma_i\Phi(X_i)\right\| \\
&\leq \frac{1}{n}\sqrt{\mathbf{E}\max_{K\in\mathcal{K}}\sigma' K\sigma} \\
&= \frac{1}{n}\sqrt{C(\mathcal{K})},
\end{aligned}$$

where $\sigma = (\sigma_1, \ldots, \sigma_{2n})$ is the vector of Rademacher random variables.

Combining gives the first part of the theorem. For the second part, consider

$$C(\mathcal{K}_B) = \mathbf{E}\max_{K\in\mathcal{K}_B}\sigma' K\sigma = \mathbf{E}\max_{\mu}\sum_{j=1}^{m}\mu_j\sigma' K_j\sigma,$$

where the max is over $\mu = (\mu_1, \ldots, \mu_m)$ for which the matrix $K = \sum_{j=1}^m \mu_j K_j$ satisfies the conditions $K \succeq 0$ and $\mathrm{trace}(K) \leq B$. Now,

$$\mathrm{trace}(K) = \sum_{j=1}^{m}\mu_j \mathrm{trace}(K_j),$$

and each trace in the sum is positive, so the supremum must be achieved for $\mathrm{trace}(K) = B$. So we can write

$$C(\mathcal{K}_B) = B\mathbf{E}\max_{K\in\mathcal{K}_B}\sum_{j=1}^{m}\sigma'\frac{K}{\mathrm{trace}(K)}\sigma.$$

Notice that $\sigma' K\sigma$ is no more than $\lambda\|\sigma\|^2 = n\lambda$, where λ is the maximum eigenvalue of K. Using $\lambda \leq \mathrm{trace}(K) = B$ shows that $C(\mathcal{K}_B) \leq Bn$.

Finally, for $\mathcal{K}_B^+$ we have

$$\begin{aligned}
C(\mathcal{K}_B^+) &= \mathbf{E}\max_{\mathcal{K}\in\mathcal{K}_B^+}\sigma' K\sigma \\
&= \mathbf{E}\max_{\mu_j}\sum_{j=1}^{m}\mu_j\sigma' K_j\sigma \\
&= \mathbf{E}\max_{j}\frac{B}{\mathrm{trace}(K_j)}\sigma' K_j\sigma.
\end{aligned}$$

Since each term in the maximum is non-negative, we can replace it with a sum to show that

$$\begin{aligned} C(\mathcal{K}_B^+) &\leq B\mathbf{E}\sigma' \left(\sum_j \frac{K_j}{\text{trace}(K_j)} \right) \sigma \\ &= Bm. \end{aligned}$$

Alternatively, we can write $\sigma' K_j \sigma \leq \lambda_j \|\sigma\| = \lambda_j n$, where λ_j is the maximum eigenvalue of K_j. This shows that

$$C(\mathcal{K}_B^+) \leq Bn \max_j \frac{\lambda_j}{\text{trace}(K_j)}.$$

Acknowledgements

We acknowledge support from ONR MURI N00014-00-1-0637 and NSF grant IIS-9988642. Sincere thanks to Tijl De Bie for helpful conversations and suggestions.

References

Andersen, E. D. and Andersen, A. D. (2000). The mosek interior point optimizer for linear programming: An implementation of the homogeneous algorithm. In Frenk, H., Roos, C., Terlaky, T., and Zhang, S., editors, *High Performance Optimization*, pages 197–232. Kluwer Academic Publishers.

Bartlett, P. L. and Mendelson, S. (2001). Rademacher and gaussian complexities: Risk bounds and structural results. Technical report, Australian National University.

Bennett, K. P. and Bredensteiner, E. J. (2000). Duality and geometry in SVM classifiers. In *Proc. 17th International Conf. on Machine Learning*, pages 57–64. Morgan Kaufmann, San Francisco, CA.

Boyd, S. and Vandenberghe, L. (2001). Convex optimization. Course notes for EE364, Stanford University. Available at `http://www.stanford.edu/class/ee364`.

Breiman, L. (1998). Arcing classifiers. *Annals of Statistics*, 26(3):801–849.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, U.K.

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., and Kandola, J. (2002). On kernel-target alignment. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA. MIT Press.

Cristianini, N., Shawe-Taylor, J., Kandola, J., and Elisseeff, A. (2001). On kernel target alignment. In *Advances in Neural Information Processing Systems 14*. MIT Press, Cambridge, MA.

De Bie, T., Lanckriet, G., and Cristianini, N. (2002). Convex optimization of the 2-norm soft margin parameter in support vector machines. Technical Report in progress, University of California, Berkeley, Dept. of EECS.

Ledoux, M. and Talagrand, M. (1991). *Probability in Banach Spaces: Isoperimetry and Processes*. Springer-Verlag, New York, NY.

Nesterov, Y. and Nemirovsky, A. (1994). *Interior point polynomial methods in convex programming: Theory and applications*. SIAM, Philadelphia, PA.

Platt, J. (1999). Using sparseness and analytic QP to speed training of support vector machines. In M. S. Kearns, S. A. Solla, D. A. C., editor, *Advances in Neural Information Processing Systems 11*, Cambridge, MA. MIT Press.

Schölkopf, B. and Smola, A. (2002). *Learning with Kernels*. MIT Press, Cambridge, MA.

Seeger, M. (2002). Pac-bayesian generalization error bounds for gaussian process classification. Technical Report EDI-INF-RR-0094, University of Edinburgh, Division of Informatics.

Shawe-Taylor, J. and Cristianini, N. (1999). Soft margin and margin distribution. In Smola, A., Schölkopf, B., Bartlett, P., and Schuurmans, D., editors, *Advances in Large Margin Classifiers*. MIT Press.

Sturm, J. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653. Special issue on Interior Point Methods (CD supplement with software).

Vandenberghe, L. and Boyd, S. (1996). Semidefinite programming. *SIAM Review*, 38(1):49–95.

Vandenberghe, L., Boyd, S., and Wu, S.-P. (1998). Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19(2):499–533.

		K_1	K_2	K_3	$\sum_i \mu_i^* K_i$	$\sum_i \mu_{i,+}^* K_i$	RBF
Breast cancer		$d=2$	$\sigma=0.5$				
HM	γ	0.0036	0.1055	-	0.1369 %	0.1219	
	TSA	92.9 %	89.0 %	-	95.5 %	94.4 %	96.1 %
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	1.90/2.35/-1.25	0.65/2.35/0	
SM1	w_{S1}^*	77.012	44.913	170.26	26.694	33.689	
	TSA	96.4 %	89.0 %	87.7 %	95.5 %	94.4 %	96.7 %
	C	1	1	1	1	1	
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	1.90/2.35/-1.25	0.65/2.35/0	
SM2	w_{S2}^*	43.138	35.245	102.51	20.696	21.811	
	TSA	96.4 %	88.5 %	87.4 %	95.4 %	94.3 %	96.8 %
	C	1	1	1	1	1	
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	2.32/2.13/-1.46	0.89/2.11/0	
SM2,C	w_{S2}^*	27.682	33.685	41.023		25.267	
	TSA	94.5 %	89.0 %	87.3 %		94.4 %	96.8 %
	C	0.3504	1.48e+8	0.3051		6.77e+7	
	$\mu_1/\mu_2/\mu_3$	1.15/0/0	0/3.99/0	0/0/0.72		0.87/3.13/0	
Ionosphere		$d=2$	$\sigma=0.5$				
HM	γ	0.0613	0.1452	-	0.1623	0.1616	
	TSA	91.2 %	92.0 %	-	94.4 %	94.4 %	93.9 %
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	1.08/2.18/-0.26	0.79/2.21/0	
SM1	w_{S1}^*	30.786	23.233	52.312	18.117	18.303	
	TSA	94.5 %	92.1 %	83.1 %	94.8 %	94.5 %	94.0 %
	C	1	1	1	1	1	
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	1.23/2.07/-0.30	0.90/2.10/0	
SM2	w_{S2}^*	18.533	17.907	31.662	13.382	13.542	
	TSA	94.7 %	92.0 %	91.6 %	94.5 %	94.4 %	94.2 %
	C	1	1	1	1	1	
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	1.68/1.73/-0.41	1.23/1.78/0	
SM2,C	w_{S2}^*	14.558	17.623	18.975		13.5015	
	TSA	93.5 %	92.1 %	90.0 %		94.6 %	94.2 %
	C	0.4144	5.8285	0.3442		0.8839	
	$\mu_1/\mu_2/\mu_3$	1.59/0/0	0/3.83/0	0/0/1.09		1.24/1.61/0	

Table 1: SVMs trained and tested with the initial kernel matrices K_1, K_2, K_3 and with the optimal kernel matrices $\sum_i \mu_i^* K_i$ and $\sum_i \mu_{i,+}^* K_i$. For hard margin SVMs (HM), the resulting margin γ is given—a dash meaning that no hard margin classifier could be found; for soft margin SVMs (SM1 = 1-norm soft margin with $C = 1$, SM2 = 2-norm soft margin with $C = 1$ and SM2,C = 2-norm soft margin with auto tuning of C) the optimal value of the cost function w_{S1}^* or w_{S2}^* is given. Furthermore, the test-set accuracy (TSA), the average weights and the average C-values are given. For c we used $c = \sum_i \operatorname{trace}(K_i)$ for HM, SM1 and SM2. The initial kernel matrices are evaluated after being multiplied by 3. This assures we can compare the different γ for HM, w_{S1}^* for SM1 and w_{S2}^* for SM2, since the resulting kernel matrix has a constant trace (i.e., everything is on the same scale). For SM2,C we use $c = \sum_i \operatorname{trace}(K_i) + \operatorname{trace}(I_n)$. This not only allows comparing the different w_{S2}^* for SM2,C but also it allows comparing w_{S2}^* between SM2 and SM2,C (since we choose $C = 1$ for SM2, we have that trace $\left(\sum_{i=1}^m \mu_i K_i + \frac{1}{C} I_n\right)$ is constant in both cases, so again, we are on the same scale). Finally, the column RBF reports the performance of the best soft margin SVM with RBF kernel, tuned using cross-validation.

		K_1	K_2	K_3	$\sum_i \mu_i^* K_i$	$\sum_i \mu_{i,+}^* K_i$	RBF
Heart		$d = 2$	$\sigma = 0.5$				
HM	γ	0.0369	0.1221	-	0.1531	0.1528	
	TSA	72.9 %	59.5 %	-	84.8 %	84.6 %	77.7 %
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	-0.09/2.68/0.41	0.01/2.60/0.39	
SM1	w_{S1}^*	58.169	33.536	74.302	21.361	21.446	
	TSA	79.3 %	59.5 %	84.3 %	84.8 %	84.6 %	83.9 %
	C	1	1	1	1	1	
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	-0.09/2.68/0.41	0.01/2.60/0.39	
SM2	w_{S2}^*	32.726	25.386	45.891	15.988	16.034	
	TSA	78.1 %	59.0 %	84.3 %	84.8 %	84.6 %	83.2 %
	C	1	1	1	1	1	
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	-0.08/2.54/0.54	0.01/2.47/0.53	
SM2,C	w_{S2}^*	19.643	25.153	16.004		15.985	
	TSA	81.3 %	59.6 %	84.7 %		84.6 %	83.2 %
	C	0.3378	1.18e+7	0.2880		0.4365	
	$\mu_1/\mu_2/\mu_3$	1.04/0/0	0/3.99/0	0/0/0.53		0.01/0.80/0.53	
Sonar		$d = 2$	$\sigma = 0.1$				
HM	γ	0.0246	0.1460	0.0021	0.1517	0.1459	
	TSA	80.9 %	85.8 %	74.2 %	84.6 %	85.8 %	84.2 %
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	-2.23/3.52/1.71	0/3/0	
SM1	w_{S1}^*	87.657	23.288	102.68	21.637	23.289	
	TSA	78.1 %	85.6 %	73.3 %	84.6 %	85.6 %	84.2 %
	C	1	1	1	1	1	
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	-2.20/3.52/1.69	0/3/0	
SM2	w_{S2}^*	45.048	15.893	53.292	15.219	15.893	
	TSA	79.1 %	85.2 %	76.7 %	84.5 %	85.2 %	84.2 %
	C	1	1	1	1	1	
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	-1.78/3.46/1.32	0/3/0	
SM2,C	w_{S2}^*	20.520	15.640	20.620		15.640	
	TSA	60.9 %	84.6 %	51.0 %		84.6 %	84.2 %
	C	0.2591	0.6087	0.2510		0.6087	
	$\mu_1/\mu_2/\mu_3$	0.14/0/0	0/2.36/0	0/0/0.02		0/2.34/0	

Table 2: See the caption to Table 1 for explanation.

		K_1	K_2	K_3	$\sum_i \mu_i^* K_i$	$\sum_i \mu_{i,+}^* K_i$	RBF
Twonorm		$d = 2$	$\sigma = 0.1$				
HM	γ	0.1436	0.1072	0.0509	0.2170	0.2169	
	TSA	94.6 %	55.4 %	94.3 %	96.6 %	96.6 %	96.3 %
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	0.03/1.91/1.06	0.06/1.88/1.06	
SM1	w_{S1}^*	23.835	43.509	22.262	10.636	10.641	
	TSA	95.0 %	55.4 %	95.7 %	96.6 %	96.6 %	97.5 %
	C	1	1	1	1	1	
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	0.03/1.91/1.06	0.06/1.88/1.06	
SM2	w_{S2}^*	16.134	32.631	11.991	7.9780	7.9808	
	TSA	95.9 %	55.4 %	95.6 %	96.6 %	96.6 %	97.2 %
	C	1	1	1	1	1	
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	0.05/1.54/1.41	0.08/1.51/1.41	
SM2,C	w_{S2}^*	16.057	32.633	7.9880		7.9808	
	TSA	96.2 %	55.4 %	96.6 %		96.6 %	97.2 %
	C	0.8213	0.5000	0.3869		0.8015	
	$\mu_1/\mu_2/\mu_3$	2.78/0/0	0/2/0	0/0/1.42		0.08/1.25/1.41	

Table 3: See the caption to Table 1 for explanation.

	$\mu_{1,+}$	$\mu_{2,+}$	$\mu_{3,+}$	$\mu_{4,+}$	$\mu_{5,+}$	C	TSA SM2,C	TSA RBF
Breast Cancer	0	0	3.24	0.94	0.82	3.6e+08	97.1 %	96.8 %
Ionosphere	0.85	0.85	2.63	0.68	0	4.0e+06	94.5 %	94.2 %
Heart	0	3.89	0.06	1.05	0	2.5e+05	84.1 %	83.2 %
Sonar	0	3.93	1.07	0	0	3.2e+07	84.8 %	84.2 %
Twonorm	0.49	0.49	0	3.51	0	2.0386	96.5 %	97.2 %

Table 4: The initial kernel matrices $\{K_i\}_{i=1}^5$ are Gaussian kernels with $\sigma = 0.01, 0.1, 1, 10, 100$ respectively. For c we used $c = \sum_i \text{trace}(K_i) + \text{trace}(I_n)$. $\{\mu_{i,+}\}_{i=1}^5$ are the average weights of the optimal kernel matrix $\sum_i \mu_{i,+}^* K_i$ for a 2-norm soft margin SVM with $\mu_i \geq 0$ and auto-tuning of C. The average C-value is given as well. The test set accuracies (TSA) of the optimal 2-norm soft margin SVM with auto-tuning of C (SM2,C) and the best soft margin SVM with RBF kernel (RBF) are reported.