Journal of Machine Learning Research 11 (2010) 1353-1390 Submitted 3/09; Revised 12/09; Published 4/10

Learning Translation Invariant Kernels for Classification

Kamaledin Ghiasi-Shirazi GHIASI@AUT.AC.IR
Reza Safabakhsh SAFA@AUT.AC.IR
Computer Engineering Department
Amirkabir University of Technology
Tehran, 15914, Iran

Mostafa Shamsi M_SHAMSI@AUT.AC.IR
Faculty of Mathematics and Computer Science
Amirkabir University of Technology
Tehran, 15914, Iran

Editor: John Shawe-Taylor

Abstract

Appropriate selection of the kernel function, which implicitly defines the feature space of an algorithm, has a crucial role in the success of kernel methods. In this paper, we consider the problem of optimizing a kernel function over the class of translation invariant kernels for the task of binary classification. The learning capacity of this class is invariant with respect to rotation and scaling of the features and it encompasses the set of radial kernels. We show that how translation invariant kernel functions can be embedded in a nested set of sub-classes and consider the kernel learning problem over one of these sub-classes. This allows the choice of an appropriate sub-class based on the problem at hand. We use the criterion proposed by Lanckriet et al. (2004) to obtain a functional formulation for the problem. It will be proven that the optimal kernel is a finite mixture of cosine functions. The kernel learning problem is then formulated as a semi-infinite programming (SIP) problem which is solved by a sequence of quadratically constrained quadratic programming (QCQP) sub-problems. Using the fact that the cosine kernel is of rank two, we propose a formulation of a QCQP sub-problem which does not require the kernel matrices to be loaded into memory, making the method applicable to large-scale problems. We also address the issue of including other classes of kernels, such as individual kernels and isotropic Gaussian kernels, in the learning process. Another interesting feature of the proposed method is that the optimal classifier has an expansion in terms of the number of cosine kernels, instead of support vectors, leading to a remarkable speedup at run-time. As a by-product, we also generalize the kernel trick to complex-valued kernel functions. Our experiments on artificial and real-world benchmark data sets, including the USPS and the MNIST digit recognition data sets, show the usefulness of the proposed method.

Keywords: kernel learning, translation invariant kernels, capacity control, support vector machines, classification, semi-infinite programming

1. Introduction

Kernel-based methods, such as support vector machines (SVM) and kernel principal component analysis (KPCA), increase the flexibility of machine learning algorithms by implicitly mapping the input data into a feature space and performing the algorithm in that space. This flexibility is achieved by a so called kernel function which substitutes the dot-product operation in an ordinary algorithm. The kernel function, by implicitly defining the feature space, plays a crucial role in the

success of kernel methods. In fact, as shown by Xiong et al. (2005), if the kernel function is not chosen appropriately, it may even worsen the performance of an algorithm. This significant impact on the performance of the kernel-based algorithms and the fact that the appropriate feature space is problem-dependent, have driven researchers to devise various algorithms to learn the kernel function from the problem data.

The earliest method for learning a kernel function is cross-validation which is very slow and is only applicable to kernels with a small number of parameters. Cristianini et al. (1998) proposed an algorithm for adapting kernel functions with only one unconstrained parameter. Instead of optimizing the parameters of a kernel, Amari and Wu (1999) suggested conformal transformation of the kernel function and proposed an algorithm for learning the parameters of the new kernel. Chapelle et al. (2002) devised a gradient-based algorithm for local optimization of a kernel with multiple unconstrained parameters. Glasmachers and Igel (2005) proposed a gradient-based method for learning the covariance matrix of Gaussian kernels (Note that since the covariance matrix of a Gaussian kernel is constrained to be positive semi-definite, the method of Chapelle et al. (2002) cannot be used for learning this matrix). Ong et al. (2005) introduced the notion of hyperkernels and used it for kernel learning. They formulated the kernel learning problem as a functional with three terms: an empirical quality functional, a regularization term that penalizes the functions in a reproducing kernel Hilbert space (RKHS), and another regularization term that penalizes the kernels in a hyper reproducing kernel Hilbert space.

A milestone in the kernel learning literature is the introduction of the multiple kernel learning (MKL) framework by Lanckriet et al. (2004). They considered the problem of finding the optimal convex combination of multiple kernels and formulated it as a quadratically constrained quadratic programming (QCQP) problem. They also introduced a generalized performance measure which encompasses the hard-margin, 1-norm soft-margin, and 2-norm soft-margin performance measures as special cases. Although these performance measures have extensively been used for learning the optimal separating hyperplane in SVMs, their use as performance measures for kernel selection was unprecedented. Since the formulation of the resulting QCQP requires storing several kernel matrices in memory, their method was only applicable to problems with a small number of training samples. Bach et al. (2004) introduced an SMO-based algorithm to widen the range of solvable MKL problems by using the Moreau-Yosida regularization technique. Sonnenburg et al. (2005, 2006) reformulated the MKL problem as a semi-infinite linear program (SILP) which was then reduced to training a sequence of classical SVMs with a single kernel for which several sophisticated large-scale algorithms exist. Rakotomamonjy et al. (2008) argued that the main difficulty with the SILP formulation of Sonnenburg et al. (2006) is that its objective function is non-smooth and introduced an equivalent convex formulation with a smooth objective function. Using convexity of the problem and the smoothness of the objective function, they proposed a reduced gradient algorithm for MKL which is also applicable to large-scale problems. The weakness of the reduced gradient algorithm is that, in contrast to to the SILP algorithm, it does not use the information collected in the previous points in the calculation of the next point. Combining the strengths of the SILP method of Sonnenburg et al. (2006) with those of the reduced gradient method of Rakotomamonjy et al. (2008), Xu et al. (2008) proposed an extended level method which is remarkably faster than both methods.

In their seminal work, Micchelli and Pontil (2005) generalized the class of admissible kernels to convex combination of an infinite number of kernels indexed by a compact set and applied their method to the problem of learning radial kernels (Argyriou et al., 2005, 2006). They used a classical

result proved by Schoenberg (1938) which states that every continuous radial kernel belongs to the convex hull of radial Gaussian kernels. They also proposed an efficient DC programming algorithm for numerically learning radial kernels in Argyriou et al. (2006). Gehler and Nowozin (2008) reformulated the optimization problem of Argyriou et al. (2006) as a semi-infinite programming problem and proposed the IKL (infinite kernel learning) framework for solving it numerically.

In this work, we consider the class of translation invariant kernel functions which encompasses the class of radial kernels as well as the class of anisotropic Gaussian kernel functions. This class contains exactly those kernels which can be defined solely based on the difference of kernel arguments; that is, the kernel functions with the property:

$$k(x,z) = \tilde{k}(x-z).$$

The general form of continuous translation invariant kernels on $\mathbb{R}^n$ was discovered by Bochner (1933). He proved that every function of the form[1]

$$k(x,z) = \tilde{k}(x-z) = \int_{\mathbb{R}^n} e^{j\gamma^T(x-z)}\, dV(\gamma) \tag{1}$$

is positive semi-definite, where V(.) is a monotonically increasing bounded function and the integration is in the Lebesgue-Stieltjes sense. He also proved that, conversely, every continuous translation invariant positive semi-definite kernel function can be represented in the above form. In statistics, the translation invariance property is referred to as the stationarity of the kernel function. Genton (2001) and Schölkopf and Smola (2002) give a list of the properties of this class along with important examples of stationary kernel functions, including the Gaussian, exponential, rational quadratic, and B_n spline kernels.

The rest of the paper proceeds as follows: Table 1 lists the choice of notations for familiar concepts in the field. Notations specific to this paper will be introduced in the course of discussions. Although the kernel learning formulation of Micchelli and Pontil (2005) contains a regularization term for controlling the complexity of the RKHS associated with the kernel function, there is no mechanism for controlling the capacity of the class of admissible kernels. In our formulation, we have provisioned a mechanism for controlling the complexity of the class of admissible kernels which is described in Section 2. The idea is to multiply a vanishing function inside the integral of Equation (1). In addition to controlling the capacity of the learning machine, this choice substitutes the compactness assumption of the integration region made by Micchelli and Pontil (2005). In Section 3, we propose a learning criterion which is essentially a reformulation of the generalized performance measure of Lanckriet et al. (2004). The proposed criterion ensures the compactness of the parameter space of SVM, and gives a probabilistic meaning to the regularization parameter of the 2-norm soft-margin SVM. The problem of finding an optimal kernel which minimizes this criterion over the class of translation invariant kernels leads to (4) which is our main variational problem.

In Section 4, we prove some important theorems which pave the way for an algorithmic solution to this problem. First, in Section 4.1 we prove the existence of an optimal solution for problem (4).

1. In this paper we will represent translation invariant kernels both as $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{C}$ with two arguments and as $\tilde{k} : \mathbb{R}^d \to \mathbb{C}$ with only one argument.

In Section 4.2, we prove that the min and max operations in (4) can be interchanged, provided that the integration region is replaced by a compact set. In addition, it will be shown that the optimal kernel is a finite mixture of the basic kernels of the form $k_\gamma(x,z) = \exp(j\gamma^T(x-z))$. In Section 4.3, it will be proved that the integration region can indeed be replaced by a compact set. To solve problem (4) numerically, we introduce a semi-infinite programming (SIP) formulation in Section 4.4. In Section 4.5, using a topological argument, the issue of including other classes of kernels in the learning process will be addressed. It is well known that the regularization parameter of the 2-norm SVM, usually denoted by τ, can be regarded as the weight of a Kronecker delta kernel function, which is incidently a translation invariant kernel, as well. In Section 4.4 we introduce the semi-infinite programming problem (19) which is our main numerical optimization problem and corresponds to the simultaneous learning of the optimal translation invariant kernel as well as the parameter τ. As another application of the discussion of Section 4.5, in Section 4.7 we will introduce a method for learning the best combination of stabilized translation invariant kernels and isotropic Gaussian kernels.

In Section 5, we address the problem of numerically solving (19) on a computer. The proposed optimization algorithm is a variant of the class of local-reduction-based algorithms for solving SIP problems. An important feature of the proposed optimization algorithm is that it does not require loading the kernel matrices into memory and so it is applicable to large-scale problems. As stated above, it will be shown in Section 4.2 that the optimal kernel is complex-valued. Since algorithms are usually designed for real Euclidean spaces, this complicates the application of the kernel trick to the optimal kernel (consider for example an algorithm that checks the sign of a dot product). In Section 6, we show that the feature space induced by the real part of a complex-valued kernel is essentially equivalent to the original complex-valued kernel and deduce that the optimal real-valued kernel is a mixture of cosines. Yet another astounding feature of the proposed method is concerned with the evaluation time of the classifier which is even faster than a classical SVM with a single Gaussian kernel. Usually, multiple kernel learning methods yield a model whose evaluation time is in the order of the number of kernels times the number of support vectors. In Section 7, we show that the evaluation time of the optimal translation invariant kernel is proportional to the number of cosine kernels, regardless of the number of support vectors. In Section 8, using a learning theory discussion, we show the necessity of controlling the complexity of the class of translation invariant kernels.

In Section 9, we will assess the practical usefulness of the proposed method on several data sets. In Section 9.1, we first perform some experiments on 13 artificial and real-world data sets collected from the UCI, DELVE, and STATLOG benchmark repositories by Rätsch et al. (2001).In Section 9.2, we perform experiments on the USPS handwritten digit recognition data set, comparing the proposed method with the MKL of Chapelle et al. (2002). In Section 9.3, we compare the proposed method with the DC method of Argyriou et al. (2006) on the MNIST handwritten digit recognition data set. In Section 9.4, we experimentally assess the role of the capacity control mechanism of Section 2. Finally, we conclude the paper in Section 10.

2. A Hierarchy of Classes for Translation Invariant Kernels

It is well-known in learning theory that to have a small generalization error, there should be a problem-dependent compromise between the complexity of the learning machine and the empirical error on the training data (Vapnik, 1998; Cucker and Zhou, 2007). In the previous section we saw

Symbol	Meaning	Symbol	Meaning
n	input space dimension	l	number of training data
l_1	number of training data with label +1	l_2	number of training data with label -1
X	input space	F	Feature space
Φ	feature map: $\Phi : X \to F$	nsv	number of support vectors
α	lagrange multipliers in SVM	C	regularization parameter in SVM
k	the kernel function	Δ	maximal margin
C^+	the set of data with label +1	C^-	the set of data with label -1
m	number of kernels in MKL framework	μ	weight of kernels in MKL
R	radius of the smallest ball surrounding	$\Re(z)$	real part of the complex number z
	the data in the feature space	j	unit imaginary number: $\sqrt{-1}$
$\bar{z}$	the complex conjugate of z	S^c	the complement of set S

Table 1: Notations

that Equation (1) captures the general form of the large class of translation invariant kernels. To obtain the best tradeoff between approximation and estimation errors, we introduce a hierarchy of classes of translation invariant kernels. The appropriate class for a specific problem is then chosen by cross-validation. It must be mentioned that the idea of controlling the complexity of the class of admissible kernels has been previously used by Ong et al. (2005) for learning the kernels with hyperkernels. Although this type of complexity control is not provisioned by Micchelli and Pontil (2005), we, based on our experiments, believe that it is an important ingredient of the framework. To restrict the class of translation invariant kernels, we propose to limit the high frequency components of the kernel function. This is similar to the use of stabilizer functions in regularization theory (see Girosi et al., 1993). But, instead of adding a stabilizer function to the objective functional, which requires the determination of both the stabilizer and the regularization parameter, we explicitly define a nested class of translation invariant kernels $\mathcal{K}_\beta$ as:

$$\mathcal{K}_\beta := \left\{ k(x,z) = \int_{\mathbb{R}^n} e^{j\gamma^T(x-z)} G_\beta(\|\gamma\|)\, dp(\gamma) \quad : \text{ p is a probablity measure on } \mathbb{R}^n \right\}$$

where β is defined on an ordered set, and $G_\beta : \mathbb{R}_+ \to [0,1]$ is a decreasing continuous function with $G_\beta(0) = 1$ and $G_\beta(\infty) = 0$. In addition, to ensure that these classes of kernels are nested, we require that $G_{\beta_1}(r) \geq G_{\beta_2}(r)$ for every $r > 0$ and $\beta_1 \leq \beta_2$. Important candidates for $G_\beta(\|\gamma\|)$ are $\exp(-\beta\|\gamma\|^2)$ and $\|\gamma\|^{-\beta}$ for $\beta > 0$. Note that we have left the choice of the norm to the application. Two important candidates are L_1 and L_2 norms. In Section 5, we will also assume that the function $G_\beta(\|\gamma\|)$ is differentiable with respect to γ.

3. Kernel Selection Criterion

In the process of learning the kernel function, one needs a criterion for choosing a kernel (or equivalently, feature space) from the class of admissible kernels. In the classification task, the ideal criterion is the misclassification error. But since the probability density of data is unknown, this criterion cannot be computed. This problem has been circumvented by proposing upper bounds on the misclassification error which hold with high probability (see for example Vapnik, 1999 and Chapter 4 of Cristianini and Shawe-Taylor, 2000). In Chapelle et al. (2002), several criteria have

been studied and the authors suggested minimizing the radius-margin bound $(R/\Delta)^2$ as the preferred criterion.

Consider the training set of a classification task consisting of the input-output pairs $\{(x_1,y_1),...,(x_l,y_l)\}$, with $y_i \in \{-1,1\}$. For a fixed kernel, support vector machines compute the maximal hard/soft margin separating hyperplane. By generalizing hard margin, 1-norm soft margin, and 2-norm soft-margin objective functions, Lanckriet et al. (2004) obtained the following generalized criterion for $1/\Delta^2$ (which must be minimized):

$$\omega'_{C',\tau'}(k) := \max_{\substack{\alpha:\ 0\le\alpha\le C',\\ \alpha^T y=0}} \left\{ 2\alpha^T e - \sum_{u=1}^{l}\sum_{v=1}^{l} \alpha_u\alpha_v y_u y_v k(x_u,x_v) - \tau'\alpha^T\alpha \right\} \tag{2}$$

where e is the $n\times 1$ vector of ones and the prime sign is used to distinguish the parameters from those used in this paper. If (C',τ') is equal to $(\infty,0)$,$(C',0)$, and $(\infty,1/C')$, one obtains hard margin, 1-norm soft margin, and 2-norm soft-margin performance measures, respectively. Our first change to this criterion is adding the constraint $\alpha^T e = 2$. It has been shown (Crisp and Burges, 1999; Mavroforakis and Theodorodis, 2006) that by adjusting the parameters C and the offset b appropriately, this new constraint does not change the separating hyperplane. However, the new constraint plus $\alpha^T y = 0$ gives $\sum_{i\in C^-}\alpha_i = \sum_{i\in C^+}\alpha_i = 1$, which makes the exposition of our method simpler. Furthermore, we divide (2) by $1+\tau'$ and define $\tau := \frac{\tau'}{1+\tau'}$. So, in this paper we use the criterion:

$$\omega_{C,\tau}(k) := \min_{\alpha\in\mathcal{A}} \left\{ (1-\tau)\sum_{u=1}^{l}\sum_{v=1}^{l} \alpha_u\alpha_v y_u y_v k(x_u,x_v) + \tau\alpha^T\alpha \right\} \tag{3}$$

where $\mathcal{A} := \left\{\alpha\in\mathbb{R}^l :\ 0\le\alpha\le C,\ \alpha^T y = 0,\ \alpha^T e = 2\right\}$. Note that in the new criterion $\max\{\frac{1}{l_1},\frac{1}{l_2}\} \le C \le 2$ and $0\le\tau\le 1$. This criterion works well for a fixed kernel by maximizing the margin Δ. But in general, to minimize the radius-margin bound $(R/\Delta)^2$, one must impose some constraint on the radius R, as well. For translation invariant kernels we have $R^2 = \|\Phi(x)\|^2 = k(x,x) = \tilde{k}(0)$. Hence, bounding the radius R is equivalent to bounding $\tilde{k}(0)$. One can easily verify that bounding $trace\{K\}$, where K is the kernel matrix in the transductive framework (see Lanckriet et al., 2004, Equation 17), also leads to a bound on $\tilde{k}(0)$. Since by exploding R and $trace\{K\}$, the margin Δ also explodes by the same amount, whilst the radius-margin bound remains constant, we from now on assume that $\tilde{k}(0) = 1$ and obtain the following optimization problem:[2]

$$\begin{aligned} \sup_k \min_{\alpha\in\mathcal{A}} & \left\{ (1-\tau)\sum_{u=1}^{l}\sum_{v=1}^{l} \alpha_u\alpha_v y_u y_v k(x_u,x_v) + \tau\alpha^T\alpha \right\} \\ \text{s.t.} \quad & k(x,z) = \int_{\mathbb{R}^n} e^{j\gamma^T(x-z)} G_\beta(\|\gamma\|)\, dV(\gamma) \\ & \int_{\mathbb{R}^n} dV(\gamma) = 1, \\ & \text{V is monotonically increasing} \end{aligned}$$

2. Note that since k is a complex-valued positive semi-definite kernel, the objective function is real-valued and non-negative.

or equivalently

$$\sup_{p\in\mathcal{P}(\mathbb{R}^n)} \min_{\alpha\in\mathcal{A}} \int_{\mathbb{R}^n} \alpha^T H(\gamma)\alpha \, dp(\gamma) \tag{4}$$

where $\mathcal{P}(\Gamma)$ denotes the set of all probability measures on Γ and $H(\gamma)$ is defined below. Let $G(\gamma)$ be the $l \times l$ matrix whose (u,v)th entry is $y_u y_v \exp(j\gamma^T(x_u - x_v))G_\beta(\|\gamma\|)$. Define $H(\gamma) := (1-\tau)G(\gamma) + \tau I_l$ where I_l is the $l \times l$ identity matrix. The following equation shows an $O(l)$ computational method for computing $\alpha^T G(\gamma)\alpha$:

$$\alpha^T G(\gamma)\alpha = \left\| \sum_{u=1}^{l} \alpha_u y_u \exp(j\gamma^T x_u) \right\|_2^2 G_\beta(\|\gamma\|). \tag{5}$$

4. Variational Optimization

In this section, we first prove the existence of a solution to problem (4). Next we prove that (4) can be written as a min-max problem with integration replaced by summation. This allows us to introduce a SIP formulation of the problem. We then introduce another SIP problem for learning the optimal kernel and parameter τ.

4.1 Replacing Sup with Max

We will prove that the sup operation in (4) can be substituted by the max operation. Note that all the variability in the choice of a probability measure p from $\mathcal{P}(\mathbb{R}^n)$ collapses to the choice of an $l \times l$ matrix $\int_{\mathbb{R}^n} H(\gamma)\, dp(\gamma)$ from $S(\mathbb{C}^l)$, where $S(\mathbb{C}^l)$ is the space of all $l \times l$ Hermitian complex-valued matrices. So, it is sufficient to prove that the set of all these matrices is compact, which ensures that any sequence of these matrices has a convergent subsequence, and subsequently, the supremum value is achieved. Furthermore, by compacting the parameter space $\mathcal{A}$ in (3) there is no need to assume that the kernel matrices are strictly positive definite, as was done in Lemma 2 of Micchelli and Pontil (2005).

Let $C_0(\mathbb{R}^n)$ denote the function space of all continuous complex-valued functions defined on $\mathbb{R}^n$ which vanish at infinity, that is, $\lim_{\|\gamma\|\to\infty} g(\gamma) = 0$ for any $g \in C_0(\mathbb{R}^n)$. By Theorem 3.17 of Rudin (1987), the function space $C_0(\mathbb{R}^n)$ with the norm

$$\|g\| := \max_{\gamma\in\mathbb{R}^n} |g(\gamma)|$$

is a Banach space. Note that the use of max operation is justified by the continuity and vanishing properties of $g \in C_0(\mathbb{R}^n)$. Considering the above discussions, we need to prove the following theorem.

Theorem 1 *For any fixed sample data $Z = \{(x_1,y_1),\ldots,(x_l,y_l)\}$, the set*

$$\mathcal{K}_Z := \left\{ \int_{\mathbb{R}^n} H(\gamma)\, dp(\gamma) \;\; : p \in \mathcal{P}(\mathbb{R}^n) \right\}$$

is a compact subset of $S(\mathbb{C}^l)$.[3]

Proof Consider any sequence (p_n) in $\mathcal{P}(\mathbb{R}^n)$. Define positive linear functionals $T_n : C_0(\mathbb{R}^n) \to \mathbb{C}$ by $T_n g := \int_{\mathbb{R}^n} g(\gamma)\, dp_n(\gamma)$. So, $T_n \in C_0'(\mathbb{R}^n)$, the dual space of $C_0(\mathbb{R}^n)$. Since for each n, $\|T_n\| = 1$, by Banach-Alaoglu theorem (see for example Theorem 3.15 of Rudin, 1991 or page 237 of Royden, 1988), there exists some $T \in C_0'(\mathbb{R}^n)$ with $\|T\| \leq 1$ and a subsequence (T_m) such that $T_m \to T$ in weak* topology. This means that for every $g \in C_0(\mathbb{R}^n)$, we have $T_m g \to Tg$. Since, each element of the matrix $H(\gamma)$ belongs to $C_0(\mathbb{R}^n)$, it follows that $\int_{\mathbb{R}^n} H(\gamma)\, dP_m(\gamma) \to TH$. One can easily prove by contradiction that $\|T\| = 1$ and T is in fact a positive linear functional. By the Riesz representation theorem (see Theorem 6.19 of Rudin, 1987), the functional T can be represented uniquely by a complex Borel measure μ ,with $\int_{\mathbb{R}^n} d|\mu| = \|T\| = 1$, in the sense that

$$Tg = \int_{\mathbb{R}^n} g\, d\mu \quad \text{for every } g \in C_0(\mathbb{R}^n).$$

The positivity of T implies that μ is a positive real measure. So, $\int_{\mathbb{R}^n} d\mu = 1$ and consequently μ is also a probability measure on $\mathbb{R}^n$. Thus,

$$\int_{\mathbb{R}^n} H(\gamma)\, dp_m(\gamma) \longrightarrow \int_{\mathbb{R}^n} H(\gamma)\, d\mu(\gamma)$$

which indicates that the set $\mathcal{K}_Z$ is compact. ■

4.2 Interchanging the min and max Operations

We first prove a theorem about interchanging the min and max operations which is an abstracted and generalized version of Theorem 20 in Micchelli and Pontil (2005).

Theorem 2 *Assume that Γ is a compact Hausdorff space and the function $g : \Gamma \times \mathbb{R}^l \to \mathbb{R}$ is continuous in the first parameter and convex and differentiable in the second parameter. Let $\mathcal{E}$ and I be finite index sets , a_i, where $i \in \mathcal{E} \cup I$, be $l \times 1$ vectors, and b_i, where $i \in \mathcal{E} \cup I$, be real-valued scalars. Considering problem (6), assume that the Slater's condition (see Boyd and Vandenberghe, 2004) holds, that is, there exists some α such that $a_i^T \alpha = b_i$ for all $i \in \mathcal{E}$ and $a_i^T \alpha < b_i$ for all $i \in I$. Then there exist a discrete probability measure $\tilde{p} \in \mathcal{P}(\Gamma)$ with at most $l+1$ atoms and some feasible point $\tilde{\alpha}$ which solve the max-min problem*

$$\max_{p \in \mathcal{P}(\Gamma)} \min_{\substack{\alpha\,:\, a_i^T \alpha = b_i \text{ for } i \in \mathcal{E}, \\ a_i^T \alpha \geq b_i \text{ for } i \in I}} \int_\Gamma g(\gamma, \alpha)\, dp(\gamma) \tag{6}$$

and the min-max problem

$$\min_{\substack{\alpha\,:\, a_i^T \alpha = b_i \text{ for } i \in \mathcal{E}, \\ a_i^T \alpha \geq b_i \text{ for } i \in I}} \max_{p \in \mathcal{P}(\Gamma)} \int_\Gamma g(\gamma, \alpha)\, dp(\gamma) \tag{7}$$

simultaneously. In addition, each atom of $\tilde{p}$ is a global maximum of $g(\gamma, \tilde{\alpha})$ as a function of γ.

3. Note that the topology of $S(\mathbb{C}^l)$ is the same as that of $\mathbb{R}^{l^2}$.An $l \times l$ hermitian matrix contains l real-valued diagonal elements and $\frac{l^2-l}{2}$ independent complex-valued off-diagonal element.

Proof Assume that $\hat{\alpha}$ and $\hat{p}$ solve the problem (7). Define the function

$$\phi : \mathbb{R}^l \to \mathbb{R}$$
$$\phi(\alpha) := \max_{\gamma \in \Gamma} \{g(\gamma, \alpha)\}$$

and the set

$$\Gamma^* := \{\gamma : \gamma \in \Gamma, g(\gamma, \hat{\alpha}) = \phi(\hat{\alpha})\}.$$

By Lemma 24 of Micchelli and Pontil (2005), the directional derivative of ϕ along the direction $d \in \mathbb{R}^l$, denoted by $\phi'_+(\alpha; d)$, is given by:

$$\phi'_+(\alpha; d) = \max_{\gamma \in \Gamma^*} \left\{ d^T \nabla_\alpha g(\gamma, \alpha) \right\}.$$

Since $\hat{\alpha}$ minimizes (7), we have

$$\phi'_+(\hat{\alpha}; d) = \max_{\gamma \in \Gamma^*} \left\{ d^T \nabla_\alpha g(\gamma, \alpha) \mid_{\alpha = \hat{\alpha}} \right\} \geq 0 \tag{8}$$

for any direction d such that $a_i^T d = 0$ for $i \in \mathcal{E}$ and $a_i^T d \geq 0$ for $i \in I^*$, where $I^* := \{i \in I : a_i^T \alpha = b_i\}$.

Let $\mathcal{M}$ be the convex hull of the set of vectors $\mathcal{N} := \{\nabla_\alpha g(\gamma, \alpha) \mid_{\alpha = \hat{\alpha}} : \gamma \in \Gamma^*\} \subseteq \mathbb{R}^l$. Since $\mathcal{M} \subseteq \mathbb{R}^l$, by the Caratheodory theorem (see for example Section 17 of Rockafellar, 1970) every vector in $\mathcal{M}$ can be expressed as a convex combination of at most $l+1$ elements of $\mathcal{N}$. We claim that the set

$$O := \left\{ \sum_{i \in \mathcal{E} \cup I^*} \lambda_i a_i : \ \lambda_i \geq 0 \text{ for all } i \in I^* \right\}$$

intersects $\mathcal{M}$. Assume, on the contrary, that $\mathcal{M}$ and O are distinct. Since $\mathcal{M}$ is convex and compact and O is convex and closed, by the strict separating hyperplane theorem (see corollary 11.4.2 of Rockafellar, 1970), there exists a separating hyperplane $w^T \alpha + b = 0$, $w \in \mathbb{R}^l$, $b \in \mathbb{R}$, such that

$$\sum_{i \in \mathcal{E} \cup I^*} \lambda_i w^T a_i + b > 0, \qquad \forall \lambda : \ \lambda_i \geq 0 \text{ for } i \in I^*$$

and

$$w^T \nabla_\alpha g(\gamma, \alpha) \mid_{\alpha = \hat{\alpha}} + b < 0, \qquad \forall \gamma \in \Gamma^*. \tag{9}$$

The first condition, for $\lambda = 0$ implies that $b > 0$ and since λ_i can take any real value for $i \in \mathcal{E}$ and any nonnegative value for $i \in I^*$, we have

$$w^T a_i = 0, \quad \forall i \in \mathcal{E},$$
$$w^T a_i \geq 0, \quad \forall i \in I^*.$$

By combining these results with (9), we get

$$\max_{\gamma \in \Gamma^*} w^T \nabla_\alpha g(\gamma, \hat{\alpha})\,|_{\alpha=\hat{\alpha}} < 0.$$

This means that w is a feasible descent direction at $\hat{\alpha}$ which contradicts (8). So, the sets O and $\mathcal{M}$ intersect. This means that there exist real numbers $\hat{\lambda}_i,\ i \in \mathcal{E}$, nonnegative numbers $\hat{\lambda}_i,\ i \in I^*$, and a discrete probability measure $\hat{p}$ with at most $l+1$ atoms such that

$$\sum_{i \in \mathcal{E} \cup I^*} \hat{\lambda}_i a_i = \int_\Gamma \nabla_\alpha g(\gamma, \hat{\alpha})\, d\hat{p}. \tag{10}$$

Now, we turn our attention to the solution of problem (6) for $p = \hat{p}$. This is a convex optimization problem. Since by assumption the Slater's condition holds, the KKT conditions provide a necessary and sufficient condition for optimality (see Boyd and Vandenberghe, 2004, page 244) and therefore a solution $\check{\alpha}$ to problem (6) is found by solving the following KKT conditions:

$$\begin{aligned}
\sum_{i \in \mathcal{E} \cup I} \check{\lambda}_i a_i &= \int_\Gamma \nabla_\alpha g(\gamma, \check{\alpha}) d\hat{p} \\
a_i^T(\check{\alpha}) &= 0, \quad i \in \mathcal{E} \\
a_i^T(\check{\alpha}) &\geq 0, \quad i \in I \\
\check{\lambda}_i &\geq 0, \quad i \in I \\
\check{\lambda}_i \left(a_i^T \check{\alpha} - b_i\right) &= 0, \quad i \in I.
\end{aligned}$$

By defining $\hat{\lambda}_i = 0$ for $i \in I \setminus I^*$, using (10), and recalling the definition of I^*, it can be seen that $\hat{\alpha}, \hat{\lambda}$ are the unique solution to the above KKT conditions. Thus, $\tilde{\alpha} = \hat{\alpha}$ and $\tilde{p} = \hat{p}$ solve problems (6) and (7) simultaneously and the theorem follows. ■

Corollary 3 *Assume that Γ is any compact subset of $\mathbb{R}^n$ and the parameter C is chosen*[4] *such that $C > \max\{\frac{1}{l_1}, \frac{1}{l_2}\}$. Then, there exist $\tilde{\alpha} \in \mathbb{R}^n$ and $\tilde{p} \in \mathcal{P}(\Gamma)$ that solve problems (11) and (12) simultaneously. Furthermore, $\tilde{p}$ is a discrete probability measure with at most $l+1$ atoms and each atom of $\tilde{p}$ is a global maximum of $\tilde{\alpha}^T H(\gamma)\tilde{\alpha}$.*

$$\max_{p \in \mathcal{P}(\Gamma)} \min_{\alpha \in \mathcal{A}} \int_\Gamma \alpha^T H(\gamma) \alpha\, dp(\gamma), \tag{11}$$

$$\min_{\alpha \in \mathcal{A}} \max_{p \in \mathcal{P}(\Gamma)} \int_\Gamma \alpha^T H(\gamma) \alpha\, dp(\gamma). \tag{12}$$

4. For a similar constraint in the context of ν-SVMs see Section 4 of Crisp and Burges (1999).

Proof It is sufficient to show that the Slater's condition holds, that is, there exists $\alpha \in \mathbb{R}^l$ such that $\alpha^T e = 2$, $\alpha^T y = 0$, $\alpha > 0$, and $\alpha < C$. One can easily verify that the choice $\alpha_i = \frac{1}{l_1}$ for $i \in C^+$ and $\alpha_i = \frac{1}{l_2}$ for $i \in C^-$ satisfies these conditions. ■

In the rest of the paper, we assume that $C > \max\{\frac{1}{l_1}, \frac{1}{l_2}\}$.

4.3 Confining Integration to a Compact Region

To write (4) as a min-max optimization problem, we first proved in Section 4.1 that the sup operation can be replaced by the max operation. In the previous section we proved Corollary 3 which asserts that if the integration region of (4) could have been replaced by a compact set, then the max and min operations could also be interchanged. In this section, we show that the integration region of (4) can safely be confined to a compact subset of $\mathbb{R}^n$. Let us first prove two useful lemmas.

Lemma 4 *For arbitrary domains X and Y and every function $f : X \times Y \to \mathbb{R}$, the following inequality holds:*

$$\sup_{x\in X} \inf_{y\in Y} f(x,y) \leq \inf_{y\in Y} \sup_{x\in X} f(x,y).$$

Proof Assume on the contrary that

$$\sup_{x\in X} \inf_{y\in Y} f(x,y) > \inf_{y\in Y} \sup_{x\in X} f(x,y).$$

Then, there exist $\tilde{x} \in X$ and $\tilde{y} \in Y$ such that

$$\inf_{y\in Y} f(\tilde{x},y) > \sup_{x\in X} f(x,\tilde{y})$$

which contradicts with the existence of $f(\tilde{x},\tilde{y})$. ■

Lemma 5 *If $\tau < 1$, then there exists some compact subset Γ_β of $\mathbb{R}^n$, independent of α, where all γ's that maximize $\alpha^T H(\gamma)\alpha$ lie in it.*[5]

Proof For each $\alpha \in \mathbb{R}^l$ we have

$$\max_{\gamma\in\mathbb{R}^n} \alpha^T H(\gamma)\alpha = (1-\tau) \max_{\gamma\in\mathbb{R}^n} \left\{ \left\| \sum_{u=1}^{l} \alpha_u y_u \exp(j\gamma^T x_u) \right\|_2^2 G_\beta(\|\gamma\|) \right\} + \tau\alpha^T\alpha. \tag{13}$$

Let $t(\alpha)$ denote the maximum value of the term in the braces in the above equation. Since G_β is continuous and $G_\beta(0) = 1$, there is an open ball around zero in $\mathbb{R}^n$ such that $G_\beta(\|\gamma\|) > 0$. This fact plus the condition $\alpha^T e = 2$, ensure that the coefficient of at least one of the exponential terms in (13) is nonzero. Hence, the term in the braces in (13), as a function of γ, is never identically zero and thus $t(\alpha) > 0$. For all values of γ with $G_\beta(\|\gamma\|) < \frac{t}{4}$ we have

5. Note that by (3), the choice $\tau = 1$ is unrealistic.

$$\alpha^T H(\gamma)\alpha - \tau\alpha^T\alpha = \left\| \sum_{u=1}^{l} \alpha_u y_u \exp(j\gamma^T x_u) \right\|^2 G_\beta(\|\gamma\|) \leq 4G_\beta(\|\gamma\|) < t(\alpha)$$

and the lemma's assertion follows from $lim_{r\to\infty} G(r) = 0$. ■

Theorem 6 *There exists an optimal solution $(\tilde{\alpha}, \tilde{p})$ to problem (4) which is also a solution to it when the integration domain is confined to the compact set Γ_β.*

Proof Let $(\tilde{\alpha}, \tilde{p})$ be a solution to problems (6) and (7) with Γ replaced by Γ_β. By Lemma 4 and Corollary 3, we have the following sequence of inequalities:

$$\begin{aligned}
\max_{p\in\mathcal{P}(\Gamma_\beta)} \min_{\alpha\in\mathcal{A}} \int_{\Gamma_\beta} \alpha^T H(\gamma)\alpha\, dp(\gamma) \leq \\
\max_{p\in\mathcal{P}(\mathbb{R}^n)} \min_{\alpha\in\mathcal{A}} \int_{\mathbb{R}^n} \alpha^T H(\gamma)\alpha\, dp(\gamma) \leq \\
\min_{\alpha\in\mathcal{A}} \max_{p\in\mathcal{P}(\mathbb{R}^n)} \int_{\mathbb{R}^n} \alpha^T H(\gamma)\alpha\, dp(\gamma) \leq \\
\min_{\alpha\in\mathcal{A}} \max_{p\in\mathcal{P}(\mathbb{R}^n)} \int_{\mathbb{R}^n} \max_{\gamma\in\mathbb{R}^n} \alpha^T H(\gamma)\alpha\, dp(\gamma) = \\
\min_{\alpha\in\mathcal{A}} \max_{\gamma\in\mathbb{R}^n} \alpha^T H(\gamma)\alpha = \\
\min_{\alpha\in\mathcal{A}} \max_{p\in\mathcal{P}(\Gamma_\beta)} \int_{\Gamma_\beta} \alpha^T H(\gamma)\alpha\, dp(\gamma).
\end{aligned}$$

However, the first and the last terms are equal by Corollary 3.

4.4 Semi-infinite Programming Formulation

We have not yet addressed the problem of how (12) is to be really solved on a machine. In this section, we reformulate (12) as a semi-infinite programming problem for which many algorithms have been proposed (see Hettich and Kortanek, 1993; Reemtsen and Görner, 1998, for two reviews on the subject).

Theorem 7 *Let $\tilde{\alpha}$ and $\tilde{t}$ be a solution to the semi-infinite programming problem (14) and define the set $\Gamma_\beta^H(\alpha) := \left\{\gamma \in \Gamma_\beta : \alpha^T H(\gamma)\alpha = max_{\gamma\in\Gamma_\beta} \alpha^T H(\gamma)\alpha\right\}$. Let (Q) be the QCQP problem that is obtained by replacing Γ_β by $\Gamma_\beta^H(\tilde{\alpha}) \equiv \{\gamma_1, ..., \gamma_m\}$ in (14) and let $\tilde{\mu}_1, ..., \tilde{\mu}_m$ be a set of Lagrange multipliers associated with the constraints $t \geq \alpha^T H(\gamma_i)\alpha$, $1 \leq i \leq m$ which optimize the dual problem of (Q). If $\tilde{p}$ is the discrete probability measure defined by $\tilde{p}(\gamma_i) := \tilde{\mu}_i$ $i = 1, ..., m$, then $\tilde{\alpha}$ and $\tilde{p}$ solve the problem (4). In addition, there exists a solution pair $(\tilde{\alpha}^*, \tilde{p}^*)$ such that $\tilde{p}^*$ contains at most $l+1$ nonzero atoms.*

$$\begin{array}{ll} \min_{\alpha,t} & t \\ s.t. & t \geq \alpha^T H(\gamma)\alpha \;\; \textit{for all} \;\; \gamma \in \Gamma_\beta \\ & 0 \leq \alpha \leq C \\ & \alpha^T y = 0 \\ & \alpha^T e = 2. \end{array} \tag{14}$$

Proof Since for all $\gamma \notin \Gamma_\beta^H(\tilde{\alpha})$ the strict inequality $\tilde{t} > \tilde{\alpha}^T H(\gamma)\tilde{\alpha}$ holds, $\tilde{\alpha}$ and $\tilde{t}$ also solve the following QCQP problem:

$$\begin{array}{ll} \min_{\alpha,t} & t \\ s.t. & t \geq \alpha^T H(\gamma_i)\alpha \;\; i = 1,...,m \\ & 0 \leq \alpha \leq C \\ & \alpha^T y = 0 \\ & \alpha^T e = 2. \end{array}$$

In addition, from $\tilde{t} = \tilde{\alpha}^T H(\gamma_1)\tilde{\alpha} = ... = \tilde{\alpha}^T H(\gamma_m)\tilde{\alpha}$, it follows that $\tilde{\alpha}$ and $\tilde{t}$ also solve the following problem:

$$\min_{\alpha \in \mathcal{A}} \max_{\mu \geq 0,\, \sum_{i=1}^m \mu_i = 1} \alpha^T \left[\sum_{i=1}^m \mu_i H(\gamma_i) \right] \alpha \tag{15}$$

By Theorem 17 of Lanckriet et al. (2004), if $\tilde{\mu}$ is chosen as specified by the statement of this theorem, then $\tilde{\alpha}$ and $\tilde{\mu}$ simultaneously solve the min-max problem (15) and the following max-min problem:

$$\max_{\mu \geq 0,\, \sum_{i=1}^m \mu_i = 1} \min_{\alpha \in \mathcal{A}} \alpha^T \left[\sum_{i=1}^m \mu_i H(\gamma_i) \right] \alpha$$

which can also be written as

$$\max_{p \in \mathcal{P}(\Gamma_\beta^H(\tilde{\alpha}))} \min_{\alpha \in \mathcal{A}} \int \alpha^T H(\gamma)\alpha \, dP(\gamma)$$

The first assertion of the theorem follows from the following inequalities:

$$\begin{aligned} \tilde{t} = \max_{p \in \mathcal{P}(\Gamma_\beta^H(\tilde{\alpha}))} \min_{\alpha \in \mathcal{A}} \int \alpha^T H(\gamma)\alpha \, dP(\gamma) \leq \\ \max_{p \in \mathcal{P}(\Gamma_\beta)} \min_{\alpha \in \mathcal{A}} \int \alpha^T H(\gamma)\alpha \, dP(\gamma) \leq \\ \min_{\alpha \in \mathcal{A}} \max_{p \in \mathcal{P}(\Gamma_\beta)} \int \alpha^T H(\gamma)\alpha \, dP(\gamma) = \tilde{t} \end{aligned}$$

where the last equality follows from a simple reformulation of problem (14). The last part of the theorem follows from Corollary 3 and reversing the above proof. ■

4.5 Including Other Kernels in the Learning Process

Although the focus of this paper is on the task of learning translation invariant kernels, it is easy to furnish the set of admissible kernels with other kernel functions. For example, one may want to find the best convex combination of stabilized translation invariant kernels along with isotropic/non-isotropic Gaussian kernels, and polynomial kernels with degrees one to five.[6] In general, assume that we have M classes of kernels

$$K_i := \left\{ k_\gamma(x,z) : \ \gamma \in \Gamma_i \right\} \qquad i = 1,...,M$$

where $\Gamma_1,...,\Gamma_M$ are distinct compact Hausdorff spaces. For $i \in 1,...,M$ let $G_i(\gamma)$ be the $l \times l$ matrix whose (u,v)'s entry is $y_u y_v k_\gamma(x_u,x_v)$ and define $H_i(\gamma) := (1-\tau)G_i(\gamma) + \tau I_l$. The problem of learning the best convex combination of kernels from these classes for classification with support vector machines can be stated as

$$\sup_{p \in \mathcal{P}(\Gamma_0)} \min_{\alpha \in \mathcal{A}} \int_{\Gamma_0} \alpha^T H_0(\gamma) \alpha \, dp(\gamma) \tag{16}$$

where $\Gamma_0 := \Gamma_1 \cup ... \cup \Gamma_M$ and

$$H_0(\gamma) := \begin{cases} H_1(\gamma) & if \gamma \in \Gamma_1 \\ H_2(\gamma) & if \gamma \in \Gamma_2 \\ \vdots & \\ H_M(\gamma) & if \gamma \in \Gamma_M \end{cases}.$$

The results of the previous sections will hold for this combined class of kernels if we prove that Γ_0 is a compact Hausdoff space and that $h_0(\gamma,\alpha) := \alpha^T H_0(\gamma)\alpha$ is continuous with respect to γ. Let $\mathcal{T}_i$ denote the topology on Γ_i for $i \in 1,...,M$. Define the set $\mathcal{T}_0$ of subsets of Γ_0 as:

$$\mathcal{T}_0 := \{ O_1 \cup O_2 ... \cup O_M : O_1 \in \mathcal{T}_1, O_2 \in \mathcal{T}_2, ..., O_M \in \mathcal{T}_M \}.$$

Proposition 8 *$\mathcal{T}_0$ is a topology.*

Proof Clearly $\Gamma_0 \in \mathcal{T}_0$. Next we must show that $\mathcal{T}_0$ is closed under arbitrary union. Let $O = \bigcup_{i \in \mathfrak{I}} O_i$ where $O_i \in \mathcal{T}_0$ and $\mathfrak{I}$ is an arbitrary index set. We have

$$O = \bigcup_{j \in \{1,...,M\}} \left(\bigcup_{i \in \mathfrak{I}} O_i \cap \Gamma_j \right)$$

6. Although the class of translation invariant kernels includes the set of isotropic/non-isotropic Gaussian kernels, it is not the case for the stabilized class $\mathcal{K}_\beta$.

which shows that $O \in \mathcal{T}_0$. Finally, we should show that finite intersection of closed sets is a closed set. Assume that $C = \bigcap_{i=1}^{r} C_i$, where $r \in \mathbb{N}$ and $C_i^c \in \mathcal{T}_0$. We have

$$C^c = \bigcup_{i\in\{1,\ldots,r\}} C_i^c = \bigcup_{i\in\{1,\ldots,r\}} \bigcup_{j\in\{1,\ldots,M\}} (C_i^c \cap \Gamma_j).$$

Since C_i^c and Γ_j are both open in $\mathcal{T}_j$, the set $C_i^c \cap \Gamma_j$ is open in $\mathcal{T}_j$. By the properties of topologies $\mathcal{T}_1, \ldots, \mathcal{T}_M$ and the definition of topology $\mathcal{T}_0$ it follows that $C^c \in \mathcal{T}_0$ which shows that C is closed in $\mathcal{T}_0$. ■

Proposition 9 *Assume that the functions $h_i(\gamma,\alpha) : \Gamma_i \times \mathbb{R}^l \to \mathbb{R}$ defined as $h_i(\gamma,\alpha) = \alpha^T H_i(\gamma)\alpha$, where $i = 1,\ldots,M$, are continuous in the first parameter. Then, the function $h_0(\gamma,\alpha) : \Gamma_0 \times \mathbb{R}^l \to \mathbb{R}$ defined as $h_0(\gamma,\alpha) = \alpha^T H(\gamma)\alpha$,where the topology of Γ_0 is $\mathcal{T}_0$, is also continuous in the first parameter.*

Proof Fix α to any value and define $\bar{h}_i(\gamma) := h_i(\gamma,\alpha)$ for $i = 0,\ldots,M$. Let O be an open subset of $\mathbb{R}$. We must show that $\bar{h}_0^{-1}(O)$ is open in topology $\mathcal{T}_0$. We have

$$\bar{h}_0^{-1}(O) = \bigcup_{i=1,\ldots,M} (\bar{h}_0^{-1}(O) \cap \Gamma_i) = \bigcup_{i=1,..,M} \bar{h}_i^{-1}(O).$$

Since the set $\bar{h}_i^{-1}(O)$ is open in $\mathcal{T}_i$ for each $i \in \{1,\ldots,M\}$, it follows from the definition of $\mathcal{T}_0$ that the union of these sets is also open in $\mathcal{T}_0$. So, $\bar{h}_0^{-1}(O)$ is open in $\mathcal{T}_0$ and the result follows. ■

Proposition 10 *The set Γ_0 with topology $\mathcal{T}_0$ is compact in itself.*

Proof Let $O = \{O_i : i \in I_0\}$ be an open covering of Γ_0, where I_0 is some index set. Let j be any number in the set $\{1,\ldots,M\}$. Since $\Gamma_j \subseteq \Gamma_0$, the set O is also an open covering for Γ_j. By compactness of Γ_j, there exists a finite index set $I_j \subseteq I_0$ such that the set $\{O_i : i \in I_j\}$ is an open subcovering of Γ_j. Thus, the set $\{O_i : i \in I_1 \cup \ldots \cup I_M\}$ is a finite open subcovering of $\mathcal{T}_0$ which proves that Γ_0 is compact. ■

Proposition 11 *Assume that topologies $\mathcal{T}_1,\ldots,\mathcal{T}_M$ are Hausdorff. Then, so is the topology $\mathcal{T}_0$.*

Proof We must prove that for any two points $\gamma_1, \gamma_2 \in \Gamma_0$ there are disjoint open sets O_1 and O_2 such that $\gamma_1 \in O_1$ and $\gamma_2 \in O_2$. If γ_1 and γ_2 belong to the same set Γ_j for some $j \in 1,\ldots,M$, then the assertion follows from the Hausdorffness property of $\mathcal{T}_j$. Without loss of generality, assume that $\gamma_1 \in \Gamma_1$ and $\gamma_2 \in \Gamma_2$. The choice $O_1 = \Gamma_1$ and $O_2 = \Gamma_2$ completes the proof. ■

Theorem 12 *Assume that $\Gamma_1,\ldots,\Gamma_M$ are compact Hausdorff spaces and the matrices H_j are as defined previously in this section. Furthermore, assume that for $j = 1,\ldots,M$ the functions $h_j(\gamma,\alpha) : \Gamma_j \to \mathbb{R}^l$ defined by $h_j(\gamma,\alpha) := \alpha^T H_j(\gamma)\alpha$ are continuous in the first parameter. Let $\tilde{\alpha}$ and $\tilde{t}$ be a solution to the semi-infinite programming problem $P(\Gamma_1,\ldots,\Gamma_M)$, which is defined as*

$$P(\Gamma_1,...,\Gamma_M) := \begin{array}{ll} \min_{\alpha,t} & t \\ s.t. & t \geq \alpha^T H_1(\gamma)\alpha \ \ \textit{for all} \ \ \gamma \in \Gamma_1 \\ & \vdots \\ & t \geq \alpha^T H_M(\gamma)\alpha \ \ \textit{for all} \ \ \gamma \in \Gamma_M \\ & 0 \leq \alpha \leq C \\ & \alpha^T y = 0 \\ & \alpha^T e = 2. \end{array} \tag{17}$$

Define the set $\Gamma_j^H(\alpha) := \left\{\gamma \in \Gamma_j : \alpha^T H_j(\gamma)\alpha = max_{\gamma\in\Gamma_j}\alpha^T H_j(\gamma)\alpha\right\}$. *Let (Q) be the QCQP problem that is obtained by replacing every* Γ_j *with* $j = 1,...,M$ *by* $\Gamma_j^H(\tilde{\alpha}) \equiv \left\{\gamma_1^j,...,\gamma_{m_j}^j\right\}$ *in (17). For any* $j \in \{1,...,M\}$ *let* $\tilde{\mu}_1^j,...,\tilde{\mu}_{m_j}^j$ *be a set of Lagrange multipliers associated with the constraints* $t \geq \alpha^T H_j(\gamma_i^j)\alpha,\ \ 1 \leq i \leq m_j$ *which optimize the dual problem of (Q). If* $\tilde{p}$ *is the discrete probability measure defined by* $\tilde{p}(\gamma_i^j) := \tilde{\mu}_i^j \ \ i = 1,...,m_j \ \ j = 1,...,M$, *then the pair* $(\tilde{\alpha},\tilde{p})$ *solves the problem (16). In addition, there exists a solution pair* $(\tilde{\alpha}^*,\tilde{p}^*)$ *such that* $\tilde{p}^*$ *contains at most* $l+1$ *nonzero atoms.*

Proof The result is immediately obtained by replacing the set Γ_β by Γ_0 in Theorem 7. ■

4.6 Automatic Adjustment of the Parameter τ

In this section, we consider the following problem:

$$\max_{0\leq\tau\leq 1, p\in\mathcal{P}(\mathbb{R}^n)} \min_{\alpha\in\mathcal{A}} \int_{\mathbb{R}^n} \alpha^T H(\gamma)\alpha \, dP(\gamma). \tag{18}$$

It is well known that the parameter τ can be envisioned as the weight of the kernel $\delta(x,z)$, where δ is the Kronecker delta function. So, the problem of learning the parameter τ is equivalent to choosing the best convex combination of the set of translation invariant kernels augmented with the delta kernel $\delta(x,z)$. By using Theorem 12 we get the following corollary.

Corollary 13 *Let* $\tilde{\alpha}$ *and* $\tilde{t}$ *be a solution to the semi-infinite programming problem* $P_\tau(\Gamma_\beta)$, *where for each compact set* Γ *the problem* $P_\tau(\Gamma)$ *is defined by (19). Define the set*

$$\Gamma_\beta^G(\alpha) := \left\{\gamma \in \Gamma_\beta : \alpha^T G(\gamma)\alpha = max_{\gamma\in\Gamma_\beta}\alpha^T G(\gamma)\alpha\right\}.$$

Let (Q) be the QCQP problem that is obtained by replacing Γ *by* $\Gamma_\beta^G(\tilde{\alpha}) \equiv \{\gamma_1,...,\gamma_m\}$ *in (19) and let* $\tilde{\mu}_1,...,\tilde{\mu}_m$ *be a set of Lagrange multipliers associated with the constraints* $t \geq \alpha^T G(\gamma_i)\alpha,\ \ 1 \leq i \leq m$ *which optimize the dual problem of (Q). In addition, let* $\tilde{\mu}_0$ *be a Lagrange multiplier associated with the constraint* $t \geq \alpha^T\alpha$ *in the dual problem of (Q). If* $\tilde{p}$ *is the discrete probability measure defined by* $\tilde{p}(\gamma_i) := \tilde{\mu}_i \ \ i = 1,...,m$ *and* $\tilde{\tau} := \tilde{\mu}_0$, *then* $\tilde{\alpha}$, $\tilde{\tau}$, *and* $\tilde{p}$ *solve the problem (18). In addition, there exist some solution* $\tilde{\alpha}^*$, $\tilde{\tau}^*$, *and* $\tilde{p}^*$ *such that* $\tilde{p}^*$ *contains at most* $l+1$ *nonzero atoms.*

$$P_\tau(\Gamma) := \begin{array}{ll} \min_{\alpha,t} & t \\ s.t. & t \geq \alpha^T G(\gamma)\alpha \quad \textit{for all} \quad \gamma \in \Gamma \\ & t \geq \alpha^T \alpha \\ & 0 \leq \alpha \leq C \\ & \alpha^T y = 0 \\ & \alpha^T e = 2. \end{array} \tag{19}$$

Proof Let $\omega_0 \notin \Gamma_\beta$ and assign the kernel $\delta(x,z)$ to this point. The corollary is proved by applying theorem 12 to the sets Γ_β and $\{\omega_0\}$. ■

4.7 Furnishing the Class of Admissible Kernels with Isotropic Gaussian Kernels

Although the class of translation invariant kernels encompasses the class of Gaussian kernels with arbitrary covariance matrices, the stabilized class of translation invariant kernels $\mathcal{K}_\beta$ does not. So, it may be advantageous to combine the class $\mathcal{K}_\beta$ with the class of Gaussian kernels. In this section, we consider learning the best convex combination of kernels of the classes $\mathcal{K}_\beta$ and the stabilized class of isotropic Gaussian kernels

$$\mathcal{K}_\eta := \left\{ k(x,z) = \int_{\mathbb{R}} e^{-\omega\|x_u - x_v\|^2} e^{-\eta\|\omega\|^2} \, dp(\omega) \quad : \text{ p is a probablity measure on } \mathbb{R} \right\}$$

where $\eta > 0$. We also learn the parameter τ automatically. The proof that there exists some compact set $\Omega_\eta \subseteq R$ where we can confine the integration to it parallels the discussion of Section 4.3 and is omitted. By using Theorem 12, it follows that the expansion of the optimal kernel along with the weight of each kernel can be obtained by solving the SIP problem $P_\tau(\Gamma_\beta, \Omega_\eta)$, where $P_\tau(\Gamma, \Omega)$ is defined as:

$$P_\tau(\Gamma,\Omega) := \begin{array}{ll} \min_{\alpha,t} & t \\ s.t. & t \geq \alpha^T G(\gamma)\alpha \quad \text{for all} \quad \gamma \in \Gamma \\ & t \geq \sum_{u=1}^{l}\sum_{v=1}^{l} \alpha_u \alpha_v y_u y_v e^{-\omega\|x-z\|^2} e^{-\eta\|\omega\|^2} \quad \text{for all} \quad \omega \in \Omega \\ & t \geq \alpha^T \alpha \\ & 0 \leq \alpha \leq C \\ & \alpha^T y = 0 \\ & \alpha^T e = 2. \end{array}$$

5. Optimization Algorithm

We now turn to the problem of numerically solving the nonlinear convex semi-infinite programming problem $P_\tau(\Gamma_\beta)$.[7] The term semi-infinite stems from the fact that whilst the number of variables is

7. Modifying the proposed algorithm to solve problem $P_\tau(\Gamma,\Omega)$ is straightforward.

finite, there is an infinite number of constraints which are indexed by the compact set Γ_β. Hopefully, for each finite set $\Gamma \subseteq \Gamma_\beta$, the problem $P_\tau(\Gamma)$ is QCQP and therefore convex. Hence, in principle, one can construct a sequence of QCQP problems with an increasing number of constraints such that their solutions converge to the solution of problem $P_\tau(\Gamma_\beta)$. This is the principle used by discretization and exchange algorithms (see Hettich and Kortanek, 1993; Reemtsen and Görner, 1998). On the other hand, by Corollary 13, only a finite number of constraints will be active in a solution. Furthermore, it is easy to show that this property is not limited to a solution point, and the active constraints at a solution also identify the active constraints in a neighborhood of it. This is the principle behind the methods based on local reduction (see Reemtsen and Görner, 1998; Hettich and Kortanek, 1993). Reemtsen and Görner (1998) proposed that to further speed up the methods based on local reduction, the set of active constraints be locally adapted. Combining these ideas with numerous experiments, we arrived at Algorithm 1. This algorithm is very similar to Algorithm 7 in Reemtsen and Görner (1998) which is based on local reduction.

5.1 Choosing the Initial Value of α

We choose the initial value of α such that maximizing the criterion (3) with respect to the kernel function k and the parameter τ correspond to maximizing the distance of the means of the two classes in a feature space. Let us first write the distance between means of two classes in the feature space of some kernel k'

$$\|m_1 - m_2\|^2 = \left\| \frac{1}{l_1} \sum_{u \in C^+} \Phi(x_u) - \frac{1}{l_2} \sum_{v \in C^-} \Phi(x_v) \right\|^2$$

$$= \frac{1}{l_1^2} \sum_{u \in C^+} \sum_{v \in C^-} k'(x_u, x_v) + 2\frac{1}{l_1 l_2} \sum_{u \in C^+} \sum_{v \in C^-} k'(x_u, x_v) + \frac{1}{l_2^2} \sum_{u \in C^-} \sum_{v \in C^-} k'(x_u, x_v).$$

By choosing $\alpha_i = 1/l_1$ for $i \in C^+$ and $\alpha_i = 1/l_2$ for $i \in C^-$, we have

$$\|m_1 - m_2\|^2 = \sum_{u=1}^{l} \sum_{v=1}^{l} \alpha_u \alpha_v y_u y_v k'(x_u, x_v).$$

Comparing the above equation with (3), we see that maximizing the criterion (3) with respect to the kernel function k and the parameter τ is equivalent to maximizing the distance between means of the samples of the two classes in the feature space of kernel $k'(x,z) = k(x,z) + \tau\delta(x,z)$. Note that this choice for α also satisfies the required conditions $\alpha^T e = 2$, $\alpha^T y = 0$, and $\max\{\frac{1}{l_1}, \frac{1}{l_2}\} \leq \alpha \leq C$; and thus $\alpha \in \mathcal{A}$.

5.2 Global Search for Local Maxima of $\alpha^T G(\gamma)\alpha$

The algorithm presented in this section attempts to gather a subset of unsatisfied constraints to be considered in the next iteration of Algorithm 1. Although at the solution point $\tilde{\alpha}$ the set of active constraints globally maximize $\tilde{\alpha}^T G(\gamma)\tilde{\alpha}$, for other choices of α it is possible that the constraints be violated by the local maxima of the function $\alpha^T G(\gamma)\alpha$. So, in Algorithm 2, we try to find the

values of γ which locally maximize the function $\alpha^T G(\gamma)\alpha$ for given α. Here we also assume that the function $G_\beta(\|\gamma\|)$ is differentiable with respect to γ. The choice of limited-memory BFGS algorithm (Nocedal, 1980; Liu and Nocedal, 1989; Nocedal and Wright, 2006) for this optimization is very important. For large-scale problems with large n, the memory needed to store the Hessian matrix and the associated computations become prohibitive. Although a gradient-ascent algorithm does not compute the Hessian matrix, its convergence rate is very slow. The limited-memory BFGS algorithm provides an excellent practical compromise between the computations at each step and the number of iterations till convergence, without storing the full Hessian matrix in memory.

Algorithm 1 General Optimization Algorithm

Require: T_1

1. $\Gamma^{(0)} \leftarrow \{\}$, $t^{(0)} \leftarrow \frac{1}{l_1} + \frac{1}{l_2}$
   {A lower bound for parameter t is the minimum value of $\alpha^T\alpha$ for $\alpha \in \mathcal{A}$}
2. Initialize $\alpha^{(0)}$ as described in Section 5.1
3. **for** $i = 1, 2, \ldots$ **do**
4. set R such that for all γ with $\|\gamma\| > R$, the relation $\alpha^T G(\gamma)\alpha < t^{(i-1)}$ holds for all α
5. $\Gamma_g^{(i)} \leftarrow GlobalSearchForLocals(\alpha^{(i-1)}, R)$
   {denote the maximum value obtained by the global search by $s^{(i)}$}
6. $\Gamma_s^{(i)} \leftarrow \Gamma^{(i-1)} \bigcup \Gamma_g^{(i)}$
7. Solve problem $P_\tau(\Gamma_s^{(i)})$ to obtain the optimal parameters $t_s^{(i)}, \alpha_s^{(i)}$ and $\mu_s^{(i)}$
   {see Section 5.3}
8. Locally adapt $\Gamma_s^{(i)}$ and $\mu_s^{(i)}$ to obtain the optimal parameters $t_l^{(i)}, \alpha_l^{(i)}, \Gamma_l^{(i)}$ and $\mu_l^{(i)}$
   {see Section 5.4}
9. $t^{(i)} \leftarrow t_l^{(i)}$, $\alpha^{(i)} \leftarrow \alpha_l^{(i)}$
10. Construct $\mu^{(i)}$ and $\Gamma^{(i)}$ by eliminating zero indices of $\mu_l^{(i)}$ along with the corresponding vectors in $\Gamma_l^{(i)}$
11. **if** $t^{(i)} - t^{(i-1)} < \varepsilon t^{(i-1)}$ or $i = T_1$ **then**
12. terminate algorithm with the kernel $k(x,z) := \sum_{j=1}^m \mu_j^{(i)} \cos\left((x-z)^T \gamma_j^{(i)}\right)$
    {we assume that $\Gamma^{(i)} = \left\{\gamma_1^{(i)}, \ldots, \gamma_m^{(i)}\right\}$ and that $\mu_j^{(i)}$ is the Lagrange multiplier associated with the constraint $\alpha^T G(\gamma_j^{(i)})\alpha \le t$ in problem $P_\tau(\Gamma^{(i)})$}
13. **end if**
14. **end for**

5.3 Solving the Problem $P_\tau(\Gamma)$ for Finite Set Γ

Let $\Gamma = \{\gamma_1, \ldots, \gamma_m\}$ be a finite subset of Γ_β. Since Γ is finite, the problem $P_\tau(\Gamma)$ can be written as the following QCQP problem:

Algorithm 2 GlobalSearchForLocals

Require: α, R, T_2, and T_3

1. $\Gamma = \{\}, i = 0$
2. **for** $j = 1, 2, \ldots$ **do**
3. generate a random point $x \in \mathbb{R}^n$
4. generate a random number $r \in [0, R]$
5. $\gamma_0 \leftarrow r \frac{x}{\|x\|}$
6. starting from γ_0 and using the limited-memory BFGS algorithm find a local maximum $\gamma^{(i)}$ for function $\alpha^T G(\gamma)\alpha$
7. **if** $\gamma^{(i)} \in \Gamma$ **then**
8. $i \leftarrow i + 1$ {count the number of repeating local maxima}
9. **end if**
10. $\Gamma \leftarrow \Gamma \cup \gamma^{(i)}$
11. **if** $(i - |\Gamma|) \geq T_2$ or $j = T_3$ **then**
12. **return** Γ
13. **end if**
14. **end for**

$$\begin{aligned} \min_{\alpha,t} \quad & t \\ s.t. \quad & t \geq \alpha^T G(\gamma_i)\alpha \quad \text{for all} \quad i = 1, \ldots, m \\ & t \geq \alpha^T \alpha \\ & 0 \leq \alpha \leq C \\ & \alpha^T y = 0 \\ & \alpha^T e = 2. \end{aligned} \tag{20}$$

This problem has been studied in Section 4.6 of Lanckriet et al. (2004) and it has been suggested to store the $l \times l$ kernel matrices $G(\gamma_1), \ldots, G(\gamma_m)$ in memory and solve the problem with general purpose software packages. But, the memory requirement of this approach limits its applicability to small-sized problems. However, the facts that

$$\alpha^T \Im \{G(\gamma)\} \alpha = 0$$

and

$$\begin{aligned} \Re \{G(\gamma)\} &= v_c(\gamma)^T v_c(\gamma) + v_s(\gamma)^T v_s(\gamma), \\ v_c(\gamma) &:= \left[y_1 cos(\gamma^T x_1),\ y_2 cos(\gamma^T x_2),\ \ldots\ , y_l cos(\gamma^T x_l) \right], \\ v_s(\gamma) &:= \left[y_1 sin(\gamma^T x_1),\ y_2 sin(\gamma^T x_2),\ \ldots\ , y_l sin(\gamma^T x_l) \right] \end{aligned}$$

where $\Re\{z\}$ and $\Im\{z\}$ are the real and imaginary parts of z, respectively, show that the kernel matrices $G(\gamma_1),...,G(\gamma_m)$ appearing in problem (20) are effectively[8] of rank two. This allows us to reformulate (20) as a new QCQP problem as follows:

$$\begin{array}{ll} \min_{\alpha,t,c,s} & t \\ s.t. & t \geq c_i^2 + s_i^2 \quad i = 1,...,m \\ & c_i = \sum_{u=1}^{l} \alpha_i y_i \cos(\gamma_i^T x_u) \quad i = 1,...,m \\ & s_i = \sum_{u=1}^{l} \alpha_i y_i \sin(\gamma_i^T x_u) \quad i = 1,...,m \\ & t \geq \alpha^T \alpha \\ & 0 \leq \alpha \leq C \\ & \alpha^T y = 0 \\ & \alpha^T e = 2. \end{array} \tag{21}$$

Now, there is no need to load the kernel matrices into memory and so general-purpose QCQP solvers such as Mosek (Andersen and Andersen, 2000) can be used to solve (21) even when the training set size is huge.

5.4 Local Adaptation

As stated in the previous section, for any finite set $\Gamma = \{\gamma_1,...,\gamma_m\} \subseteq \Gamma_\beta$, the problem $P_\tau(\Gamma)$ is convex and so every local solution is also globally optimal. But, if we consider the values $\gamma_1,...,\gamma_m$ as points in the space $\mathbb{R}^n$, we get the following non-convex optimization problem:

$$\max_{\substack{\mu \geq 0, \mu^T e = 1, \\ \gamma_1,...,\gamma_m \in \mathbb{R}^n}} \min_{\alpha \in \mathcal{A}} \sum_{i=1}^{m} \mu_i \alpha^T G(\gamma_i) \alpha + \mu_0 \alpha^T \alpha. \tag{22}$$

Now, we can use the the solution of the problem $P_\tau(\Gamma)$ obtained in the previous section as the starting point for problem (22) and locally improve it by an ascent method. By unrolling (22), we obtain the following optimization problem:

$$\max_{\substack{\mu \in \mathbb{R}^{m+1}, \\ \gamma_1,...,\gamma_m \in \mathbb{R}^n}} \hat{J}(\mu, \gamma_1,...,\gamma_m) \quad s.t. \ \mu \geq 0, \mu^T e = 1 \tag{23}$$

where

8. In this paper, we say that a matrix G is effectively of rank r if there exists some matrix H of rank r such that for all vectors $\alpha \in \mathcal{A}$ we have $\alpha^T G \alpha = \alpha^T H \alpha$.

$\hat{J}(\mu,\gamma_1,...,\gamma_m) :=$

$$\min_{\alpha\in\mathcal{A}} \left\{ \sum_{u=1}^{l}\sum_{v=1}^{l} \alpha_u\alpha_v y_u y_v \sum_{i=1}^{m} \mu_i G_\beta(\|\gamma_i\|)\cos(\gamma_i^T(x_u-x_v)) + \mu_0 \sum_{u=1}^{l} \alpha_u^2 \right\}. \tag{24}$$

Problem (23), corresponds to adapting the kernel parameters $\gamma_1,...,\gamma_m$ and $\mu_0,...,\mu_m$ of the kernel function $k(x,z) = \sum_{i=1}^{m} \mu_i \cos(\gamma_i^T(x-z)) + \mu_0\delta(x-z)$ for the task of SVM classification, where δ denotes the Kronecker delta function. This problem has been previously studied by Chapelle et al. (2002) for general kernel functions with unconstrained parameters and they proved that the function $\hat{J}(.)$ is differentiable provided that problem (24) has a unique solution.[9] They also proposed a simple gradient-based iterative algorithm for adapting the kernel parameters. Recently, Rakotomamonjy et al. (2008) performed a more detailed analysis of this problem in MKL and proposed a reduced gradient algorithm with line search. They reasoned that since the computation of the function $\hat{J}$ is costly,[10] the overhead of a line search preserves the effort.

To avoid the difficulties of the constrained optimization, we replace the constrained vector μ by the unconstrained vector ρ, connected by the relation $\mu_i = \frac{\rho_i^2}{\rho^T\rho}$, $i = 0,...m$, and rewrite (23) as the following problem:

$$\max_{\substack{\rho\in\mathbb{R}^{m+1},\\ \gamma_1,...,\gamma_m\in\mathbb{R}^n}} \quad J(\rho,\gamma_1,...,\gamma_m) \tag{25}$$

where

$J(\rho,\gamma_1,...,\gamma_m) :=$

$$\min_{\alpha\in\mathcal{A}} \frac{1}{\rho^T\rho} \left\{ \sum_{u=1}^{l}\sum_{v=1}^{l} \alpha_u\alpha_v y_u y_v \sum_{i=1}^{m} \rho_i^2 G_\beta(\|\gamma_i\|)\cos(\gamma_i^T(x_u-x_v)) + \rho_0^2 \sum_{u=1}^{l} \alpha_u^2 \right\}. \tag{26}$$

We use the limited-memory BFGS algorithm to numerically solve (25). Our experiments on MKL tasks show that the method proposed in this section is several times fatser than the reduced gradient algorithm of Rakotomamonjy et al. (2008).[11] It has been also stated by Rakotomamonjy et al. (2008) that their method could be improved if the Hessian matrix could be computed efficiently. This is not the case for the problem (25) with $m\times(n+1)$ variables; where, even for moderate size problems, the storage of the Hessian matrix requires lots of memory.

5.5 Solving the Intermediate SVM Problem and Its Gradient

To compute the function $J(.)$ defined by Equation (26), we have to solve the following constrained quadratic programming problem:

9. Truely speaking, the proof should be credited to Danskin (1966).
10. Although the definition of the function J in Rakotomamonjy et al. (2008) differs from (24), computation of both functions corresponds to training a single-kernel SVM.
11. We leave this comparison along with some theoretical results to another paper.

$$\begin{aligned} \min_{\alpha} \quad & \alpha^T \left(\sum_{i=1}^{m} \frac{\rho_i^2}{\rho^T \rho} G(\gamma_i) \right) \alpha + \frac{\rho_0^2}{\rho^T \rho} \alpha^T \alpha \\ s.t. \quad & 0 \leq \alpha \leq C \\ & \alpha^T y = 0 \\ & \alpha^T e = 2. \end{aligned} \tag{27}$$

Although the traditional algorithms for solving quadratic programming problems, such as active set methods, are fast, they need to store the kernel matrix in memory which prevents their application in large-scale problems. So, various algorithms for large-scale training of SVMs, such as SMO (Platt, 1999) or SVM^{Light} (Joachims, 1999), have been proposed. Again, since the effective rank of the kernels $G(\gamma_1), ..., G(\gamma_m)$ is two, we can re-state the problem (27) in a memory efficient manner as:

$$\begin{aligned} \min_{\alpha,c,s} \quad & \frac{\rho_0^2}{\rho^T \rho} \alpha^T \alpha + \sum_{i=1}^{m} \frac{\rho_i^2}{\rho^T \rho} \left(c_i^2 + s_i^2 \right) \\ s.t. \quad & c_i = \sum_{u=1}^{l} \alpha_u y_u \cos(\gamma_i^T x_u) \quad i = 1, ..., m \\ & s_i = \sum_{u=1}^{l} \alpha_u y_u \sin(\gamma_i^T x_u) \quad i = 1, ..., m \\ & 0 \leq \alpha \leq C \\ & \alpha^T y = 0 \\ & \alpha^T e = 2. \end{aligned}$$

In our experiments we have used the optimization software Mosek (Andersen and Andersen, 2000) to solve this problem. After computing the value of the function $J(.)$ and obtaining a solution $\tilde{\alpha}$ to (26), we compute the gradient using the following formulas:

$$\nabla_{\gamma_j} J = \frac{\rho_j^2}{\rho^T \rho} \nabla_{\gamma} \left\{ \tilde{\alpha}^T G(\gamma) \tilde{\alpha} \right\} |_{\gamma = \gamma_j} \quad j = 1, ..., m, \tag{28}$$

$$\nabla_{\rho_j} J = 2 \frac{\rho_j}{\rho^T \rho} \left\{ \tilde{\alpha}^T G(\gamma_j) \tilde{\alpha} - \sum_{i=1}^{m} \frac{\rho_i^2}{(\rho^T \rho)} \tilde{\alpha}^T G(\gamma_i) \tilde{\alpha} - \frac{\rho_0^2}{(\rho^T \rho)} \tilde{\alpha}^T \tilde{\alpha} \right\} \quad j = 1, ..., m. \tag{29}$$

Note that, in general, the computational complexity of computing formulas (28) and (29) is $O(m \times n \times nsv^2)$ as was pointed out by Rakotomamonjy et al. (2008).[12] The following formulas show an $O(m \times (nsv + n))$ method for computing the gradient of function $J(.)$.

$$\nabla_{\gamma_j} J = \frac{\rho_j^2}{\rho^T \rho} \left(c_j^2 + s_j^2 \right) \nabla_{\gamma} G_{\beta}(\|\gamma\|)|_{\gamma = \gamma_j} + 2 \frac{\rho_j^2}{\rho^T \rho} \left(s_j' s_j + c_j' c_j \right) G_{\beta}(\|\gamma_j\|) \quad j = 1, ..., m,$$

$$\nabla_{\rho_j} J = 2 \frac{\rho_j}{\rho^T \rho} \left\{ (c_j^2 + s_j^2) G_{\beta}(\|\gamma_j\|) - \sum_{i=1}^{m} \frac{\rho_i^2}{\rho^T \rho} (c_j^2 + s_j^2) G_{\beta}(\|\gamma_j\|) - \frac{\rho_0^2}{\rho^T \rho} \tilde{\alpha}^T \tilde{\alpha} \right\} \quad j = 1, ..., m$$

where

12. Note that in Rakotomamonjy et al. (2008) the function $J(.)$ has only m variables, while here the number of variables is $m \times (n+1)$.

$$c_j = \sum_{u=1}^{nsv} \tilde{\alpha}_u y_u \cos(\gamma_j^T x_u) \quad j = 1, ..., m,$$
$$s_j = \sum_{u=1}^{nsv} \tilde{\alpha}_u y_u \sin(\gamma_j^T x_u) \quad j = 1, ..., m,$$
$$c'_j = -\sum_{u=1}^{nsv} \tilde{\alpha}_u y_u x_u \sin(\gamma_j^T x_u) \quad j = 1, ..., m,$$
$$s'_j = \sum_{u=1}^{nsv} \tilde{\alpha}_u y_u x_u \cos(\gamma_j^T x_u) \quad j = 1, ..., m.$$

5.6 Convergence Analysis

In this section, we study the convergence properties of Algorithm 1. We hope the contents of this section help the reader to get a better feeling of this algorithm. For any finite or infinite set Γ, we denote the solution of problem $P_\tau(\Gamma)$ by $S(\Gamma)$. Let us first prove a useful lemma.

Lemma 14 *If the loop inside Algorithm 1 is executed for the i'th iteration, then $S(\Gamma_l^{(i)}) = S(\Gamma^{(i)})$. In other words, removing the constraints where their associated Lagrange multipliers are zero, does not change $S(\Gamma_l^{(i)})$.*

Proof Assume that $\Gamma_l^{(i)} = \{\gamma_1, ..., \gamma_m\}$. Without loss of generality, we assume that $\Gamma^{(i)} = \{\gamma_1, ..., \gamma_{m'}\}$, where $m' < m$. Denote the Lagrangian of $P_\tau(\Gamma_l^{(i)})$ by $\mathcal{L}(\alpha, t, \mu, \lambda)$, where $\mu_1, ..., \mu_m$ are the Lagrange multipliers associated with the constraints $\alpha^T G(\gamma_1)\alpha \leq t$, ...,$\alpha^T G(\gamma_m)\alpha \leq t$, respectively, and $\lambda \in \Lambda$ denotes the Lagrange multipliers associated with all other constraints. Since $P_\tau(\Gamma_l^{(i)})$ is convex, by the strong duality we have

$$S(\Gamma_l^{(i)}) = \max_{\mu \geq 0, \lambda \in \Lambda} \min_{\alpha, t} \mathcal{L}(\alpha, t, \lambda, \mu) = \mathcal{L}(\alpha^*, t^*, \lambda^*, \mu^*)$$

where it is assumed that α^*, t^*, λ^*, and μ^* are a solution to problem $S(\Gamma_l^{(i)})$. Since $\mu^*_{m'+1}, ..., \mu^*_m$ are zero, we also have

$$S(\Gamma_l^{(i)}) = \max_{\mu_1 \geq 0, ..., \mu_{m'} \geq 0, \lambda \in \Lambda} \min_{\alpha, t} \mathcal{L}(\alpha, t, \lambda, \mu).$$

Since the strong duality also holds for $P_\tau(\Gamma^{(i)})$, the last expression is equal to $S(\Gamma^{(i)})$ and the lemma follows. ■

Now, we prove that for any $\varepsilon > 0$ the Algorithm 1 converges, even without limiting the maximum number of iterations.

Proposition 15 *The sequence of numbers $t^{(0)}, t^{(1)}, ...$ generated by Algorithm 1 is increasing and bounded.*

Proof By steps 6 and 8 of Algorithm 1, it follows that $S(\Gamma^{(i-1)}) \leq S(\Gamma_s^{(i)})$ and $S(\Gamma_s^{(i)}) \leq S(\Gamma_l^{(i)})$. By Lemma 14, $S(\Gamma_l^{(i)}) = S(\Gamma^{(i)})$. So, $t^{(i-1)} = S(\Gamma^{(i-1)}) \leq S(\Gamma^{(i)}) = t^{(i)}$. By Equation (5) and the fact that $\alpha^T e = 2$, it follows that $\alpha^T \alpha \leq 4$ and $\alpha^T G(\gamma)\alpha \leq 4$ for all choices of α and γ; and thus, the sequence is bounded. ■

Define $g(\alpha,\gamma) := \alpha^T G(\gamma)\alpha$. The following theorem is essentially Theorem 7.2 of Hettich and Kortanek (1993), where its proof is reconstructed here for the sake of completeness.

Theorem 16 *Assume that in every run of step 5 of Algorithm 1 at least one global maximizer of the function $g(\alpha^{(i-1)},\gamma)$ is found and that the steps 10-13 of the algorithm are omitted (Note that the key point is the omission of step 10). Let $\bar{\alpha}$ be any accumulation point of the sequence $\alpha^{(0)},\alpha^{(1)},...$ and assume that $t^{(i)} \nearrow \bar{t}$. Then the pair $(\bar{\alpha},\bar{t})$ is a solution of $P_\tau(\Gamma_\beta)$.*

Proof First note that since $\alpha^{(i)} \in \mathcal{A}$ and $\mathcal{A}$ is compact, a point of accumulation for the sequence $\alpha^{(0)},\alpha^{(1)},...$ always exists. Recall the definition of Ω_β from Section 4.6 and define the function $g(\alpha)$ as:

$$g(\alpha) := \max_{\gamma\in\Omega_\beta} \alpha^T G(\gamma)\alpha$$

For simplicity, assume that $\alpha^{(i)} \to \bar{\alpha}$. Let (α^*,t^*) denote a solution of problem $P_\tau(\Gamma_\beta)$. Clearly $\bar{t} \leq t^*$. If $\bar{t} = t^*$ then the theorem is proved. Assume on the contrary that $\bar{t} < t^*$. Then, there exists $\bar{\gamma} \in \Omega_\beta$ such that $\bar{t} < g(\bar{\alpha},\bar{\gamma}) = g(\bar{\alpha})$. But, since $\bar{\alpha}^T\bar{\alpha} \leq \bar{t}$, it follows that $\bar{\gamma} \in \Gamma_\beta$. For $i = 0,1,...$ choose $\gamma^{(i)} \in \Omega_\beta$ such that $g(\alpha^{(i)},\gamma^{(i)}) = g(\alpha^{(i)})$. We have

$$g(\bar{\alpha}) - \bar{t} = \left[g(\alpha^{(i)}) - \bar{t}\right] + \left[g(\bar{\alpha}) - g(\alpha^{(i)})\right] = \left[g(\alpha^{(i)},\gamma^{(i)}) - \bar{t}\right] + \left[g(\bar{\alpha}) - g(\alpha^{(i)})\right] \tag{30}$$

On the other hand, by omission of step 10 from Algorithm 1, all constraints of the previous iterations will continue to appear in the next iterations. Since $g(\alpha,\gamma)$ is continuous, $\bar{\alpha}$ is a feasible point for all problems $P_\tau(\Gamma^{(i)})$, where $i = 0,1,...,\infty$. Therefore,

$$g(\bar{\alpha},\gamma^{(i)}) \geq \sup_{j=1,...,\infty} t^{(j)} = \bar{t} \quad \text{for all } i = 0,1,... \tag{31}$$

Using (31) in (30) we obtain

$$\begin{aligned} g(\bar{\alpha}) - \bar{t} &= \left[g(\alpha^{(i)},\gamma^{(i)}) - \bar{t}\right] + \left[g(\bar{\alpha}) - g(\alpha^{(i)})\right] \\ &\leq \left[g(\alpha^{(i)},\gamma^{(i)}) - g(\bar{\alpha},\gamma^{(i)})\right] + \left[g(\bar{\alpha}) - g(\alpha^{(i)})\right]. \end{aligned} \tag{32}$$

By continuity of $g(.,.)$, the right hand side of (32) tends to zero, which contradicts the assumption $\bar{t} < g(\bar{\alpha})$. ■

6. Generalizing the Kernel Trick to Complex-valued Kernels

Consider a machine learning algorithm designed for a real-valued Euclidean space. The kernel trick for real-valued kernels states that if all geometric concepts of an algorithm are defined solely based on the dot-product operation, then by replacing all of these dot-products by a kernel function k, we arrive at a version of the very algorithm running in a feature space associated with kernel k. For complex-valued kernels, the dot-product of any two vectors may be complex-valued which makes the application of these kernels to machine learning algorithms more tricky. We now introduce a generalization of the kernel trick for complex-valued kernels. Assume that $k(x,z)$ is a complex-valued kernel. Then there exists at least one complex feature space, say F, and a mapping $\Phi : X \to F$ such that $k(x,z) = \langle \Phi(x), \Phi(z) \rangle_F$. Each axis in the complex feature space F can be substituted by two real-valued axes, one representing the real part and the other the imaginary part. Let us call this real-valued space G. To use the kernel trick, we replace the complex feature space F with the equivalent real feature space G. Now, we show that the dot product between elements of G can be computed by the real-valued kernel function $\Re\{k(x,z)\}$,[13] where $\Re\{z\}$ is the real part of z.

Theorem 17 *Let F be a complex Hilbert space of dimension N (possibly infinite) and G the corresponding $2N$-dimensional Hilbert space obtained by representing real and imaginary parts of F in separate real axes. Then $\langle x', z' \rangle_G = \Re\{\langle x, z \rangle_F\}$, where x' is the $2N$-dimensional vector obtained by concatenating real and imaginary parts of x.*

Proof For finite N we have

$$\begin{aligned}\Re\{\langle x,z\rangle_F\} &= \Re \sum_{i=1}^{N} x_i \overline{z_i} = \Re \sum_{i=1}^{N} (x_i^{re} + j x_i^{im})(z_i^{re} - j z_i^{im}) \\ &= \sum_{i=1}^{N} (x_i^{re} z_i^{re} + x_i^{im} z_i^{im}) = \sum_{i=1}^{2N} x'_i z'_i = \langle x', z' \rangle_G\end{aligned}$$

If F is infinite dimensional, then it has an orthonormal basis (see Kreyszig, 1989 p.168) and x and z can have at most countably many nonzero elements (see Kreyszig, 1989 p.165) which we indicate by index set $\mathfrak{I}$. So,

$$\begin{aligned}\Re\{\langle x,z\rangle_F\} &= \Re \sum_{i\in\mathfrak{I}} x_i \overline{z_i} = \Re \sum_{i\in\mathfrak{I}} (x_i^{re} + j x_i^{im})(z_i^{re} - j z_i^{im}) \\ &= \sum_{i\in\mathfrak{I}} (x_i^{re} z_i^{re} + x_i^{im} z_i^{im}) = \sum_{i\in\mathfrak{I}} x'_i z'_i = \langle x', z' \rangle_G.\end{aligned}$$

■

After fully developing the paper based on the complex-valued form of translation invariant kernels, one of the reviewers introduced us to the real-valued form of these kernels as was discovered by Bochner (1955). He proved that every continuous real-valued translation invariant positive definite kernel in $\mathbb{R}^n$ has the general form

13. The fact that real part of a complex kernel is a real kernel is not new (see Schölkopf and Smola, 2002 page 31). But, as far as we know, the relation between the corresponding Hilbert spaces, as stated in the theorem, is new.

$$k(x,z) = \tilde{k}(x-z) = \int_{\mathbb{R}^n} cos\gamma^T (x-z)\, dV(\gamma).$$

It is interesting that after applying the appropriate kernel trick to both real-valued and complex-valued forms of translation invariant kernels, the optimal kernel is found to be a mixture of cosines.

7. The Method at Runtime

One frequently denounced feature of SVMs is that the resulting classifier has an expansion based on support vectors. Although support vectors are considered to be sparse in the training set, the resulting classifier is usually slower than other competing methods such as neural networks (Schölkopf et al., 1998). In general, the computation of a support vector classifier requires $O(n \times nsv)$ steps. The problem becomes more severe when the kernel function becomes a combination of several kernels, where the computational complexity of evaluating the classifier grows up to $O(m \times n \times nsv)$. For some kernels, such as the Gaussian kernel with isotropic covariance matrix, the computation time can be reduced to $O((m+n) \times nsv)$. Our method has the eminent property that the resulting classifier is not expanded based on support vectors at all. Considering the SVM classifier $f(x) = \sum_{u=1}^{nsv} \alpha_u y_u k(x, x_u) + b$, we have

$$f(x) = \sum_{u=1}^{nsv} \alpha_u y_u k(x,x_u) + b = \sum_{u=1}^{nsv} \alpha_u y_u \left(\sum_{i=1}^{m} \mu_i \cos(\gamma_i^T (x - x_u)) G_\beta(\|\gamma_i\|) \right) + b$$

$$= \sum_{i=1}^{m} \mu_i G_\beta(\|\gamma_i\|) \sum_{u=1}^{nsv} \alpha_u y_u \left(\cos(\gamma_i^T x)\cos(\gamma_i^T x_u) + \sin(\gamma_i^T x)\sin(\gamma_i^T x_u) \right) + b$$

$$= \sum_{i=1}^{m} \mu_i G_\beta(\|\gamma_i\|) \left[\left(\sum_{u=1}^{nsv} \alpha_u y_u \cos(\gamma_i^T x_u) \right) \cos(\gamma_i^T x) + \left(\sum_{u=1}^{nsv} \alpha_u y_u \sin(\gamma_i^T x_u) \right) \sin(\gamma_i^T x) \right] + b.$$

But $\sum_{u=1}^{nsv} \alpha_u y_u \cos(\gamma_i^T x_u)$ and $\sum_{u=1}^{nsv} \alpha_u y_u \sin(\gamma_i^T x_u)$ are constant values. So, the computational complexity of evaluating the classifier of the proposed method is $O(m \times n)$. Note that the classifier has an expansion based on the number of kernels, instead of support vectors. In addition, by Theorem 2, the number of kernels is limited to $l+1$. Furthermore, since the deletion of non-support vector samples from the training set has no effect on the optimal classifier, it follows that $m \leq nsv + 1$. Although, theoretically, the number of kernels can reach the number of support vectors, our experiments show that the number of kernels is usually a fraction of the number of support vectors.

8. A Learning Theory Perspective

A common feature between the class of radial kernels, considered by Micchelli and Pontil (2005), and the class of translation invariant kernels, considered here, is that the kernels of both classes

have the property that $k(x,x) = 1$ for every $x \in \mathbb{R}^n$.[14] Micchelli et al. (2005b) used this feature along with a result from Yiming and Zhou (2007) to obtain a probably approximately correct (PAC) upper bound on the generalization error of their kernel learning framework over the class of radial kernel functions. They concluded that the regularization parameter of a single-kernel learning machine is sufficient for controlling the complexity of the class of radial kernels, rejecting the use of an auxiliary method for controlling the complexity of the class of radial kernels.

But, the situation for translation invariant kernels is completely different. It is well-known that the VC-dimension of the class of cosine functions with arbitrary frequencies is infinite (see Vapnik, 1998, page 160). In addition, the finiteness of the VC-dimension is a necessary and sufficient condition for distribution independent learning of binary classification tasks (see Vapnik, 1998, Theorem 4.5). So, controlling the complexity of the class of translation invariant kernels is a necessary ingredient of our framework. This discussion will be experimentally verified in Section 9, where we will show the vital role of the complexity control mechanism of Section 2.

9. Experimental Results

In this section we report the results of our experiments on several artificial and real-world benchmark data sets. In addition, we will experimentally investigate the role of the complexity control mechanism of Section 2. In all the experiments we have set $C = \infty$, $T_1 = 1000$, $T_2 = 4$, $T_3 = 500$, $G_\beta(\|\gamma\|_2) = \exp(-\beta\|\gamma\|_2^2))$ and the parameter τ is automatically learnt according to Algorithm 1. The implementation of this paper is packaged in the SIKL (Stabilized infinite kernel learning) toolbox and is available at *http://www.mloss.org*. We obtained the implementation of the limited memory BFGS algorithm from the website *http://www.chokkan.org/software/liblbfgs* which is a C++ translation of the original implementation made available by Nocedal in Fortran 77. For limited-memory BFGS algorithm, the 17 most recent curvature information are used and the maximum number of line-search tries is set to 20. We also changed the stopping condition of the algorithm from $\frac{\|\nabla x\|}{\|x\|} < \varepsilon$ to $\|\nabla x\| < \varepsilon$ to avoid the degradation of the accuracy of the global search algorithm for points far from the origin. The QCQP sub-problem of Algorithm 1 and the QP problem of Section 5.5 are solved by the optimization software Mosek (Andersen and Andersen, 2000). All the experiments have been performed on a 2.8GHz Pentium D computer with 2GB memory and running the Linux operating system.

9.1 Experiments on Small-size Benchmark Data Sets

In this section, we report our experiments on the benchmark data sets prepared by Rätsch et al. (2001). This benchmark consists of 13 data sets and there exist 100 splits of each data set into training and test sets. The classification error for each data set is obtained by averaging the classification error over these splits. For this experiment we set $\varepsilon = 0.001$. The comparison is among the following methods:

- **Single Gaussian (SG)** Rätsch et al. (2001) performed experiments with a single isotropic Gaussian kernel. The variance parameter σ of the isotropic Gaussian kernel and the parameter C of the 1-norm soft-margin SVM are optimized by performing 5-fold cross-validation on the first five instances of the training set.

14. In fact, the classes of radial/translation invariant kernel functions contain kernels with arbitrary positive values for $k(x,x)$. But, the constraint $k(x,x) = 1$ is imposed for reasons stated in Section 3.

- **Gaussian Mixture (GM)** A generalization of the method of Gehler and Nowozin (2008) is implemented and used for learning the optimal kernel over the class $\mathcal{K}_\eta$. The number of Gaussian kernels $\bar{m}$, their parameters, and the parameter τ are learnt automatically. The parameter η is set to 0.001 and the parameter C is learnt by performing 5-fold cross validation on the first five instances of the training set.

- **Cosine Mixture (CM)** Here, the method of Section 4.6 is used. The number of cosine kernels m, their parameters, and the parameter τ are learnt automatically. The parameter C is fixed to ∞ and the parameter β is optimized by performing 5-fold cross validation on the first five instances of the training set.

- **Cosine and Gaussian Mixture (CGM)** Here, the method of Section 4.7 is used. The number of cosine kernels m, the number of Gaussian kernels $\bar{m}$, the parameters of cosine and Gaussian kernels, and the parameter τ are learnt automatically. The parameter C is set to ∞ and the parameter η is set to 0.03. The parameter β is optimized by performing 5-fold cross validation on the first five instances of the training set.

To compare the training and evaluation times of these methods, we repeated the experiments of Rätsch et al. (2001) on our machine. For training a single-kernel SVM we used the implementation of SMO algorithm (Platt, 1999) contained in the Statistical Pattern Recognition Toolbox.[15] To keep the results reported by Rätsch et al. (2001) as reference, we neglect the accuracies obtained by the SG method.

Table 2 summarizes the test error rates and training times of the methods on each data set. It can be seen that the GM method has the worst performance and does not provide any improvement over other methods. The only benefit of the GM over SG is that while the latter requires specifying the kernel function by hand, GM learns the kernel function automatically. The SG and CM are the only methods of this experiment that do not store the kernel matrices in memory; and thus are applicable to large-scale problems. In addition, they have also the best training times. To our surprise, although the CM method solves a musch more difficult problem than SG, it has also improved the training time in some data sets. Considering the test error rates, the CGM method has the best overall performance. But, the SG method on the *F.Solar* data set, and CM method on the *Thyroid* data set provide significantly better results. For the *Ringnorm* data set, the CM method has obtained a high error rate of 8.5%. Interestingly, the number of training and testing samples of the *Ringnorm* data set are exactly equal to the *Twonorm* data set, for which the CM method has even improved the accuracy. The essential difference between the *Twonorm* and the *Ringnorm* data sets, where in both data sets each class has a multivariate normal distribution, is that in the *Twonorm* data set the classes have separate means, whilst in the *Ringnorm* data set the classes have separate covariance matrices. So, it seems that the Gaussian kernel is inherently much more suitable for solving the *Ringnorm* data set than the cosine kernel. In fact, this is exactly why combining several kernels is important. By combining the cosine and Gaussian kernels, the CGM method provides the best performance.

Table 3 compares the methods in terms of the evaluation time. For each method, the factors that influence the evaluation time are also reported. As can be seen, except for the *Ringnorm* data set, the CM is significantly faster at run-time than all other methods, including a classical SVM with Gaussian kernel. The best speedup is for the *Twonorm* data set for which, in addition to a lower test

15. Available at *http://cmp.felk.cvut.cz/cmp/software/stprtool*.

Data Set	Single Gaussian		Gaussian Mixture		Cosine Mixture		Cosine& Gaussian Mixture	
	error (%)	training (*sec*)	error (%)	training (*sec*)	error (%)	training (*sec*)	error (%)	training (*sec*)
Banana	11.5 ± 0.7	**1.1**	10.5 ± 0.5	26.4	10.7 ± 0.5	3.5	$\mathbf{10.4 \pm 0.5}$	21.3
B. Cancer	26.0 ± 4.7	**0.2**	26.7 ± 5.0	2.7	26.2 ± 4.9	1.2	$\mathbf{25.8 \pm 4.7}$	4.2
Diabetis	23.5 ± 1.7	**1.6**	23.7 ± 1.7	7.5	$\mathbf{23.2 \pm 1.9}$	2.3	$\mathbf{23.2 \pm 1.8}$	16.5
F. Solar	$\mathbf{32.4 \pm 1.8}$	9.1	35.4 ± 1.7	15.4	33.3 ± 1.8	**1.2**	33.9 ± 1.8	57.7
German	$\mathbf{23.6 \pm 2.1}$	4.4	25.3 ± 2.5	35.4	24.1 ± 2.2	**3.5**	23.7 ± 2.2	53.9
Heart	16.0 ± 3.3	**0.1**	17.0 ± 3.2	1.4	$\mathbf{15.6 \pm 3.2}$	1.1	16.0 ± 3.2	2.8
Image	3.0 ± 0.6	**16.0**	3.6 ± 1.3	178.9	$\mathbf{2.5 \pm 0.5}$	1057.1	$\mathbf{2.5 \pm 0.5}$	779.6
Ringnorm	$\mathbf{1.7 \pm 0.1}$	**2.9**	$\mathbf{1.7 \pm 0.1}$	9.7	8.5 ± 0.9	192.7	$\mathbf{1.7 \pm 0.1}$	17.7
Splice	10.9 ± 0.7	445.4	11.1 ± 0.7	91.8	9.7 ± 0.4	**43.3**	$\mathbf{9.3 \pm 0.5}$	187.0
Thyroid	4.8 ± 2.2	**0.1**	4.6 ± 2.2	1.8	$\mathbf{3.7 \pm 2.2}$	3.4	4.8 ± 2.1	3.1
Titanic	$\mathbf{22.4 \pm 1.0}$	**0.1**	23.2 ± 1.3	1.0	22.9 ± 1.2	0.9	22.9 ± 1.2	2.6
Twonorm	3.0 ± 0.2	**0.4**	2.7 ± 0.2	7.9	$\mathbf{2.4 \pm 0.1}$	5.4	2.7 ± 0.2	17.9
Waveform	9.9 ± 0.4	5.8	9.8 ± 0.4	8.5	10.0 ± 0.5	**2.6**	$\mathbf{9.7 \pm 0.4}$	17.7

Table 2: Test errors and training times of SG, GM, CM, and CGM methods on the data sets collected by Rätsch et al. (2001)

Data Set	Single Gaussian		Gaussian Mixture			Cosine Mixture		Cosine & Gaussian Mixture			
	testing (*ms*)	nsv	testing (*ms*)	nsv	$\bar{\mathbf{m}}$ *Gauss*	testing (*ms*)	m *cos*	testing (*ms*)	nsv	m *cos*	$\bar{\mathbf{m}}$ *Gauss*
Banana	144.7	153.1	615.9	375.7	2.1	**13.4**	13.6	611.3	393.1	2.0	2.5
B. Cancer	2.2	122.3	4.8	200.0	1.8	**0.4**	3.6	0.5	200.0	4.3	0.1
Diabetis	19.9	263.3	47.6	464.8	2.1	**0.8**	7.0	3.8	466.2	5.8	0.2
F. Solar	49.9	507.9	354.0	666.0	1.6	**0.6**	2.1	1.3	666	11.0	0.0
German	31.4	426.0	120.7	696.4	2.9	**1.0**	6.6	76.3	700.0	11.6	1.8
Heart	2.2	84.3	6.4	163.1	1.9	**0.3**	1.8	5.1	169.8	1.6	1.5
Image	205.4	700.7	765.1	1030.4	4.4	**56.6**	160.2	503.2	712.7	65.2	3.8
Ringnorm	**120.2**	64.4	487.0	156.2	1.9	228.5	91.7	610.8	200.2	0.0	2.0
Splice	357.7	385.4	1647.7	879.5	2.3	**51.4**	30.8	1293.6	755.7	13.5	1.6
Thyroid	0.5	22.0	3.1	84.3	3.0	**0.4**	6.4	3.1	112.0	1.0	2.1
Titanic	35.9	89.5	78.5	150.0	1.5	**1.5**	2.5	14.8	150.0	2.5	0.3
Twonorm	332.6	167.4	455.7	180.6	1.5	**3.5**	1.0	582.1	226.2	0.0	1.5
Waveform	134.3	104.5	587.4	290.5	1.9	**6.0**	2.9	731.9	374.7	1.0	1.6

Table 3: Experimentally measured evaluation times along with the parameters that theoretically determine the evaluation times of SG, GM, CM, and CGM methods on the data sets collected by Rätsch et al. (2001)

error rate, the evaluation of the CM method is 95 times faster than a classical SVM with Gaussian kernel. This speedup in the evaluation of the classifiers can be very useful for applications targeted at small computers with limited computational power.

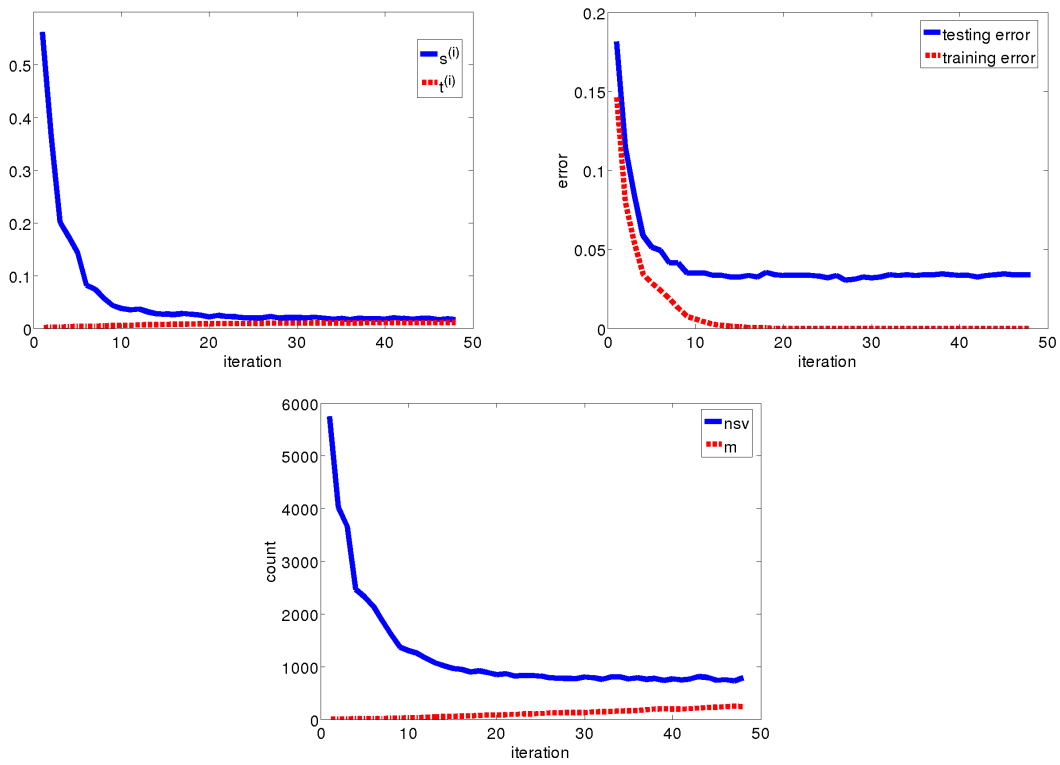

Figure 1: Evolution of the values m, nsv, $t^{(i)}$, $s^{(i)}$, training error, and test error during training of the proposed method on the USPS data set.

9.2 Experiments on the USPS Data Set

In this section, we show the applicability of the proposed method[16] to a real-world digit recognition problem. We consider the problem of classifying digits 0-4 against 5-9 on the USPS handwritten digit recognition data set as considered by Chapelle et al. (2002) for evaluating their kernel learning method. This data set consists of 7291 training examples and 2007 test examples of digit images of size 16×16. With polynomial kernel and 256 scaling factors, Chapelle et al. (2002) were able to get a test error rate of 9.0%. We trained the proposed method with the parameters $\varepsilon = 0.001$, and $\beta = 3.0$. After two hours of training, the algorithm produced a model with 244 cosine kernels and 790 support vectors. It took 1.3 of a second to test the model on the 2007 test samples and we obtained a test error rate of 3.4% which is significantly better than the 9.0% result reported by Chapelle et al. (2002). Figure 1 shows the evolution of the values m, nsv, $t^{(i)}$, $s^{(i)}$, training error, and test error during training of the USPS data set, where $s^{(i)}$ and $t^{(i)}$ are defined in Algorithm 1.

9.3 Experiments on the MNIST Data Set

While many algorithms for kernel learning consider the combination of a finite number of kernels, learning translation invariant kernels corresponds to combining an infinite number of kernels. In

16. From this section onward, the proposed method refers to the cosine mixture method.

this section, we compare the proposed algorithm with the DC method proposed by Argyriou et al. (2006) which is based on the theory developed by Micchelli and Pontil (2005). They considered the problem of finding an optimal kernel over the whole class of radial kernels which is equivalent to the problem of learning the best convex combination of Gaussian kernels with isotropic covariance matrices.

Argyriou et al. (2006) performed a series of experiments on the MNIST data set by using the first 500 training examples for training and the first 1000 test examples for evaluation. The MNIST data set contains 28×28 images of handwritten digits which are divided into 60,000 training and 10,000 test examples. In addition, the results reported by the DC method have been obtained by splitting each image into four sub-images which is a use of extra information. In the first experiment, we use the first $3,000$ training examples[17] for training both systems and evaluate them on the whole test set. As Argyriou et al. (2006), we consider the tasks of classifying digits 3 vs. 8, 4 vs. 7, and odds vs. evens. We downloaded the implementation of the DC method from *http://www.cs.ucl.ac.uk/staff/a.argyriou/code/dc* and chose the range $[75, 25000]$ for the parameter σ of the Gaussian kernel which is the largest range considered by Argyriou et al. (2006). The parameter μ of the DC method and the parameter β of the CM method were optimized by hand. The parameter ε was set to the value 0.001. The first three rows of Table 4 show the results of this experiment. It can be seen that the main benefit of the DC method is its short training time, while the CM method has superiority in terms of the evaluation time.

Another remarkable feature of the CM method is its applicability to large-scale problems. To illustrate this fact, we increased the size of the training set of the previous experiments from $3,000$ to $10,000$. We also increased the parameter ε from 0.001 to 0.01 to decrease the training time. The DC method could not handle this size of training samples and ran out of memory. The last three rows and columns of Table 4 show the results of the experiments with the CM method. Figure 2 depicts the evolution of the values *m*, *nsv*, $t^{(i)}$, $s^{(i)}$, training error, and test error during training of CM algorithm on the ods vs even task with $10,000$ training samples. Note that the model produced by the CM method on the larger training set is more accurate and faster-to-evaluate than the best model that the DC algorithm could produce. We think that the capabilities of the CM method and DC method are complementary. The DC method works with full-rank matrices, is not large-scale, converges fast, and its model takes more time to compute. On the other hand, the CM method works with low-rank matrices, is large-scale, converges slowly, and its model can be evaluated very fast. One open problem is that whether these methods can be combined in a way that the benefits of both methods are achieved.

9.4 Assessing the Effect of the Proposed Complexity Control Mechanism

In Section 8 we provided theoretical support for the necessity of controlling the complexity of the class of translation invariant kernels. Here we support this claim by experimenting on the *Heart* data set chosen from the benchmark produced by Rätsch et al. (2001). This data set contains 170 train patterns and 100 test patterns of dimension 13. The experiments of the previous section show that the proposed method was completely successful in obtaining a low test error rate on this data set. In addition, the mean error rate of the proposed method on the train set is 14.0% which is close to the mean error rate of 15.5% obtained on the test set. The left plot of Figure 3 illustrates the trajectories of the train and test errors of the proposed method during training on the *Heart* data set.

17. This is approximately the largest possible train-set size where the DC method did not ran out of memory.

Data Set		DC method			Cosine mixture		
Task	**#tr**	**error** (%)	**train** (*min*)	**test** (*sec*)	**error** (%)	**train** (*min*)	**test** (*sec*)
odd vs even	3000	3.1	11	60.1	3.2	37	10.1
3 vs 8	3000	0.7	9	24.8	0.8	192	2.6
4 vs 7	3000	0.4	16	26.1	0.5	122	2.5
odd vs even	10000	-	-	-	2.0	63	9.1
3 vs 8	10000	-	-	-	0.4	1133	5.3
4 vs 7	10000	-	-	-	0.2	649	3.1

Table 4: Test errors of the proposed method (CM) and the DC method on different tasks on the MNIST data set. The dash sign indicates running out of memory.

In another experiment we disabled our complexity control mechanism by setting $\beta = 10^{-6}$ and instead tried to control the capacity of the learning machine by adjusting the parameter C.[18] We optimized the parameter C using the 5-fold cross-validation method described in the previous section. After testing on all the 100 splits of the *Heart* data set we obtained a mean test error rate of 20.8% and a mean train error rate of 17.4%. The right plot of Figure 3 illustrates the trajectories of the train and test error rates of this experiment on the *Heart* data set. This experiment confirms the usefulness of controlling the complexity of the class of translation invariant kernels, as was claimed in Section 8.

10. Conclusions

In this paper we addressed the problem of learning a translation invariant kernel function for the task of binary classification with SVM. We proposed a mechanism for controlling the complexity of the class of translation invariant kernels which was found to be very useful in practice. The criterion proposed by Lanckriet et al. (2004) was modified to ensure the compactness of the parameter space of SVM and to give a probabilistic meaning to the regularization parameter of the 2-norm SVM. We then introduced a semi-infinite programming formulation of the problem. The proposed method can automatically learn the regularization parameter of the 2-norm SVM, as well. We have also shown that how other classes of kernels can be included in the learning process. To numerically solve the SIP problem on a computer, we introduced a large-scale algorithm which is applicable to problems with both huge number of training samples and large number of features. Since the optimal translation invariant kernel is complex-valued, we then introduced a method for applying the kernel trick to complex-valued kernels. It revealed that the optimal translation invariant kernel is a mixture of cosine kernels. An interesting feature of the proposed method is that there is a very fast way for evaluating the classifier at run-time. While an ordinary MKL algorithm with m kernels requires $O(m \times nsv \times n)$ steps for computing the classifier, the optimal classifier of the proposed method can be computed in $O(m \times n)$ steps.

In continuation of this work, we plan to extend it in several directions. First, we intend to generalize the proposed kernel learning method from binary classification to other learning problems, including regression, multiclass classification, clustering, and kernel PCA. Second, we will try to propose a novel large-scale algorithm that combines the benefits of the full-rank Gaussian and the

18. Setting $\beta = 0$ (exactly) allowed the global search algorithm to find points at infinity which caused problems.

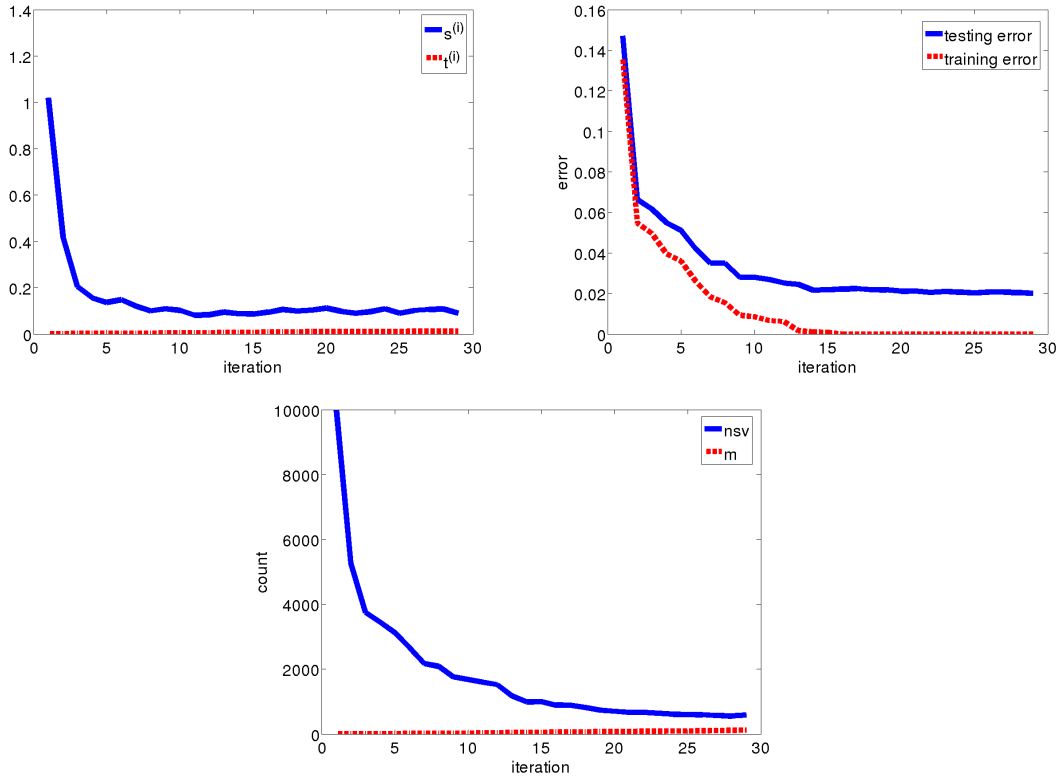

Figure 2: Evolution of the values m, nsv, $t^{(i)}$, $s^{(i)}$, training error, and testing error during training of the proposed method with the first $10,000$ samples of the MNIST data set for the task of classifying odds vs. evens.

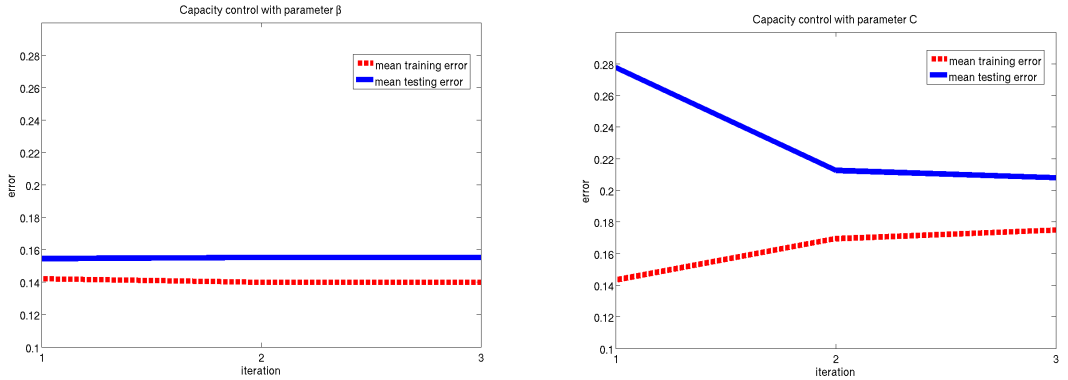

Figure 3: Comparison between the parameters β and C for controlling the capacity of the class of translation invariant kernels on the *Heart* data set.

low-rank cosine kernels. Third, we intend to investigate the applicability of the idea of Xu et al. (2008) about adding a regularization term that smoothes the fluctuating behaviour of SILP algorithms, to the proposed SIP algorithm. We hope that this study would greatly decrease the training time of the proposed method. Another direction is to support the complexity control mechanism of Section 2 by introducing upper bounds for the generalization error of the proposed method. Our long time plan is to investigate the use of other low-rank kernels and make it a competing popular technology.

Acknowledgments

The first author would like to thank professor Abdolhamid Riazi for allowing him to participate in his functional analysis class and helping him to get acquainted with the concepts of this graduate course. We also want to thank Stephen Boyd, Andreas Argoriou, and Peter Gehler for providing us with their valuable materials.

References

S. Amari and S. Wu. Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, 12(6):783–789, 1999.

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

A. Argyriou, C.A. Micchelli, and M. Pontil. Learning convex combinations of continuously parameterized basic kernels. In *Proceedings of the 18th Conference on Learning Theory*, volume 18, pages 338–352, 2005.

A. Argyriou, R. Hauser, C.A. Micchelli, and M. Pontil. A DC-programming algorithm for kernel selection. In *Proceedings of the* 23^{rd} *International Conference on Machine Learning*, volume 23, pages 338–352, Pittsburgh, PA, 2006.

F.R. Bach, G.R.G. Lanckriet, and M.I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the 21st International Conference on Machine Learning*, volume 21, pages 41–48, Banff, Canada, 2004. Omnipress.

S. Bochner. Monotone functions, Stieltjessche integrale und harmonische analyse. *Mathematische Annalen*, 108:378–410, 1933.

S. Bochner. *Harmonic Analysis and the Theory of Probability*. University of California Press, Los Angeles, California, 1955.

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

O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3):131–159, 2002.

D.J. Crisp and C.J.C. Burges. A geometric interpretation of ν-svm classifiers. In *Advances in Neural Information Processing Systems*, volume 12, pages 244–250, Cambridge, MA:, 1999. MIT Press.

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

Nello Cristianini, Colin Campbell, and John Shawe-taylor. Dynamically adapting kernels in support vector machines. In *Advances in Neural Information Processing Systems 11*, pages 204–210. MIT Press, 1998.

F. Cucker and D.X. Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press, Cambridge, UK, 2007.

J. M. Danskin. The theory of max-min, with applications. *SIAM Journal on Applied Mathematics*, 14(4):641–664, 1966.

P. V. Gehler and S. Nowozin. Infinite kernel learning. In *Proceedings of the NIPS 2008 Workshop on "Kernel Learning: Automatic Selection of Optimal Kernels"*, pages 1–4, 2008.

M.G. Genton. Classes of kernels for machine learning: A statistics perspective. *Journal of Machine Learning Research*, 2:299–312, 2001.

F. Girosi, M. Jones, and T. Poggio. Priors, stabilizers and basis functions: from regularization to radial, tensor and additive splines. A.I. Memo 1430, Artificial Intelligence Labratory, Massachusetts Institute of Technology, June 1993.

T. Glasmachers and C. Igel. Gradient-based adaptation of general Gaussian kernels. *Neural Computation*, 17:2099–2105, 2005.

R. Hettich and K.O. Kortanek. Semi-infinite programming: theory, methods, and applications. *SIAM Review*, 35(3):380–429, 1993.

T. Joachims. Making large-scale support vector machine learning practical. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods- Support Vector Learning*, pages 169–184. MIT Press, 1999.

E. Kreyszig. *Introductory Functional Analysis with Applications*. Wiley, New York, 1989.

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

D.C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming B*, 45(3):503–528, 1989.

M.E. Mavroforakis and S. Theodorodis. A geometric approach to support vector machine (SVM) classification. *IEEE Transactions on Neural Networks*, 17(3):671–682, 2006.

C.A. Micchelli and M. Pontil. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6:1099–1125, 2005.

C.A. Micchelli, M. Pontil, Q. Wu, and D.X. Zhou. Error bounds for learning the kernel. Research Note RN/05/09, Department of Computer Science, University College London, June 2005b.

J. Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35 (151):773–782, 1980.

J. Nocedal and S.J. Wright. *Numerical Optimization, second edition*. Springer Science+Business Media, LLC, New York, USA, 2006.

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

J.C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods- Support Vector Learning*, pages 185–208. MIT Press, 1999.

A. Rakotomamonjy, F.R. Bach, S. Canu, and V. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

G. Rätsch, T. Onoda, and K.R. Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3): 287–320, 2001.

R. Reemtsen and S. Görner. Numerical methods for semi-infinite programming: A survey. In R. Reemtsen and Rückmann, editors, *Semi-infinite Programming*, pages 195–275. Kluwer Academic Publishers, 1998.

R.T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, 1970.

H.L. Royden. *Real Analysis, 3rd edition*. Macmillan Publishing Company, New York, 1988.

W. Rudin. *Functional Analysis, 2nd edition*. McGraw-Hill, New York, 1991.

W. Rudin. *Real & Complex Analysis, 3rd edition*. McGraw-Hill, New York, 1987.

I.J. Schoenberg. Metric spaces and completely monotone functions. *Annals of Mathematics*, 39(4): 811–841, 1938.

B. Schölkopf and A. Smola. *Learning with Kernels- Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, MA, 2002.

B. Schölkopf, P. Knirsch, A. Smola, and C. Burges. Fast approximation of support vector kernel expansions, and an interpretation of clustering as approximation in feature spaces. In *DAGM Symposium Mustererkennung*. Springer Lecture Notes in Computer Science, 1998.

S. Sonnenburg, G. Rätsch, and C. Schafer. A general and efficient multiple kernel learning algorithm. In *Advances in Neural Information Processing Systems*, volume 18, pages 1275–1282, Cambridge MA, 2005. MIT Press.

S. Sonnenburg, G. Rätsch, C. Schafer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1567, 2006.

V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

V.N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.

H. Xiong, M.N.S. Swamy, and M.O. Ahmad. Optimizing the kernel in the empirical feature space. *IEEE Transactions on Neural Networks*, 16(2):460–474, 2005.

Z. Xu, R. Jin, I. King, and M.R. Lyu. An extended level method for efficient multiple kernel learning. In *Advances in Neural Information Processing Systems*, volume 21, pages 1825–1832, British Columbia, Canada, 2008. MIT Press.

Y. Yiming and D.X. Zhou. Learnability of Gaussians with flexible variances. *Journal of Machine Learning Research*, 8:249–276, 2007.