Journal of Machine Learning Research 20 (2019) 1-25 Submitted 7/19; Published 10/19

A Convex Parametrization of a New Class of Universal Kernel Functions

Brendon K. Colbert BRENDON.COLBERT@ASU.EDU
Department of Mechanical and Aerospace Engineering
Arizona State University
Tempe, AZ 85281-4322, USA

Matthew M. Peet MPEET@ASU.EDU
Department of Mechanical and Aerospace Engineering
Arizona State University
Tempe, AZ 85281-1776, USA

Editor: Mehryar Mohri

Abstract

We propose a new class of universal kernel functions which admit a linear parametrization using positive semidefinite matrices. We refer to kernels of this class as Tessellated Kernels (TKs) due to the observation that if applied to kernel-based learning algorithms, the resulting discriminants are defined by continuous piecewise-polynomial functions with hyper-rectangular domains whose vertices are determined by the training data. While TKs are defined by polynomials they, unlike polynomial kernels, are universal in the sense that the resulting discriminants occupy a hypothesis space which is dense in L_2. In addition TKs have only one user defined hyper-parameter–the maximum degree of the piecewise-polynomials. This implies that the use of TKs for learning the kernel (aka kernel learning) can obviate the need for Gaussian kernels and associated problem of selecting bandwidth - a conclusion verified through extensive numerical testing on soft margin Support Vector Machine (SVM) problems. Furthermore, our results show that when the ratio of the number of training data to features is high, the proposed method will significantly outperform other algorithms for learning the kernel. Finally, TKs can be integrated efficiently with existing Multiple Kernel Learning (MKL) algorithms such as SimpleMKL.

Keywords: Kernel Functions, Multiple Kernel Learning, Semi-definite Programming, Supervised Learning, Universal Kernels

1. Introduction

This paper addresses the problem of automated selection of an optimal kernel function for a given kernel-based machine learning problem (e.g. soft margin SVM). Kernel functions implicitly define a linear parametrization of nonlinear candidate maps $y = f(x)$ from vectors x to scalars y. Specifically, for a given kernel, the 'kernel trick' allows optimization over a set of candidate functions in the kernel-associated hypothesis space without explicit representation of the space itself. The kernel selection process, then, is critical for determining the class of hypothesis functions and, as a result, is a well-studied topic with common kernels including polynomials, Gaussians, and many variations of the Radial Basis Function. In addition, specialized kernels include string kernels as in Lodhi et al. (2002); Eskin et al. (2003), graph kernels as in Gärtner et al. (2003), and convolution kernels as in Haussler (1999); Collins and Duffy (2002). The kernel selection process heavily influences the accuracy of the resulting fit and hence significant research has gone into optimization of these kernel functions in order to select the hypothesis space which most accurately represents the underlying physical process.

Recently, there have been a number of proposed kernel learning algorithms. For support vector machines, the methods proposed in this paper are heavily influenced by the SDP approach proposed by Lanckriet et al. (2004) which directly imposed kernel matrix positivity using a linear subspace of candidate kernel functions (as in MKL). There have been several extensions of the SDP approach, including the hyperkernel method of Ong et al. (2005). However, because of the complexity of semidefinite programming, more recent work has focused on Alignment methods for MKL as in, e.g. Cortes et al. (2012) or gradient methods for convex and non-convex parameterizations of positive linear combinations of candidate kernels, as in SimpleMKL Rakotomamonjy et al. (2008) or the several variations in Sonnenburg et al. (2010). These MKL methods rely on kernel operations (addition, multiplication, convolution) to generate large numbers of parameterized kernel functions as in Cortes et al. (2009). Examples of non-convex parameterizations include GMKL as introduced in Jain et al. (2012), and LMKL Gönen and Alpaydin (2008). Work focused on regularization includes the group sparsity metric defined in Subrahmanya and Shin (2010) and the enclosing ball approach in Gai et al. (2010). See, e.g. Gönen and Alpaydın (2011) for a comprehensive review of MKL algorithms.

In this paper, we focus on the class of "Universal Kernels" formalized in Micchelli et al. (2006). For a given compact metric space (input space), $\mathcal{X}$, it is said that a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a Positive Kernel (PK) if for any $N \in \mathbb{N}$ and any $\{x_i\}_{i=1}^N \subset \mathcal{X}$, the matrix defined elementwise by $K_{ij} = k(x_i, x_j)$ is symmetric and Positive SemiDefinite (PSD).

Definition 1 *A kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is said to be universal on the compact metric space $\mathcal{X}$ if it is continuous and there exists an inner-product space $\mathcal{W}$ and feature map, $\Phi : \mathcal{X} \to \mathcal{W}$ such that $k(x,y) = \langle \Phi(x), \Phi(y) \rangle_{\mathcal{W}}$ and where the unique Reproducing Kernel Hilbert Space (RKHS),*

$$\mathcal{H} := \{f \ : \ f(x) = \langle v, \Phi(x) \rangle, \ v \in \mathcal{W}\}$$

with associated norm $\|f\|_{\mathcal{H}} := \inf_v \{\|v\|_{\mathcal{W}} \ : \ f(x) = \langle v, \Phi(x) \rangle\}$ is dense in $\mathcal{C}(\mathcal{X}) := \{f : \mathcal{X} \to \mathbb{R} \ : f is\ continuous\}$ where $\|f\|_{\mathcal{C}} := \sup_{x \in X} |f(x)|$.

Note that for an given PD kernel, $\mathcal{H}$ exists, is unique, and can be characterized using the Riesz representation theorem Sun (2005) as the closure of span$\{k(y, \cdot) \ : \ y \in \mathcal{X}\}$ with inner product defined for any $f(x) = \sum_{i=1}^n c_i k(y_i, x)$ and $g(x) = \sum_{i=1}^m d_i k(z_i, x)$ as

$$\langle f, g \rangle_{\mathcal{H}} := \sum_{i=1}^n \sum_{j=1}^m c_i d_i k(y_i, z_i).$$

The most well-known example of a universal kernel is the Gaussian (generalized in Zanaty and Afifi (2011)). However, many other common kernels are not universal, including, significantly, the polynomial class of kernels (this is significant because polynomials admit a linear parameterization).

In this paper, we propose a new class of universal kernel functions which are not polynomials, yet which are defined by polynomials and admit a convex parametrization. The class of kernels can be represented using polynomials on hyper-rectangular sets whose vertices are defined by the input data $\{x_i\}_{i=1}^{p}$.

In this way, each data point further divides the domains which it intersects, resulting in increasing numbers of disjoint sub-domains, each with associated polynomial classifier. This new class of universal kernels thus retains the properties of a polynomial classifier, are universal like Gaussian kernels yet have only one hyper parameter.

The paper is organized as follows; in Section 2 we provide an overview of the MKL problem. Section 3 proposes a framework by which the class of TK functions can be parameterized by positive matrices, and Section 4 proves general properties such as universality for the class of TK functions. Section 5 and 6 show how the class of TK functions can be rigorously incorporated into the SDP MKL framework and into SimpleMKL's framework respectively. In Section 7 we discuss the complexity of incorporating TK functions into both the SDP MKL framework and the SimpleMKL framework. Finally in Section 8 we provide numerical results that illustrate improved performance using TK functions on a number of UCI repository data sets.

2. An overview of the optimal kernel learning problem for the 1-norm SVM problem

We begin this section by posing the kernel-learning problem as a convex optimization problem for the particular case of the 1-norm soft margin support vector machine. Next, in Subsections A and B, we present two standard algorithms for solving the kernel learning problem. These algorithms are general in the sense that they apply to any given linear parameterization of kernel functions. The adaptation of these algorithms to the special case of Tessellated Kernels (TKs) will then be described in Section 3.

Suppose we are given a set of m *training data* points $\{x_i\}_{i=1}^{m} \subset \mathbb{R}^n$, each with associated *label* $y_i \in \{-1, 1\}$ for $i = 1, \cdots, m$. For a given "penalty" parameter $C \in \mathbb{R}^+$, we define the *linear* 1-norm soft margin problem as

$$\begin{aligned} \min_{w \in \mathbb{R}^n, \zeta \in \mathbb{R}^{+m}, b \in \mathbb{R}} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^{m} \zeta_i \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \zeta_i, \end{aligned} \tag{1}$$

where the learned map (*classifier*) from inputs to outputs is then $f : \mathbb{R}^n \to \{-1, 1\}$ where

$$f(x) = \text{sign}(w^T x + b).$$

If we desire the classifier to be defined by a nonlinear function, we may introduce a positive kernel function, k.

Definition 1 *We say a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a* **positive kernel function** *if*

$$\int_{\mathcal{X}} \int_{\mathcal{X}} f(x) k(x, y) f(y) dx dy \geq 0$$

for any function $f \in L_2[\mathcal{X}]$.

In this case the the primal problem becomes,

$$\begin{aligned}\min_{w\in\mathbb{R}^n,\zeta\in\mathbb{R}^{+m},b\in\mathbb{R}} \quad & \frac{1}{2}w^T w + C\sum_{i=1}^m \zeta_i \\ \text{s.t.} \quad & y_i(\langle w, \Phi(x_i)\rangle + b) \geq 1 - \zeta_i, \end{aligned} \tag{2}$$

where, $k(x,y) = \langle \Phi(x), \Phi(y)\rangle$ and the classifier can be represented as,

$$f(z) = \text{sign}\left(\langle w, \Phi(z)\rangle + b\right).$$

The primal form of SVM is rarely used, however, as the function $\Phi(\cdot)$ may be infinite dimensional or difficult to compute. More common is the dual formulation,

$$\begin{aligned}\max_{\alpha\in\mathbb{R}^m} \quad & \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i=1}^m\sum_{j=1}^m \alpha_i\alpha_j y_i y_j \langle \Phi(x_i), \Phi(x_j)\rangle \qquad (3) \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C \ \ \forall \ \ i = 1, ..., m.\end{aligned}$$

In this case we may eliminate Φ from the optimization problem using $\langle \Phi(x_i), \Phi(x_j)\rangle = k(x_i, x_j)$ where the elements $k(x_i, x_j)$ define the kernel matrix. In this case, the resulting classifier is also only a function of k and becomes

$$f(z) = \text{sign}\left(\sum_{i=1}^m \alpha_i y_i k(x_i, z) + b\right).$$

Note that b can be found a posteriori as the average of $y_j - \sum_{i=1}^m \alpha_i y_i k(x_j, x_i)$ for all j such that $0 < \alpha_j < C$ - See Schölkopf et al. (2002). This implies that the primal variable w is not explicitly required for the calculation of b, and that the resulting learned classifier, f, may be expressed solely in terms of α and the kernel function.

Commonly used positive kernel functions include the gaussian kernel $k_1(x, y) = \mathrm{e}^{(-\beta||x-y||^2)}$, where β is the bandwidth (and must be chosen a priori) and the polynomial kernel $k_2(x, y) = (1 + x^T y)^d$ where d is the degree of the polynomial.

Unfortunately optimization problem 3 requires that the kernel function, $k(x, y)$, be chosen a priori and furthermore, this choice significantly influences the accuracy of the resulting classifier f. We therefore alter the optimization problem by considering the kernel itself to be an optimization variable, constrained to lie in a given convex set of candidate kernel functions, $\mathcal{K}$. In this case, we have the following convex optimization problem.

$$\begin{aligned}\min_{k\in\mathcal{K}}\max_{\alpha\in\mathbb{R}^m} \quad & \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i=1}^m\sum_{j=1}^m \alpha_i\alpha_j y_i y_j k(x_i, x_j) \qquad (4) \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C \ \ \forall \ \ i = 1, ..., m.\end{aligned}$$

Having formulated the kernel learning problem, we now present two standard approaches to parameterizing the set of candidate kernels, $\mathcal{K}$, and solving the resulting convex optimization problem.

2.1 Formulating the kernel optimization problem for linear combinations of kernel functions

We first consider the method of Lanckriet et al. (2004), wherein positive matrices were used to parameterize $\mathcal{K}$ for a given set of candidate kernels $\{k_i\}_{i=1}^l$ as

$$\mathcal{K} := \left\{ k(x,y) = \sum_{i=1}^{l} \mu_i k_i(x,y) \, : \, \mu \in \mathbb{R}^l, \; K_{ij} = k(x_i, x_j), \; K \geq 0 \right\},$$

where the $\{x_i\}_{i=1}^m \subset \mathbb{R}^n$ are the training points of the SVM problem and the k_i were chosen a priori to be Gaussian and polynomial kernels. It is significant to note that the PSD constraint on the kernel matrix K, enforces that the kernel matrix is PSD for the set of training data, but does not necessarily enforce that the kernel function itself is PD - meaning that kernels in $\mathcal{K}$ are not necessarily positive kernels.

Using this parameterized $\mathcal{K}$, the kernel optimization problem for the 1-norm soft margin support vector machine was formulated as the following semi-definite program,

$$\begin{aligned} \min_{\mu \in \mathbb{R}^l, G \in \mathbb{R}^{m \times m}, t \in \mathbb{R}, \gamma \in \mathbb{R}, \nu \in \mathbb{R}^m, \delta \in \mathbb{R}^m} \quad & t, \\ \text{subject to:} \quad & \begin{pmatrix} G & e + \nu - \delta + \gamma y \\ (e + \nu - \delta + \gamma y)^T & t - \frac{2}{m\lambda} \delta^T e \end{pmatrix} \geq 0 \\ \nu \geq 0, \qquad \delta \geq 0, \qquad & G_{ij} = k(x_i, x_j) y_i y_j \\ k(x,y) = \sum_{i=1}^{l} \mu_i k_i(x,y) & \end{aligned} \tag{5}$$

Note that here the original constraint $K \geq 0$ in $\mathcal{K}$ has been replaced by an equivalent constraint on G. This problem can now be solved using well-developed interior-point methods as in Alizadeh et al. (1998) with implementations such as MOSEK ApS (2015).

In Optimization Problem (5), the size of the SDP constraint is $(m+1) \times (m+1)$ which is problematic in that the complexity of the resulting SDP grows as a polynomial in the number of training data. Our parameterization, introduced in Section 3, avoids this computational bottleneck by proposing kernel positivity tests whose complexity is independent of the amount of training data. Furthermore, our method does not require the a priori selection of a set of basis kernels.

2.2 Formulating the kernel learning optimization problem for positive linear combinations of kernel functions

In this subsection, we again take a set of basis kernels $\{k_i\}_{i=1}^l$ and consider the set of positive linear combinations,

$$\mathcal{K} := \left\{ k \; : \; k(x,y) = \sum_{i=1}^{l} \mu_i k_i(x,y), \; \mu_i \geq 0 \right\}.$$

Any element of this set is a positive kernel, replacing the matrix positivity constraint by an LP constraint.

$$\begin{aligned} \min_{\mu \geq 0} \max_{\alpha \in \mathbb{R}^m} \quad & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{l} \mu_k \alpha_i \alpha_j y_i y_j k_k(x_i, x_j) \\ \text{s.t.} \quad & \sum_{i=1}^{m} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C \ \ \forall \ \ i = 1, ..., m. \end{aligned}$$

Use of this formulation is generally referred to as Multiple Kernel Learning (MKL). It has the disadvantage that it is non-convex in native form. Recently, however, a number of highly efficient two-step methods have been proposed to solve the associated kernel learning problem, including SimpleMKL as in Rakotomamonjy et al. (2008). These methods first fix μ_i and optimize over α, then fix α and optimize over μ, adding the constraint that $\sum_i \mu_i = 1$ using a projected gradient descent. Other two-step solvers such as Gönen and Alpaydın (2011) solve the second step using LP. Two-step MKL solvers typically have a significantly reduced computational complexity compared with SDP-based approaches and can typically handle thousands of data points and thousands of basis kernels.

In Section 3, we propose a parameterization of kernels using positive matrices which avoids the need for the selection of basis kernels. Moreover, we show that this parameterization can be combined with MKL algorithms either directly in SimpleMKL through the use of a randomly generated basis of kernels, or through a new algorithm which modifies the second step to optimize over the set of positive matrices.

3. Positive "tessellated" kernel functions can be parameterized by positive matrices

In this section, we propose a general framework for using positive matrices to parameterize a class of TK functions. The following result is based on a parametrization of positive integral operators initially proposed in Recht (2006).

Lemma 2 *Let N be any bounded measurable function $N : \mathcal{X} \times X \to \mathbb{R}^q$ on compact $\mathcal{X}$ and X and $P \in \mathbb{R}^{q \times q}$ be a positive matrix $P \geq 0$. Then*

$$k(x, y) = \int_{\mathcal{X}} N(z, x)^T P N(z, y) dz \tag{6}$$

is a positive kernel function.

Proof Since N is bounded and measurable, $k(x, y)$ is bounded and measurable. Since $P \geq 0$, there exists $P^{\frac{1}{2}}$ such that $P = (P^{\frac{1}{2}})^T P^{\frac{1}{2}}$. Now define

$$g(z) = \int_X P^{\frac{1}{2}} N(z, x) f(x) dx.$$

Then

$$\begin{aligned}\int_X \int_X f(x)k(x,y)f(y)dxdy =&\int_X \int_X \int_{\mathcal{X}} f(x)N(z,x)^T PN(z,y)f(y)dzdxdy\\ =&\int_{\mathcal{X}} \left(\int_X P^{\frac{1}{2}}N(z,x)f(x)dx \right)^T \int_X N(z,y)P^{\frac{1}{2}}f(y)dxdydz\\ =&\int_{\mathcal{X}} g(z)^T g(z)dz \geq 0.\end{aligned}$$

■

Generalized Polynomial Kernels Let $X = \mathbb{R}^n$ and $\mathcal{X} = \mathbb{R}^p$ and define $Z_d : \mathbb{R}^n \to \mathbb{R}^q$ to be the vector of monomials of degree d. In this case, it was shown in Peet et al. (2009) that k is a degree $2d$ positive polynomial kernel if and only if there exists some $P \geq 0$ such that

$$k(x,y) = Z_d(x)^T PZ_d(y). \tag{7}$$

This implies that a representation of the form of Equation (6) is necessary and sufficient to represent all positive polynomial kernels. Unfortunately, polynomial kernels are not universal and hence we propose the following universal class of TKs, each of which is defined by polynomials, but which are not polynomial.

Tessellated Kernels As defined in Gohberg et al. (2013), a kernel $k(x,y)$ is *semi-separable* if there exist functions A_i such that

$$k(x,y) = \begin{cases} A_1(x)A_2(y), & \text{if } x > y\\ A_3(x)A_4(y), & \text{otherwise.}\end{cases}$$

Semi-separable kernels define a broad class of integral operators include, e.g. the Volterra operators. TKs are a generalization of semi-separable kernels to multiple orthants. Specifically, if we define the partition of $\mathbb{R}^n \times \mathbb{R}^n$ into 2^n orthants - parameterized as $\{X_\beta\}_{\beta \in \{0,1\}^n}$ where

$$X_\beta := \left\{ (x,y) \in \mathbb{R}^n \times \mathbb{R}^n \;:\; \substack{x_j \geq y_j \text{ for all } j:\beta_j=0,\\ y_i \geq x_i \text{ for all } i:\beta_i=1} \right\}, \tag{8}$$

then TKs have the form

$$k(x,y) = \Big\{ A_{1,\beta}(x)A_{2,\beta}(y), \quad \text{if } x-y \in X_\beta.$$

To begin, let us denote the positive orthant as S_1 and the negative orthant as S_2. Now define the indicator function on a set S as

$$I_S(z,x) = \begin{cases} 1 & z-x \in S\\ 0 & \text{otherwise,}\end{cases}$$

Now let $X = \mathcal{X} = \mathbb{R}^n$ and define $Z_d : \mathcal{X} \times X \to \mathbb{R}^q$ to be the vector of monomials of degree d in $\mathbb{R}^{2n}$. We now propose the following choice of $N : \mathcal{X} \times X \to \mathbb{R}^{2q}$.

$$N(z,x) = \begin{bmatrix} Z_d(z,x)I_{S_1}(z,x)\\ Z_d(z,x)I_{S_2}(z,x)\end{bmatrix}. \tag{9}$$

Combining this definition of N with Eqn. (6), we will now show that the associated kernel function

$$k(x,y)=\int_{\mathcal{X}} N(z,x)^T P N(z,y)dz.$$

has the tessellated structure indicated above.

A Partition of the Tessellated Kernel The following result shows that for our choice of N, the resulting kernel is piecewise polynomial, with a polynomial k_β associated with each domain X_β.

Lemma 3 *Suppose that for $a<b\in\mathbb{R}^n$, $X=\mathcal{X}=[a,b]$, N is as defined in Eqn. (9),*

$$P=\begin{bmatrix}P_{11} & P_{12}\\ P_{21} & P_{22}\end{bmatrix}>0$$

k is as defined in Eqn. (6) and $\{X_\beta\}_{\beta\in\{0,1\}^n}$ is defined in Eqn. (8). Then

$$k(x,y)=\Big\{k_\beta(x,y) \quad \textit{if} \;\; (x,y)\in X_\beta. \tag{10}$$

where the k_β are polynomials defined as

$$k_\beta(x,y)=\prod_{i:\beta_i=0}\int_{z_i=x_i}^{b_i}\prod_{j:\beta_j=1}\int_{z_j=y_j}^{b_j} Z_d(z,x)^T Q_1 Z_d(z,y)dz+k_0(x,y),$$

where,

$$k_0(x,y)=\int_x^b Z_d(z,x)^T Q_2 Z_d(z,y)dz+\int_y^b Z_d(z,x)^T Q_3 Z_d(z,y)dz+\int_a^b Z_d(z,x)^T P_{22} Z_d(z,y)dz,$$

and

$$Q_1{=}P_{11}-P_{12}-P_{21}+P_{22},\quad Q_2{=}P_{12}-P_{22},\quad Q_3{=}P_{21}-P_{22}.$$

Proof

Given N as defined above, if we partition $P=\begin{bmatrix}P_{11} & P_{12}\\ P_{21} & P_{22}\end{bmatrix}$ into equal-sized blocks, we have

$$k(x,y)=\int_{\mathcal{X}} N(z,x)^T P N(z,y)dz=\sum_{i,j=1}^{2}\int_{(x,y,z)\in\mathcal{X}_{ij}} Z_d(z,x)^T P_{i,j} Z_d(z,y)dz$$

where

$$\mathcal{X}_{ij}:=\{z\in\mathbb{R}^{3n}\;:\; I_{S_i}(z,x)I_{S_j}(z,y)=1\}.$$

From the definition of $\mathcal{X}_{ij}$ we have that,

$$\begin{aligned}
\mathcal{X}_{11}=&\{z\in Z\;:\; z_i\geq p_i^*(x,y),\; i=1,\cdots,n\}\\
\mathcal{X}_{12}=&\{z\in Z\;:\; z_i\geq x_i,\; i=1,\cdots,n\}/X_{11}\\
\mathcal{X}_{21}=&\{z\in Z\;:\; z_i\geq y_i,\; i=1,\cdots,n\}/X_{11}\\
\mathcal{X}_{22}=&Z/\left(X_{11}\cup X_{12}\cup X_{21}\right).
\end{aligned}$$

where $p_i^*(x,y) = \max\{x_i, y_i\}$. By the definitions of $\mathcal{X}_{11}, \mathcal{X}_{12}, \mathcal{X}_{21}$, and $\mathcal{X}_{22}$ we have that,

$$\begin{aligned} k(x,y) =& \int_{p^*(x,y)}^{b} Z_d(z,x)^T (P_{11} - P_{12} - P_{21} + P_{22})\, Z_d(z,y) dz + \int_{x}^{b} Z_d(z,x)^T (P_{12} - P_{22})\, Z_d(z,y) dz \\ &+ \int_{y}^{b} Z_d(z,x)^T \left(P_{21} - P_{22}\right) Z_d(z,y) dz + \int_{a}^{b} Z_d(z,x)^T P_{22} Z_d(z,y) dz. \end{aligned} \tag{11}$$

■

Note that the number of domains X_β used to define the piecewise polynomial k is 2^n, which does not depend on q (the dimension of P_{ij}). Thus, even if $Z_d = 1$, the resulting kernel is partitioned into 2^n domains. The length of $Z_d(x,y) \in \mathbb{R}^q$ only influences the degree of the polynomial defined on each domain.

The significance of the partition does not lie in the number of domains, however. Rather, the significance lies in the resulting classifier, which is defined by the input data $\{x_i\}_{i=1}^m$ and has the form

$$\begin{aligned} f(z) &= \sum_{i=1}^{m} \alpha_i y_i k(x_i, z) + b \\ &= \begin{cases} \sum_{i=1}^{m} \alpha_i y_i k_\beta(x_i, z) & \text{if } \ (x_i, z) \in X_\beta. \end{cases} \\ &= \begin{cases} f_{i,\beta}(z) & \text{if } \ z \in X_{i,\beta}, \end{cases} \end{aligned}$$

where,

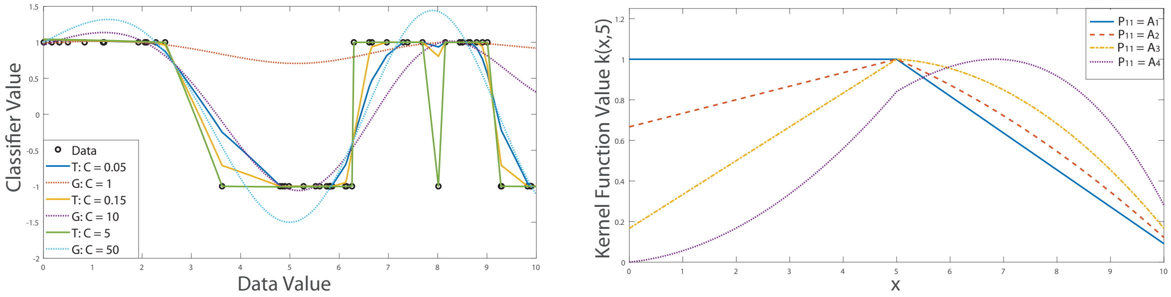
$$\begin{aligned} f_{i,\beta}(z) &= \sum_{i=1}^{m} \alpha_i y_i k_\beta(x_i, z) \\ X_{i,\beta} &:= \left\{ z \ : \ \substack{(x_i)_j \geq z_j \text{ for all } j:\beta_j = 0, \\ z_k \geq (x_i)_k \text{ for all } k:\beta_i = 1} \right\} \end{aligned}$$

where the $f_{i,\beta}$ are polynomials. In this way, each data point further divides the domains which it intersects, resulting in $(m+1)^n$ disjoint sub-domains, each with associated polynomial classifier.

Thus we see that the number of domains of definition grows quickly in m, the number of training data points. For instance with n = 2 there are 100 sub-domains for just 9 data points. This growth is what makes TKs universal - as will be seen in Section IV.

In Figure 1(a) we see the function, $f(z) = \sum_{i=1}^{m} \alpha_i y_i k(x_i, z) + b$, for a degree 1 TK function trained for a 1-dimensional labeling problem as compared with a Gaussian kernel. We see that the TK is continuous, and captures the shape of the generator better than the Gaussian. However, the kernel is not continuously differentiable; a property imposed using the inverse regularity weight C in the objective function on Eqn (2). In Figure 1(a), as C decreases we see that the changes in slope at edges of the domain decrease.

To illustrate the basis function $k_\beta(x_i, z)$, in Figure 1(b) we plot the value of the kernel function in one dimension with training datum $x_i = 5$, for a selection of different positive



(a) Optimal classifier, $f(z)$ for labelling a 1 dimensional dataset using a degree one TK (solid lines), and a positive combination of Gaussian kernels (dotted lines) with three different penalty weights C.

(b) Normalized kernel function $k_{\{1,1\}}(5, z)$ using $P_{1,1} = A_i$ from (12) and $P_{1,2} = P_{2,1} = P_{2,2} = 0$..

Figure 1: This figure depicts the optimal classifier for labelling a 1-dimensional dataset compared to Gaussian classifiers as well as the normalized kernel function basis, $k_{\{1,1\}}(5, z)$, using an assortment of different P matrices.

matrices where $P_{1,2} = P_{2,1} = P_{2,2} = 0$ and $P_{1,1} = A_i$ for $i = 1, \dots, 4$ where

$$A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 0 \\ 0 & .1 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, A_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{12}$$

In the first three cases the monomial basis is of degree 1, while in the fourth case the monomial basis is of degree 2. These different matrices all illustrate changes in slope which occur at the training datum. In addition, Figure 1(b) basis functions vary significantly based on the selected positive matrix.

4. Properties of the tessellated class of kernel functions

In this section we prove that TK functions are both continuous and universal, even in the case of degree $d = 0$. Let us begin by recalling that for any $P \geq 0$ and $N(z, x)$,

$$k(x, y) = \int_Z N(z, x)^T P N(z, y) dz$$

is a positive kernel and recall that for the TKs, we have

$$N(z, x) = \begin{bmatrix} Z_d(z, x) I_{S_1}(z, x) \\ Z_d(z, x) I_{S_2}(z, x) \end{bmatrix}.$$

By the representer theorem this implies that the classifiers consists of functions of the form

$$f(y) = \sum_{i=1}^{m} \alpha_i \int_{\mathcal{X}} N(x_i, z)^T P N(y, z) dz.$$

The following theorem establishes that such functions are necessarily continuous.

Theorem 4 *Suppose that for $a < b \in \mathbb{R}^n$, $X = \mathcal{X} = [a,b]$, $P \geq 0$, N is as defined in Eqn. (9) for and $d \geq 0$ and k is as defined in Eqn. (6). Then for any $\{x_i\}_{i=1}^m$, the function*

$$f(z) = \sum_{i=1}^{m} \alpha_i k(x_i, z),$$

is continuous.

Proof Partition P as follows

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} > 0.$$

To prove that $f(z)$ is continuous we need only prove that $k(x,y)$ is continuous. Applying Lemma 4 we may define $k(x,y)$ as

$$k(x,y) = \Big\{ k_\beta(x,y) \quad \text{if } \ (x,y) \in X_\beta. \tag{13}$$

where the k_β are polynomials defined as

$$k_\beta(x,y) = \prod_{i:\beta_i=0} \int_{z_i=x_i}^{b_i} \prod_{j:\beta_j=1} \int_{z_j=y_j}^{b_j} Z_d(z,x)^T Q_1 Z_d(z,y) dz + k_0(x,y),$$

where $Q_1 = P_{11} - P_{12} - P_{21} + P_{22}$, and $k_0(x,y)$ is a polynomial and thus continuous. To expand $k_\beta(x,y)$, we use multinomial notation for the monomials in Z_d. Specifically, we index the elements of Z_d as $Z_d(x,z)_i = x^{\alpha_i} z^{\gamma_i}$ where $\alpha_i, \gamma_i \in \mathbb{N}^n$ for $i = 1, \cdots, q$. Then

$$\begin{aligned} \prod_{i:\beta_i=0} \int_{z_i=x_i}^{b_i} \prod_{j:\beta_j=1} \int_{z_j=y_j}^{b_j} Z_d(z,x)^T Q_1 Z_d(z,y) dz &= \sum_{k,l} (Q_1)_{k,l} \prod_{i:\beta_i=0} \int_{z_i=x_i}^{b_i} \prod_{j:\beta_j=1} \int_{z_j=y_j}^{b_j} x^{\alpha_k} z^{\gamma_k} z^{\gamma_l} y^{\alpha_l} dz \\ &= \sum_{k,l} (Q_1)_{k,l} x^{\alpha_k} y^{\alpha_l} \prod_{i:\beta_i=0} \int_{z_i=x_i}^{b_i} \prod_{j:\beta_j=1} \int_{z_j=y_j}^{b_j} z^{\gamma_k + \gamma_l} dz. \end{aligned} \tag{14}$$

Expanding the integrals in (14), each has the form

$$\begin{aligned} \prod_{i:\beta_i=0} \int_{z_i=x_i}^{b_i} \prod_{j:\beta_j=1} \int_{z_j=y_j}^{b_j} z^{\alpha} dz &= \prod_{i:\beta_i=0} \int_{z_i=x_i}^{b_i} z_i^{\alpha_i} dz_i \prod_{j:\beta_j=1} \int_{z_j=y_j}^{b_j} z_j^{\alpha_j} dz_j \\ &= \prod_{i:\beta_i=0} \frac{1}{\alpha_i + 1}(b_i - x_i^{\alpha_i+1}) \prod_{j:\beta_j=1} \frac{1}{\alpha_j+1}(b_j - y_j^{\alpha_j+1}) \\ &= \prod_{k=1}^{n} \frac{1}{\alpha_k + 1} \prod_{i:\beta_i=0} (b_i - x_i^{\alpha_i+1}) \prod_{j:\beta_j=1} (b_j - y_j^{\alpha_j+1}) \\ &= \prod_{j=1}^{n} \frac{b_j - (\frac{1}{2}(x_j + y_j + |x_j - y_j|))^{\alpha_j+1}}{\alpha_j + 1} \end{aligned}$$

where we have used the fact that

$$\frac{1}{2}(x+y+|x-y|) = \begin{cases} x & x>y \\ y & y>x. \end{cases}$$

We conclude that $k(x,y)$ is the product and summation of continuous functions and therefore k and the resulting classifiers are both continuous. ■

In addition to continuity, we show that any kernel of this form for $P>0$ has the universal property. Recall the following definition of universality.

Definition 5 *A kernel $k:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ is said to be universal on the compact metric space $\mathcal{X}$ if it is continuous and there exists an inner-product space $\mathcal{W}$ and feature map, $\Phi:\mathcal{X}\to\mathcal{W}$ such that $k(x,y)=\langle\Phi(x),\Phi(y)\rangle_{\mathcal{W}}$ and where the unique Reproducing Kernel Hilbert Space (RKHS),*

$$\mathcal{H} := \{f \;:\; f(x)=\langle v,\Phi(x)\rangle,\; v\in\mathcal{W}\}$$

with associated norm $\|f\|_{\mathcal{H}} := \inf_v\{\|v\|_{\mathcal{W}} \;:\; f(x)=\langle v,\Phi(x)\rangle\}$ is dense in $\mathcal{C}(\mathcal{X}) := \{f : \mathcal{X}\to\mathbb{R} \; : f is\ continuous\}$ where $\|f\|_{\mathcal{C}} := \sup_{x\in X}|f(x)|$.

The following theorem shows that any TK with $P>0$ is necessarily universal.

Theorem 6 *Suppose k is as defined in Eqn. (6) for some $P>0$, $d\in\mathbb{N}$ and N as defined in Eqn. (9). Then k is universal for $X=\mathcal{X}=[a,b]$, $a<b\in\mathbb{R}^n$.*

Proof Without loss of generality, we assume $X=\mathcal{X}=[0,1]^n$. If $P>0$, then there exist ϵ_i such that $P=P_0+\sum_i\epsilon_iP_i$ where $P_0>0$ and

$$P_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} e_1, 0, \dots, 0 \end{bmatrix}$$

where $\{e_1\}$ is the first canonical basis of $\mathbb{R}^n$. In this case

$$k(x,y) = k_0(x,y) + \underbrace{\prod_{i=1}^{n}\epsilon_i\min\{x_i,y_i\}}_{k_1(x,y)},$$

where k_0 is a positive kernel. Since the hypothesis space satisfies the additive property (See Wang et al. (2013) Borgwardt et al. (2006)), if k_1 is a universal kernel, then k is a universal kernel.

Recall that for a given kernel, the hypothesis space, $\mathcal{H}$, can be characterized as the closure of $\text{span}\{k(y,\cdot) \;:\; y\in\mathcal{X}\}$. Now, consider

$$\text{span}\{k_1(y,\cdot) \;:\; y\in\mathcal{X}\}$$

which consists of all functions of the form

$$f(x) = \sum_j c_j \prod_{i=1}^{n} f_{ij}(x_i)$$

where

$$f_{ij}(x) = \min\{x, y_{ij}\} = \begin{cases} x, & \text{if } x \le y_{ij} \\ y_{ij}, & \text{otherwise.} \end{cases}$$

For $n = 1$, we may construct a triangle function centered at y_2 as

$$f(x) = \sum_{i=1}^{3} \alpha_i k(y_i, x) = \begin{cases} 0, & \text{if } x < y_1 \\ \delta(x - y_1), & \text{if } y_1 \le x < y_2 \\ 1 - \delta(x - y_2), & \text{if } y_2 \le x < y_3 \\ 0, & \text{if } y_3 < x \end{cases}$$

where $\delta = y_1 - y_2 = y_2 - y_3$, and

$$\alpha_1 = -\delta, \quad \alpha_2 = 2\delta, \quad \alpha_3 = -\delta.$$

By taking the product of triangle functions in each dimension, we obtain the pyramid functions which are known to be dense in the space of continuous functions on a compact domain (See Shekhtman (1982)). We conclude that k_1 is a universal kernel and hence k is universal.

■

Notation For convenience, we denote the TKs by saying $k \in \mathcal{K}_T^d$ if there exists some $P \ge 0$ such that k is as defined in Equation (6) where N is as defined in Eqn (9) using Z_d.

This theorem implies that even if the degree of the polynomials is small, the kernel is still universal. Specifically, in the case when $n = 1$ and $d = 0$, the set $\mathcal{K}_T^0$ contains only three parameters (elements of P).

In addition, unlike the Gaussian kernel, elements of the kernel matrix associated with a TK may be negative. To demonstrate the necessity of negative entries for finding the optimal kernel matrix, we analytically solve the following SDP derived from Optimization Problem (5) which determines the optimal kernel matrix (K^*) for a specific kernel learning problem, but with no constraint on the form of the kernel function (other than it be PD).

$$\begin{aligned} &\min_{t\in\mathbb{R}, K\in\mathbb{R}^{m\times m}, \gamma\in\mathbb{R}, \nu\in\mathbb{R}^m, \delta\in\mathbb{R}^m} \quad t, \\ &\text{subject to: } \begin{pmatrix} G & e + \nu - \delta + \gamma y \\ (e + \nu - \delta + \gamma y)^T & t - C\delta^T e \end{pmatrix} \ge 0 \\ &\nu \ge 0, \qquad \delta \ge 0, \qquad K \ge 0, \qquad \text{trace}(K) = m, \qquad G_{i,j} = y_i K_{i,j} y_j \end{aligned} \tag{15}$$

The following theorem finds an analytic solution for a specific instantiation of this problem.

Theorem 7 *Let $y_i \in \{1, -1\}$ for $i = 1, \cdots, m$ and $Y = diag(y)$. If $C \ge \frac{2}{m}$, then the solution to Optimization Problem 15 is,*

$$\nu^* = 0, \quad \gamma^* = -\sum_{i=1}^{m} \frac{y_i}{m}, \quad \delta^* = 0, \quad t^* = \frac{\|e - \gamma^* y\|_2}{m}$$

and, $K^* = \frac{m}{\|e+\gamma^* y\|_2^2} Y(e + \gamma^* y)(e + \gamma^* y)^T Y$.

Proof We first show that $K^* = U\Sigma U^T$, where

$$U = Y\begin{bmatrix} \frac{(e+\nu^*-\delta^*+\gamma^* y)}{\|(e+\nu^*-\delta^*+\gamma^* y)\|_2} & \cdots \end{bmatrix}, \Sigma = \begin{bmatrix} m & 0 \\ 0 & 0 \end{bmatrix}.$$

Optimization Problem (15) is equivalent to

$$\min_{K\in\mathbb{R}^{m\times m}, \gamma\in\mathbb{R}, \nu\in\mathbb{R}^m, \delta\in\mathbb{R}^m} \quad (e+\nu-\delta+\gamma y)^T (YKY)^{-1}(e+\nu-\delta+\gamma y) + 2C\delta^T e, \tag{16}$$
$$\text{subject to:} \quad \nu \geq 0, \qquad \delta \geq 0, \qquad K \geq 0, \qquad \text{trace}(K) = m.$$

This problem can be separated into subproblems as

$$\min_{\substack{\gamma\in\mathbb{R}, \nu\in\mathbb{R}^m, \delta\in\mathbb{R}^m \\ \nu\geq 0,\, \delta\geq 0,}} \quad \min_{\substack{K\in\mathbb{R}^{m\times m}, \\ K\geq 0,\, \text{trace}(K)=m}} \quad (e+\nu-\delta+\gamma y)^T (YKY)^{-1}(e+\nu-\delta+\gamma y) + 2C\delta^T e.$$

Now, for any feasible K, we have that $K \geq 0$ and $\bar{\sigma}(K) \leq m$ and hence

$$\begin{aligned}(e+\nu-\delta+\gamma y)^T (YKY)^{-1}(e+\nu-\delta+\gamma y) &\geq \frac{1}{\bar{\sigma}(K)}\|e+\nu-\delta+\gamma y\|_2^2 \\ &\geq \frac{1}{m}\|e+\nu-\delta+\gamma y\|_2^2.\end{aligned}$$

Now, Let $K = U\Sigma U^T$, where

$$U = Y\begin{bmatrix} \frac{(e+\nu^*-\delta^*+\gamma^* y)}{\|(e+\nu^*-\delta^*+\gamma^* y)\|_2} & V \end{bmatrix}, \Sigma = \begin{bmatrix} m & 0 \\ 0 & 0 \end{bmatrix}.$$

and V is any unitary completion of the matrix U. Then $K \geq 0$, $\text{trace}(K) = m$ and

$$\begin{aligned}&(e+\nu-\delta+\gamma y)^T Y\,(YKY)^{-1} Y^T (e+\nu-\delta+\gamma y) \\ &= (e+\nu-\delta+\gamma y)^T Y \left(Y \begin{bmatrix} \frac{(e+\nu-\delta+\gamma y)}{\|(e+\nu-\delta+\gamma y)\|_2} \, V \end{bmatrix} \begin{bmatrix} m & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{(e+\nu-\delta+\gamma y)}{\|(e+\nu-\delta+\gamma y)\|_2} \, V \end{bmatrix}^T Y \right)^{-1} Y^T (e+\nu-\delta+\gamma y) \\ &= \frac{\|(e+\nu-\delta+\gamma y)\|_2^2}{m}.\end{aligned}$$

We conclude that this K solves the first sub-problem and hence Optimization Problem (15) reduces to

$$\min_{\substack{\gamma\in\mathbb{R}, \nu\in\mathbb{R}^m, \delta\in\mathbb{R}^m \\ \nu\geq 0,\, \delta\geq 0,}} \quad \frac{\|(e+\nu-\delta+\gamma y)\|_2^2}{m} + 2C\delta^T e. \tag{17}$$

Now let $\nu^*, \delta^*, \gamma^*$ be as defined in the theorem statement. For the convex objective

$$f(\delta, \nu, \gamma) = \frac{\|e+\nu-\delta+\gamma y\|_2^2}{m} + 2C\delta^T e$$

we have that

$$\frac{\partial f}{\partial \nu_i}(\nu^*, \delta^*, \gamma^*) = \frac{2 + 2\bar{y}y_i}{m} \geq \frac{2 + -2(1)(1)}{m} \geq 0,$$

and for $C \geq \frac{2}{m}$

$$\frac{\partial f}{\partial \delta_i}(\nu^*, \delta^*, \gamma^*) = \frac{2mC - 2 - 2\bar{y}y_i}{m} \geq \frac{4 - 2 - 2(1)(1)}{m} \geq 0.$$

Finally

$$\frac{\partial f}{\partial \gamma}(\nu^*, \delta^*, \gamma^*) = \frac{1}{m}\sum_{i=1}^{m} 2y_i - 2\bar{y} = -2\bar{y} + 2\frac{1}{m}\sum_{i=1}^{m} y_i = 0.$$

Hence the KKT conditions are satisfied and since the optimization problem is convex, $(\nu^*, \delta^*, \gamma^*)$ is optimal. ■

This result shows that for binary labels, the optimal kernel matrix has an analytic solution. Furthermore, if we consider the case where $\sum_{i=1}^{m} y_i = 0$, then $\lambda^* = 0$ and hence $K^*_{i,j} = y_i y_j$. This implies that the optimal kernel matrix consists of an equal number of positive and negative entries - meaning that kernels functions with globally positive values will not be able to approximate the optimal kernel matrix well. Furthermore note that for large numbers of data points the value of C chosen will almost always be greater than $\frac{2}{m}$. For values of C less than $\frac{2}{m}$ we find numerically that the same kernel matrix is still optimal - only the values of δ^* and γ^* are different.

Next, we confirm numerically that Gaussian kernels perform poorly and polynomial/TK kernels perform well at approximating the kernel matrix. Specifically, we now pose the kernel learning problem solely as the ability to approximate the optimal kernel matrix for a given set of data and parameterization of kernels using both the element-wise matrix $\|\cdot\|_1$ and $\|\cdot\|_\infty$ norms.

$$\min_{k \in \mathcal{K}} \frac{\|K - K^*\|_1}{n^2} \quad s.t. \;\; K_{i,j} = k(x_i, x_j) \tag{18}$$

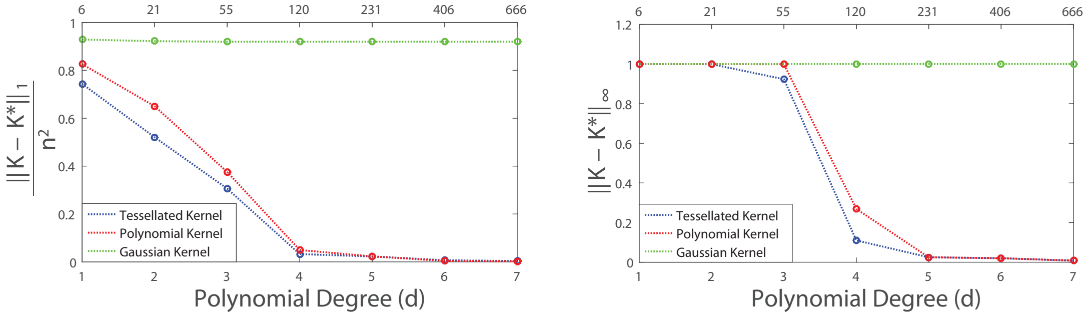
$$\min_{k \in \mathcal{K}} \|K - K^*\|_\infty \quad s.t. \;\; K_{i,j} = k(x_i, x_j) \tag{19}$$

We consider parameterizations of the Gaussian, polynomial and TKs so that $\mathcal{K} \in \{\mathcal{K}_G^\gamma,\ \mathcal{K}^d,\ \mathcal{K}_T^d\}$ where

$$\mathcal{K}^d := \{k \ : \ k(x,y) = Z_d(x)^T P Z_d(y) \ : \ P \geq 0\}$$

$$\mathcal{K}_G^\gamma := \{k \ : \ k(x,y) = \sum_{\gamma_i \in \gamma} \mu_i \mathrm{e}^{\frac{||x-y||_2^2}{\gamma_i}} \ : \ \mu_i > 0\}.$$

We now solve Optimization Problems (18) and (19) for $\mathcal{K}_G^\gamma$, $\mathcal{K}^d$,and $\mathcal{K}_T^d$ as a function of the degree of the polynomials and the number of bandwidths selected. For this test, we use the spiral data set with 20 samples and corresponsing labels such that $\sum_{i=1}^{m} y_i = 0$. Since



(a) $\frac{\|K-K^*\|_1}{n^2}$ for the TK and polynomial kernels of degree d and for a positive combination of m Gaussian kernels.

(b) $\|K-K^*\|_\infty$ for the TK and polynomial kernel of degree d and for a positive combination of m Gaussian kernels.

Figure 2: The objective of Optimization Problem 18 and 19 for the TK and polynomial kernel of degree d and for a positive combination of m Gaussian kernels with bandwidths ranging from .01 to 10. The number of bandwidths is selected so that the number of decision variables match in the Gaussian and in the TK function case.

half of the entries in K^* are -1, and since the Gaussian kernel is globally positive, it is easy to see that for $\mathcal{K} = \mathcal{K}_G^\gamma$ the minimum objective values of Optimization Problems (18) and (19) are lower bounded by 0.5 and 1 respectively, irrespective of the choice of bandwidths, γ_i. In Figs. 2(a) and 2(b) we numerically show the change in the objective value of Optimization Problems (18) and (19) for the optimal Gaussian, Polynomial, and TK function as we increase the complexity of the kernel function. For the TK and polynomial kernel functions we increase the complexity of the kernel function by increasing the degree of the monomial basis while scaling the x-axis to ensure equivalent computational complexity. We show numerically that, in this case, the Gaussian Kernel saturates with an objective value significantly larger than the lower bound of 0.5 for the 1-norm and exactly at 1 for the ∞-norm (the projected lower bound). Meanwhile, as the degree increases, both polynomial and TKs are able to approximate the kernel matrix arbitrarily well, with almost no error at degree $d = 7$. Furthermore, the TKs converge faster.

Having demonstrated finite convergence numerically for both polynomial kernels and TKs, we prove that this finite convergence always holds for an arbitrary optimal kernel matrix, K^*. Specifically, we prove this result for polynomial kernels, and use the following lemma to show that polynomial kernels are a subset of TKs.

Lemma 8 $\mathcal{K}^d \subset \mathcal{K}_T^d$

Proof If $k_p \in \mathcal{K}^d$, there exists a $P_1 \geq 0$ such that $k_p(x,y) = Z_d(x)^T P_1 Z_d(y)$. Now let J be the matrix such that $JZ_d(z,x) = Z_d(x)$ and define

$$P = \frac{1}{b-a}\begin{bmatrix} J^T P_1 J & J^T P_1 J \\ J^T P_1 J & J^T P_1 J \end{bmatrix} \geq 0.$$

Now let k be as defined in Equation (6). Then $k \in \mathcal{K}_T^d$ and

$$\begin{aligned}
k(x,y) =& \frac{1}{b-a} \int_{p^*(x,y)}^b Z_d(z,x)^T J^T \left(P_1 - P_1 - P_1 + P_1\right) J Z_d(z,y) dz \\
&+ \frac{1}{b-a} \int_x^b Z_d(z,x)^T J^T \left(P_1 - P_1\right) J Z_d(z,y) dz \\
&+ \frac{1}{b-a} \int_y^b Z_d(z,x)^T J^T \left(P_1 - P_1\right) J Z_d(z,y) dz \qquad (20) \\
&+ \int_a^b Z_d(z,x)^T J^T P_1 J Z_d(z,y) dz \\
=& \frac{1}{b-a} \int_a^b Z_d(x)^T P_1 Z_d(y) dz \\
=& k_p(x,y). \qquad (21)
\end{aligned}$$

■

We will now prove that polynomial kernels can generate a kernel matrix which is arbitrarily close to any given kernel matrix. Since Lemma 8 proved that polynomial kernels are a subset of the TKs the following theorem holds for TKs as well.

Theorem 9 *For any kernel matrix K^* and any finite set $\{x_i\}_{i=1}^m$, there exists a $d \in \mathbb{N}$ and $k \in \mathcal{K}^d$ such that if $K_{i,j} = k(x_i, x_j)$, then $K = K^*$.*

Proof Since $K^* \geq 0$, $K^* = M^T M$ for some M. Since $k \in \mathcal{K}^d$, k has the form

$$k(x,y) = Z_d(x)^T P Z_d(y).$$

Choose $P = Q^T Q$ where, using multivariate polynomial interpolation (as in Gasca and Sauer (2001))), for sufficiently large d, we may choose Q such that

$$Q \begin{bmatrix} Z_d(x_1) & \cdots & Z_d(x_m) \end{bmatrix} = M.$$

Partition M as

$$M = \begin{bmatrix} m_1 & \cdots & m_m \end{bmatrix}.$$

Then $Q Z_d(x_i) = m_i$ and hence

$$\begin{aligned}
K_{ij} &= Z_d(x_i)^T Q^T Q Z_d(x_j) \\
&= m_i^T m_j \\
&= K_{ij}^*.
\end{aligned}$$

■

5. SDP Formulation of the Kernel Learning Problem

Section 2 detailed general optimization methods by which we may search for an optimal kernel function, $k \in \mathcal{K}$, given that the set of kernel functions has a linear parameterization. We will now formulate specific methods for learning an optimal TK function using either the SDP method of Optimization Problem (5), or using a two-step method like SimpleMKL. Using the representation of TKs ($\mathcal{K} = \mathcal{K}_T^d$) following Theorem 4 in Section 4, Optimization Problem (4) may be expressed as

$$\begin{aligned}
&\min_{t\in\mathbb{R},\gamma\in\mathbb{R},\nu\in\mathbb{R}^m,\delta\in\mathbb{R}^m} \quad t, \\
&\text{subject to:} \begin{pmatrix} G(P) & e+\nu-\delta+\gamma y \\ (e+\nu-\delta+\gamma y)^T & t-\frac{2}{m\lambda}\delta^T e \end{pmatrix} \geq 0 \\
&\nu \geq 0, \qquad \delta \geq 0, \qquad P \geq 0, \qquad \operatorname{trace}(P) \leq 1 \\
&G_{ij}(P) = \sum_{i,j=1,2} \sum_{k,l} (P_{i,j})_{k,l}\, x_i^{\alpha_k} x_j^{\alpha_l} \int_{\mathcal{X}_{ij}} z^{\gamma_k+\gamma_l} dz y_i y_j.
\end{aligned} \tag{22}$$

The set $\mathcal{X}_{ij}$ is as defined in the proof of Lemma 3. The α_k and γ_k are the multinomials defined in the proof of 4. The integral term in the expression for G, then, is integration of a monomial over a rectangular domain and may be calculated a priori based on the choice of domain. Optimization Problem (22), then, is an SDP and can therefore be solved efficiently using standard SDP solvers such as ApS (2015). Note that we use the trace constraint to ensure the kernel function is bounded.

Typically SDP problems require roughly p^2n^2 number of operations, where p is the number of decision variables and n is the dimension of the SDP constraint (Doherty et al. (2004)). The number of decision variables in (22) is moderate, increasingly linearly in the number of training data points and the size of P. However, this optimization problem has a semi-definite matrix constraint whose dimension is linear in m. As we will see in Section 7, this limits the amount of training data which can be processed using Optimization Problem (22). To improve the scability of the algorithm, we therefore turn to variations on SimpleMKL.

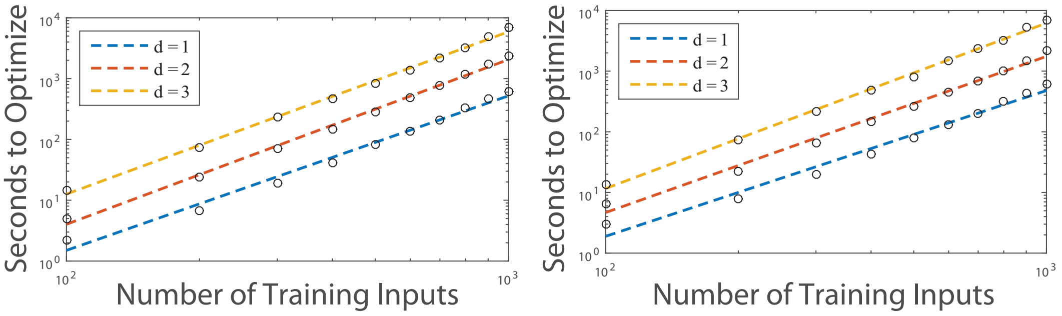
6. SimpleMKL Formulation of the Kernel Learning Problem

Recall that SimpleMKL searches for an optimal linear combination of kernel functions - it returns a vector of weights μ, on the a priori selected kernel functions. Here we discuss how SimpleMKL as implemented in Rakotomamonjy et al. (2008) can be used to find optimal combinations of TK functions that perform well in practice.

Specifically, we randomly generating a set of l positive semi-definite matrices, P_i for $i = 1, \ldots, l$ and use SimpleMKL to find the optimal linear combination of the TKs defined by each matrix P_i.

While the current use of randomly generated matrices is somewhat heuristic, it may be avoided through the development of a dedicated two-step algorithm - wherein the first step optimized α for a fixed P and the second step fixes α and searches over the positive matrices.

We will next consider a complexity analysis of both the SDP and SimpleMKL implementations.



(a) Complexity Scaling for Identification of Circle

(b) Complexity Scaling for Identification of Spiral

Figure 3: Log-Log Plot of Computation Time vs number of training data for 2-feature kernel learning.

7. Implementation and complexity analysis

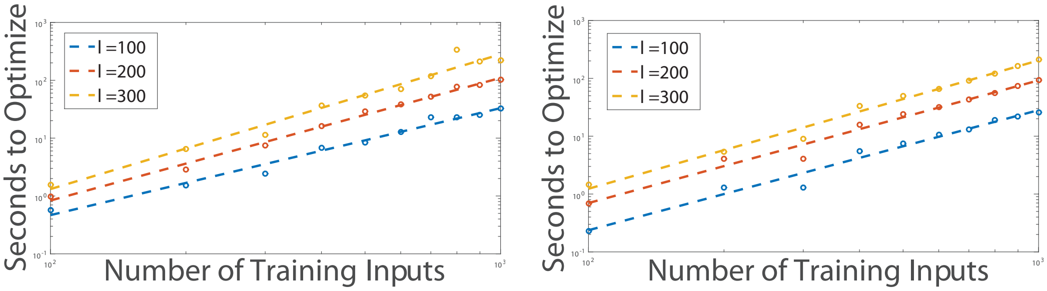
In this paper, we have proposed a new class of kernel functions defined by piecewise polynomials. In this section we analyze the complexity of Optimization Problem (22) with respect to the number of training points as well as the selected degree of the TK function.

The constraint that the kernel be a positive TK can be expressed as an LMI constraint with variables P_{ij}. Using Optimization Problem (22), if $P \in \mathbb{R}^{q \times q}$, and m is the number of training data, with a Mosek implementation, we find experimentally that the complexity of the resulting SDP scales as approximately $m^{2.6} + q^{1.9}$ as can be seen in Fig. 3 and is similar to the complexity of other methods such as the hyperkernel approach in Ong et al. (2005). These scaling results are for training data randomly generated by two standard 2-feature example problems (circle and spiral - See Fig. 6) for degrees $d = 1$, 2, 3 and where d defines the length of Z_d (and hence q) which is the vector of all monomials in 2 variables of degree d or less.

Note that the length of Z_d scales with the degree and number of features, n, as $q = \frac{(n+d-1)!}{n!d!}$. For a large number of features and high degree, the size of Z_d will become unmanageably large. Note, however, that, as indicated in the Section 4, even when $d = 0$, the kernels are universal.

Using SimpleMKL with the TK we need to select a number of random matrices to generate. If we have l random positive semi-definite matrices and m training data points then we find experimentally that the complexity of the resulting SDP scales as approximately $m^{2.1} + l^{1.6}$ as can be seen in Fig. 4. These scaling results are, as in the results for the SDP method, for training data randomly generated by two standard 2-feature example problems (circle and spiral - See Fig. 6). We select the number of training data m, to vary between 100 and 1000 points and select the number of random matrices to be $l = 100, 200, 300$.

Note that the complexity of the SimpleMKL version is largely independent of the selected degree of the polynomial. However, a larger degree means that the matrices P are larger, and therefore a larger number of random positive semi-definite matrices, l, should be selected. Recall again, that all of the TK are universal, therefore, so any of the positive semi-definite matrices parameterize a universal kernel.



(a) Complexity Scaling for Identification of Circle using SimpleMKL and TKs

(b) Complexity Scaling for Identification of Spiral using SimpleMKL and TKs

Figure 4: Log-Log Plot of Computation Time vs number of training data for 2-feature kernel learning using SimpleMKL and TKs.

8. Accuracy and comparison with existing methods

In this section, we evaluate the relative accuracy of SDP and SimpleMKL implementations.

To evaluate accuracy, we applied 5 variations of the kernel learning problem to 5 randomly selected benchmark data sets from the UCI Machine learning Data Repository - Liver, Cancer, Heart, Pima, and Ionosphere. In all evaluations of Test Set Accuracy (TSA), the data is partitioned into 80% training data and 20% testing and this partition is repeated 30 times to obtain 30 sets of training and testing data. For all numerical tests we use the soft-margin problem with regularization parameter C, where C is selected from a set of values picked a priori by 5-fold cross-validation. To perform 5-fold cross-validation we split the training data set into five groups, solve the optimization problem using each potential value of C on four of the five groups and test the optimal classifier performance on the remaining group. We repeat this process using each of the five groups as the test set and select the value of C which led to the best average performance.

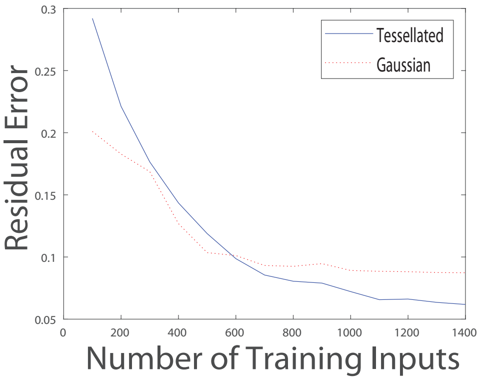


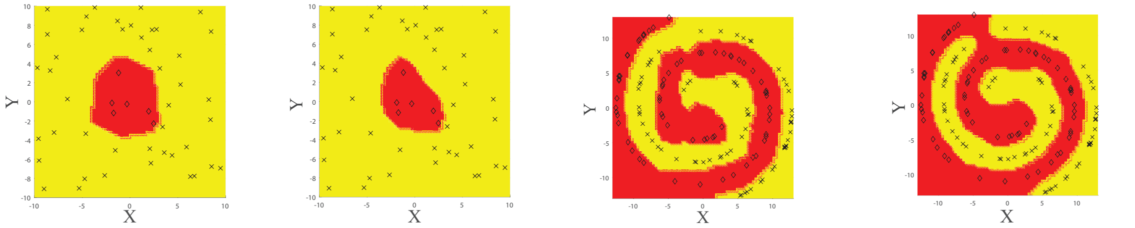
Figure 5: TSA compared with SimpleMKL for spiral dataset with artificial additive noise.

The 5 variations on the kernel learning problem are

[Tessellated] We use the SDP algorithm in (22) using $d = 1$ (Except Ionosphere, which uses $d = 0$); To determine the integral in (22), we first scaled the data so that $x_i \in [0, 1]^n$, and then set $\mathcal{X} := [0 - \epsilon, 1 + \epsilon]^n$, where $\epsilon > 0$ was chosen by 5-fold cross-validation.

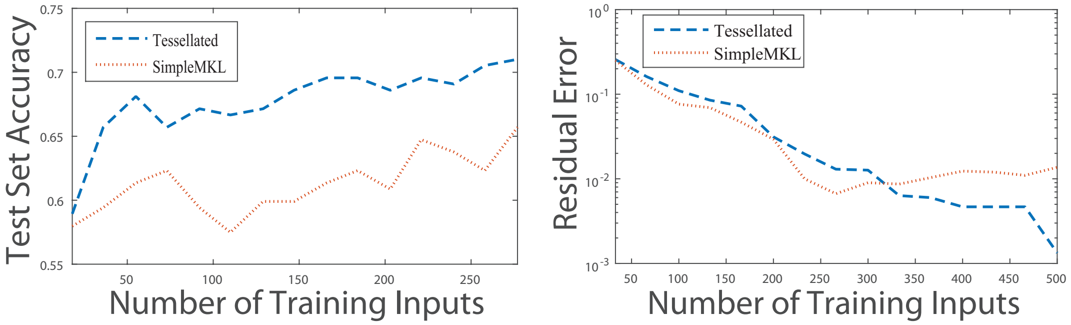
[SimpleMKL] We use SimpleMKL with a standard selection of Gaussian and polynomial kernels with bandwidths arbitrarily chosen between .5 and 10 and polynomial degrees one through three - yielding approximately $13(n + 1)$ kernels;

[SimpleMKL Tess.] We randomly generated a sequence of 300 positive semidefinite matrices and use these as the SimpleMKL library of kernels;



(a) Circle with T [n=50] (b) Circle with S [n=50] (c) Spiral with T [n=150] (d) Spiral with S [n=150]

Figure 6: Discriminant Surface for Circle and Spiral Separator using Tesselated kernel [T] as Compared with SimpleMKL [S] for n training data.



(a) Average test set accuracy on the Liver dataset vs. the number of training data for the proposed method compared to SimpleMKL

(b) Semilog plot of residual error on generated 2D spiral data vs. number of training data for proposed method compared to SimpleMKL. .

Figure 7: Plots demonstrating the change in accuracy of the TK method and SimpleMKL with respect to the number of training inputs. The residual error is defined as 1-TSA where TSA is the test set accuracy.

[Combined] We combined the libraries in [SimpleMKL] and [SimpleMKL Tess.] into a single SimpleMKL implementation;
[Neural Net] We use 3 layer neural network with 50 hidden layers using MATLABs patternnet implementation.

In Table 1, we see the average TSA for these four approaches as applied to several randomly selected benchmark data sets from the UCI Machine learning Data Repository. In all cases, the TK met or in some cases significantly exceeded the accuracy of SimpleMKL.

In addition to the standard battery of tests, we performed a secondary analysis to demonstrate the advantages of the TK class when the ratio of training data to number of features is high. For this analysis, we use the liver data set (6 features) and the spiral discriminant (Lang (1988)) with 2 features (x and y) (we also briefly examine the unit circle). For the liver data set, in Figure 8, we see a semilog plot of the residual error (i.e. 1-TSA) as the size of the training data increases as compared with SimpleMKL. This figure shows consistent improvement of the tessellated class over standard usage of SimpleMKL. For the spiral case, in Figure 8 we again see a semilog plot of the residual error as the size of the training data increases as compared with SimpleMKL. In this case, both methods converge

Table 1: TSA comparison for algorithms a), b), c), d), and e). The maximum TSA for each data set is bold. The average TSA, standard deviation of TSA and time to compute are shown below. m is size of dataset and n the number of features.

Data Set	Method	Accuracy	Time	Data Features
	TK	**72.32 ± 4.92**	95.75 ± 2.68	
Liver	SimpleMKL	65.51 ± 5.10	2.61 ± 0.42	m = 346
	SimpleMKL Tess.	70.58 ± 4.69	8.37 ± 0.30	n = 6
	Combined	70.53 ± 4.79	14.70 ± 0.76	
	Neural Net	66.32 ± 7.46	0.14 ± 0.04	
	TK	**97.18 ± 1.48**	636.17 ± 25.43	
Cancer	SimpleMKL	96.55 ± 1.34	14.74 ± 1.33	m = 684
	SimpleMKL Tess.	96.89 ± 1.43	45.84 ± 4.28	n = 9
	Combined	96.89 ± 1.42	65.08 ± 10.52	
	Neural Net	96.67 ± 1.30	0.18 ± 0.06	
	TK	83.46 ± 4.56	221.67 ± 29.63	
Heart	SimpleMKL	83.70 ± 4.77	3.09 ± 0.19	m = 271
	SimpleMKL Tess.	**84.38 ± 4.34**	55.48 ± 2.67	n = 13
	Combined	83.64 ± 4.54	13.23 ± 2.70	
	Neural Net	78.64 ± 5.19	0.12 ± 0.01	
	TK	76.32 ± 3.10	1211.66 ± 27.01	
Pima	SimpleMKL	76.00 ± 3.33	19.04 ± 2.33	m=769
	SimpleMKL Tess.	**76.75 ± 2.81**	34.65 ± 23.28	n = 8
	Combined	76.57 ± 2.72	96.20 ± 30.42	
	Neural Net	75.35 ± 2.98	0.24 ± 0.19	
	TK	**93.24 ± 3.04**	6.69 ± 0.27	
Ionosphere	SimpleMKL	92.16 ± 2.78	26.24 ± 2.78	m = 352
	SimpleMKL Tess.	87.65 ± 2.88	8.28 ± .16	n = 34
	Combined	92.16 ± 2.78	50.77 ± 2.98	
	Neural Net	90.85 ± 3.42	0.16 ± 0.02	

well with the TK showing significant improvement over SimpleMKL only for very large training data sets.

To further explore this scenario, we generated a new, 1400 point training data set with additive noise of zero mean and $\sigma = .1$. The results are seen in Figure 5. In this case we see that the TKs significantly outperform SimpleMKL, beginning at 600 data points.

Finally, as illustration, we plotted the discriminant surface for both the spiral and unit circle data sets using both the TK and SimpleMKL using 150 training data points. These 2D surfaces are found in Figure 6.

9. Conclusion

In this paper, we have proposed a new class of universal kernel functions. This set of kernels can be parameterized directly using positive matrices or indirectly using positive coefficients combined with randomly generated positive matrices. Furthermore, any element of this class is universal in the sense that the hypothesis space is dense in L_2, giving it comparable performance to and properties of the Gaussian kernels. However, unlike the Gaussian, the TK does not require a set of bandwidths to be chosen a priori. Indeed, by increasing the degree of the monomial basis, we have shown that TKs can approximate any kernel matrix arbitrarily well.

We have demonstrated the effectiveness of the tessellated class of kernel on several datasets from the UCI repository. We have shown that the computational complexity is comparable to other SDP-based kernel learning methods. Furthermore, by using a randomized basis for the positive matrices, we have shown that the tessellated class can be readily integrated with existing multiple kernel learning algorithms such as Simple MKL - yielding similar results with less computational complexity. In most cases, either the optimal TK, or the MKL learned sub-optimal tessellated kernel will out perform or match an MKL approach using Gaussian and polynomial kernels with respect to the Test Set Accuracy. Finally, we note that this universal class of kernels can be trivially extended to matrix-valued kernels for use in, e.g. multi-task learning Caponnetto et al. (2008).

References

F. Alizadeh, J.-P. Haeberly, and M. Overton. Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results. *SIAM Journal on Optimization*, 1998.

MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28).*, 2015.

K.M. Borgwardt, A. Gretton, M.J. Rasch, H. Kriegel, B. Schölkopf, and A.J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 2006.

A. Caponnetto, C. Micchelli, M. Pontil, and Y. Ying. Universal multi-task kernels. *Journal of Machine Learning Research*, 2008.

M. Collins and N. Duffy. Convolution kernels for natural language. In *Advances in neural information processing systems*, 2002.

C. Cortes, M. Mohri, and A. Rostamizadeh. Learning non-linear combinations of kernels. In *Advances in neural information processing systems*, 2009.

C. Cortes, M. Mohri, and A. Rostamizadeh. Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 2012.

A.C. Doherty, P.A. Parrilo, and F.M. Spedalieri. Complete family of separability criteria. *Physical Review A*, 2004.

E. Eskin, J. Weston, W. Noble, and C. Leslie. Mismatch string kernels for SVM protein classification. In *Advances in neural information processing systems*, 2003.

K. Gai, G. Chen, and C.-S. Zhang. Learning kernels with radiuses of minimum enclosing balls. In *Advances in neural information processing systems*, 2010.

T. Gärtner, P. Flach, and S. Wrobel. On graph kernels: Hardness results and efficient alternatives. In *Learning Theory and Kernel Machines*. 2003.

M. Gasca and T. Sauer. On the history of multivariate polynomial interpolation. In *Numerical Analysis: Historical Developments in the 20th Century*, pages 135–147. 2001.

I. Gohberg, S. Goldberg, and M. Kaashoek. *Classes of linear operators*. Birkhäuser, 2013.

M. Gönen and E. Alpaydin. Localized multiple kernel learning. In *Proceedings of the International Conference on Machine learning*, 2008.

M. Gönen and E. Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 2011.

D. Haussler. Convolution kernels on discrete structures. Technical report, University of California in Santa Cruz, 1999.

A. Jain, S. Vishwanathan, and M. Varma. Spf-gmkl: generalized multiple kernel learning with a million kernels. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, 2012.

G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine learning research*, 2004.

K. Lang. Learning to tell two spirals apart. In *Proceedings of the Connectionist Models Summer School*, 1988.

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2002.

C. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 2006.

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

V. Paulsen and M. Raghupathi. *An introduction to the theory of reproducing kernel Hilbert spaces*. Cambridge University Press, 2016.

M.M. Peet, A. Papachristodoulou, and S. Lall. Positive forms and stability of linear time-delay systems. *SIAM Journal on Control and Optimization*, 2009.

A. Rakotomamonjy, F.R. Bach, S. Canu, and Y. Grandvalet. Simplemkl. *Journal of Machine Learning Research*, 2008.

B. Recht. *Convex Modeling with Priors*. PhD thesis, Massachusetts Institute of Technology, 2006.

B. Schölkopf, A.J. Smola, and F. Bach. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

B. Shekhtman. Why piecewise linear functions are dense in c [0, 1]. *Journal of Approximation Theory*, 1982.

SĆ Sonnenburg, S. Henschel, C. Widmer, J. Behr, A. Zien, F. de Bona, A. Binder, C. Gehl, V. Franc, et al. The shogun machine learning toolbox. *Journal of Machine Learning Research*, 2010.

N. Subrahmanya and Y. Shin. Sparse multiple kernel learning for signal processing applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

H Sun. Mercer theorem for rkhs on noncompact sets. *Journal of Complexity*, 2005.

H. Wang, Q. Xiao, and D. Zhou. An approximation theory approach to learning with ℓ_1 regularization. *Journal of Approximation Theory*, 2013.

E. Zanaty and A. Afifi. Support vector machines (SVMs) with universal kernels. *Applied Artificial Intelligence*, 2011.