A Convex Parametrization of a New Class of Universal Kernel Functions for use in Kernel Learning

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

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

Editor:

Abstract

We propose a new class of universal kernel functions which admit a linear parametrization using positive semidefinite matrices. These kernels are generalizations of the Sobolev kernel, are parameterized by piecewise-polynomial functions, and do not require an a priori selection of basis kernels (as in MKL). 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. The number of parameters used to define these TKs is determined by the length of an associated monomial basis. However, even for a single monomial basis function (e.g. a constant yields 4 parameters) and for any choice of parameters, the TKs are universal in the sense that the resulting discriminants occupy a hypothesis space which is dense in $L_2$. This implies that the use of TKs for learning the kernel (aka kernel learning) can eliminate the need for Gaussian kernels and thus the associated ad-hoc and heuristic approaches to the choice of bandwidth - a conclusion verified through extensive numerical testing on soft margin Support Vector Machine (SVM) problems. Furthermore, our results imply that when the ratio of the number of training data to features is high, this 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 using a randomized basis for the positive matrix parameters.

Keywords: Kernel Functions, Multiple Kernel Learning,

1. Introduction

This paper addresses the problem of automated selection of an optimal kernel function for a given kernel-based machine learning problem (i.e. soft margin SVM). Kernel functions implicitly define a linear parametrization of nonlinear candidate maps $y = f(x)$ from features $x$ 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 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 methods rely on kernel set operations (addition, multiplication, convolution) to generate large numbers of parameterized kernel functions as in Cortes et al. (2009). When the parameterization is non-convex, gradient-based methods find local minima and include GMKL as introduced in Jain et al. (2012). Other variations include LMKL Gönen and Alpaydin (2008) (gating), polynomial kernel combinations and the Alignment and Centered Alignment MKL in, e.g. Cortes et al. (2012). For kernel learning, regularization is particularly important and interesting approaches to this problem include the group sparsity metric 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), $X$, it is said that a function $k : X \times X \rightarrow \mathbb{R}$ is a Positive Kernel (PK) if for any $N \in \mathbb{N}$ and any $\{x_i\}_{i=1}^N \subset X$, the matrix defined elementwise by $K_{ij} = k(x_i, x_j)$ is symmetric and Positive SemiDefinite (PSD).

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

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

with associated norm $\|f\|_H := \inf_{v \in W} \{ \|v\|_W : f(x) = \langle v, \Phi(x) \rangle \}$ is dense in $C(X) := \{ f : X \rightarrow \mathbb{R} : f is continuous \}$ where $\|f\|_C := \sup_{x \in X} |f(x)|$.

Note that for an given PD kernel, $\mathcal{H}$ exists, is unique, and can be characterized using the Riesz representation theorem as the closure of $\text{span}\{k(y, \cdot) : y \in X\}$ 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

$$\langle f, g \rangle_H := \sum_{i=1}^n \sum_{j=1}^m c_i d_j k(y_i, z_i).$$
A New Class of Universal Kernel Functions

The most well-known example of a universal kernel is the Gaussian (generalized in Zanaty and Afifi (2011)). However, most 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 are defined by polynomials and admit a convex parametrization. Specifically, if 

$$X := \{ x \in \mathbb{R}^n : x_i \in [a_i, b_i] \}$$

and the inequality > is defined by the positive orthant, we consider kernels

$$k(x,y) = \int_X I(z,x)Z(z,x)^T P Z(z,y)I(z,y)dz,$$

where

$$I(z,x) = \begin{cases} 1, & \text{if } z > x \\ 0, & \text{if } z \not> x \end{cases},$$

where

$$Z : X \times X \to \mathbb{R}^m$$

is a vector of monomials and 

$$P \in \mathbb{S}^m.$$ We show in Section 3 that if 

$$P > 0,$$

then

$$k$$

is a PK, continuous and universal.

Furthermore, 

$$f$$

can be represented using polynomials,

$$f_{i,\beta}(x),$$

on hyper-rectangular sets whose vertices are defined by the input data 

$$\{x_i\}_{i=1}^m.$$ 

$$f(z) = \sum_{i=1}^m \alpha_i y_i k(x_i, z) + b = f_{i,\beta}(z) \quad z \in X_{i,\beta} := \{ z : (x_i)_{j \geq z_j} \text{ for all } j: \beta_j = 0, \\
( x_i)_{k \geq z_k} \text{ for all } k: \beta_i = k \}$$

where the 

$$f_{i,\beta}$$

are polynomials. 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.

Specifically, we show that these kernels are generalizations of the class of Sobolev kernels in Paulsen and Raghupathi (2016) and have a hypothesis space dense in and 

$$C(Z)$$

and 

$$L_2(Z).$$ In contrast to the Gaussian kernels, however, this universal class of “tesselated kernels” (TKs) are linearly parametrization using positive matrices, need not be pointwise positive, and are piecewise-polynomial, making them significantly more robust and useful for learning the kernel. To illustrate, we show how this class of kernel can be rigorously incorporated into both the SDP kernel learning framework and the MKL framework for SVM soft margin problems. In the numerical results we illustrate this improved performance performance 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, for a given linear parameterization of kernel functions, in Subsections A and B, we then present two standard algorithms for solving the kernel learning problem. These algorithms will then be applied in Section 3 to our class of Tessellated Kernels (TKs).

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 

$$i = 1, \ldots, m.$$ For a given “penalty” parameter 

$$C \in \mathbb{R}^+,$$

we define the linear 1-norm soft margin problem as

$$\min_{w \in \mathbb{R}^n, \zeta \in \mathbb{R}^m, b \in \mathbb{R}} \frac{1}{2} w^T w + C \sum_{i=1}^m \zeta_i$$

s.t. 

$$y_i (w^T x_i + b) \geq 1 - \zeta_i, \quad (1)$$
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 primal problem becomes,
\[
  \min_{w \in \mathbb{R}^n, \zeta \in \mathbb{R}^m, b \in \mathbb{R}} \frac{1}{2} w^T w + C \sum_{i=1}^m \zeta_i
  \quad \text{s.t.} \quad y_i(\langle w, \Phi(x_i) \rangle + b) \geq 1 - \zeta_i,
\]
where, $k(x, y) = \langle \Phi(x), \Phi(y) \rangle$ and the classifier can be represented as,
\[
  f(z) = \text{sign} (\langle w, \Phi(z) \rangle + b).
\]

However, in some cases the function $\Phi(\cdot)$ may be infinite dimensional or computationally difficult to compute. In these cases it is beneficial to solve the SVM problem in it’s dual form,
\[
  \max_{\alpha \in \mathbb{R}^m} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \langle \Phi(x_i), \Phi(x_j) \rangle
  \quad \text{s.t.} \quad \sum_{i=1}^m \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C \quad \forall \ i = 1, ..., m,
\]
because $\langle \Phi(x_i), \Phi(x_j) \rangle = k(x_i, x_j)$, and we can solve the dual problem by computing $k(x_i, x_j)$ instead of explicitly computing $\Phi(x_i)$ and $\Phi(x_j)$. In this case, the 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 $y_j - \sum_{i=1}^m \alpha_i y_i k(x_i, x_i)$ 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 $\alpha$ and the kernel function.

Commonly used positive kernel functions include the gaussian kernel $k_1(x, y) = e^{-\beta ||x-y||^2}$, where $\beta$ 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. The selection of a kernel function, however, can have a large effect on the accuracy of the resulting classifier \( f \). We therefore consider methods for selecting an optimal kernel function from a convex set of kernel functions \( K \). In this case, we have the following convex optimization problem.

\[
\min_{k \in K} \max_{\alpha \in \mathbb{R}^m} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\
\text{s.t. } \sum_{i=1}^m \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C \quad \forall \ i = 1, ..., m.
\]

In the following two subsections we present two standard approaches to parameterizing \( K \) 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 \( K \) for a given set of candidate kernels \( \{k_i\}_{i=1}^l \) as

\[
K := \{k(x, y) = \sum_{i=1}^l \mu_i k_i(x, y) : \mu \in \mathbb{R}^l, K_{ij} = k(x_i, x_j), K \succeq 0\},
\]

where the \( x_i \) 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 kernel in \( K \) are not necessarily positive kernels.

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

\[
\min_{\mu \in \mathbb{R}^l, G \in \mathbb{R}^{m \times m}, \mathbf{e} \in \mathbb{R}^m, \mathbf{y} \in \mathbb{R}^m, \mathbf{b} \in \mathbb{R}^m, t} \quad t, \\
\text{subject to: } \left( \begin{array}{c}
Ge + \mathbf{v} - \mathbf{\delta} + \gamma \mathbf{y} \\
(e + \mathbf{v} - \mathbf{\delta} + \gamma \mathbf{y})^T & t - \frac{1}{m} \mathbf{\delta}^T \mathbf{e}
\end{array} \right) \succeq 0 \\
\mathbf{v} \succeq 0, \quad \mathbf{\delta} \succeq 0, \quad G_{ij} = k(x_i, x_j) y_i y_j
\]

\[
k(x, y) = \sum_{i=1}^l \mu_i k_i(x, y)
\]

Note that here the original constraint \( K \succeq 0 \) in \( K \) 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
scaling 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,

\[
K := \{k : k(x, y) = \sum_{i=1}^l \mu_i k_i(x, y), \mu_i \geq 0\}.
\]

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

\[
\min_{\mu \geq 0} \max_{\alpha \in \mathbb{R}^m} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^l \mu_k \alpha_i \alpha_j y_i y_j k_k(x_i, x_j)
\]

s.t. \( \sum_{i=1}^m \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C \quad \forall \quad i = 1, ..., m. \)

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 Rakotomamonjy et al. (2008). These methods first fix \( \mu_i \) and optimize over \( \alpha \), then fix \( \alpha \) and optimize over \( \mu \), 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 Rakotomamonjy et al. (2008) 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 tessellated kernel functions. The following result is based on a parametrization of positive integral operators initially proposed in Recht (2006).

**Theorem 2** Let \( N \) be any bounded measurable function \( N : X \times X \rightarrow \mathbb{R}^q \) on compact \( X \) and \( X \) and \( P \in \mathbb{R}^{q \times q} \) be a positive matrix \( P \succeq 0 \). Then

\[
k(x, y) = \int_X N(z, x)^T P N(z, y)dz
\]

6
is a positive kernel function.

**Proof** Since $N$ is bounded and measurable, $k(x, y)$ is bounded and measurable. Since $P \succeq 0$, there exists $P^\perp$ such that $P = (P^\perp)^TP^\perp$. Now define

$$g(z) = \int_X P_{\perp}^T N(z, x) f(x) dx.$$ 

Then

$$\int_X \int_X f(x) k(x, y) f(y) dxdy = \int_X \int_X \int_X f(x) N(z, x)^T P N(z, y) f(y) dxdydz$$

$$= \int_X \left( \int_X P_{\perp}^T N(z, x) f(x) dx \right)^T \int_X N(z, y) P_{\perp}^T f(y) dxdydz$$

$$= \int_X g(z)^T g(z) dz \geq 0.$$

**Polynomial Kernels** Let $X = \mathbb{R}^n$ and $\mathcal{X} = \mathbb{R}^p$ and define $Z_d : \mathbb{R}^n \rightarrow \mathbb{R}^q$ 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 $P \succeq 0$ such that

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

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 tessellated kernels, 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 broader class of integral operators include, e.g. the Volterra operators. To parameterize such as class of kernels, we first replace $x > y$ with the constraints $x - y \in S_1 \subset \mathbb{R}^n$ and $x - y \in S_2 \subset \mathbb{R}^n$ where the $S_1$ is the positive orthant and $S_2$ is the negative orthant. We now define the following indicator function

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

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

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

(7)
Using Eqn. (6), the associated kernel function is,

\[ k(x, y) = \int_X N(z, x)^T P N(z, y) dz. \]

**A Partition of the Tessellated Kernel** In this part, we partition the domain \( X \) into \( 2^n \) orthants and by expanding the integral and show that a tessellated kernel is piecewise polynomial, using polynomial \( k_\beta \) indexed to each domain \( X_\beta \).

**Lemma 3** Suppose that for \( a < b \in \mathbb{R}^n \), \( X = \mathcal{X} = [a, b] \), \( N \) is as defined in Eqn. (7)

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

and \( k \) is as defined in Eqn. (6). Then 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

\[ X_\beta := \left\{ (x, y) \in \mathbb{R}^n \times \mathbb{R}^n : \begin{array}{l} x_j \geq y_j \text{ for all } j: \beta_j = 0 \\ y_i \geq x_i \text{ for all } i: \beta_i = 1 \end{array} \right\}, \]

we have that

\[ k(x, y) = \begin{cases} k_\beta(x, y) & \text{if } (x, y) \in X_\beta. \end{cases} \quad (8) \]

where the \( k_\beta \) are polynomials defined as

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

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

\[ k(x, y) = \int_X N(z, x)^T P N(z, y) dz \]

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

where

\[ \mathcal{X}_{ij} := \{ z \in \mathbb{R}^{3n} : I_{S_i}(z, x) I_{S_j}(z, y) = 1 \}. \]
Figure 1: Optimal classifier, \( f(z) \) for labelling a 1 dimensional dataset using a degree one tessellated kernel (solid lines), and a positive combination of Gaussian kernels (dotted lines) with three different penalty weights \( C \). Note that as \( C \) increases so to does the maximum slope of \( f(z) \) for the tessellated kernel and the maximum value of \( f(z) \) for the Gaussian kernel.

From the definition of \( X_{ij} \) we have that,

\[
X_{11} = \{ z \in Z : z_i \geq p^*_i(x,y), \ i = 1, \cdots, n \} \\
X_{12} = \{ z \in Z : z_i \geq x_i, \ i = 1, \cdots, n \} \cap X_{11} \\
X_{21} = \{ z \in Z : z_i \geq y_i, \ i = 1, \cdots, n \} \cap X_{11} \\
X_{22} = Z / (X_{11} \cup X_{12} \cup X_{21}).
\]

where \( p^*_i(x,y) = \max \{ x_i, y_i \} \). By the definitions of \( X_{11}, X_{12}, X_{21}, \) and \( X_{22} \) we have that,

\[
k(x,y) = \int_{P^*(x,y)}^{b} Z_d(z,x)^T (P_{11} - P_{12} - P_{21} + P_{22}) Z_d(z,y)dz + \int_{x}^{b} Z_d(z,x)^T (P_{12} - P_{22}) Z_d(z,y)dz \\
+ \int_{y}^{b} Z_d(z,x)^T (P_{21} - P_{22}) Z_d(z,y)dz + \int_{a}^{b} Z_d(z,x)^T P_{22} Z_d(z,y)dz. \tag{9}
\]

Note that the number of domains \( X_{\beta} \) 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

\[ f(z) = \sum_{i=1}^{m} \alpha_i y_i k(x_i, z) + b \]

\[ = \begin{cases} \sum_{i=1}^{m} \alpha_i y_i k(x_i, z) & \text{if } (x_i, z) \in X_\beta. \\ f_{i,\beta}(z) & \text{if } z \in X_{i,\beta}, \end{cases} \]

where,

\[ f_{i,\beta}(z) = \sum_{i=1}^{m} \alpha_i y_i k(x_i, z) \]

\[ X_{i,\beta} := \left\{ z : \begin{array}{l} (x_i)_j \geq z_j \text{ for all } j; \beta_j = 0, \\ z_k \geq (x_i)_k \text{ for all } k; \beta_i = k \end{array} \right\} \]

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 the number of training data points \( m \). For instance with \( n = 2 \) there are 100 sub-domains for just 9 data points. This growth is what makes tessellated kernels universal - as will be seen in Section IV.

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

To determine what the basis of these polynomial functions are, in Figure 2 we plot the value of the kernel function in one dimension with training datum \( x_i = 5 \), for a selection of different positive matrices where \( P_{1,2} = P_{2,1} = P_{2,2} = 0 \) and \( P_{1,1} \) is equal to either matrix \( A_1, A_2, A_3 \) or \( A_4 \),

\[
\begin{align*}
A_1 &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, & A_2 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, & A_3 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, & A_4 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{align*}
\]

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 the discontinuity that occurs at the training datum for the kernel function. In addition, Figure 2 illustrates that the class of tessellated kernel functions vary significantly based on the selected positive matrix, thus causing significant variations in the learned support vector machine.
A New Class of Universal Kernel Functions

Figure 2: Normalized kernel function $k_i(x, y)$ where $y = 5$, defined on the domain $[0,10]$ where the $P_{1,1}$ used to parameterize $k_i(x, y)$ is the matrix $A_i$ from (10) and $P_{1,2} = P_{2,1} = P_{2,2} = 0$.

### 4. Properties of the tessellated class of kernel functions

In this section we will prove that the tessellated kernel functions are both continuous and universal, even in the simplest case of degree $d = 0$. Let us begin by recalling that for any $P \succeq 0$ 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 tessellated kernel, 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 Schölkopf et al. (2001) this implies that the classifiers consists of functions of the form

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

The following theorem establishes that such functions are necessarily continuous.

**Theorem 4** Suppose that for $a < b \in \mathbb{R}^n$, $X = \mathcal{X} = [a, b]$, $P \succeq 0$, $N$ is as defined in Eqn. (7) 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

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

$$k(x, y) = \begin{cases} k_\beta(x, y) & \text{if } (x, y) \in X_\beta. \end{cases}$$

where the $k_\beta$ are polynomials defined as

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

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 $Z_d(x, z) = x^{\alpha_i} z^{\gamma_i}$ where $\alpha_i, \gamma_i \in N^n$ for $i = 1, \cdots, q$. Then

$$\prod_{i; \beta_i = 0_{z_i = x_i}}^{b_i} \int_{j; \beta_j = 1_{z_j = y_j}}^{b_j} Z_d(z, x) T Q_1 Z_d(z, y) dz = \sum_{k, l} (Q_1)_{k, l} \prod_{i; \beta_i = 0_{z_i = x_i}}^{b_i} \int_{j; \beta_j = 1_{z_j = y_j}}^{b_j} x^{\alpha_k} y^{\gamma_k} z^{\gamma_i} dz$$

$$= \sum_{k, l} (Q_1)_{k, l} x^{\alpha_k} y^{\alpha_l} \prod_{i; \beta_i = 0_{z_i = x_i}}^{b_i} \int_{j; \beta_j = 1_{z_j = y_j}}^{b_j} z^{\gamma_k + \gamma_l} dz.$$

Expanding the integrals in (12), each has the form

$$\prod_{i; \beta_i = 0_{z_i = x_i}}^{b_i} \int_{j; \beta_j = 1_{z_j = y_j}}^{b_j} z^{\alpha} dz = \prod_{i; \beta_i = 0_{z_i = x_i}}^{b_i} \int_{j; \beta_j = 1_{z_j = y_j}}^{b_j} z^{\alpha_i} dz_i \prod_{j; \beta_j = 1_{z_j = y_j}}^{b_j} z^{\alpha_j} dz_j$$

$$= \prod_{i; \beta_i = 0_{z_i = x_i}}^{b_i} \int_{j; \beta_j = 1_{z_j = y_j}}^{b_j} (b_i - x_i^{\alpha_i + 1}) \prod_{j; \beta_j = 1_{z_j = y_j}}^{b_j} \frac{1}{\alpha_j + 1} (b_j - y_j^{\alpha_j + 1})$$

$$= \prod_{k=1}^{n} \frac{1}{\alpha_k + 1} \prod_{i; \beta_i = 0_{z_i = x_i}}^{b_i} (b_i - x_i^{\alpha_i + 1}) \prod_{j; \beta_j = 1_{z_j = y_j}}^{b_j} (b_j - y_j^{\alpha_j + 1})$$

$$= \prod_{j=1}^{n} b_j - \frac{1}{2} (x_j + y_j + |x_j - y_j|) \alpha_j + 1.$$

where we have used the fact that

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

Therefore $k(x, y)$ is the product and summation of continuous functions and thus $k(x, y)$ can be defined by a single continuous function over every domain. We conclude that $k$ and therefore 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. We use the following definition of universal kernel as can be found in, e.g. Micchelli et al. (2006).

**Definition 5** A kernel $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ 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, $\Phi : \mathcal{X} \rightarrow \mathcal{W}$ 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 $C(\mathcal{X}) := \{ f : \mathcal{X} \rightarrow \mathbb{R} : \text{f is continuous} \}$ where $\|f\|_C := \sup_{x \in \mathcal{X}} |f(x)|$.

Recall that $\mathcal{H}$ can be characterized as the closure of span\{ $k(y, \cdot) : y \in \mathcal{X}$ \}

The following theorem shows that any tessellated kernel 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. (7). Then $k$ is universal for $X = \mathcal{X} = [a, b] \subseteq \mathbb{R}^n$.

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

$$P_i = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \otimes e_i$$

where $\{e_i\}$ is the canonical basis for $\mathbb{R}^n$. In this case

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

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

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

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

13
For \( n = 1 \), we may construct a triangle function centered at \( y_2 \) as

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

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 Shekhtman (1982). We conclude that \( k_1 \) is a universal kernel and hence \( k \) is universal.

**Notation** For convenience, we denote the positive Tessellated Kernels by saying \( k \in \mathcal{K}_d[T] \) if there exists some \( P \succeq 0 \) such that \( k \) is as defined in Equation (6) where \( N \) is as defines in Eqn (7) 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 \( \mathcal{K}_T \) contains only three parameters (elements of \( P \)).

In addition, unlike the Gaussian kernel, entries of the kernel matrix may be negative when using a tessellated kernel function. To demonstrate negative entries are necessary for finding the optimal kernel matrix, we analytically solve the following SDP to determine what the optimal kernel matrix, \( K^* \), would be if the kernel matrix itself is the decision variable of our optimization problem. We find that the solution of the following SDP,

\[
\begin{aligned}
\min_{t \in \mathbb{R}, K \in \mathbb{R}^{m \times m}, \gamma \in \mathbb{R}^m, \nu \in \mathbb{R}^m, \delta \in \mathbb{R}^m} & t, \\
\text{subject to:} & G (e + \nu - \delta + \gamma y)_T e + \nu - \delta + \gamma y < C e, \\
& \nu \geq 0, \quad \delta \geq 0, \quad K \succeq 0, \quad \text{trace}(K) = m, \quad G_{i,j} = y_i K_{i,j} y_j
\end{aligned}
\]

is \( K^*_{i,j} = y_i y_j \).

**Theorem 7** Let \( y_i \in \{1, -1\} \) for \( i = 1, \cdots, m \) and \( Y = \text{diag}(y) \). If \( C \geq \frac{2}{m} \), then the solution to Optimization Problem 13 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 will first perform a change of variables to represent the optimal value of $K$, as $K^*(\nu, \delta, \gamma)$. We will then prove that $\gamma^*, \nu^*$, and $\delta^*$ is the globally optimal point of Optimization Problem 13. First note that the LMI in Optimization Problem 13, 

$$
\begin{bmatrix}
KY

(e + \nu - \delta + \gamma y)^T

\nu

t - C\delta^T e
\end{bmatrix} \succeq 0
$$

implies that

$$
t \geq (e + \nu - \delta + \gamma y)^T (KY)^{-1} (e + \nu - \delta + \gamma y) + 2C\delta^T e
$$

Since $Y$ is unitary $KY \succeq 0$ is equivalent to $K \succeq 0$ and Optimization Problem 13 is equivalent to,

$$
\min_{K \in \mathbb{R}^{m \times m}, \gamma \in \mathbb{R}, \mu \in \mathbb{R}^m, \delta \in \mathbb{R}^m} (e + \nu - \delta + \gamma y)^T (KY)^{-1} (e + \nu - \delta + \gamma y) + 2C\delta^T e
$$

subject to: $\nu \geq 0, \quad \delta \geq 0, \quad K \succeq 0, \quad \text{trace}(K) \leq m.$

First note that since $Y$ is unitary, $K \succeq 0$ and $\text{trace}(K) \leq m$ we have that,

$$
(e + \nu - \delta + \gamma y)^T (KY)^{-1} (e + \nu - \delta + \gamma y) \geq \frac{1}{\sigma(K)} \|e + \nu - \delta + \gamma y\|_2^2 \\
\geq \frac{1}{m} \|e + \nu - \delta + \gamma y\|_2^2.
$$

Therefore if we find a $K^*(\nu, \delta, \gamma)$ such that $(e + \nu - \delta + \gamma y)^T (KY)^{-1} (e + \nu - \delta + \gamma y) = \frac{1}{m} \|e + \nu - \delta + \gamma y\|_2^2$, then such a $K^*$ is optimal for any values of $\nu, \delta$, and $\gamma$ and we would therefore have a change of variables for Optimization Problem 13. Let, 

$$K^*(\nu, \delta, \gamma) = U(\nu, \delta, \gamma) \Sigma U(\nu, \delta, \gamma)^T,
$$

where $U$ is unitary and,

$$U(\nu, \delta, \gamma) = Y \left[ \frac{(e+\nu-\delta+\gamma y)}{\|e+\nu-\delta+\gamma y\|_2} \cdots \right], \Sigma = \left[ \begin{array}{cc} m & 0 \\ 0 & 0 \end{array} \right].$$

Then, 

$$(e + \nu - \delta + \gamma y)^T (KY)^{-1} (e + \nu - \delta + \gamma y)$$

is equivalent to

$$(e + \nu - \delta + \gamma y) Y \left[ \frac{(e+\nu-\delta+\gamma y)}{\|e+\nu-\delta+\gamma y\|_2} \cdots \right] \left[ \begin{array}{cc} m & 0 \\ 0 & 0 \end{array} \right] \left[ \frac{(e+\nu-\delta+\gamma y)}{\|e+\nu-\delta+\gamma y\|_2} \cdots \right]^T Y^{-1} (e + \nu - \delta + \gamma y).
$$

But since $U$ is unitary and $Y$ is symmetric and unitary, we have that Eqn. (15) is equivalent to

$$
\left[ \frac{\|e+\nu-\delta+\gamma y\|^2}{\|e+\nu-\delta+\gamma y\|_2^2} \cdots \right] \left[ \begin{array}{cc} m & 0 \\ 0 & 0 \end{array} \right]^{-1} \left[ \frac{\|e+\nu-\delta+\gamma y\|^2}{\|e+\nu-\delta+\gamma y\|_2^2} \cdots \right] = \left[ \frac{\|e+\nu-\delta+\gamma y\|^2}{m} \right].
$$

Therefore $K^*(\nu, \delta, \gamma)$ must be the optimal matrix that minimizes the objective function for any values of $\nu, \gamma$ and $\delta$.
Now perform the variable substitution $K = K^*(\nu, \delta, \gamma)$ to Optimization Problem (13) and we have,

$$\min_{\gamma \in \mathbb{R}, \nu \in \mathbb{R}^m, \delta \in \mathbb{R}^m} \frac{\|e + \nu - \delta + \gamma y\|_2^2}{m} + 2C\delta^t e,$$

subject to: $\nu \geq 0$, $\delta \geq 0$.

The objective function of Optimization Problem 16 is equivalent to the quadratic,

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

$$= \frac{1}{m} \sum_{i=1}^{m} [(1 + \nu_i - \delta_i + \gamma y_i)^2 + 2mC\delta_i]$$

$$= \frac{1}{m} \sum_{i=1}^{m} 1 + 2\nu_i + (2mC - 2)\delta_i + 2\gamma y_i + \nu_i^2 - 2\delta_i\nu_i + 2\nu_i\gamma y_i + \delta_i^2 - 2\delta_i\gamma y_i + \gamma^2.$$

Thus $f(\nu, \delta, \gamma)$ is convex, and has partial derivatives

$$\frac{\partial f}{\partial \nu_i} = \frac{2 + 2\bar{y}y_i}{m}, \quad \frac{\partial f}{\partial \delta_i} = \frac{2mC - 2 - 2\bar{y}y_i}{m}, \quad \frac{\partial f}{\partial \gamma} = \frac{1}{m} \sum_{i=1}^{m} 2y_i + 2\gamma$$

At the point $\delta^* = 0, \nu^* = 0, \gamma^* = -\bar{y}$ the partial derivatives with respect to the decision variables are,

$$\frac{\partial f}{\partial \nu_i} = \frac{2 + 2\bar{y}y_i}{m}, \quad \frac{\partial f}{\partial \delta_i} = \frac{2mC - 2 - 2\bar{y}y_i}{m}, \quad \frac{\partial f}{\partial \gamma} = \frac{1}{m} \sum_{i=1}^{m} 2y_i + 2\gamma$$

First note that $\bar{y} \in (-1, 1)$, then we have that,

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

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

$$\frac{\partial f}{\partial \delta_i} = \frac{2mC - 2 - 2\bar{y}y_i}{m} \geq \frac{4 - 2 - 2(1)(1)}{m} \geq 0.$$

Finally we have that

$$\frac{\partial f}{\partial \gamma} = \frac{1}{m} \sum_{i=1}^{m} 2y_i - 2\bar{y}$$

$$= -2\bar{y} + 2\frac{1}{m} \sum_{i=1}^{m} y_i$$

$$= 0.$$
there exists a $P_{K}$ and we will consider the following two optimization problems, where $k$ is a tessellated kernel function in $\epsilon$ bounds (see Fig. 3(a)). However, for the tessellated kernel, given an $\epsilon$ respectively. In practice we find that the objective value can be larger than these lower bounds. The absolute minimum objective value of Optimization Problem’s 17 and 18 is 0 and thus $K^{*}_{\epsilon}$ functions and tessellated kernel functions can approach $K^{*}$ matrices generated by positive combinations of Gaussian kernel functions, polynomial kernel functions and tessellated kernel functions.

Therefore, we have that,

$$\nu^{*} = 0, \quad \gamma^{*} = \sum_{i=1}^{m} -\frac{y_{i}}{m}, \quad \delta^{*} = 0, \quad t^{*} = \frac{\|e + \gamma^{*} y\|^{2}}{m}$$

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

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 experimentally that the same kernel matrix is selected, and that only the values of $\delta^{*}$ and $\gamma^{*}$ are different.

Without loss of generality we will consider the case where $\sum_{i=1}^{m} y_{i} = 0$. In this case we have that $\lambda^{*} = 0$ and thus $K^{*}_{ij} = y_{i} y_{j}$. Using $K^{*}_{ij} = y_{i} y_{j}$, we will analyze whether kernel matrices generated by positive combinations of Gaussian kernel functions, polynomial kernel functions and tessellated kernel functions can approach $K^{*}$. To analyze these kernel matrices we will consider the following two optimization problems,

$$\min_{k} \frac{\|K - K^{*}\|}{n^{2}} \quad s.t. \quad K_{ij} = k(x_{i}, x_{j}) \quad (17)$$

$$\min_{k} \|K - K^{*}\|_{\infty} \quad s.t. \quad K_{ij} = k(x_{i}, x_{j}) \quad (18)$$

where $k \in \mathcal{K}^{d}$, $\mathcal{K}_{P}^{d}$, $\mathcal{K}_{G}^{m}$ with

$$\mathcal{K}^{d} := \{k(x_{i}, x_{j}) = Z_{d}(x_{i})^{T} P Z_{d}(x_{j}) : P \succeq 0\} \quad \mathcal{K}_{P}^{d} := \{k(x_{i}, x_{j}) = \sum_{\gamma_{i} \in \Gamma_{P}} \mu_{i} e^{\frac{||x_{i} - x_{j}||^{2}}{\gamma_{i}}}, \quad \mu_{i} > 0\}$$

Since half of the entries in $K^{*}$ are $-1$, and since the Gaussian kernel is globally positive, the absolute minimum objective value of Optimization Problem’s 17 and 18 is 0.5 and 1 respectively. In practice we find that the objective value can be larger than these lower bounds (see Fig. 3(a)). However, for the tessellated kernel, given an $\epsilon$, there exists a $d$, and a tessellated kernel function in $K_{\epsilon}^{d}$ such that $\|K - K^{*}\|_{\infty} \leq \epsilon$.

**Theorem 8** Suppose $k$ is as defined in Eqn. (9) and we are given a set of points $\{x_{i}\}_{i=1}^{m} \subset \mathbb{R}^{n}$, with associated labels, $y_{i} \in \{-1, 1\}$ for $i = 1, \ldots, m$. For an $N$ as defined in Eqn. (7), there exists a $P \succeq 0$, and a $d \in \mathbb{N}$, such that $\|K - K^{*}\|_{\infty} \leq \epsilon$ for any $\epsilon > 0$ where $K^{*}_{ij} = y_{i} y_{j}$ and $K_{ij} = k(x_{i}, x_{j})$.

**Proof** First select $P_{22} \succeq 0$, and define $P_{11} = P_{12} = P_{21} = P_{22}$. By the Schur complement since $P_{22} \succeq 0$, then

$$P^{-1} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}^{-1} = P_{22} - P_{12}^{T} P_{11}^{-1} P_{21} = P_{22} - P_{22} = 0,$$
(a) $\| K - K^* \|_1$ for the tessellated kernel of degree $d$ and for a positive combination of $m$ Gaussian kernels.

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

Figure 3: The objective of Optimization Problem 17 and 18 for the tessellated and polynomial kernel of degree $d$ and for a positive combination of $m$ Gaussian kernels with bandwidths ranging from 0.01 to 10. The number of bandwidths is selected so that the number of decision variables match in the Gaussian and in the tessellated kernel function case.

so $P \succeq 0$ and we have that the kernel function $k$ is,

$$k(x, y) = \int Z_d(z, x)^T (P_{11} - P_{12} - P_{21} + P_{22}) Z_d(z, y) dz + \int Z_d(z, x)^T (P_{12} - P_{22}) Z_d(z, y) dz + \int Z_d(z, x)^T P_{22} Z_d(z, y) dz$$

$$= \int Z_d(z, x)^T P_{22} Z_d(z, y) dz$$

$$= C p(x)^T p(y)$$

(19)

where $p(\cdot) : \mathbb{R}^q \rightarrow \mathbb{R}^q, P_{22} \in \mathbb{R}^{q \times q}, p(\cdot) = Z_d(\cdot) M^T$ for $P_{22} = M^T M$ and $C = \int_a^b Z_d(z)^T Z_d(z) dz$.

Since polynomials are dense in $L_\infty$, and since any polynomial can be parameterized by an $M$ and $d$, select an $M$ and $d$ such that,

$$\| p(x_i) - y_i \left[ \frac{1}{\sqrt{C}} \ 0 \ \ldots \ 0 \right]^T \|_\infty \leq \sqrt{\frac{\epsilon}{Cq}} \text{ for all } i = 1, \ldots, m.$$

Then we have that,

$$\| K^* - K \|_\infty = \max_{i, j \in [0, m]} \| y_i y_j - C p(x_i)^T p(x_j) \|_\infty$$

$$\leq \max_{i, j \in [0, m]} \| y_i y_j - C \left( y_i \left[ \frac{1}{\sqrt{C}} \ 0 \ \ldots \ 0 \right]^T \left[ \frac{1}{\sqrt{C}} \ 0 \ \ldots \ 0 \right]^T y_j + q \sqrt{\frac{\epsilon}{Cq}} \sqrt{\frac{\epsilon}{Cq}} \right) \|_\infty$$

$$\leq \max_{i, j \in [0, m]} \| y_i y_j - C \left( y_i y_j \frac{\epsilon}{C} + \frac{\epsilon}{C} \right) \|_\infty$$

$$\leq \epsilon$$
To prove Theorem 8 we selected a simplified tessellated kernel function where we recover a polynomial kernel function. Since polynomial kernels are included in the class of tessellated kernel functions, the kernel matrix generated by the optimal tessellated kernel function is guaranteed to approach $K^*$ at least as quickly as an optimal polynomial kernel, though numerical experiments show it can approach $K^*$ more quickly than a polynomial kernel.

In Fig. 3(a) and 3(b) we numerically show the change in the objective value of Optimization Problems 17 and 18 for the optimal Gaussian, Polynomial, and Tessellated kernel function for the spiral data set with 20 points as we increase the number complexity of the kernel function. For the tessellated and polynomial kernel functions we increase the complexity of the kernel function by increasing the degree of the monomial basis. For the Gaussian kernel function we increase the number of bandwidths, $\gamma$, selected. We show numerically that, in this case, the Gaussian Kernel saturates with an error value significantly larger than the lower bound of 0.5 for the 1-norm and exactly at 1 for the $\infty$-norm (the projected lower bound). In addition the tessellated kernel initially approaches an error of 0 at a faster rate than the polynomial kernel.

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 tessellated kernel function using either the SDP method of Optimization Problem (5), or using a two-step method like SimpleMKL. Using the representation of Tessellated Kernels ($\mathcal{K} = \mathcal{K}_T$) in Theorem 4 in Section 4, Optimization Problem (4) may be expressed as

$$\min_{t \in \mathbb{R}, \gamma \in \mathbb{R}, \nu, \delta \in \mathbb{R}^m} t, \quad (20)$$

subject to:

$$G(P) \begin{pmatrix} (e + \nu - \delta + \gamma y)^T \nu - \delta + \gamma y \\ t - \frac{2}{m\lambda} \delta^T e \end{pmatrix} \succeq 0$$

$$\nu \geq 0, \quad \delta \geq 0, \quad P \succeq 0, \quad \text{trace}(P) \leq 1$$

$$G_{ij}(P) = \sum_{i,j=1,2} \sum_{k,l} (P_{i,j})_{k,l} x_i^{\alpha_k} x_j^{\alpha_l} \int_{X_{ij}} z^{\gamma_k + \gamma_l} dz x_i y_j.$$

Optimization Problem (20) is an SDP and can therefore be solved efficiently using standard SDP solver such as ApS (2015). Note that we use the trace constraint to ensure the kernel function is bounded.

Typically SDP problems require roughly $p^2 n^2$ 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 (20) 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 (20). To improve the scalability 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, that is it returns a vector of weights $\mu$, on the a-priori selected kernel functions. Here we discuss how SimpleMKL can be used to find optimal combinations of tessellated kernel functions that perform well in practice.

Since tessellated kernel functions have a linear parameterization, the positive sum of multiple tessellated kernel functions, parameterized by the positive semi-definite matrices $P_i$, is equivalent to a single tessellated kernel function, represented by the matrix $P = \sum_{i=1}^{k} P_i$. Therefore, by randomly generating a set of $l$ positive semi-definite matrices, $P_i$ for $i = 1, \ldots, l$, we may use SimpleMKL to find the optimal linear combination of tessellated kernels defined by each matrix $P_i$. The optimal tessellated kernel function may then be approximated as the tessellated kernel function parameterized by the matrix,

$$ P = \sum_{i=1}^{l} \mu_i P_i. $$

Where $\mu$ is the vector of weights returned by SimpleMKL. In practice, we find that this randomized approach performs well in terms of accuracy on test data sets. Note that the complexity of SimpleMKL approximately increases linearly with the number of kernel functions, and superlinearly with respect to $m$, the number of data points Rakotomamonjy et al. (2008).

Finally, we mention that we may avoid the heuristic use of randomized matrices by noting that SimpleMKL is a two-step method - where the second step fixes $\alpha$ and searched over $\mu_i$. Since our parameterization of tessellated kernels is linear, this second step may be used to search over the entire space of tessellated kernels. However, implementation of this approach is left for future work.

We will next consider an experimental complexity analysis of the SDP method before comparing the accuracy of the two proposed methods.

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 (20) with
A New Class of Universal Kernel Functions

Number of Training Inputs

Seconds to Optimize

Figure 5: Log-Log Plot of Computation Time vs number of training data for 2-feature kernel learning.

(a) Complexity Scaling for Identification of Circle

(b) Complexity Scaling for Identification of Spiral

respect to the number of training points as well as the selected degree of the tessellated kernel function.

The constraint that the kernel be a positive tessellated kernel can be expressed as an LMI constraint with variables $P_{ij}$. Using Optimization Problem (20), 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. 5 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. 4) 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 tessellated kernel 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. 6. 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. 4). 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 tessellated kernels are universal, therefore, so any of the positive semi-definite matrices parameterize a universal kernel.
Figure 6: Log-Log Plot of Computation Time vs number of training data for 2-feature kernel learning using SimpleMKL and tessellated kernels.

8. Accuracy and comparison with existing methods

In this section, we evaluate the relative accuracy of 1) Optimization Problem (20); 2) SimpleMKL as defined in Rakotomamonjy et al. (2008) using polynomial and Gaussian kernels three different sets of kernel basis $\mathcal{K}$ - the first contains polynomial and Gaussian kernels, the second is the parameterization of tessellated kernels from Section 6, and the last contains both tessellated, polynomial and Gaussian kernels. To determine the set $\mathcal{X}$ for the integral of the kernel function, we first scale the data so that $x_i \in [0, 1]^n$, and then set $\mathcal{X} := [0 - \epsilon, 1 + \epsilon]^n$, where $\epsilon$ is chosen by 5-fold cross-validation.

For the numerical tests we use the soft-margin problem with regularization parameter $C$ also determined by 5-fold cross-validation and compare the following methods: a) For the tessellated kernel, in all cases we choose $d = 1$ (Except Ionosphere, which uses $d = 0$); b) For SimpleMKL, we use the standard kernel selection of combined Gaussian and polynomial kernels with bandwidths arbitrarily chosen between .5 and 10 and degrees of degree one through three - yielding approximately $13(n + 1)$ kernels; c) To illustrate the effect of combining the proposed kernel with SimpleMKL, we randomly generated a sequence of 300 positive semidefinite matrices and used these as the SimpleMKL library of kernels; Finally, in d) We combined the SimpleMKL library of kernels mentioned earlier with the 300 randomly generated tessellated library of kernels. 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. 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
Table 1: TSA comparison for algorithms a), b), c), and d). 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.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Method</th>
<th>Accuracy</th>
<th>Time</th>
<th>Data Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liver</td>
<td>Tessellated</td>
<td>72.32 ± 4.92</td>
<td>95.75 ± 2.68</td>
<td>m = 346</td>
</tr>
<tr>
<td></td>
<td>SimpleMKL</td>
<td>65.51 ± 5.10</td>
<td>2.61 ± 0.42</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SimpleMKL Tess.</td>
<td>70.58 ± 4.69</td>
<td>8.37 ± 0.30</td>
<td>n = 6</td>
</tr>
<tr>
<td></td>
<td>Combined</td>
<td>70.53 ± 4.79</td>
<td>14.70 ± 0.76</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Neural Net</td>
<td>66.32 ± 7.95</td>
<td>0.08 ± 0.02</td>
<td></td>
</tr>
<tr>
<td>Cancer</td>
<td>Tessellated</td>
<td>97.18 ± 1.48</td>
<td>636.17 ± 25.43</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SimpleMKL</td>
<td>96.55 ± 1.34</td>
<td>14.74 ± 1.33</td>
<td>m = 684</td>
</tr>
<tr>
<td></td>
<td>SimpleMKL Tess.</td>
<td>96.89 ± 1.43</td>
<td>45.84 ± 4.28</td>
<td>n = 9</td>
</tr>
<tr>
<td></td>
<td>Combined</td>
<td>96.89 ± 1.42</td>
<td>65.08 ± 10.52</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Neural Net</td>
<td>96.50 ± 1.51</td>
<td>0.08 ± 0.01</td>
<td></td>
</tr>
<tr>
<td>Heart</td>
<td>Tessellated</td>
<td>83.46 ± 4.56</td>
<td>221.67 ± 29.63</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SimpleMKL</td>
<td>83.70 ± 4.77</td>
<td>3.09 ± 0.19</td>
<td>m = 271</td>
</tr>
<tr>
<td></td>
<td>SimpleMKL Tess.</td>
<td>84.38 ± 4.34</td>
<td>55.48 ± 2.67</td>
<td>n = 13</td>
</tr>
<tr>
<td></td>
<td>Combined</td>
<td>83.64 ± 4.54</td>
<td>13.23 ± 2.70</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Neural Net</td>
<td>80.67 ± 5.74</td>
<td>0.07 ± 0.01</td>
<td></td>
</tr>
<tr>
<td>Pima</td>
<td>Tessellated</td>
<td>76.32 ± 3.10</td>
<td>1211.66 ± 27.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SimpleMKL</td>
<td>76.00 ± 3.33</td>
<td>19.04 ± 2.33</td>
<td>m = 769</td>
</tr>
<tr>
<td></td>
<td>SimpleMKL Tess.</td>
<td>76.75 ± 2.81</td>
<td>34.65 ± 23.28</td>
<td>n = 8</td>
</tr>
<tr>
<td></td>
<td>Combined</td>
<td>76.57 ± 2.72</td>
<td>96.20 ± 30.42</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Neural Net</td>
<td>75.97 ± 2.88</td>
<td>0.09 ± 0.02</td>
<td></td>
</tr>
<tr>
<td>Ionosphere</td>
<td>Tessellated</td>
<td>93.24 ± 3.04</td>
<td>6.69 ± 0.27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SimpleMKL</td>
<td>92.16 ± 2.78</td>
<td>26.24 ± 2.78</td>
<td>m = 352</td>
</tr>
<tr>
<td></td>
<td>SimpleMKL Tess.</td>
<td>87.65 ± 2.88</td>
<td>8.28 ± 1.16</td>
<td>n = 34</td>
</tr>
<tr>
<td></td>
<td>Combined</td>
<td>92.16 ± 2.78</td>
<td>50.77 ± 2.98</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Neural Net</td>
<td>89.15 ± 3.86</td>
<td>0.09 ± 0.01</td>
<td></td>
</tr>
</tbody>
</table>

Data Repository. In all cases, the tessellated kernel 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 tessellated kernel 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 \((x \text{ and } y)\) (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 well with the tessellated kernel showing significant improvement over SimpleMKL only for very large training data sets.
Figure 8: Plots demonstrating the change in accuracy of the tessellated kernel 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.

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 7. In this case we see that the tessellated kernels 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 Tessellated kernel and SimpleMKL using 150 training data points. These 2D surfaces are found in Figure 4.

9. Conclusion
In this paper, we have proposed a new class of universal kernel functions based on a generalization of the Sobolev kernel. 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 tessellated kernel does not require a set of bandwidths to be chosen a priori. Indeed, by increasing the degree of the monomial basis, it may be possible to show that the tesselated kernels can approximate any universal kernel 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 tessellated kernel, 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).
A New Class of Universal Kernel Functions

References


