Positive Definite Kernels in Machine Learning

Marco Cuturi*

May 6, 2010

Abstract

This survey is an introduction to positive definite kernels and the set of methods they have inspired in the machine learning literature, namely kernel methods. We first discuss some properties of positive definite kernels as well as reproducing kernel Hibert spaces, the natural extension of the set of functions $\{k(x,\cdot), x \in \mathcal{X}\}$ associated with a kernel k defined on a space $\mathcal{X}$. We discuss at length the construction of kernel functions that take advantage of well-known statistical models. We provide an overview of numerous data-analysis methods which take advantage of reproducing kernel Hilbert spaces and discuss the idea of combining several kernels to improve the performance on certain tasks. We also provide a short cook-book of different kernels which are particularly useful for certain data-types such as images, graphs or speech segments.

Remark: This report is a draft. I apologize in advance for the numerous mistakes, typos, and not always well written parts it contains. Comments and suggestions will be highly appreciated.

Summary

We provide in this survey a short introduction to positive definite kernels and the set of methods they have inspired in machine learning, also known as kernel methods. Kernel methods are a successful family of data-analysis tools that have become popular in the late 90's. Kernel methods are for lazy people: they provide simple and principled answers to perform a large variety of tasks on different datasets with minimal effort and little tuning. Still, the success of such methods relies on a key ingredient, the kernel. This ingredient will be the particular focus of this survey.

The main mathematical idea behind kernel methods is the following. Most data-inference tasks aim at defining an appropriate decision function f on a

*mcuturi@princeton.edu. Currently with ORFE- Princeton University. A large share of this work was carried out while the author was working at the Institute of Statistical Mathematics, Tokyo, Japan. In particular, this research was supported by the Function and Induction Research Project, Transdisciplinary Research Integration Center, Research Organization of Information and Systems.

set of objects of interest $\mathcal{X}$. When $\mathcal{X}$ is a vector space of dimension d, say $\mathbb{R}^d$, linear functions $f_a(x) = a^T x$ are one of the easiest and better understood choices, notably for regression, classification or dimensionality reduction. Given a positive definite kernel k on $\mathcal{X}$, that is a real-valued function on $\mathcal{X} \times \mathcal{X}$ which quantifies effectively how similar two points x and y are through the value $k(x, y)$, kernel methods are algorithms which estimate functions f of the form

$$f : x \in \mathcal{X} \to f(x) = \sum_{i \in I} \alpha_i k(x_i, x), \tag{1}$$

where $(x_i)_{i \in I}$ is a family of known points paired with $(\alpha_i)_{i \in I}$, a family of real coefficients. Kernel methods are often referred to as

- *data-driven* since the function f described in Equation (1) is an expansion of evaluations of the kernel k on points observed in the sample I, as opposed to a linear function $a^T x$ which only has d parameters;
- *non-parametric* since the vector of parameters (α_i) is indexed on a set I which is of variable size;
- *non-linear* since k can be a non-linear function such as the gaussian kernel $k(x, y) = \exp(-\|x - y\|^2/(2\sigma^2))$, and result in non-linear compounded functions f.
- *easily handled through convex programming* since many of the optimization problems formulated to propose suitable choices for the weights α involve quadratic constraints and objectives, which typically involve terms of the sort $\alpha^T K \alpha$ where K is a positive semi-definite matrix of kernel evaluations $[k(x_i, x_j)]$.

The problem of defining all of the elements introduced above, from the kernel k to the index set I and most importantly the weights α_i has spurred a large corpus of literature. We propose a survey of such techniques in this document. Our aim is to provide both theoretical and practical insights on positive definite kernels.

This survey is structured as follows:

- We start this survey by giving an overall introduction to kernel methods in Section 1 and highlight their specificity.
- We provide the reader with the theoretical foundations that underlie positive definite kernels in Section 2, introduce reproducing kernel Hilbert spaces theory and provide a discussion on the relationships between positive definite kernels and distances.
- Section 3 describes different families of kernels which have been covered in the literature of the last decade. We also describe a few popular techniques to encode prior knowledge on objects when defining kernels.

- We follow with the exposition in Section 4 of popular methods which, paired with the definition of a kernel, provide estimation algorithms to define the weights α_i of Equation (1).
- Selecting the right kernel for a given application is a practical hurdle when applying kernel methods in practice. We provide a few techniques to do so in Section 5, notably parameter tuning and the construction of linear mixtures of kernels, also known as multiple kernel learning.
- We close the survey by providing a brief cookbook of kernels in Section 6, that is a short description of kernels for complex objects such as strings, texts, graphs and images.

This survey is built on earlier references, notably (Schölkopf and Smola, 2002; Schölkopf et al., 2004; Shawe-Taylor and Cristianini, 2004). Whenever adequate we have tried to enrich this presentation with slightly more theoretical insights from (Berg et al., 1984; Berlinet and Thomas-Agnan, 2003), notably in Section 2. Topics covered in this survey overlap with some of the sections of (Muller et al., 2001) and more recently (Hofmann et al., 2008). The latter references cover in more detail kernel machines, such as the support vector machine for binary or multi-class classification. This presentation is comparatively tilted towards the study of positive definite kernels, notably in Sections 2 and 3

Contents

1 Introduction

The automation of data collection in most human activities, from industries, public institutions to academia, has generated tremendous amounts of observational data. In the same time, computational means have expanded in such a way that massive parallel clusters are now an affordable commodity for most laboratories and small companies. Unfortunately, recent years have seen an increasing gap of efficiency between our ability to produce and store these databases and our the analytical tools that are needed to infer knowledge from them. This long quest to understand and analyze such databases has spurred in the last decades fertile discoveries at the intersection of mathematics, statistics and computer science.

One of the most interesting changes brought forward by the abundance of data in recent years lies arguably in the increasing diversity of data structures practitioners are now faced with. Some complex data types that come from real-life applications do not translate well into simple vectors of features, which used to be a *de facto* requirement for statistical analysis up to four decades ago. When the task on such data types can be translated into elementary subtasks that involve for instance *regression, binary or multi-class classification, dimensionality reduction, canonical correlation analysis* or *clustering*, a novel class of algorithms popularized in the late nineties and known as kernel methods have proven to be effective, if not reach state-of-the art performance on many of these problems.

statistics, functional analysis and computer science: the mathematical machinery of kernel methods can be traced back to the seminal presentation of reproducing kernel Hilbert spaces by Aronszajn (1950) and its use in non-parametric statistics by Parzen (1962). However, their recent popularity in machine learning comes from recent innovations in both the *design of kernels* geared towards specific applications such as the one we cover in Section 6, paired with efficient *kernel machines* as introduced in Section 4. Examples of the latter include algorithms such as gaussian processes with sparse representations (Csató and Opper, 2002) or the popular support vector machine (Cortes and Vapnik, 1995). The theoretical justifications for such tools can be found in the statistical learning literature (Cucker and Smale, 2002; Vapnik, 1998) but also in subsequent convergence and consistency analysis carried out for specific techniques (Fukumizu et al., 2007; Vert and Vert, 2005; Bach, 2008b). Kernel design embodies the research trend pionneered in Jaakkola and Haussler (1999); Haussler (1999); Watkins (2000) of incorporating contextual knowledge on the objects of interest to define kernels.

Two features of kernel methods have been often quoted to explain the practical success of kernel methods. First, kernel methods can handle efficiently complex data types through the definition of appropriate kernels. Second, kernel methods can handle data which have multiple data representations, namely multimodal data. Let us review these claims before introducing the mathematical definition of kernels in the next section.

1.1 Versatile Framework for Structured Data

Structured objects such as (to cite a few) strings, 3D structures, trees and networks, time-series, histograms, images, and texts have become in an increasing number of applications the *de facto* inputs for data analysis algorithms. The originality of kernel methods is to address this diversity through a single approach.

from n points to $n \times n$ similarity matrices: using kernel methods on a dataset usually involves choosing first a family of similarity measures between pairs of objects. Irrespective of the initial complexity of the considered objects, dealing with a learning problem through kernels is equivalent to translating a set of n data points into a symmetric and positive definite $n \times n$ similarity matrix. This matrix will be the sole input used by the kernel algorithm, as schematically shown on Figure 1.1. This is very similar to the k-nearest neighbor (k-NN) framework (see [§13](Hastie et al., 2009) for a survey) where only distances between points matter to derive decision functions. On the contrary, parametric approaches used in statistics and neural networks impose a functional class beforehand (e.g. a family of statistical models or a neural architecture), which is either tailored to fit vectorial data – which in most cases requires a feature extraction procedure to avoid large or noisy vectorial representations – or tailored to fit a particular data type (hidden Markov models with strings, Markov random fields with images, parametric models for time series with given lags and seasonal corrections etc.). In this context, practitioners usually give kernel methods different credits, among them the fact that

- Defining kernel functions is in general easier that designing an accurate generative model and the estimation machinery that goes along with it, notably the optimization mechanisms and/or bayesian computational schemes that are necessary to make computations tractable.
- Efficient kernel machines, that is algorithm which use directly as an input Kernel matrices, such as the SVM or kernel-PCA, are numerous and the subject of separate research. Their wide availability under the form of software packages, makes them simple to use once a kernel has been defined.
- Kernel methods share initially the conceptual simplicity of k-nearest neighbors which make them popular when dealing with high-dimensional and challenging datasets for which little is known beforehand, such as the study of long sequences in bioinformatics Vert (2006). On the other hand, kernel algorithms offer a wider scope than the regression/classification applications of k-NN and also provide a motivated answer to control the bias/variance tradeoff of the decision function through penalized estimation, as explained in Section 4.3.

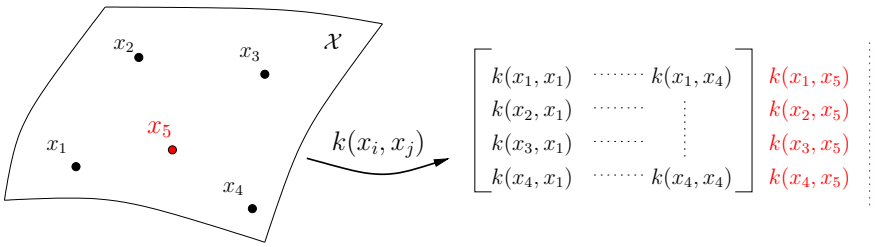

Figure 1: Given a dataset in a given space $\mathcal{X}$, represented as $\{x_1, x_2, x_3, x_4\}$ in the figure above, the kernel approach to data analysis involves representing these points through a positive-definite symmetric matrix of inter-similarities between points, as in the matrix $K_{4\times4}$ in the figure on the right. Given a new point x_5, any prediction with respect to x_5 (as in regression or classification for instance) will be a direct function of the similarity of x_5 to the learning set $\{x_1, x_2, x_3, x_4\}$. Thus, and in practice, kernel methods rely exclusively, both in the training phase and the actual use of the decision function, on similarity matrices.

1.2 Multimodality and Mixtures of Kernels

In most applications currently studied by practitioners, datasets are increasingly *multimodal*. Namely, described objects of interest through the lens of different representations.

For instance, a protein can be seen as an amino-acid sequence, a macromolecule with a 3D-structure, an expression level in a DNA-chip, a node in a biological pathway or in a phylogenetic tree. A video segment might be characterized by its images, its soundtrack, or additional information such as when it was broadcasted and on which channel. The interrelations between these modalities and the capacity to integrate them is likely to prove helpful for most learning tasks. Kernel methods provide an elegant way of integrating multimodalities through convex kernel combinations. This combination takes usually place before using a kernel machine as illustrated in Figure 1.2. This stands in stark contrast to other standard techniques which usually aggregate decision functions trained on the separated modalities. A wide range of techniques have been designed to do so through convex optimization and the use of unlabelled data (Lanckriet et al., 2004; Sindhwani et al., 2005). Kernels can thus be seen as atomic elements that focus on certain types of similarities for the objects, which can be combined through so-called multiple kernel learning methods as will be exposed more specifically in Section 5.2.

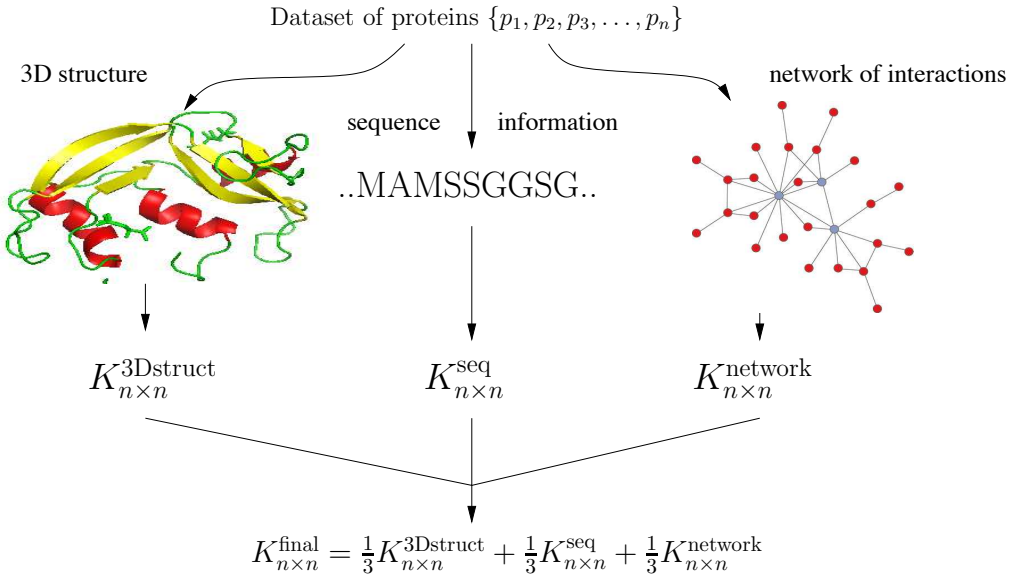

Figure 2: A dataset of proteins can be regarded in (at least) three different ways: as a dataset of 3D structures, a dataset of sequences and a set of nodes in a network which interact with each other. A different kernel matrix can be extracted from each datatype, using known kernels on 3D shapes, strings and graphs. The resulting kernels can then be combined together with arbitrary weights, as is the case above where a simple average is considered, or estimated weights, which is the subject of Section 5.2

2 Kernels: a Mathematical Definition

2.1 Positive Definiteness

Let us start this section by providing the reader with a definition for kernels, since the term "kernel" itself is used in different branches of mathematics, from linear algebra where it is used as a synonym for the nullspace of a linear operator to the theory of integral operators and density estimation. Some common kernels used in non-parametric statistics, such as the Epanechnikov kernel[1], are not, for instance, kernels in the sense of the terminology adopted in this report. We develop in this section elementary insights on kernels, combining different presentations given in (Berlinet and Thomas-Agnan, 2003; Berg et al., 1984; Schölkopf and Smola, 2002) to which the reader may refer for a more complete exposition.

basic mathematic definition: let $\mathcal{X}$ be a non-empty set sometimes referred to as the index set, and k a symmetric real-valued[2] function on $\mathcal{X} \times \mathcal{X}$. For practitioners of kernel methods, a kernel is above all a positive definite function in the following sense:

Definition 1 (Real-valued Positive Definite Kernels) *A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite (p.d.) kernel on $\mathcal{X}$ if*

$$\sum_{i,j=1}^{n} c_i c_j \, k\left(x_i, x_j\right) \geq 0, \tag{2}$$

holds for any $n \in \mathbb{N}, x_1, \ldots, x_n \in \mathcal{X}$ and $c_1 \ldots, c_n \in \mathbb{R}$.

kernel matrices derived from kernel functions: one can easily deduce from Definition 1 that the set of p.d. kernels is a closed, convex pointed cone[3]. Furthermore, the positive definiteness of kernel functions translates in practice into the positive definiteness of so called *Gram* matrices, that is matrices of kernel evaluations built on a sample of points $X = \{x_i\}_{i \in I}$ in $\mathcal{X}$,

$$K_X = \left[k(x_i, x_j)\right]_{i,j \in I}.$$

Elementary properties of the set of kernel functions such as its closure under pointwise and tensor products are directly inherited from well known results in Kronecker and Schur (or Hadamard) algebras of matrices (Bernstein, 2005, §7).

[1] for $h > 0$, $k_h(x, y) = \frac{3}{4}\left(1 - \left(\frac{x-y}{h}\right)^2\right)^+$

[2] kernels are usually complex valued in the mathematical literature; we only consider the real case here, which is the common practice in machine learning.

[3] A set C is a cone if for any $\lambda \geq 0, x \in C \Rightarrow \lambda x \in C$, pointed if $x \in C, -x \in C \Rightarrow x = 0$

kernel matrices created using other kernel matrices: kernel matrices for a sample X can be obtained by applying transformations r that conserve positive definiteness to a prior Gram matrix K_X. In such a case the matrix $r(K_X)$ can be used directly on that subspace, namely without having to define explicit formulas for the constructed kernel on the whole space $\mathcal{X} \times \mathcal{X}$. A basic example is known as the empirical kernel map, where the square map $r : M \to M^2$ can be used on a matrix (Schölkopf et al., 2002). More complex constructions are the computation of the diffusion kernel on elements of a graph through its Laplacian matrix (Kondor and Lafferty, 2002), or direct transformations of the kernel matrix through unlabelled data (Sindhwani et al., 2005).

strict and semi-definite positiveness: functions for which the sum in Equation (2) is (strictly) positive when $c \neq 0$ are sometimes referred to as positive definite functions, in contrast with functions for which this sum is only non-negative, which are termed positive *semi*-definite. We will use for convenience throughout this report the term positive definite for kernels that simply comply with non-negativity, and will consider indifferently positive semi-definite and positive definite functions. Most theoretical results that will be presented in this report are also indifferent to this distinction, and in numerical practice definiteness and semi-definiteness will be equivalent since most estimation procedures consider a regularization of some form on the matrices to explicitly lower bound their conditioning number[4].

the importance of positive definiteness : Equation (2) distinguishes general measures of similarity between objects and a kernel function. The requirement of Equation (2) is important when seen from (at least) two perspective. First, the usage of positive definite matrices is a key assumption in convex programming Boyd and Vandenberghe (2004). In practice the positive definiteness of kernel matrices ensures that kernel algorithms such as Gaussian processes or support vector machines converge to a relevant solution[5] Second, the positive definiteness assumption is also a key assumption of the functional view described below in reproducing kernel Hilbert spaces theory.

2.2 Reproducing Kernels

Kernels can be also viewed from the functional analysis viewpoint, since to each kernel k on $\mathcal{X}$ is associated a Hilbert space $\mathcal{H}_k$ of real-valued functions on $\mathcal{X}$.

Definition 2 (Reproducing Kernel) *A real-valued function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a reproducing kernel of a Hilbert space $\mathcal{H}$ of real-valued functions on $\mathcal{X}$ if and only if*

$$i)\ \forall t \in \mathcal{X},\ \ k(\cdot, t) \in \mathcal{H};$$
$$ii)\ \forall t \in \mathcal{X}, \forall f \in \mathcal{H},\ \ \langle f, k(\cdot, t)\rangle = f(t).$$

[4] that is the ratio of the biggest to the smallest eigenvalue of a matrix

[5] (Haasdonk, 2005; Luss and D'Aspremont, 2008) show however that arbitrary similarity measures can be used with slightly modified kernel algorithms

Condition (ii) above is called the *reproducing property*. A Hilbert space that is endowed with such a kernel is called a reproducing kernel Hilbert space (RKHS) or a proper Hilbert space. Conversely, a function on $\mathcal{X} \times \mathcal{X}$ for which such a Hilbert space $\mathcal{H}$ exists is a reproducing kernel and we usually write $\mathcal{H}_k$ for this space which is unique. It turns out that both Definitions 1 and 2 are equivalent, a result known as the Moore-Aronszajn theorem (Aronszajn, 1950). First, a reproducing kernel is p.d., since it suffices to write the expansion of Equation (2) to obtain the squared norm of the function $\sum_{i=1}^{n} c_i k(x_i, \cdot)$, that is

$$\sum_{i,j=1}^{n} c_i c_j \, k\left(x_i, x_j\right) = \left\| \sum_{i=1}^{n} c_i k(x_i, \cdot) \right\|_{\mathcal{H}}^2, \tag{3}$$

which is non-negative. To prove the opposite in a general setting, that is not limited to the case where $\mathcal{X}$ is compact which is the starting hypothesis of the Mercer representation theorem (Mercer, 1909) reported in (Schölkopf and Smola, 2002, p.37), we refer the reader to the progressive construction of the RKHS associated with a kernel k and its index set $\mathcal{X}$ presented in (Berlinet and Thomas-Agnan, 2003, §1.3). In practice, the RKHS boils down to the completed linear span of elementary functions indexed by $\mathcal{X}$, that is

$$\mathcal{H}_k \stackrel{\text{def}}{=} \overline{\text{span}}\{k(x, \cdot), x \in \mathcal{X}\},$$

whereby completeness we mean that all Cauchy sequences of functions converge.

the parallel between a kernel and a RKHS: Definition 2 may seem theoretical at first glance, but its consequences are are however very practical. Defining a positive definite kernel k on any set $\mathcal{X}$ suffices to inherit a Hilbert space of functions $\mathcal{H}_k$ which may be used to pick candidate functions for a given data-analysis task. By selecting a kernel k, we hope that the space $\mathcal{H}_k$ – though made up of *linear* combinations of elementary functions – may contain useful functions with low norm. This is in many ways equivalent to defining a space of low degree polynomials and its dot-product in order to approximate an arbitrary function of interest on a given interval $[a, b]$ on the real line with a polynomial of low norm.

functional norm in a RKHS: another crucial aspect of RKHS is the simplicity of their induced norms and dot-products which are both inherited from the reproducing kernel. The fact that this norm is easy to compute for finite expansions, as seen in Equation (3), is an important property which has direct implications when considering regularized estimation schemes, introduced in Section 4.3 and more precisely Equation (15). Additionally, the dot-product between two functions in the RKHS can be expressed as

$$\left\langle \sum_{i \in I} a_i k(x_i, \cdot), \sum_{j \in J} b_j k(y_j, \cdot) \right\rangle = \sum_{i \in I, j \in J} a_i b_j k(x_i, y_j).$$

which only depend on kernel evaluations on pairs (x_i, y_j) and on the weights a_i and b_j. The fact that in $\mathcal{H}_k$ the dot-product $\langle k(x,\cdot), k(y,\cdot)\rangle_{\mathcal{H}_k}$ is equal to $k(x,y)$ illustrates an alternative view, namely that a kernel is a disguised dot-product.

2.3 Kernels as Feature Maps

The theorem below (Berlinet and Thomas-Agnan, 2003, p.22) gives an interpretation of kernel functions, seen as dot-products between feature representations of their arguments in a space of sequences.

Theorem 1 *A function k on $\mathcal{X} \times \mathcal{X}$ is a positive definite kernel if and only if there exists a set T and a mapping ϕ from $\mathcal{X}$ to $l^2(T)$, the set of real sequences $\{u_t, t \in T\}$ such that $\sum_{t\in T} |u_t|^2 < \infty$, where*

$$\forall (x, y\,) \in \mathcal{X} \times \mathcal{X}\,,\; k(x,y) = \sum_{t\in T} (\phi(x))_t (\phi(y))_t = \langle \phi(x), \phi(y)\rangle_{l^2(X)}$$

The proof is derived from the fact that for any Hilbert space (notably $\mathcal{H}_k$) there exists a space $l^2(X)$ to which it is isometric. As can be glimpsed from this sketch, the feature map viewpoint and the RKHS one are somehow redundant, since

$$x \mapsto k(x,\cdot),$$

is a feature map by itself. If the RKHS is of finite dimension, functions in the RKHS are exactly the dual space of the Euclidian space of feature projections. Although closely connected, it is rather the feature map viewpoint than the RKHS one which actually spurred most of the initial advocation for kernel methods in machine learning, notably the SVM as presented in (Cortes and Vapnik, 1995; Schölkopf and Smola, 2002). The the latter references present kernel machines as mapping data-entries into high-dimensional feature spaces,

$$\{x_1, \cdots, x_n\} \mapsto \{\phi(x_1), \cdots, \phi(x_n)\},$$

to find a linear decision surface to separate the points in two distinct classes of interest. This interpretation actually coincided with the practical choice of using polynomial kernels[6] on vectors, for which the feature space is of finite dimension and well understood as products of monomials up to degree d.

The feature map approach was progressively considered to be restrictive in the literature, since it imposes to consider first the extracted features and then compute the kernel that matches them. Furthermore, useful kernels obtained directly from a similarity between objects do not always translate into feature maps which can be easily described, as in diffusion kernels on graphs for instance (Kondor and Lafferty, 2002). Kernels without explicit feature maps may also be obtained through the polynomial combination of several kernels. The feature map formulation, particularly advocated in the early days of SVM's, also misled some observers into thinking that the kernel mapping was but a

[6] $k(x,y) = (\langle x, y\rangle + b)^d, d \in \mathbb{N}, b \in \mathbb{R}^+$

piece of the SVM machinery. Instead, the SVM should be rather seen as an efficient computational approach – among many others – deployed to select a "good" function f in the RKHS $\mathcal{H}_k$ given a learning sample, as presented in Section 4.3.

2.4 Kernels and Distances, a Discussion

We discuss in this section possible parallels between positive definite kernels and distances. Kernel methods are often compared to distance based methods such as nearest neighbors. We would like to point out a few differences between their two respective ingredients, kernels k and distances d.

Definition 3 (Distances) *Given a space $\mathcal{X}$, a nonnegative-valued function d on $\mathcal{X} \times \mathcal{X}$ is a distance if it satisfies the following axioms, valid for all elements x, y and z of $\mathcal{X}$:*

- *$d(x, y) \geq 0$, and $d(x, y) = 0$ if and only if $x = y$.*
- *$d(x, y) = d(y, x)$ (symmetry),*
- *$d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)*

Distances which do not satisfy the requirement that $d(x, y) = 0$ only if $x = y$ are called pseudo-metrics. We write $\mathcal{D}(\mathcal{X})$ for the set of distances in $\mathcal{X}$. The missing link between kernels and distances is given by a particular type of kernel function, which includes all negations of positive definite kernels as a particular case,

Definition 4 (Negative Definite Kernels) *A symmetric function $\psi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a negative definite (n.d.) kernel on $\mathcal{X}$ if*

$$\sum_{i,j=1}^{n} c_i c_j \, \psi(x_i, x_j) \leq 0 \tag{4}$$

holds for any $n \in \mathbb{N}, x_1, \ldots, x_n \in \mathcal{X}$ and $c_1 \ldots, c_n \in \mathbb{R}$ such that $\sum_{i=1}^{n} c_i = 0$.

A matrix interpretation of this is that for any set of points $x_1, \cdots, x_n$ and vectors of weights $c \in \mathbb{R}^n$ in the hyperplane $\left\{ y \mid 1^T y = 0 \right\}$. we necessarily have that $c^T \Psi c \leq 0$ with $\Psi = [\psi(x_i, x_j)]_{i,j}$. A particular family of distances known as Hilbertian norms can be considered as negative definite kernels as pointed out in Hein and Bousquet (2005). This link is made explicitly by (Berg et al., 1984, Proposition 3.2) given below

Proposition 1 *Let $\mathcal{X}$ be a nonempty set and $\psi : \mathcal{X} \times \mathcal{X}$ be a negative definite kernel. Then there is a Hilbert space H and a mapping $x \mapsto \phi(x)$ from X to H such that*

$$\psi(x, y) = \|\phi(x) - \phi(y)\|^2 + f(x) + f(y), \tag{5}$$

where $f : \mathcal{X} \to \mathbb{R}$ is a real-valued function on X. If $\psi(x, x) = 0$ for all $x \in \mathcal{X}$ then f can be chosen as zero. If the set of pairs such that $\psi(x, y) = 0$ is exactly $\{(x, x), x \in \mathcal{X}\}$ then $\sqrt{\psi}$ is a distance.

negative definite kernels and distances: the parallel between negative definite kernels and distances is thus clear: whenever a n.d. kernel vanishes on the *diagonal*, that is the set $\{(x,x), x \in \mathcal{X}\}$, and is zero only on the diagonal, then its square root is a distance for $\mathcal{X}$. More generally, to each negative definite kernel corresponds a decomposition (5) which can be exploited to recover a Hilbertian distance by substracting to $\psi(x,y)$ the part corresponding to $f(x)$ and $f(y)$, $\psi(x,x)/2 + \psi(y,y)/2$. This distance may however be degenerated if the resulting function is null outside the diagonal. On the other hand, to each distance does not correspond necessarily a negative definite kernel and there are numerous examples of distances which are not Hilbertian metrics such as the Monge-Kantorovich distance (Naor and Schechtman, 2007) or most common variations of the edit distance (Vert et al., 2004).

negative definite kernels and positive definite kernels: on the other hand, n.d. kernels can be identified with a subfamily of p.d. kernels known as infinitely divisible kernels. A nonnegative-valued kernels k is said to be infinitely divisible if for every $n \in \mathbb{N}$ there exists a positive definite kernel k_n such that $k = (k_n)^n$ and k_n is positive definite. More simply put, k is infinitely divisible if k^t is positive definite for all $t > 0$.

Example 1 *The Gaussian kernel between two vectors of $\mathbb{R}^d$ is infinitely divisible. The identity*

$$k_\sigma(x,y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}} = \left(e^{-\frac{\|x-y\|^2}{2n\sigma^2}}\right)^n,$$

suffices to prove this property.

Here follows a slightly simplified version of (Berg et al., 1984, Proposition 2.7) which provides a key interpretation:

Proposition 2 *For a p.d. kernel $k \geq 0$ on $\mathcal{X} \times \mathcal{X}$, the following conditions are equivalent*

(i) $-\log k \in \mathcal{N}(\mathcal{X})$,

(ii) k^t *is positive definite for all* $t > 0$.

If k satisfies either, k is said to be infinitely divisible,

Figure 3 provides a schematic view on the relationships between distances, negative definite kernels and positive definite kernels. The reader should also keep in mind that

(i) $\mathcal{D}(\mathcal{X})$ is a cone;

(ii) $\mathcal{N}(\mathcal{X})$ is a cone. Additionally, if $\psi(x,x) \geq 0$ for all $x \in \mathcal{X}$, $\psi^\alpha \in \mathcal{N}(\mathcal{X})$ for $0 < \alpha < 1$ since

$$\psi^\alpha = \frac{\alpha}{\Gamma(1-\alpha)} \int_0^\infty t^{-\alpha-1}(1 - e^{-t\psi})dt$$

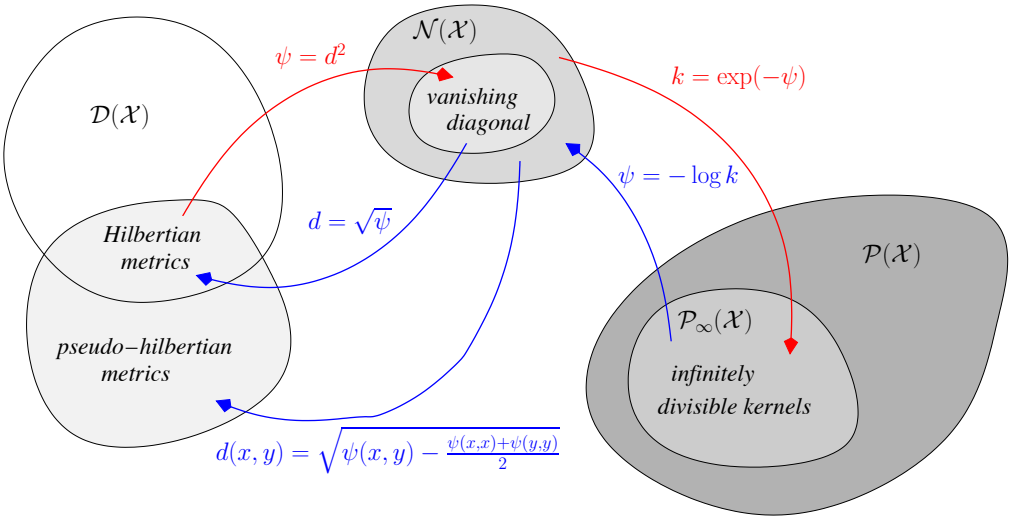

Figure 3: A schematic view of the relationships between $\mathcal{D}(\mathcal{X})$ on the left, its subset of Hilbertian metrics and their one-to-one mapping with negative-definite kernels with vanishing diagonal, itself contained in the set $\mathcal{N}(\mathcal{X})$. Note that the set of negative-definite kernels is in direct correspondence with the set $\mathcal{P}_\infty(\mathcal{X})$, the subset of infinitely divisible positive definite kernels of $\mathcal{P}(\mathcal{X})$.

and $\log(1+\psi) \in \mathcal{N}(\mathcal{X})$ since

$$\log(1+\psi) = \int_0^\infty (1 - e^{-t\psi}) \frac{e^{-t}}{t} dt.$$

Additionally, if $\psi > 0$, then $\log(\psi) \in \mathcal{N}$ since

$$\log(\psi) = \lim_{c\to\infty} \log\left(\psi + \frac{1}{c}\right) = \lim_{c\to\infty} \log(1 + c\psi) - \log c.$$

These results are presented in (Berg et al., 1984, Corollary 2.10) along others.

(iii) $\mathcal{P}(\mathcal{X})$ is a cone. Additionally, the pointwise product $k_1 k_2$ of two p.d. kernels is also in $\mathcal{P}(\mathcal{X})$, and as a consequence so is k^n for $n \in \mathbb{N}$;

and may refer to (Berg et al., 1984, p.79) for other numerous results and important examples.

3 Designing Kernels from Statistical Knowledge

We follow the relatively theoretical exposition of the previous chapter with a more practical exposition. Although the mathematical elements presented above explain most of the desirable properties of kernel machines, notably the convexity of most optimization carried out when estimating kernel machines, the view taken by practitioners on kernels is most often linked with that of a *similarity* measure between objects. Namely that for two objects x and y the value $k(x, y)$ can be a reliable quantification of how x and y are similar. This similarity may be chosen arbitrarily, incorporating as much prior knowledge on the objects as possible without any connection to the task itself, or rather considered under the light of a given task.

Example 2 *The usual criterion for two texts to be similar might be that they share the same languages and/or topics of interest and/or overall length, but a very specialized algorithm might solely focus on the occurrence of a single word within their body. Two photographs might be qualified as similar if they display similar colors or shapes. For other tasks, rather than the image itself their date or their location might be the key criterion, regardless of their pictorial content. Two videos might be qualified as similar if they display the same person for a fixed length of time or if they were broadcasted through the same television channel.*

The following sections start with the relatively simple example of defining kernels on vectors. We address objects with more complex structures later in this chapter using statistical modeling.

3.1 Classical Kernels for Vectors

3.1.1 Vectors in $\mathbb{R}^n$

Finite dimensional Vectors are a fundamental tool to represent natural phenomenon as numeric data. Vectors are known to be easy to manipulate by both algorithms and computer codes, and as such positive definite kernels taking vector arguments can be easily constructed. The canonical dot-product on a vector-space of finite dimensions, also known as the linear kernel $k(x, y) = x \cdot y$ is the most fundamental example. We will use three lemmas to show the reader how most classical kernels can be easily reconstructed through the linear kernel. For a family of kernels $k_1, \ldots, k_n, \ldots$

- The sum $\sum_{i=1} n\lambda_i k_i$ is positive definite, given $\lambda_1, \ldots, \lambda_n \geq 0$
- The product $k_1^{a_1} \cdots k_n^{a_n}$ is positive definite, given $a_1, \ldots, a_n \in \mathbb{N}$
- The limit $k \stackrel{\text{def}}{=} \lim_{n\to\infty} k_n$ is positive definite if the limit exists.

Using these properties listed in (Berg et al., 1984), we can reconstruct

- the polynomial kernel $k_p(x,y) = (x \cdot y + b)^d, b > 0, d \in \mathbb{N}$, simply because $b > 0$ is a p.d. kernel, and so is $(x \cdot y + b)$ as a consequence of the first property, and so is $(x \cdot y + b)^d$ as a consequence of the second.

- the Gaussian kernel $k_\sigma(x,y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$ which can be rewritten in the following form

$$k_\sigma(x,y) = \left[e^{-\frac{\|x\|^2}{2\sigma^2}} e^{-\frac{\|y\|^2}{2\sigma^2}}\right] \cdot \left[\sum_{i=0}^{\infty} \frac{(x \cdot y)^i}{i!}\right]$$

  The term in the first brackets is trivially a kernel, and so is the term in the second part as a limit of positive definite kernels.

- the all-subset Gaussian kernel in $\mathbb{R}^d$,

$$k(x,y) = \prod_{i=1}^{d} (1 + a e^{-b(x_i - y_i)^2}) = \sum_{I \subset \{1,\cdots,d\}} a^{\#(J)} e^{-b\|x_J - y_J\|^2}$$

  where x_J is the vector of size $\#(J)$, the cardinal of J, with coordinates $x_i, i \in J$.

3.1.2 Vectors in $\mathbb{R}^n_+$ and Histograms

Histograms are frequently encountered in applications of machine learning to real-life problems. Indeed, most natural phenomena produce visible data, which the practitioner is likely to count to describe reality. As a consequence, most observations are usually available under the form of nonnegative vectors of counts, which, if normalized, yield histograms of frequencies. Metrics or divergences for general probability measures, the obvious generalization of histograms, is the object of study of information geometry Amari and Nagaoka (2001). However, as hinted in Section 2.4, a proper understanding of metrics and divergences for a certain class of objects cannot be immediately applied to define positive definite kernels. Indeed, the Kullback-Leibler divergence, which has a fundamental importance in information geometry, cannot be used as such in kernel methods as it is neither symmetric nor positive/negative definite.

elementary kernels on positive measures: it has been shown however in Hein and Bousquet (2005) that the following family of squared metrics, respectively the Jensen Divergence, the χ-square, Total Variation and two variations of the Hellinger distance, are all negative-definite kernels:

$$\psi_{JD}(\theta,\theta') = h\left(\frac{\theta+\theta'}{2}\right) - \frac{h(\theta)+h(\theta')}{2},$$

$$\psi_{\chi^2}(\theta,\theta') = \sum_i \frac{(\theta_i - \theta'_i)^2}{\theta_i + \theta'_i}, \quad \psi_{TV}(\theta,\theta') = \sum_i |\theta_i - \theta'_i|,$$

$$\psi_{H_2}(\theta,\theta') = \sum_i |\sqrt{\theta_i} - \sqrt{\theta'_i}|^2, \quad \psi_{H_1}(\theta,\theta') = \sum_i |\sqrt{\theta_i} - \sqrt{\theta'_i}|.$$

As a consequence, these metrics can all be used to definite positive definite kernels using Proposition 2 and the following formula:

$$k(\theta, \theta') = e^{-\frac{1}{t}\psi},$$

with $t > 0$. Although histograms appear frequently in the study of objects such as images, through histograms of colors, and texts, through bags-of-words representations, their usage alone is restrictive when studying objects that carry a finer structure. In such a case, probability distributions that are tailored to capture better the interdependencies between smaller components in those objects can be used to define kernels as presented in the next section.

3.2 Statistical Modeling and Kernels

Fisher kernel: Jaakkola and Haussler (1999) first thought of using generative models to build kernels that would provide in turn the necessary inputs of *discriminative* machines, that is kernel classifiers. Although the principle outlined in the next lines can be applied to different pairs of datatypes/generative models, we follow the original presentation of their paper which focused on sequences. Jaakkola and Haussler (1999) observed that the hidden Markov model (HMM), which is known to capture efficiently the behaviour of amino-acid sequences can be used as an efficient feature extractor. The authors did so by defining for each considered sequence a vector of features derived from an estimated HMM model, namely the Fisher score. Given a measurable space $(\mathcal{X}, \mathcal{B}, \nu)$ and a parametric family of absolutely continuous measures of $\mathcal{X}$ represented by their densities $\{p_\theta, \theta \in \Theta \subset \mathbb{R}^d\}$, the Fisher kernel between two elements x, y of $\mathcal{X}$ is

$$k_{\hat{\theta}}(x, y) = \left(\frac{\partial \ln p_\theta(x)}{\partial \theta}\Big|_{\hat{\theta}} \right)^T J_{\hat{\theta}}^{-1} \left(\frac{\partial \ln p_\theta(y)}{\partial \theta}\Big|_{\hat{\theta}} \right),$$

where $\hat{\theta}$ is a parameter selected beforehand to match the whole training set, in maximum likelihood sense for instance, and $J_{\hat{\theta}}$ is the Fisher information matrix computed in $\hat{\theta}$. The statistical model not only acts as a *feature extractor* through the score vectors, but also defines the *Mahalanobis metric* associated with these vectors through $J_{\hat{\theta}}$. Note that Jaakkola et al. (1999) also provide the following alternative formulation

$$k_{\hat{\theta}}(x, y) = e^{-\frac{1}{\sigma^2}\left(\nabla_{\hat{\theta}} \ln p_\theta(x) - \nabla_{\hat{\theta}} \ln p_\theta(y)\right)^T J_{\hat{\theta}}^{-1} \left(\nabla_{\hat{\theta}} \ln p_\theta(x) - \nabla_{\hat{\theta}} \ln p_\theta(y)\right)}. \qquad (6)$$

extensions to the Fisher kernel: the proposal of the Fisher kernel fostered further research, notably in (Tsuda et al., 2002a; Smith and Gales, 2002). The motivation behind these contributions was to overcome the limiting assumption that the parameter $\hat{\theta}$ on which the score vectors are evaluated is unique and fits the whole set of points at hand. Rather, Tsuda et al. (2002a) and Smith and Gales (2002) proposed simultaneously to incorporate in the context of binary

classification two parameters $\hat{\theta}_1$ and $\hat{\theta}_2$ for each class respectively, and consider the score vector of the likelihood ratio between the two classes evaluated in x,

$$\phi_{\hat{\theta}_1,\hat{\theta}_2} : x \mapsto \left(\frac{\partial \ln \frac{p_{\theta_1}(x)}{p_{\theta_2}(x)}}{\partial \vartheta} \Big|_{\hat{\vartheta}=(\hat{\theta}_1,\hat{\theta}_2)} \right),$$

where $\vartheta = (\theta_1, \theta_2)$ is in Θ^2, to propose instead the kernel

$$(x, y) \mapsto \phi_{\hat{\theta}_1,\hat{\theta}_2}(x)^T \phi_{\hat{\theta}_1,\hat{\theta}_2}(y).$$

The Fisher kernel was also studied from a theoretical perspective when used in conjunction with a logistic regression (Tsuda et al., 2004).

mutual information kernels: the Fisher kernel is related to a wider class of kernels coined down as mutual information kernels by Seeger (2002). Starting also from a set of distributions $\{p_\theta, \theta \in \Theta\}$ where Θ is measurable, and from a given prior $\omega \in L_2(\Theta)$, the mutual information kernel k_ω between two elements x and y is defined as

$$k_\omega(x, y) = \int_\Theta p_\theta(x) p_\theta(y)\, \omega(d\theta). \tag{7}$$

As noted in (Seeger, 2002), the Fisher kernel can be regarded as a maximum *a posteriori* approximation of the mutual information kernel, by setting the prior ω to the multivariate Gaussian density $\mathcal{N}(\hat{\theta}, J_{\hat{\theta}}^{-1})$, following the approximation of Laplace's method. Let us review this claim in more details: given an object x and a parameter $\hat{\theta}$, the following approximation

$$\log p_\theta(x) \approx \log p_{\hat{\theta}}(x) + \nabla_{\hat{\theta}} \ln p_\theta(x)^T (\theta - \hat{\theta})$$

can be rewritten using the notation

$$\Phi(x) = \nabla_{\hat{\theta}} \ln p_\theta(x) = \frac{\partial \ln p_\theta(x)}{\partial \theta}\Big|_{\hat{\theta}}$$

as

$$\log p_\theta(x) \approx \log p_{\hat{\theta}}(x) + \Phi(x)(\theta - \hat{\theta}).$$

Using a Gaussian approximation for ω yields a change in Equation 7 as

$$\begin{aligned} k(x, y) &= \int_\Theta p_\theta(\mathbf{x}) p_\theta(\mathbf{y})\, \omega(d\theta), \\ &\approx C \int_\Theta e^{\log p_{\hat{\theta}}(x) + \Phi(x)^T(\theta-\hat{\theta})} e^{\log p_{\hat{\theta}}(y) + \Phi(y)^T(\theta-\hat{\theta})}\ e^{-(\theta-\hat{\theta})^T J_{\hat{\theta}}(\theta-\hat{\theta})} d\theta \\ &= C p_{\hat{\theta}}(x) p_{\hat{\theta}}(y) \int_\Theta e^{(\Phi(x)+\Phi(y))^T(\theta-\hat{\theta}) + (\theta-\hat{\theta})^T J_{\hat{\theta}}(\theta-\hat{\theta})} d\theta \\ &= C' p_{\hat{\theta}}(x) p_{\hat{\theta}}(y) e^{\frac{1}{2}(\Phi(x)+\Phi(y))^T J_{\hat{\theta}}^{-1}(\Phi(x)+\Phi(y))} \end{aligned} \tag{8}$$

it is then easy to check that the kernel

$$\tilde{k}(x,y) = \frac{k(x,y)}{\sqrt{k(x,x)k(y,y)}}$$

is equal to the Fisher kernel given in its form of Equation (6). Cuturi and Vert (2005) propose an example of a mutual information kernel defined on strings that can be computed exactly. In the latter work the set of distributions $\{p_\theta, \theta \in \Theta\}$ is a set of Markov chain densities on sequences with finite depths. The prior ω is a combination of branching process priors for the structure of the chain and mixtures of Dirichlet priors for the transition parameters. This setting yields closed computational formulas for the kernel through previous work led in universal coding (Willems et al., 1995; Catoni, 2004). The computations can be carried in a number of elementary operations that is linear in the lengths of the inputs x and y.

marginalized kernels: in the framework of sequence analysis first (Tsuda et al., 2002b), and then in comparisons of graphs (Kashima et al., 2003), further attention was given to latent variable models to define kernels in a way that also generalized the Fisher kernel. In a latent variable model, the probability of emission of an element x is conditioned by an unobserved latent variable $s \in \mathcal{S}$, where $\mathcal{S}$ is a finite space of possible states. When a string is considered under the light of a hidden Markov model, to its chain $x = x_1 \cdots x_n$ of letters is associated a similar sequence $s = s_1 \cdots s_n$ of states that is not usually observed. When the sequence of states s is known, the probability of x under such a model is then determined by the marginal probabilities $p(x_i|s_i)$. Building adequate transition structures for the emitting states, and their corresponding emission probabilities is one of the goals of HMM estimations. The marginalized kernel assumes that this sequence is not known for objects x and y, but it performs, given an available structure of states, an averaging

$$k(x,y) = \sum_{s \in \mathcal{S}} \sum_{s' \in \mathcal{S}} p(s|x)\, p(s'|y)\, \kappa\left((x,s),(y,s')\right)$$

of arbitrary kernel evaluations κ weighted by posterior probabilities which are estimated from data. In this setting, κ can be any arbitrary kernel on $\mathcal{X} \times \mathcal{S}$. For particular choices of κ the kernel can be computed in closed form, both on sequences and graphs (Mahé et al., 2004).

kernels defined on maximum-likelihood parameters: the previous approaches make different uses of a statistical model $p_\theta(x)$. In mutual information kernels $p_\theta(x)$ is treated as a feature indexed by a large set of parameters $\theta \in \Theta$. For marginalized kernels an unseen, latent variable is added to p_θ, $p_\theta(x,s)$ and while θ is kept constant the integration of $p_\theta(x,s)p_\theta(y,s')$ is led over all possible combinations of latent variables (s,s'). A third approach, conceptually simpler, compares two objects by considering directly the parameters θ and θ' that fits

them better respectively, that is, map first

$$x \mapsto \hat{\theta}_x, \quad y \mapsto \hat{\theta}_y,$$

through maximum likelihood estimation for instance, and then compare x and y through a kernel k_Θ on Θ,

$$k(x, y) = k_\Theta(\hat{\theta}_x, \hat{\theta}_y).$$

Under this form, the topic of defining interesting functions k on $(\hat{\theta}_x, \hat{\theta}_y)$ is loosely connected with information geometry (Amari and Nagaoka, 2001) and one may use for simple densities some of the kernels presented in Section 3.1.2. For more complex spaces of parameters Θ one may refer to Jebara et al. (2004) which presents the family of kernels

$$k_\beta(x, y) = \int_{\mathcal{X}} p_{\hat{\theta}_x}(z)^\beta p_{\hat{\theta}_y}(z)^\beta dz$$

for $\beta > 0$, the case $\beta = \frac{1}{2}$ being the well known Bhattacharrya affinity between densities. The authors review a large family of statistical models for which these kernels can be computed in closed form, ranging from graphical models, Gaussian multivariate densities, multinomials and hidden Markov models.

information diffusion kernel: aiming also at computing kernels of interest on multinomials, Lafferty and Lebanon (2005) propose to follow Kondor and Lafferty (2002) and use diffusion processes to define kernels. To do so they express solutions for the heat equation in the Riemannian manifold induced by the Fisher metric of the considered statistical models, inspired again by information geometry Amari and Nagaoka (2001). They derive *information diffusion* kernels out of such solutions which, when specialized to multinomials, that is elements of the simplex[7], boil down to kernels of the form

$$k_{\Sigma_d}(\theta, \theta') = e^{-\frac{1}{t}\arccos^2(\sqrt{\theta}\cdot\sqrt{\theta'})}, \tag{9}$$

where $t > 0$ is the diffusion parameter. Note that the squared arc-cosine in Equation (9) is the squared geodesic distance between θ and θ' seen as elements from the unit sphere (that is when each θ_i is mapped to $\sqrt{\theta_i}$). Based on the seminal work of Schoenberg (1942), Zhang et al. (2005) rather advocate the direct use of the geodesic distance:

$$k_{\Sigma_d}(\theta, \theta') = e^{-\frac{1}{t}\arccos(\sqrt{\theta}\cdot\sqrt{\theta'})},$$

They prove that the geodesic distance is a negative definite kernel on the *whole sphere*, while its square used in Equation (9) is not. If the points θ and θ' are restricted to lie in the positive orthant, which is the case for multinomials, both approaches yield however positive definite kernels.

[7] writing Σ_d for the canonical simplex of dimension d, i.e., $\Sigma_d = \{\xi = (\xi_i)_{1 \leq i \leq d} : \xi_i \geq 0, \sum \xi_i = 1\}$.

3.3 Semigroup Kernels and Integral Representations

Most positive definite kernels on groups, which includes kernels on vectors of $\mathbb{R}^n$ as described in Section (3.1.1) can be considered as semigroup kernels. A semigroup is an algebraic structure that is simple enough to fit most datatypes and rich enough to allow for a precise study of the kernels defined on them. Most of the material of this section is taken from (Berg et al., 1984), but the interest reader may consult the additional references (Devinatz, 1955; Ehm et al., 2003). Let us start this section with the following definitions.

semigroups: a semigroup $(\mathcal{S}, +)$ is a nonempty set $\mathcal{S}$ endowed with an *associative composition* $+$ which admits a neutral element 0, that is such that $\forall x \in \mathcal{S}, x + 0 = x$. An involutive semigroup $(\mathcal{S}, +, *)$ is a semigroup endowed with an involution $*$ which is a mapping $\mathcal{S} \to \mathcal{S}$ such that for any x in $\mathcal{S}$ $(x^*)^* = x$. Let us provide some examples of semigroups:

- $\mathcal{S}$ is the set of strings formed with letter from a given alphabet, $+$ is the concatenation operation, 0 is the empty string and $*$ is either the identity or the operation which inverses the order of the letters of a string.
- $\mathcal{S}$ is a group, and $*$ is the inverse operation of the group. $(\mathbb{R}, +, -)$ is a typical example.
- $\mathcal{S}$ is the positive orthant $\mathbb{R}^+$ endowed with the usual addition and $*$ is the identity.

Note that most semigroups considered in the machine learning literature are *abelian* that is operation $+$ is commutative.

semigroup kernels: a semigroup kernel is a kernel k defined through a complex-valued function φ defined on $\mathcal{S}$ such that

$$k(x, y) \stackrel{\text{def}}{=} \varphi(x + y^*).$$

A function φ is a positive definite function if the kernel that can be derived from it as $\varphi(x + y^*)$ is itself positive definite. When $\mathcal{S}$ is a vector space, and hence a group, for two elements x, y of $\mathcal{S}$ one can easily check that most elementary kernels are either defined as

$$k(x, y) = \varphi(x - y),$$

or

$$k(x, y) = \psi(x + y), \tag{10}$$

respectively when $*$ is the minus operation and $*$ is the identity. Kernels build on the former structure will typically emphasize the difference between two elements, given this difference can be computed, and include as their most important example radial basis functions (RBF) and the Gaussian kernel. When

a subtraction between elements cannot be defined as is the case with strings, histograms and nonnegative measures, the form of Equation (10) is better suited as can be seen in some of the examples of Section 3.1.2 and studied in (Cuturi et al., 2005). In this work, the authors study a family of kernels for probability measures μ and μ' by looking at their average $(\mu + \mu')/2$. They narrow down their study to kernels defined through the variance matrix $\Sigma(\frac{\mu+\mu'}{2})$ of their average, and show that

$$k(\mu, \mu') \stackrel{\text{def}}{=} \frac{1}{\sqrt{\det \Sigma \left(\frac{\mu+\mu'}{2}\right)}},$$

is a positive definite kernel between the two measures. This result can be further extended through reproducing kernel Hilbert space theory, yielding a kernel between two clouds of points $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_m\}$ which only depends on the kernel similarity matrices $K_{XY} = [\kappa(x_i, y_j)], K_X = [\kappa(x_i, x_j)]$ and $K_Y = [\kappa(y_i, y_j)]$.

integral representations: semigroup kernels can be expressed as sums of semicharacters, a family of elementary functions on $\mathcal{S}$. A real-valued function ρ on an Abelian semigroup $(S, +)$ is called a semicharacter if it satisfies

(i) $\rho(0) = 1$,

(ii) $\forall s, t \in \mathcal{S},\ \rho(s + t) = \rho(s)\overline{\rho(t)}$,

(iii) $\forall s \in \mathcal{S},\ \rho(s) = \overline{\rho(s^*)}$.

The set of semicharacters defined on S is written S^* while the set of bounded semicharacters can be written as $\hat{S}$. It is trivial to see that every semicharacter is itself a positive definite function. The converse is obviously not true, but it is possible to show that bounded semicharacters are the extremal points of the cone of bounded positive definite functions, therefore providing the following result given by Berg et al. (1984):

Theorem 2 (Integral representation of p.d. functions) *A bounded function $\varphi : S \to \mathbb{R}$ is p.d. if and only if it there exists a non-negative measure ω on $\hat{S}$ such that:*

$$\varphi(s) = \int_{\hat{S}} \rho(s)\ d\omega(\rho).$$

In that case the measure ω is unique.

When S is the Euclidian space $\mathbb{R}^d$ the following results due originally to Bochner (Bochner, 1933; **?**) and Bernstein respectively allow us to characterize kernels for two vectors x and y that depend respectively on $(x - y)$ and $(x + y)$.

identical involution let a kernel k be such that $k(x, y) = \varphi(x - y)$. Then there exists a unique non-negative measure ω on $\mathbb{R}^d$ such that

$$\varphi(x) = \int_{\mathbb{R}^d} e^{ix^T r} d\omega(r);$$

In other words, φ is the Fourier transform of a non-negative measure ω on $\mathbb{R}^d$.

opposite involution let a bounded kernel k be such that $k(x, y) = \psi(x + y)$. Then there exists a unique non-negative measure ω on $\mathbb{R}^d$ such that

$$\psi(x) = \int_{\mathbb{R}^d} e^{-x^T r} d\omega(r);$$

or in other words ψ is the Laplace transform of a non-negative measure ω on $\mathbb{R}^d$.

4 Kernel Machines

Kernel machines are algorithms that select functions with desirable properties in a pre-defined reproducing kernel Hilbert space (RKHS) given sample data. All kernel estimation procedures define first a criterion that is a combination of possibly numerous and different properties. Subsequently, the element f of the RKHS that is the optimum with respect to this criterion is selected following an optimization procedure. Before presenting such algorithms, let us mention an important theoretical challenge that appears when dealing with the estimation of functions in RKHS.

Let $\mathcal{X}$ be a set endowed with a kernel k and $\mathcal{H}_k$ its corresponding RKHS. Choosing a function in an infinite dimension space such $\mathcal{H}$ can become an ill-defined problem when the criterion used to select the function does not have a unique minimizer. The representer theorem formulated below provides a practical answer to this problem when a regularization term is used along with a convex objective.

4.1 The Representer Theorem

Most estimation procedures presented in the statistical literature to perform dimensionality reduction or infer a decision function out of sampled points rely on the optimization of a criterion which is usually carried out over a class of linear functionals of the original data. Indeed, PCA, CCA, logistic regression and least-square regression and its variants (lasso or ridge regression) all look for linear transformations of the original data points to address the learning task. When these optimizations are led instead on an infinite dimensional space of functions, namely in the RKHS $\mathcal{H}_k$, the optimization can be performed in finite subspaces of $\mathcal{H}_k$ if the criterion only depends on a criterion computed on a finite sample of points. This result is known as the representer theorem and explains why so many linear algorithms can be "kernelized" when trained on finite datasets.

Theorem 3 (Representer Theorem (Kimeldorf and Wahba, 1971)) *Let $\mathcal{X}$ be a set endowed with a kernel k and $\mathcal{H}_k$ its corresponding RKHS. Let $\{x_i\}_{1\leq i\leq n}$ be a finite set of points of $\mathcal{X}$ and let $\Psi : \mathbb{R}^{n+1} \to \mathbb{R}$ be any function that is strictly increasing with respect to its last argument. Then any solution to the problem*

$$\min_{f\in\mathcal{H}_k} \Psi\left(f(x_1), \cdots, f(x_n), \|f\|_{\mathcal{H}_k}\right)$$

is in the finite dimensional subspace $\operatorname{span}\{k(x_i,\cdot), 1 \leq i \leq n\}$ *of $\mathcal{H}_k$.*

The theorem in its original form was cast in a more particular setting, where the term $\|f\|_{\mathcal{H}_k}$ would be simply added to an empirical risk as often used in Section 4.3. This generalized version is however important to deal with an unsupervised setting.

4.2 Eigenfunctions in a RKHS of Sample Data Points

For some applications, practitioners are interested first in summarizing the information contained in sample data without a classification or regression problem in mind. This task is broadly categorized as dimensionality reduction and can be seen as a data-dependent way to summarize the information contained in each datapoint to a few numbers. Using a sample $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ of observations of a random variable X taking values in $\mathbb{R}^d$, an *unsupervised* dimensionality reduction technique finds m functions $f_1, \cdots, f_m$ that jointly maximize a statistical fidelity criterion. The classical principal component analysis (Jolliffe, 2002) technique uses variance for example. Other criterions such as the kurtosis of each projection $f_i(X)$ are studied in the more general framework of projection pursuit (Friedman and Tukey, 1974; Huber, 1985). This criterion was adapted to high-dimensional spaces, functional spaces by Ramsay and Dalzell (1991) and more detailed considerations for RKHS by Schölkopf et al. (1998). We study in detail this last approach in this section, as well a canonical correlation analysis in RKHS. Let us first describe these techniques in the simple finite-dimensional case.

Principal component analysis (PCA) 's aim is to define an orthonormal basis $\{v_1, \cdots, v_r\}$ where $r \leq \dim(\mathcal{X})$ of a subspace of dimension r of $\mathcal{X}$ such that for $1 \leq j \leq \dim(\mathcal{X})$,

$$v_j = \underset{v \in \mathcal{X}, \|v\|_{\mathcal{X}}=1, v \perp \{v_1, \cdots, v_{j-1}\}}{\operatorname{argmax}} \operatorname{var}_X[v^T], \tag{11}$$

where for any function $f : \mathcal{X} \to \mathbb{R}$, $\operatorname{var}_X[f]$ denotes the variance functional, which to any real-valued function o $\mathcal{X}$ associates f to $E_X[(f - E_X[f])^2]$. Since we assume that $\mathcal{X}$ is Euclidian, this operator can be written as

$$\operatorname{var}_X[v^T] = v^T \Sigma v$$

where $\Sigma = \mathbb{E}_X[\mathbf{x}\mathbf{x}^T]$. This latter operator can be approximated by the empirical estimator $\operatorname{var}_X^n[v^T] = v^T \Sigma_n v$ where

$$\Sigma_n = \frac{1}{n-1} \sum_{i=1}^{n} \left(x_i - \tfrac{1}{n} \sum_{j=1}^{n} x_j \right) \left(x_i - \tfrac{1}{n} \sum_{j=1}^{n} x_j \right)^T.$$

By the Courant-Fisher-Weyl minmax theorem, the r vectors $v_1, \cdots, v_r$ can be estimated as the r first eigenvectors $\hat{v}_1, \cdots, \hat{v}_r$ of Σ_n.

Canonical correlation analysis (CCA) quantifies linear relationships between two random variables X in $\mathcal{X}$ and Y taken in a set $\mathcal{Y}$, using a sample $\{(x_1, y_1), \cdots, (x_n, y_n)\}$ of joint observations of X and Y. Such tasks appear typically when each index i refers to the same underlying object cast in different modalities (Vert and Kanehisa, 2003). CCA looks for meaningful relationships

between X and Y by focusing on linear projections of X and Y, $\alpha^T X$ and $\beta^T Y$, such that the correlation between $\alpha^T X$ and $\beta^T Y$ is high on average, namely

$$\begin{aligned}(\alpha, \beta) &= \underset{\xi \in \mathcal{X}, \zeta \in \mathcal{Y}}{\operatorname{argmax}} \underset{X,Y}{\mathbf{corr}}[\alpha^T, \beta^T] \\ &= \underset{\alpha \in \mathcal{X}, \beta \in \mathcal{Y}}{\operatorname{argmax}} \frac{\mathbf{cov}_{X,Y}[\alpha^T, \beta^T]}{\sqrt{\operatorname{var}_X[\alpha^T] \operatorname{var}_Y[\beta^T]}}\end{aligned} \tag{12}$$

where for two real valued functions $f : \mathcal{X} \to \mathbb{R}$ and $g : \mathcal{Y} \to \mathbb{R}$ we write

$$\begin{aligned}\operatorname{var}_X[f] &= E_X(f(x) - E_X[f(x)])^2, \\ \operatorname{var}_Y[g] &= E_X(g(y) - E_Y[g(y)])^2, \\ \underset{X,Y}{\mathbf{cov}}[f, g] &= E_{X,Y}[(f(x) - E_X[f(x)])(g(y) - E_Y[g(y)])].\end{aligned}$$

These three quantities can be approximated, or vector projections α^T and β^T or f and g respectively, using the empirical estimators

$$\begin{aligned}\Sigma_X^n &= \tfrac{1}{n-1} \textstyle\sum_{i=1}^n \left(x_i - \tfrac{1}{n} \sum_{j=1}^n x_j\right) \left(x_i - \tfrac{1}{n} \sum_{j=1}^n x_j\right)^T, \\ \Sigma_Y^n &= \tfrac{1}{n-1} \textstyle\sum_{i=1}^n \left(y_i - \tfrac{1}{n} \sum_{j=1}^n y_j\right) \left(y_i - \tfrac{1}{n} \sum_{j=1}^n y_j\right)^T, \\ \Sigma_{XY}^n &= \tfrac{1}{n-1} \textstyle\sum_{i=1}^n \left(x_i - \tfrac{1}{n} \sum_{j=1}^n x_j\right) \left(y_i - \tfrac{1}{n} \sum_{j=1}^n y_j\right)^T,\end{aligned}$$

as $\operatorname{var}_X^n[\alpha^T] = \alpha^T \Sigma_X^n \alpha, \operatorname{var}_Y^n[\beta^T] = \beta^T \Sigma_Y^n \beta$ and $\operatorname{var}_{XY}^n[\alpha^T, \beta^T] = \alpha^T \Sigma_{XY}^n \beta$, yielding the estimator

$$\underset{X,Y}{\overset{n}{\mathbf{corr}}}[\alpha^T, \beta^T] \frac{\alpha^T \Sigma_{XY}^n \beta}{\sqrt{\alpha^T \Sigma_X^n \alpha} \sqrt{\beta^T \Sigma_Y^n \beta}}.$$

α is the leading eigenvector of the matrix

$$(\Sigma_{XX}^n)^{-1/2} \Sigma_{XY}^n (\Sigma_{YY}^n)^{-1} (\Sigma_{XY}^n)^T (\Sigma_{XX}^n)^{-1/2}.$$

β can be recovered as $(\Sigma_{YY}^n)^{-1/2} (\Sigma_{XY}^n)^T (\Sigma_{XX}^n)^{-1/2} \alpha$ with a norm normalized to 1.

Generalization to functions in a RKHS: The "kernelization" of PCA and CCA is intuitive when considering the same criterions on the mappings in $\mathcal{H}$ of the random variables X and Y. We write for convenience $\mathcal{H}_\mathcal{X}$ and $\mathcal{H}_\mathcal{Y}$ for the RKHS associated with $\mathcal{X}$ and $\mathcal{Y}$ with respective kernels $k_\mathcal{X}$ and $k_\mathcal{Y}$. If we cast now the problem as that of estimating a functions f in $\mathcal{H}_\mathcal{X}$ and a couple of functions (f, g) in $\mathcal{H}_\mathcal{X}$ and $\mathcal{H}_\mathcal{Y}$ respectively, we are now looking for vectors in such spaces – that is real-valued functions on $\mathcal{X}$, and $\mathcal{X}$ and $\mathcal{Y}$ respectively – that are directions of interest in the sense that they have adequate values according to the criterions defined in Equations (11) and (12). When considered on the

finite subspaces of $\mathcal{H}_k$ spanned by the datapoints, the two previous optimizations become

$$f_j = \underset{f \in \mathcal{H}_\mathcal{X}, \|f\|_{\mathcal{H}_\mathcal{X}}=1, f \perp \{f_1, \cdots, f_{j-1}\}}{\operatorname{argmax}} \operatorname{var}_X[\langle f, k_\mathcal{X}(x, \cdot)\rangle_{\mathcal{H}_\mathcal{X}}],$$

for $1 \leq j \leq n$ and

$$(f, g) = \underset{f \in \mathcal{H}_\mathcal{X}, g \in \mathcal{H}_\mathcal{Y}}{\operatorname{argmax}} \frac{\mathbf{cov}_{X,Y}[\langle f, k_\mathcal{X}(x, \cdot)\rangle_{\mathcal{H}_\mathcal{X}}, \langle g, k_\mathcal{Y}(y, \cdot)\rangle_{\mathcal{H}_\mathcal{X}}]}{\sqrt{\operatorname{var}_X[\langle f, k_\mathcal{X}(x, \cdot)\rangle_{\mathcal{H}_\mathcal{X}}] \operatorname{var}_Y[\langle g, k_\mathcal{Y}(y, \cdot)\rangle_{\mathcal{H}_\mathcal{Y}}]}}. \tag{13}$$

kernel PCA: the first problem has been termed kernel-PCA by Schölkopf et al. (1998) and boils down to the decomposition of the operator var_X^n into n eigenfunctions. Ramsay and Dalzell (1991) considered the more general case of arbitrary functional spaces. This decomposition can be carried out by considering the $n \times n$ kernel matrix K_X of the n observations, or more precisely its centered counterpart

$$\bar{K}_X = (I_n - \frac{1}{n}\mathbb{1}_{n,n})K_X(I_n - \frac{1}{n}\mathbb{1}_{n,n}).$$

The eigenfunctions f_i can be recovered by considering the eigenvalue/eigenvector pairs (e_i, d_i) of $\bar{K}_X$, that is such that

$$\bar{K}_X = EDE^T$$

where $D = \mathbf{diag}(d)$ and E is an orthogonal matrix. Writing $U = ED^{-1/2}$ we have that

$$f_j(\cdot) = \sum_{i=1}^{n} U_{i,j} k(x_i, \cdot) \tag{14}$$

with $\operatorname{var}_X^n[f_j(x)] = \frac{d_j}{n}$.

kernel CCA: the second optimization, first coined down kernel-CCA by Akaho (2001), is ill-posed if Equation (13) is used directly with a finite sample, and requires a regularization as explained in (Bach and Jordan, 2002; Fukumizu et al., 2007). This approach was hinted earlier by Leurgans et al. (1993) in the framework of functional analysis. The direct maximization

$$(f, g) = \underset{f \in \mathcal{X}, g \in \mathcal{Y}}{\operatorname{argmax}} \frac{\mathbf{corr}_{X,Y}^n[f, g]}{\sqrt{\operatorname{var}_X^n[f] \operatorname{var}_Y^n[g]}}$$

is ill posed: any function f or g for which $\operatorname{var}_X^n[f]$ or $\operatorname{var}_Y^n[g]$ is zero, suffices to maximize the ratio above. Instead, the criterion below, where the variance has been regularized,

$$(f, g) = \underset{f \in \mathcal{X}, g \in \mathcal{Y}}{\operatorname{argmax}} \frac{\mathbf{corr}_{X,Y}^n[f, g]}{\sqrt{(\operatorname{var}_X^n[f] + \lambda\|f\|^2)(\operatorname{var}_Y^n[g] + \lambda\|g\|^2)}},$$

is known to converge to a meaningful solution when λ decreases to zero as n grows with the proper convergence speed (Fukumizu et al., 2007). The finite sample estimates f^n and g^n can be recovered as

$$f^n(\cdot) = \sum_{i=1}^{n} \xi_i \varphi_i(\cdot),$$
$$g^n(\cdot) = \sum_{i=1}^{n} \zeta_i \psi_i(\cdot)$$

where ξ and ζ are the solutions of

$$(\xi, \zeta) = \underset{\xi,\zeta \in \mathbb{R}^n, \xi^T(\bar{K}_X^2 + n\lambda \bar{K}_X)\xi = \zeta^T(\bar{K}_Y^2 + n\lambda \bar{K}_Y)\zeta = 1}{\operatorname{argmax}} \zeta^T \bar{K}_Y \bar{K}_X \xi$$

and

$$\varphi_i(\cdot) = k_{\mathcal{X}}(x_i, \cdot) - \frac{1}{n}\sum_{j=1}^{n} k_{\mathcal{X}}(x_i, \cdot),$$
$$\psi_i(\cdot) = k_{\mathcal{Y}}(y_i, \cdot) - \frac{1}{n}\sum_{j=1}^{n} k_{\mathcal{Y}}(y_i, \cdot),$$

are the centered projections of (x_i) and (y_j) in $\mathcal{H}_{\mathcal{X}}$ and $\mathcal{H}_{\mathcal{Y}}$ respectively. The topic of supervised dimensionality reduction, explored in (Fukumizu et al., 2004), is also linked to the kernel-CCA approach. The author look for a sparse representation of the data that will select an effective subspace for $\mathcal{X}$ and delete all directions in $\mathcal{X}$ that are not correlated to paired observations in $\mathcal{Y}$, based on two samples X and Y. In linear terms, such a sparse representation can be described as a projection of the points of $\mathcal{X}$ into a subspace of lower dimension while conserving the correlations observed with corresponding points in $\mathcal{Y}$.

4.3 Regression, Classification and other Supervised Tasks

Suppose that we wish to infer now from what is observed in the samples X and Y a causal relation between all the points of $\mathcal{X}$ and $\mathcal{Y}$. This type of inference is usually restricted to finding a mapping f from $\mathcal{X}$ to $\mathcal{Y}$ that is consistent with the collected data and has desirable smoothness properties so that it appears as a "natural" decision function seen from a prior perspective. If $\mathcal{X}$ is Euclidian and $\mathcal{Y}$ is $\mathbb{R}$, the latter approach is a well studied field of mathematics known as approximation theory, rooted a few centuries ago in polynomial interpolation of given couples of points, and developed in statistics through spline regression (Wahba, 1990) and basis expansions (Hastie et al., 2009, §5).

empirical risk minimization: statistical learning theory starts its course when a probabilistic knowledge about the generation of the points (x, y) is assumed, and the reader may refer to (Cucker and Smale, 2002) for a valuable review. We skip its rigorous exposition, and favour intuitive arguments next.

A sound guess for the learning rule f would be a function with a low empirical risk,

$$R_c^{\text{emp}}(f) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} c\left(f(x_i), y_i\right),$$

quantified by a cost function $c : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ that penalizes wrong predictions and which is nil on the diagonal. Minimizing directly R_c^{emp} given training sets X and Y is however unlikely to give interesting functions for f. If the function class $\mathcal{F}$ from which f is selected is large, the problem becomes ill-posed in the sense that many solutions to the minimization exist, of which few will prove useful in practice. On the contrary, if the function class is too restricted, there will be no good minimizer of the empirical risk that may serve in practice. To take that tradeoff into account, and rather than constraining $\mathcal{F}$, assume that $J : \mathcal{F} \to \mathbb{R}$ is a function that quantifies the roughness of a function which is used to penalize the empirical risk,

$$R_c^{\lambda}(f) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} c\left(f(x_i), y_i\right) + \lambda J(f). \tag{15}$$

Here $\lambda > 0$ balances the tradeoff between two desired properties for the function f, that is a good fit for the data at hand and a smoothness as measured by J. This formulation is used in most regression and classification settings to select a good function f as the minimizer of

$$\hat{f} = \underset{f \in \mathcal{F}}{\operatorname{argmin}} R_c^{\lambda}. \tag{16}$$

kernel classifiers and regressors: we recover through the formulation of Equation (15) a large variety of methods, notably when the penalization is directly related to the norm of the function in a RKHS:

- When $\mathcal{X}$ is Euclidian and $\mathcal{Y} = \mathbb{R}$, $\mathcal{F} = \mathcal{X}^*$, the dual of $\mathcal{X}$ and $c(f(x), y) = (y - f(x))^2$, minimizing R_c^{λ} is known as least-square regression when $\lambda = 0$; ridge regression (Hoerl, 1962) when $\lambda > 0$ and J is the Euclidian 2-norm; the lasso (Tibshirani, 1996) when $\lambda > 0$ and J is the 1-norm.

- When $\mathcal{X} = [0, 1]$, $\mathcal{Y} = \mathbb{R}$, $\mathcal{F}$ is the space of m-times differentiable functions on $[0, 1]$ and $J = \int_{[0,1]} \left(f^{(m)}(t)\right)^2 dt$, we obtain regression by natural splines of order m. This setting actually corresponds to the usage of thin-base splines which can also be regarded as a RKHS type method (Wahba, 1990), see (Girosi et al., 1995, Table 3) for other examples.

- When $\mathcal{X}$ is an arbitrary set endowed with a kernel k and $\mathcal{Y} = \{-1, 1\}$, $\mathcal{F} = \mathcal{H}_k$, $J = \|\cdot\|_{\mathcal{H}_k}$ and the hinge loss $c(f(x), y) = (1 - yf(x))^+$ is used, we obtain the support vector machine (Cortes and Vapnik, 1995). Using the cost function $c(f(x), y) = \ln(1 + e^{-yf(x)})$, yields an extension of logistic regression known as kernel logistic regression (Zhu and Hastie, 2002).

- When $\mathcal{X}$ is an arbitrary set endowed with a kernel k and $\mathcal{Y} = \mathbb{R}$, $\mathcal{F} = \mathcal{H}_k$, $J = \|\cdot\|_{\mathcal{H}_k}$ and $c(f(x), y) = (|y - f(x)| - \varepsilon)^+$, the ε-insensitive loss function, the solution to this program is known as support vector regression (Drucker et al., 1997).

Note that by virtue of the representer theorem, recalled above as Theorem 3, that whenever $\mathcal{F}$ is set to be a RKHS $\mathcal{H}$, the mathematical program of Equation (16) reaches its minima in the subspace $\mathcal{H}_n$ spanned by the kernel functionals evaluated on the sample points, that is

$$\hat{f} \in \operatorname{span} k(x_i, \cdot),$$

hence the function f in Equation (15) can be explicitly replaced by a finite expansion

$$f = \sum_{i=1}^{n} a_i k(x_i, \cdot), \tag{17}$$

and the corresponding set of feasible solutions $f \in \mathcal{H}$ by $f \in \mathcal{H}_n$ and more simply $a \in \mathbb{R}^n$ using Equation (17). The reader may consult (Steinwart and Christmann, 2008) for an exhaustive treatment.

kernel graph inference: we quote another example of a supervized RKHS method. In the context of supervised graph inference, Vert and Yamanishi (2005) consider a set of connected points $\{x_i\}_{1 \leq i \leq n}$ whose connections are summarized in the combinatorial Laplacian matrix L of their graph, that is for $i \neq j$, $L_{i,j} = -1$ if i and j are connected and 0 otherwise, and $L_{i,i} = -\sum_{j \neq i} L_{i,j}$. The authors look for a sequence of functions $\{f_i\}_{1 \leq i \leq d}$ of a RKHS $\mathcal{H}_k$ to map the original points in $\mathbb{R}^d$, and hope to recover the structure of the original graph through this representation. Namely, the projection is optimized such that the points, once projected in $\mathbb{R}^d$, will have graph interactions in that metric (that is by linking all nearest neighbours up to some distance threshold) that will be consistent with the original interactions. This leads to successive minimizations that may recall those performed in kernel-PCA, although different in nature through the addition of a regularization term proportional to λ:

$$f_j = \underset{f \in \mathcal{H}_k, f \perp \{f_1, \cdots, f_{j-1}\}}{\operatorname{argmax}} \frac{f_X^T L f_X + \lambda \|f\|_{\mathcal{H}_k}}{f_X^T f_X}.$$

where the vector f_X is defined as

$$f_X \overset{\text{def}}{=} (f(x_1), \cdots, f(x_n))^T.$$

The term $f_X^T L f_X$ above can be interpreted as a cost function with respect to the observable graph L, which penalizes functions f that are for which the values of $f(x_i)$ and $f(x_j)$ are very different for two connected nodes.

kernel discriminant analysis: we recall briefly the ideas behind the Fisher linear discriminant Fisher (1936) for classification. Given a sample $X = (x_1, \cdots, x_n)$ of points in $\mathbb{R}^d$ and assume that to each point x_i corresponds a binary variable $y_i \in \{0, 1\}$ which is equal to 0 if x_i belong to a first class and 1 when x_i belongs to a second class. Fisher discriminant analysis (LDA) assumes that the conditional distributions $p_0(X) = p(X|Y = 0)$ and $p_1(X) = p(X|Y = 1)$ are both normal densities. If the mean and variances μ_0, μ_1 and Σ_0 and Σ_1 respectively of p_0 and p_1 were known, the Bayes optimal rule would be to classify any observation x according to the value of its probability ratio $\frac{p_1(x)}{p_0(x)}$ and predict it is in class 0 whenever that ratio is below a certain threshold

$$(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0) \ + \ln|\Sigma_0| \ - \ (x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) - \ln|\Sigma_1| < \beta.$$

If the two classes are homoscedastic, that is $\Sigma_0 = \Sigma_1 = \Sigma$, then the decision can be simplified to testing whether whether $\omega^T x < c$ where ω is defined as $\omega = \Sigma^{-1}(\mu_1 - \mu_0)$. This later case is known in the literature as Linear Discriminant Analysis (LDA). When the latter assumption is not valid, Fisher proposed to find a vector ω that separates the two classes by optimizing the ratio

$$r(\omega) = \frac{\left(\omega^T\mu_0 - \omega^T\mu_1\right)^2}{\omega^T\Sigma_0\omega + \omega^T\Sigma_1\omega} = \frac{\left(\omega^T(\mu_0 - \mu_1)\right)^2}{\omega^T\Sigma_0\omega + \omega^T\Sigma_1\omega}$$

The ratio r is a Rayleigh quotient whose maximum is the only nonzero eigenvalue of the generalized eigenvalue problem $\left((\mu_0 - \mu_1)(\mu_0 - \mu_1)^T, \Sigma_0 + \Sigma_1\right)$ which corresponds to the eigenvector

$$\omega = (\Sigma_0 + \Sigma_1)^{-1}(\mu_0 - \mu_1).$$

In practice all quantities Σ_i and μ_i are replaced by empirical estimators. As shown in Mika et al. (1999), the criterion r can be conveniently cast as a quadratic problem in an arbitrary RKHS $\mathcal{H}$ corresponding to a set $\mathcal{X}$. In this new setting, a sample $X = (x_1, \cdots, x_n)$ of n points in $\mathcal{X}$ is paired with a set of labels $(y_1, \cdots, y_n)$. Instead of looking for a vector ω, namely a linear function, kernel discriminant analysis looks for a function $f \in \mathcal{H}_n$ such that

$$r(f) = \frac{(f(\mu_0) - f(\mu_1))^2}{\operatorname{var}_0 f + \operatorname{var}_1 f},$$

Let us write n_0 and n_1 for the numbers of elements of X of class 0 and 1 respectively, where $n_0 + n_1 = n$. For functions $f \in \mathcal{H}_n$, namely functions which can be written as $f(\cdot) = \sum_{i=1}^n a_i k(x_i, \cdot)$, we have that

$$r(f) = \frac{\left(a^T m_0 - a^T m_1\right)^2}{a^T S_0 a + a^T S_1 a}$$

where writing

$$\begin{aligned} K &= [k(x_i, x_j)]_{1 \le i,j \le n}, \\ K_0 &= [k(x_i, x_j)]_{1 \le i,j \le n, y_j = 0}, \\ K_1 &= [k(x_i, x_j)]_{1 \le i,j \le n, y_j = 1}, \end{aligned}$$

allows us to express means and variances in of X evaluated in functions of $\mathcal{H}_n$ as

$$\begin{aligned} m_0 &= K_0 \mathbf{1}_{n_0}, \\ m_1 &= K_1 \mathbf{1}_{n_1}, \\ S_0 &= K_0 (I - \frac{1}{n_0} \mathbf{1}_{n_0 m n_0}) K_0^T, \\ S_1 &= K_1 (I - \frac{1}{n_1} \mathbf{1}_{n_1, n_1}) K_1^T. \end{aligned}$$

Following the approach used above for linear functionals in Euclidian spaces, the vector a of weights could be recovered as the (only) nonzero eigenvalue of the n dimensional generalized eigenvalue problem $\big((m_0 + m_1)(m_0 + m_1)^T, (S_0 + S_1)\big)$. However, as the reader may easily check, the matrix $S_0 + S_1$ is not-invertible in the general case. Adding a regularization term λI_n is a hack that makes the problem computationally tractable. It can also be motivated from a regularization point of view. Indeed, since we are looking to maximize the ratio, this modification is equivalent to adding the RKHS norm of f_α to its denominator and hence favor functions with low norm. Although this explanation is not as motivated as the empirical regularization scheme discussed in Section 4.3, it is the one provided in the original work of Mika et al. (1999). Note in particular that the represent theorem does not apply in this setting and hence looking for function in $\mathcal{H}_n$ is itself an arbitrary choice. The kernel discriminant is thus the function f such that

$$f(\cdot) = \sum_{i=1}^{n} a_i k(x_i, \cdot), \quad a = (S_0 + S_1 + \lambda I_n)^{-1} (m_0 + m_1).$$

4.4 Density Estimation and Novelty Detection

A density estimator is an estimator based on a data sample of points drawn independently and identically distributed according to an unobservable underlying probability density function. The level sets of the estimator are the sets of points in $\mathcal{X}$ for which the density of the estimator has values below or above a given threshold. Estimating level sets rather than a density estimator taken within a set of candidate densities is the nonparametric direction taken by the one-class support vector machine presented below.

one-class SVM: taking advantage of the support vector machine formulation to minimize the penalized empirical risk of Equation (15), Schökopf et al. (1999) proposed the reformulation

$$R_c^\lambda(f) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} c\left(f(x_i)\right) + \nu \|f\|_{\mathcal{H}}.$$

where the labels of all points are set to 1 to estimate a function f that is positive on its support and that takes smaller values on areas of lower densities.

c can be any convex function differentiable at 0 and such that $c'(0) < 0$. In particular, Schökopf et al. (1999) solves the following mathematical program

$$\begin{array}{ll} \text{minimize} & \nu\|f\|_{\mathcal{H}} + \sum_{i=1}^{n}(\xi_i - \rho) \\ \text{subject to} & f \in \mathcal{H}_n \\ & f(x_i) \leq \rho - \xi_i, \xi_i \geq 0 \end{array}$$

novelty detection and kernel-PCA: Novelty detection refers to the task of detecting patterns in a given data set that do not conform to an established normal behavior (Chandola et al., 2009). novelty detection can be implemented in practice by using the level sets of a density estimator. A new observation is intuitively labelled as abnormal if it lies within a region of low density of the estimator granted this new observation has been drawn from the same distribution. Another approach to novelty detection is given by a spectral analysis implemented through the study of the principal components of a data sample. This approach can be naturally generalized to a "kernelized algorithm".

Principal component analysis can be used as a novelty detection tool for multivariate data (Jolliffe, 2002, §10.1) assuming the underlying data can be reasonably approximated by a Gaussian distribution. Given a sample $X = (x_1, \cdots, x_n)$ of n points drawn i.d.d from the distribution of interest in $\mathbb{R}^d$, the p first eigenvectors of PCA are defined as the p first orthonormal eigenvectors $e_1, \cdots, e_p$ with corresponding eigenvalues $\lambda_1, \cdots, \lambda_p$ of the sample variance matrix $\Sigma_n = \frac{1}{n-1}\sum(x_i - m)(x_i - m)^T$ where $m = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the sample mean. An observation y is labelled as abnormal whenever its projection in the space spanned by the p eigenvectors is markedly outside the ellipsoid defined by the semi-axes (e_i, λ_i), namely when

$$\sum_{i=1}^{p} \frac{e_i^T(y-m)}{\lambda_i^2} \geq \beta,$$

or alternatively when the contribution of the first p eigenvectors to the total norm of y is low compared to the weight taken by the other directions of lower variance,

$$\frac{\|y\|^2 - \sum_{i=1}^{p} \frac{e_i^T(y-m)}{\lambda_i^2}}{\|y\|^2} \geq \alpha.$$

This idea has been extended in the case of kernel-PCA in (Hoffmann, 2007) by using a kernel k on $\mathcal{X}$. In that case the linear functionals $e_i^T \cdot$ are replaced by the evaluations of eigenfunctions $f_i(\cdot)$ introduced in Equation (14) and the norm of y itself is taken in the corresponding RKHS $\mathcal{H}$ and can be recovered as $k(y, y)$.

5 Kernel Selection and Kernel Mixture

An important issue that arises when using kernel machines in practice is to select an adequate kernel. In most practical cases choices are abundant if not infinite. We review three different families of techniques designed to cope with this situation. The following section is set in the usual classification setting where the dataset of interest is composed of n pairs of points and labels, i.e. $\{(x_i, y_i)\}_{i=1..n}$ where each $x_i \in \mathcal{X}$ and $y_i \in \{-1, 1\}$.

5.1 Parameter Selection

When the kernel can be parameterized by a few variables, a brute force approach that examines the cross-validation error over a grid of acceptable parameters is a reasonable method which often yields satisfactory results. This approach is non-tractable when the number of parameters reaches but a few values. (Chapelle et al., 2002; Bousquet and Herrmann, 2003; Frölich et al., 2004) and more recently (Keerthi et al., 2007) have proposed different schemes to tune the parameters of a Gaussian kernel on $\mathbb{R}^d$. The authors usually assume a setting where the weights σ_i assigned to each feature of two vectors x and y in $\mathbb{R}^d$ need to be tuned, that is consider kernels of the form

$$k(x, y) = \exp\left(-\sum_{i=1}^{d} \frac{(x_i - y_i)^2}{\sigma_i^2}\right).$$

Finding an adequate parameter choice $(\sigma_1, \cdots, \sigma_d)$ implies defining first a criterion to discriminate good from bad choices for such parameters. Chapelle et al. (2002) consider the leave-one-out error of the regularized empirical risk formulation of Equation (15). Given a kernel k_σ parameterized by a set of parameters $(\sigma_1, \cdots, \sigma_d)$, the leave-one-out error is the sum

$$\mathcal{E}_{\text{LOO}}(\sigma) = \frac{1}{n} \sum \mathbf{1}(f_{-i}(x_i) \neq y_i),$$

where we use a generic regularized empirical risk minimizer estimated on all points of the sample but one:

$$f_{-i} = \text{argmin}_{f \in \mathcal{H}} \frac{1}{n} \sum_{j=1, j \neq i}^{n} c\left(f(x_j), y_j\right) + \lambda J(f).$$

Evgeniou et al. (2004) show that the leave-one-out error is a good way to quantify the performance of a class of classifiers. If $\mathcal{E}_{\text{LOO}}$ was a tractable and analytical function of σ, it would thus seem reasonable to select σ as a minimizer of $\mathcal{E}_{\text{LOO}}$. This is not the case unfortunately. The authors of both (Chapelle et al., 2002; Bousquet and Herrmann, 2003) propose to consider instead upperbounds on $\mathcal{E}_{\text{LOO}}$ which are tractable and design algorithms to minimize such upperbounds through gradient descent methods. Keerthi et al. (2007) generalize this approach by considering other proxies of the performance of a kernel on a given problem.

5.2 Multiple Kernel Learning

Rather than looking for a single kernel in a large set of candidates, a research trend initiated by Lanckriet et al. (2004) proposes to consider instead combinations of candidate kernels. As recalled in Section 2.4 positive definite kernels can be combined multiplicatively (under point-wise multiplication) and linearly (through positive linear combinations). Since the pioneering work of Lanckriet et al. (2004), which relied on expensive semi-definite programming to compute optimal linear combinations of kernels, the shift of study has progressively evolved towards computationally efficient alternatives to define useful additive mixtures as in (Bach et al., 2004; Sonnenburg et al., 2006; Rakotomamonjy et al., 2007). A theoretical foundation for this line of research can be found in Micchelli and Pontil (2006). We follow the exposition used in (Rakotomamonjy et al., 2007). Recall, as exposed in Section 4.3, that given a kernel k, kernel classifiers or regressors yield decision functions of the form

$$f(x) = \sum_{i=1}^{n} \alpha_i^\star y_i k(x, xi) + b^\star, \tag{18}$$

where both the family $(\alpha_i^\star)$ and $b^\star$ stand for optimized parameters. When not one, but a family of m kernels $k_1, \ldots, k_m$ kernels can be combined in a convex manner to yield a composite kernel $k = \sum_{l=1}^{m} d_l k_l$ with $\sum d_l = 1$, the task consisting in learning both the coefficients α_i, b and the weights d_l in a single optimization problem is known as the multiple kernel learning (MKL) problem (Bach et al., 2004). Writing $\mathcal{H}$ for the rkhs corresponding to kernel k, The penalized SVM-type optimization framework for the estimation of a function f in $\mathcal{H}$ is

$$\begin{array}{ll} \text{minimize} & \|f\|_{\mathcal{H}}^2 + C \sum \xi_i \\ \text{subject to} & f \in \mathcal{H}, \;\; b \in \mathbb{R}, \\ & \forall i, y_i(f(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{array}$$

When the kernel k is a mixture of m kernels, the authors propose the following optimization scheme,

$$\begin{array}{ll} \text{minimize} & \sum_l \frac{1}{d_l} \|f_l\|_{\mathcal{H}_l}^2 + C \sum \xi_i \\ \text{subject to} & f_l \in \mathcal{H}_l, \;\; b \in \mathbb{R}, d_l, \xi_i, \\ & \forall i \leq n, \; y_i(\sum_l f_l(x_i) + b) \geq 1 - \xi_i, \\ & \sum_l d_l = 1; \;\; \xi_i, d_l \geq 0. \end{array}$$

In the formulation above, each value d_l controls the importance given to squared norm of f_l in the objective function. A bigger d_l favors functions whose component in $\mathcal{H}_l$ may have a larger norm. If the weight d_l goes to zero, the corresponding function f_l can only be zero as shown by the authors, which is indeed equivalent to not taking into account kernel k_l in the sum $\sum_k d_l k_l$. The solution to the problem above can actually be decomposed into a two-step procedure,

namely by minimizing an objective function $J(d)$ defined on the weights $d = (d_l)$ and which is itself computed through a SVM optimization, namely:

$$\begin{array}{lll} \text{minimize} & J(d), & \\ \text{subject to} & \sum_l d_l = 1;\ d_l \geq 0, & \\ & J(d) = \begin{array}{ll} \text{minimize} & \sum_l \frac{1}{d_l} \|f_l\|^2_{\mathcal{H}_l} + C \sum \xi_i \\ \text{subject to} & f_l \in \mathcal{H}_l, b \in \mathbb{R}, \xi_i \\ & \forall i, y_i(\sum_l f_l(x_i) + b) \geq 1 - \xi_i; \xi_i \geq 0. \end{array} & \end{array} \tag{19}$$

The authors iterate between the computation of the objective function J, itself a SVM optimization, and the optimization of J, carried out using projected Gradient methods. Each iteration of this loop involves the computation of the Gradient's directions $\frac{\partial J}{\partial d_l}$ which the authors show are simple functions of the weights $\alpha_i^\star$ retrieved during the SVM-computation conducted to compute J, namely

$$\frac{\partial J}{\partial d_l} = -\frac{1}{2} \sum_{i,j=1}^{n} \alpha_i^\star \alpha_j^\star y_i y_j k_l(x_i, x_j)$$

The algorithm boils down to the following loop.

- initialize all weights d_l to $1/m$,
- Loop :
  - compute an SVM-solution to the problem with fixed weights d. This gives J, as well as its associated Gradient directions $\frac{\partial J}{\partial d_l}$.
  - Optimize J with respect to d, that is replace the current weights family d by $d + \gamma D$ where D is the vector of descent direction computed from the Gradient (reducing it and projecting it) such that the new d satisfies the simplex constraints, and γ is an optimal step size determined by line search.
  - Check for optimality conditions initially set, and if reached get out of the loop.

By the end of the convergence, both the weights α and b that arise from the last computation of J, that is the SVM-computation step, and the weights d obtained in the end provide the parameters needed to define f in Equation (18). An additional property of the algorithm is that it tends to produce sparse patterns for d, which can be be helpful to interpret which kernels are the most useful for the given task.

5.3 Families of Kernels Labeled on a Graph

The approach taken in the latter section assumes that all kernels are independently selected. The optimization of Equation (19) is carried out on the linear subspace formed by all linear combinations of these m kernels. Rather than

treating all kernels uniformly and mixing them linearly, (Cuturi and Fukumizu, 2007) consider a setting with two particular features.

First, an a-priori knowledge on the structure on the kernels themselves can be used, namely a hierarchical structure under the form of a tree. Kernels are indexed by labels on a directed acyclic graph (DAG) $\{\alpha \in \mathcal{T}\}$, and each kernel k_α is related to its siblings, that is $k_\beta, \beta \in s(\alpha)$ where $s(\alpha)$ stands for the sons of a node α. An example of such an approach can be seen in Figure 4 where the hierarchy is a dyadic partition of the surface of an image.

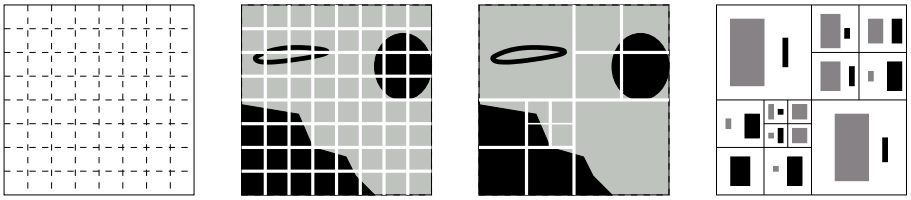

Figure 4: The leftmost image represents the final nodes of the hierarchy considered, that is the 4^3 squares present in the grid. The ancestors of such nodes correspond iteratively to the 16 larger square obtained when grouping 4 small windows, then to the image divided into 4 equal parts and finally the whole image. The hierarchy has thus a depth of 3. Any set of nodes taken in the hierarchy can in turn be used to compare two images under the light of those local color histograms displayed in the right most image, which reduces in the case of two-color images to binary histograms as illustrated in the right-most image.

Second, the hierarchy can be used not only to combine kernels additively but also multiplicatively. More precisely, the authors define the space of candidate kernels as the space $\mathcal{S}$ of all complete subtrees t of $\mathcal{T}$ starting with the same root. Such a tree t is uniquely characterized by its set of final nodes $f(t)$, and the kernel associated to such a subtree is the product of the kernels associated to each final node, that is

$$k_t = \prod_{\alpha \in f(t)} k_\alpha.$$

Note that the number of potential subtrees grows super-exponentially, hence yielding a number of candidate kernels far superior to the total number of node kernels.

Grounded on these two assumptions, Cuturi and Fukumizu (2007) use a prior weight on subtree kernels to propose a fast computation of a kernel k as

$$k = \sum_{t \in \mathcal{S}} d_t k_t.$$

The weight d_t penalizes the complexity of a given subtree t by considering its number of nodes. In practice the weights d_t are defined with an analogy to branching process priors for trees (Jagers, 1975). Bach (2008a) proposed a similar setting to optimize directly the weights d_t using a variation of the

Multiple Kernel Learning framework. The originality of the approach is to take advantage of the hierarchy between kernels to adaptively explore subtrees which may fit better the task at stake.

6 Kernel Cookbook

We review in this section practical guidelines that can apply to the selection of a kernel given a dataset of structured objects.

6.1 Advanced Kernels on Vectors

Vectors of $\mathbb{R}^d$ can be directly approached by considering the linear dot-product as a kernel, which amounts to performing an alternative penalized regression and optimizing it in the dual space as is described in Chapelle (2007).

Beyond the use of the linear kernel, the array of positive definite kernels defined on vectors if not on scalars is very large, and include functions of all possible shapes as illustrated in (Berg et al., 1984, Exercise 2.12, p.79). Although some specific kernels have been used for their precise invariance properties Fleuret and Sahbi (2003), most practitioners limit themselves to the use of Gaussian and polynomial kernels. Once a family of kernel has been selected, the topic of choosing adequate parameters for this kernel is itself one of the biggest challenges when using kernel methods on vectorial data, as hinted in (Hastie et al., 2009, Section 12.3.4). For polynomial kernels searches are usually limited to the offset and exponent parameters. In the case of Gaussian kernels, usually favored by practitioners, the more general use of Mahalanobis distances instead of the simple Euclidian distance, that is kernels of the form

$$k_\Sigma(x, y) = e^{-\frac{1}{2}(x-y)^T \Sigma (x-y)}$$

where Σ is a $d \times d$ symmetric positive definite matrix, has also been investigated to fit better data at hand and to insist on the possible correlations or importance of the described features. The simplest type of matrices Σ which can be used is one with a diagonal structure, and pre-whitening the data might be considered as such an approach. More advanced tuning strategies have been covered in Section 5.1.

6.2 Kernels on Graphs

Labeled graphs are widely used in computer science to model data linked by discrete dependencies, such as social networks, molecular pathways or patches in images. Designing kernels for graphs is usually done with this wide applicability in mind.

A graph G is described by a finite set of vertices $\mathcal{V}$ and a hence finite set of edges $E = \mathcal{V} \times \mathcal{V}$. Graphs are sometimes labelled. In that case there exists a function of E to the set of labels $\mathcal{L}$, or alternatively $\mathcal{V}$ to $\mathcal{L}$ that assigns a label to a node or an edge.

Given an arbitrary subgraph f and a graph of interest G, the feature $f(G)$ measuring how many subgraphs of G have the same structure as graph f is a useful elementary feature. The original paper by Kashima et al. (2003) presented in Section 3.2 uses for the set of subgraphs f simple random walks and counts

their co-occurrences to provide a kernel, an approach that had also been studied in the case of trees (Vert, 2002). The work has found extensions in (Mahe et al., 2005) to take better into account similarity between not only the graph structure but also the labels that populate it. More advanced sets of features, which rely on algebraic descriptions of graphs have been recently considered in (Kondor et al., 2009). We refer the reader to the exhaustive review of Vishwanathan et al. (2008).

6.3 Kernels on Images

Technically speaking, an image can be seen as a long 3-dimensional vector of RGB intensities. It is however unlikely that treating images as vectors and applying Gaussian kernels on them will yield any interesting result. In that sense, the definition of kernels for images are build on higher-level properties of images and images contents, such as the invariance to slight translations in both color intensities and patterns positions in the image. These properties can be translated into the following kernels.

color histograms: numerous approaches have stemmed from the use of color histograms to build kernel on images, starting with the seminal experiments carried out by Chapelle et al. (1999). By representing an image I by an arbitrary color histogram $\theta_I \in \Sigma_d$, where d stands for the color depth (typically 256, 4096 or 16 million), the authors follow by designing a kernel on two images using kernels on multinomials such as those presented in Section 3.1.2, typically

$$k(I_1, I_2) = e^{-\lambda \|\theta_{I_1} - \theta_{I_2}\|}.$$

Note that this approach assumes a *total invariance* under pixel translation, which is usually a drastic loss of information on the structure and the content of the image itself, as illustrated in Figure 5.

Further developments have tried to cope with this this limitation. Rather than considering a single histogram for each image Grauman and Darrell (2005) and Cuturi and Fukumizu (2007) divide the image into local patches and compare the resulting families of local histograms. These approaches provide substantial improvements at a low computational cost.

taking shapes taken into account: note first that by counting elementary shapes in each image, rather than colours, the techniques described in the paragraph above can be applied to define kernels that focus on shape similarities. However, when larger shapes are the main criterion to discriminate two images, histogram representations have obvious limitations. Haasdonk and Keysers (2002) propose a kernel which exponentiates the opposite of the tangent distance between two images to quantify their similarity. Since the computation of the tangent distance requires the optimization of a criterion, the framework is related to other attempts at designing a kernel from of distance, e.g (Watkins, 2000; Shimodaira et al., 2002a; Vert et al., 2004; Cuturi, 2007; Cuturi et al.).

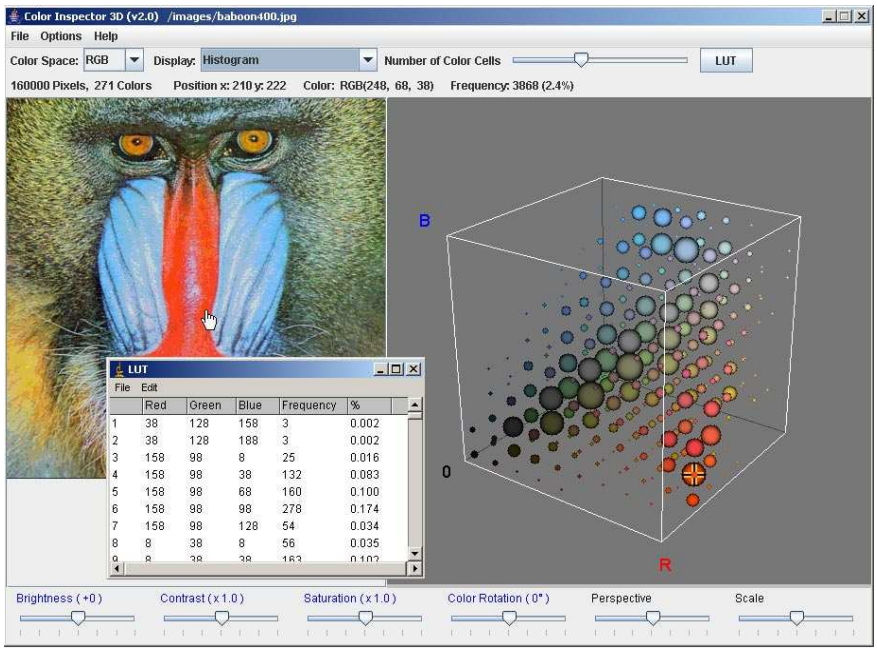

Figure 5: A complex image such as the monkey above can be summarized through color histograms, represented, as above, as a 3D histogram of red/green/blue intensities. Although this representation inquires considerable loss of information, it is often used for image retrieval.

The tangent distance (Simard et al., 1998) is a distance computed between two shapes x and y to assess how different they are to each other by finding an optimal series of elementary transformations (rotations, translations) that produces y when starting from x. Since the distance is not negative-definite the tangent distance does not yield directly a positive definite kernel, but might be used in practice with most kernel machines after an adequate correction.

shapes seen as graphs: taking such limitations into account but still willing to incorporate a discrimination based on shapes, Harchaoui and Bach (2007) have exploited existing graph kernels to adapt them to images. Images can indeed be seen as large graphs of interconnected color dots. Harchaoui and Bach (2007) propose to segment first the images through standard techniques (in the quoted paper the authors use the watershed transform technique) into large areas of homogeneous colors, and then treat the resulting interconnections between colored areas as smaller graphs labeled with those simplified colors. The two graphs are subsequently compared using standard graph kernels, notably a variation proposed by the authors. When the number of active points in the images is low, Bach (2008c) focus on a specific category of graph kernels tailored for point clouds taking values in 2D or 3D spaces.

6.4 Kernels on Variable-Length Sequential Data

Variable-length sequence data-types are ubiquitous in most machine learning applications. They include the observation sampled from a discrete-time processes, texts as well as long strings such as protein and DNA codes. One of the challenges of designing kernels on such objects is that such kernels should be able to compare sequences of different lengths, as would be the case when comparing two speech segments with different sampling frequencies or overall recorded time, two texts, or two protein with different total number of amino acids.

kernels on texts: most kernels used in practice on texts stem from the use of the popular bag-of-words (BoW) representations, that is sparse word count vectors taken against very large dictionaries. The monograph (Joachims, 2002) shows how the variations of the BoW can be used in conjunction with simple kernels such as the ones presented in Section 3.1.2. From a methodological point of view, much of the approach relies rather on choosing efficient BoW representations and on the contrary usually boil down to the use of simple kernels.

histograms of transitions: when tokens are discreet and few, the easiest approach is arguably to map them as histograms of shorter substrings, also known as n-grams, and compare those histograms directly. This approach was initially proposed by (Leslie et al., 2002) with subsequent refinements to either incorporate more knowledge about the tokens transitions (Cuturi and Vert, 2005; Leslie

et al., 2003) or improve computational speed (Teo and Vishwanathan, 2006).

higher level transition modeling with HMM's: rather than using simple n-gram counts descriptors, Jaakkola et al. (2000) use more elaborate statistical models to define kernels between strings which can be modelled as HMM. The interested reader may refer to Section 3.2 for a review of the Fisher kernel to see how the HMM model is used to build a feature vector to compare strings directly.

edit distances: a different class of kernels can be build using transformations on the sequences themselves, in a form that echoes with the Tangent distance kernel presented in an earlier section. Intuitively, if by successive and minor changes one can map a sequence x to another sequence y, then the overall cost (which remains to be defined) needed to go from x to y can be seen as a good indicator of how related they are to each other. As with the tangent distance reviewed above, Shimodaira et al. (2002b) take into account the optimal route from x to y, whose total cost is known depending on the application field as the edit distance, the Smith-Waterman score, Dynamic-Time-Warping or Levenshtein distance, to define a kernel which is not necessarily positive-definite but which performs reasonably well on a speech discrimation task. Vert et al. (2004) argue that by taking a weighted average of the costs associated to all possible transformations mapping x to y, one can obtain a kernel who is positive definite and which usually performs better on the set of proteins they consider in their study. Up to a few subtleties, a similar approach is presented in (Cuturi et al.) which shows good performance on speech data.

References

Akaho, S. (2001). A kernel method for canonical correlation analysis. In *Proceedings of International Meeting on Psychometric Society (IMPS2001)*.

Amari, S.-I. and Nagaoka, H. (2001). *Methods of Information Geometry*. AMS vol. 191.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337 – 404.

Bach, F. (2008a). Exploring large feature spaces with hierarchical multiple kernel learning. In *Advances in Neural Information Processing Systems 21*. to appear.

Bach, F. and Jordan, M. (2002). Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48.

Bach, F. R. (2008b). Consistency of the group lasso and multiple kernel learning. *J. Mach. Learn. Res.*, 9:1179–1225.

Bach, F. R. (2008c). Graph kernels between point clouds. In Cohen, W. W., McCallum, A., and Roweis, S. T., editors, *ICML*, volume 307 of *ACM International Conference Proceeding Series*, pages 25–32. ACM.

Bach, F. R., Lanckriet, G. R. G., and Jordan, M. I. (2004). Multiple kernel learning, conic duality, and the smo algorithm. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 6, New York, NY, USA. ACM.

Berg, C., Christensen, J. P. R., and Ressel, P. (1984). *Harmonic Analysis on Semigroups*. Number 100 in Graduate Texts in Mathematics. Springer Verlag.

Berlinet, A. and Thomas-Agnan, C. (2003). *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers.

Bernstein, D. S. (2005). *Matrix Mathematics: Theory, Facts, and Formulas with Application to Linear Systems Theory*. Princeton University Press.

Bochner, S. (1933). A theorem on Fourier-Stieltjes integrals. *Mathematische Annalen*, 108:378–410.

Bousquet, O. and Herrmann, D. (2003). On the complexity of learning the kernel matrix. *Advances in neural information processing systems*, pages 415–422.

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

Catoni, O. (2004). *Statistical learning theory and stochastic optimization*. Number 1851 in LNM. Springer Verlag.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*.

Chapelle, O. (2007). Training a support vector machine in the primal. *Neural Comput.*, 19(5):1155–1178.

Chapelle, O., Haffner, P., and Vapnik, V. (1999). SVMs for histogram based image classification. *IEEE Transactions on Neural Networks*, 10(5):1055.

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

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273.

Csató, L. and Opper, M. (2002). Sparse on-line gaussian processes. *Neural Computation*, 14(3):641–668.

Cucker, F. and Smale, S. (2002). On the mathematical foundations of learning. *BAMS: Bulletin of the American Mathematical Society*, 39.

Cuturi, M. (2007). Permanents, transportation polytopes and positive-definite kernels on histograms. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI) 2007*, pages 732 – 737.

Cuturi, M. and Fukumizu, K. (2007). Kernels on structured objects through nested histograms. In Schölkopf, B., Platt, J., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA.

Cuturi, M., Fukumizu, K., and Vert, J.-P. (2005). Semigroup kernels on measures. *Journal of Machine Learning Research*, 6:1169–1198.

Cuturi, M. and Vert, J.-P. (2005). The context-tree kernel for strings. *Neural Networks*, 18(8).

Cuturi, M., Vert, J.-P., Birkenes, Øystein., and Matsui, T. A kernel for time series based on global alignments. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume II.

Devinatz, A. (1955). The representation of functions as Laplace-Stieltjes integrals. *Duke Mathematical Journal*, 24:481–498.

Drucker, H., Burges, C., Kaufman, L., Smola, A., and Vapnik, V. (1997). Support vector regression machines. *Advances in neural information processing systems*, pages 155–161.

Ehm, W., Genton, M., and Gneiting, T. (2003). Stationary covariances associated with exponentially convex functions. *Bernoulli*, pages 607–615.

Evgeniou, T., Pontil, M., and Elisseeff, A. (2004). Leave one out error, stability, and generalization of voting combinations of classifiers. *Machine Learning*, 55(1):71–97.

Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188.

Fleuret, F. and Sahbi, H. (2003). Scale-invariance of support vector machines based on the triangular kernel. In *In 3rd International Workshop on Statistical and Computational Theories of Vision*.

Friedman, J. and Tukey, J. (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. Comput.*, 23:881–890.

Frölich, H., Chapelle, O., and Schökopf, B. (2004). Feature selection for support vector machines using genetic algorithms. *International Journal on Artificial Intelligence Tools*, 13(4):791–800.

Fukumizu, K., Bach, F., and Gretton, A. (2007). Statistical consistency of kernel canonical correlation analysis. *The Journal of Machine Learning Research*, 8:361–383.

Fukumizu, K., Bach, F., and Jordan, M. (2004). Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 5:73–99.

Girosi, F., Jones, M., and Poggio, T. (1995). Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269.

Grauman, K. and Darrell, T. (2005). The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, pages 1458–1465. IEEE Computer Society.

Haasdonk, B. (2005). Feature space interpretation of SVMs with indefinite kernels. *IEEE Trans. Pattern Anal. Mach. Intell*, 27(4):482–492.

Haasdonk, B. and Keysers, D. (2002). Tangent distance kernels for support vector machines. In *Proceedings of the International Conference on Pattern Recognition (2)*, pages 864–868.

Harchaoui, Z. and Bach, F. (2007). Image classification with segmentation graph kernels. In *CVPR*.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd edition)*. Springer Verlag.

Haussler, D. (1999). Convolution kernels on discrete structures. Technical report, UC Santa Cruz. USCS-CRL-99-10.

Hein, M. and Bousquet, O. (2005). Hilbertian metrics and positive definite kernels on probability measures. In Ghahramani, Z. and Cowell, R., editors, *Proceedings of AISTATS 2005*.

Hoerl, A. (1962). Application of ridge analysis to regression problems. *Chemical Engineering Progress*, 58(3):54–59.

Hoffmann, H. (2007). Kernel PCA for novelty detection. *Pattern Recognition*, 40(3):863–874.

Hofmann, T., Scholkopf, B., and Smola, A. (2008). Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171.

Huber, P. (1985). Projection pursuit. *The annals of Statistics*, 13(2):435–475.

Jaakkola, T., Diekhans, M., and Haussler, D. (2000). A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1,2):95–114.

Jaakkola, T., Diekhaus, M., and Haussler, D. (1999). Using the fisher kernel method to detect remote protein homologies. *7th Intell. Sys. Mol. Biol.*, pages 149–158.

Jaakkola, T. S. and Haussler, D. (1999). Exploiting Generative Models in Discriminative Classifiers. In Kearns, M. S., Solla, S. A., and Cohn, D. A., editors, *Advances in Neural Information Processing Systems 11*. MIT Press.

Jagers, P. (1975). *Branching processes with biological applications*. John Wiley & Sons.

Jebara, T., Kondor, R., and Howard, A. (2004). Probability product kernels. *Journal of Machine Learning Research*, 5:819–844.

Joachims, T. (2002). *Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms*. Kluwer Academic Publishers.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer, New York, NY, USA.

Kashima, H., Tsuda, K., and Inokuchi, A. (2003). Marginalized kernels between labeled graphs. In Faucett, T. and Mishra, N., editors, *Proceedings of the Twentieth International Conference on Machine Learning*, pages 321–328. AAAI Press.

Keerthi, S., Sindhwani, V., and Chapelle, O. (2007). An efficient method for gradient-based adaptation of hyperparameters in SVM models. *Advances in Neural Information Processing Systems*, 19:673.

Kimeldorf, G. S. and Wahba, G. (1971). Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82–95.

Kondor, R. and Lafferty, J. (2002). Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 315–322.

Kondor, R. I., Shervashidze, N., and Borgwardt, K. M. (2009). The graphlet spectrum. In *ICML*, page 67.

Lafferty, J. and Lebanon, G. (2005). Diffusion kernels on statistical manifolds. *JMLR*, 6:129–163.

Lanckriet, G. R. G., Bie, T. D., Cristianini, N., Jordan, M. I., and Noble, W. S. (2004). A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635.

Leslie, C., Eskin, E., and Noble, W. S. (2002). The spectrum kernel: a string kernel for svm protein classific ation. In *Proc. of PSB 2002*, pages 564–575.

Leslie, C., Eskin, E., Weston, J., and Noble, W. S. (2003). Mismatch string kernels for svm protein classification. In Becker, S., Thrun, S., and Obermayer, K., editors, *NIPS 15*. MIT Press.

Leurgans, S. E., Moyeed, R. A., and Silverman, B. W. (1993). Canonical correlation analysis when the data are curves.

Luss, R. and D'Aspremont, A. (2008). Support vector machine classification with indefinite kernels. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems 20*, pages 953–960. MIT Press, Cambridge, MA.

Mahe, P., Ueda, N., Akutsu, T., Perret, J.-L., and Vert, J.-P. (2005). Graph kernels for molecular structure-activity relationship analysis with support vector machines. *Journal of Chemical Information and Modeling*, 45(4):939–951.

Mahé, P., Ueda, N., Akutsu, T., Perret, J.-L., and Vert, J.-P. (2004). Extensions of marginalized graph kernels. In Greiner, R. and Schuurmans, D., editors, *Proceedings of the Twenty-First International Conference on Machine Learning (ICML 2004)*, pages 552–559. ACM Press.

Mercer, T. (1909). Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London, Series A*, 209:415–446.

Micchelli, C. and Pontil, M. (2006). Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6(2):1099.

Mika, S., Ratsch, G., Weston, J., Schökopf, B., and Muller, K. (1999). Fisher discriminant analysis with kernels. *Neural networks for signal processing IX*, pages 41–48.

Muller, K., Mika, S., Ratsch, G., Tsuda, K., and Scholkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks*, 12(2):181–201.

Naor, A. and Schechtman, G. (2007). Planar earthmover is not in l_1. *SIAM J. Comput.*, 37(3):804–826.

Parzen, E. (1962). Extraction and detection problems and reproducing kernel Hilbert spaces. *Journal of the Society for Industrial and Applied Mathematics. Series A, On control*, 1:35–62.

Rakotomamonjy, A., Bach, F., Canu, S., and Grandvalet, Y. (2007). More efficiency in multiple kernel learning. In *International Conference on Machine Learning (ICML)*. IDIAP-RR 07-18.

Ramsay, J. O. and Dalzell, C. J. (1991). Some tools for functional data analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3):539–572.

Schoenberg, I. (1942). Positive definite functions on spheres. *Duke Math. J.*, 9:96–108.

Schökopf, B., Platt, J. C., Shawe-taylor, J., Smola, A. J., and Williamson, R. C. (1999). Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:2001.

Schölkopf, B., Smola, A., and Müller, K. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization , Optimization, and Beyond*. MIT Press.

Schölkopf, B., Tsuda, K., and Vert, J.-P. (2004). *Kernel Methods in Computational Biology*. MIT Press.

Schölkopf, B., Weston, J., Eskin, E., Leslie, C., and Noble, W. S. (2002). A kernel approach for learning from almost orthogonal patterns. In Elomaa, T., Mannila, H., and Toivonen, H., editors, *Proceedings of ECML 2002, 13th European Conference on Machine Learning, Helsinki, Finland, August 19-23, 2002*, volume 2430 of *Lecture Notes in Computer Science*, pages 511–528. Springer.

Seeger, M. (2002). Covariance kernels from bayesian generative models. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, pages 905–912. MIT Press.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge Univ Pr.

Shimodaira, H., Noma, K.-I., Nakai, M., and Sagayama, S. (2002a). Dynamic time-alignment kernel in support vector machine. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA. MIT Press.

Shimodaira, H., Noma, K.-I., Nakai, M., and Sagayama, S. (2002b). Dynamic time-alignment kernel in support vector machine. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA. MIT Press.

Simard, P. Y., Cun, Y. A. L., Denker, J. S., and Victorri, B. (1998). Transformation invariance in pattern recognition – tangent distance and tangent propagation. In *Lecture Notes in Computer Science*, pages 239–274. Springer.

Sindhwani, V., Niyogi, P., and Belkin, M. (2005). Beyond the point cloud: from transductive to semi-supervised learning. In *Proceedings of ICML '05: Twenty-second international conference on Machine learning*. ACM Press.

Smith, N. and Gales, M. (2002). Speech recognition using svms. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA. MIT Press.

Sonnenburg, S., Rätsch, G., Schäfer, C., and Schölkopf, B. (2006). Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565.

Steinwart, I. and Christmann, A. (2008). *Support Vector Machines*. Springer Verlag.

Teo, C. H. and Vishwanathan, S. V. N. (2006). Fast and space efficient string kernels using suffix arrays. In *In Proceedings, 23rd ICMP*, pages 929–936. ACM Press.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Tsuda, K., Akaho, S., Kawanabe, M., and Müller, K.-R. (2004). Asymptotic properties of the fisher kernel. *Neural Computation*, 16(1):115–137.

Tsuda, K., Kawanabe, M., Rätsch, G., Sonnenburg, S., and Müller, K.-R. (2002a). A new discriminative kernel from probabilistic models. *Neural Computation*, 14(10):2397–2414.

Tsuda, K., Kin, T., and Asai, K. (2002b). Marginalized kernels for biological sequences. *Bioinformatics*, 18(Suppl 1):268–275.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley.

Vert, J.-P. (2002). A tree kernel to analyze phylogenetic profiles. *Bioinformatics*, 18:S276–S284.

Vert, J.-P. (2006). Classification of biological sequences with kernel methods. *Lecture Notes in Computer Science*, 4201:7.

Vert, J.-P. and Kanehisa, M. (2003). Graph-driven features extraction from microarray data using diffusion kernels and kernel cca. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15*. MIT Press.

Vert, J.-P., Saigo, H., and Akutsu, T. (2004). Local alignment kernels for protein sequences. In Schölkopf, B., Tsuda, K., and Vert, J.-P., editors, *Kernel Methods in Computational Biology*. MIT Press.

Vert, J.-P. and Yamanishi, Y. (2005). Supervised graph inference. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*. MIT Press.

Vert, R. and Vert, J.-P. (2005). Consistency and convergence rates of one-class svm and related algorithms. Technical Report 1414, LRI, Université Paris Sud.

Vishwanathan, S., Borgwardt, K., Kondor, I., and Schraudolph, N. (2008). Graph kernels. *Journal of Machine Learning Research*, 9:1–37.

Wahba, G. (1990). *Splines Models for Observational Data*. Series in Applied Mathematics, Vol. 59, SIAM.

Watkins, C. (2000). Dynamic alignment kernels. In Smola, A., Bartlett, P., Schölkopf, B., and rmans, D. S., editors, *Advances in Large Margin Classifiers*, pages 39–50. MIT Press.

Willems, F. M. J., Shtarkov, Y. M., and Tjalkens, T. J. (1995). The context-tree weighting method: basic properties. *IEEE Transactions on Information Theory*, pages 653–664.

Zhang, D., Chen, X., and Lee, W. S. (2005). Text classification with kernels on the multinomial manifold. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 266–273. ACM Press.

Zhu, J. and Hastie, T. (2002). Kernel logistic regression and the import vector machine. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, pages 1081–1088, Cambridge, MA. MIT Press.