2 Kernels

In Chapter 1, we described how a kernel arises as a similarity measure that can be thought of as a dot product in a so-called feature space. We tried to provide an intuitive understanding of kernels by introducing them as similarity measures, rather than immediately delving into the functional analytic theory of the classes of kernels that actually admit a dot product representation in a feature space.

In the present chapter, we will be both more formal and more precise. We will study the class of kernels k that correspond to dot products in feature spaces $\mathcal{H}$ via a map Φ,

$$\begin{aligned}\Phi : \mathcal{X} &\to \mathcal{H} \\ x &\mapsto \mathbf{x} := \Phi(x),\end{aligned} \tag{2.1}$$

that is,

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle . \tag{2.2}$$

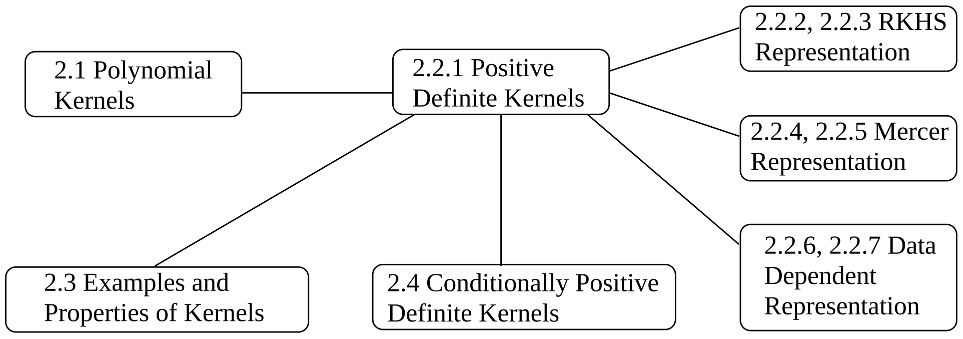
Regarding the input domain $\mathcal{X}$, we need not make assumptions other than it being a set. For instance, we could consider a set of discrete objects, such as strings.

A natural question to ask at this point is what kind of functions $k(x, x')$ admit a representation of the form (2.2); that is, whether we can always construct a dot product space $\mathcal{H}$ and a map Φ mapping into it such that (2.2) holds true. We shall begin, however, by trying to give some motivation as to why kernels are at all useful, considering kernels that compute dot products in spaces of monomial features (Section 2.1). Following this, we move on to the questions of how, given a kernel, an associated feature space can be constructed (Section 2.2). This leads to the notion of a Reproducing Kernel Hilbert Space, crucial for the theory of kernel machines. In Section 2.3, we give some examples and properties of kernels, and in Section 2.4, we discuss a class of kernels that can be used as dissimilarity measures rather than as similarity measures.

Overview

Prerequisites

The chapter builds on knowledge of linear algebra, as briefly summarized in Appendix B. Apart from that, it can be read on its own; however, readers new to the field will profit from first reading Sections 1.1 and 1.2.


2.1 Polynomial Kernels
2.2.1 Positive Definite Kernels
2.2.2, 2.2.3 RKHS Representation
2.2.4, 2.2.5 Mercer Representation
2.2.6, 2.2.7 Data Dependent Representation
2.3 Examples and Properties of Kernels
2.4 Conditionally Positive Definite Kernels

2.1 Product Features

In this section, we think of $\mathcal{X}$ as a subset of the vector space $\mathbb{R}^N$, $(N \in \mathbb{N})$, endowed with the canonical dot product (1.3).

Monomial Features

Suppose we are given patterns $x \in \mathcal{X}$ where most information is contained in the dth order products (so-called monomials) of entries $[x]_j$ of x,

$$[x]_{j_1} \cdot [x]_{j_2} \cdots [x]_{j_d}, \tag{2.3}$$

where $j_1, \ldots, j_d \in \{1, \ldots, N\}$. Often, these monomials are referred to as *product features*. These features form the basis of many practical algorithms; indeed, there is a whole field of pattern recognition research studying *polynomial classifiers* [484], which is based on first extracting product features and then applying learning algorithms to these features. In other words, the patterns are preprocessed by mapping into the feature space $\mathcal{H}$ of all products of d entries. This has proven quite effective in visual pattern recognition tasks, for instance. To understand the rationale for doing this, note that visual patterns are usually represented as vectors whose entries are the pixel intensities. Taking products of entries of these vectors then corresponds to taking products of pixel intensities, and is thus akin to taking logical "and" operations on the pixels. Roughly speaking, this corresponds to the intuition that, for instance, a handwritten "8" constitutes an eight if there is a top circle *and* a bottom circle. With just one of the two circles, it is not half an "8," but rather a "0." Nonlinearities of this type are crucial for achieving high accuracies in pattern recognition tasks.

Let us take a look at this feature map in the simple example of two-dimensional patterns, for which $\mathcal{X} = \mathbb{R}^2$. In this case, we can collect all monomial feature extractors of degree 2 in the nonlinear map

$$\Phi : \mathbb{R}^2 \to \mathcal{H} = \mathbb{R}^3, \tag{2.4}$$

$$([x]_1, [x]_2) \mapsto ([x]_1^2, [x]_2^2, [x]_1[x]_2). \tag{2.5}$$

This approach works fine for small toy examples, but it fails for realistically sized