Chapter 1

Statistical Measures of Dependence

In this chapter we describe the mathematical foundations on which the Hilbert-Schmidt independence criterion is prescribed. For a further comprehensive review the reader may be interested in (Muandet et al 2016).

1.1 Kernels

The goal of almost every machine learning problem is to discover a *funtion* which relates input data to output data. In the simplest case, the data is given in the form

$$(x_1, y_1), ..., (x_m, y_m) \in \mathbb{X} \times \mathbb{R}^N \tag{1.1}$$

where $\{x_i\}$ are input vectors and $\{y_i\}$ are (typically) output scalars. In order to learn a function that reliably assigns output labels to input values we must deal with two (related) problems: (1) how do we compute the similarity between inputs x_j and x_k, and (2) do the inputs permit a boundary that can be found by our choice of algorithm? In the simplest case we could use the Euclidean dot product as a similarity measure on the inputs and assign labels to test values according to the training inputs they lie closest to. This amounts to finding a hyperplane that separates the input cases, and is easily visualised as a line separating each class of input in two-dimensions. Unfortunately, however, this example requires that the input data is *linearly separable*, which is never guaranteed.

One approach to the linear separability problem is to use neural networks, which are often effective at learning a non-linear decision boundary that separates classes. Whilst both popular, and effective, neural networks will not be the focus of this work. Rather, we will explore the field of *reproducing kernel Hilbert spaces* (RKHS), which allows us to perform what is colloquially known as the *kernel trick.* Rather than operating in the original input space we project each of the input examples to a higher (and possibly infinite) dimensional space where we perform inference. Obviously, however, it is likely that operating in this high dimensional space is intractable, therefore we choose the space in such a way that we can compute inner products without explicitly computing the high dimensional features. Formally, we wish to find a map

$$\Phi \colon \mathbb{X} \to \mathbb{H}, \tag{1.2}$$

and a function that allows

$$k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle \tag{1.3}$$

such that $\Phi(x)$ is a mapping of the point x to a feature space and $k(x, x)$ is a function that computes dot products between features. Given the inputs $x_1, ..., x_m \in \mathbb{X}$ we refer to $k \colon \mathbb{X}^2 \to \mathbb{R}$ as a *kernel* and to $K_{ij} := k(x_i, x_j)$ as its associated $m \times m$ *Gram matrix.*

References

Duda, R. O., Hart, P. E., and Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.

Renyi, A. (1961). On measures of entropy and information.

Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, *5*(1), 3–55.

Wells, W. M., Viola, P., Atsumi, H., Nakajima, S., and Kikinis, R. (1996). Multimodal volume registration by maximization of mutual information. *Medical image analysis*, *1*(1), 35–51.