ColumbiaX: Machine Learning: Week1 : Building blocks for Machine Learning

dataset - model - infer hidden variables - goal(predict&explore)

generative models and discriminative models.

The basic ideas of generative models is to learn the probability distribution that generates the observed data from a large enough training set. Some generative models can give you an estimation of the parameter that defines the probability distribution. While others can give you new examples generated from the same paobability distribution. One advantage of generative models is that it can take variable-length input, for example, for an image, using SIFT, we may get many different local descriptors in vector length. And for hidden markov model, it can handle sequences of varying length.

The discriminative models have outweighted generative models in discriminative tasks.

And sometimes we can combine both of the strength of generative models and discriminative models using kernerls.

For a nonlinear feature mapping $\varnothing(\mathbf{x})$, the kernel function is given by

$$\mathbf{k}(\mathbf{x}, \mathbf{x}') = \varnothing(\mathbf{x})^T \varnothing(\mathbf{x})$$

The concept of a kernel formulated as an inner product in a feature space allows us to build interesting extensions of many well-known algorithms by making use of the kernel trick, also known as kernel substitution. With kernels, we can avoid explicit introduction of the feature vector $\varnothing(\mathbf{x})$, which allows us implicitly to use feature spaces of high, even infinite, dimensionality. a kernel expresses the appropriate form of similarity between $\mathbf{x}$ and $\mathbf{x}'$ according to the intended application

Constructing kernels

One approach is to choose a feature space mapping $\varnothing(\mathbf{x})$ and then use this to find the correspongding kernel. An alternative approach is to construct kernel functions directly. In this case, we must ensure that the function we choose is a valid kernel, inother words that it corresponds to a scalar product in some (perhaps infinite dimensional) feature space. We don't have to construct the function $\varnothing(\mathbf{x})$ explicitly. A necessary and sufficient condition for a function $\mathbf{k}(\mathbf{x}, \mathbf{x}‘)$ to be a valid kernel is that the Gram Matrix $\mathbf{K}$ should be positive semidefinite. The elements of Gram matrix is given by $\mathbf{k}_{ij} = \varnothing(\mathbf{x})^T \varnothing(\mathbf{x})$