8803 Machine Learning Theory

Maria Florina Balcan Lecture 17: October 20, 2011

Kernels

Definition 1 *A pairwise function $k(\cdot,\cdot)$ is a kernel is it corresponds to a legal definition of a dot product.*

As discussed last time, one can easily construct new kernels from previously defined kernels. Suppose k_1 and k_2 are valid (symmetric, positive definite) kernels on X. Then, the following are valid kernels:

1. $k(x,z) = \alpha k_1(x,z) + \beta k_2(x,z)$, for $\alpha, \beta \geq 0$.
2. $k(x,z) = k_1(x,z)k_2(x,z)$.
3. $k(x,z) = k_1(f(x), f(z))$, where $f : X \to X$.
4. $k(x,z) = g(x)g(z)$, for $g : X \to R$.
5. $k(x,z) = f(k_1(x,z))$, where f is a polynomial with positive coefficients.

   Proof: Since each polynomial term is a product of kernels with a positive coefficient, the proof follows by applying 1 and 2.
6. $k(x,z) = exp(\tilde{k}(x,z))$

   Proof: We have $\exp(x) = \lim_{i\to\infty}(1 \ldots + x^i/i!)$. The proof follows from 5 and the fact that: $k(x,z) = \lim_{i\to\infty} k_i(x,z)$.
7. $k(x,z) = \exp\left(\frac{-||x-z||^2}{\sigma^2}\right)$

   Proof We have:

$$\begin{aligned} k(x,z) &= \exp\left(\frac{-||x-z||^2}{\sigma^2}\right) = \exp\left(\frac{-||x||^2 - ||z||^2 + 2x^T z}{\sigma^2}\right) = \\ &= \exp\left(\frac{-||x||^2}{\sigma^2}\right)\exp\left(\frac{-||z||^2}{\sigma^2}\right)\exp\left(\frac{2x^T z}{\sigma^2}\right) = \\ &= (g(x)g(z))\exp(k_1(x,z)) \end{aligned}$$

   Clearly, $g(x)g(z)$ is a kernel according to 4, and $\exp(k_1(x,z))$ is a kernel according to 6. According to 2, the product of two kernels is a valid kernel. All these imply that $k(x,z) = \exp\left(\frac{-||x-z||^2}{\sigma^2}\right)$ is a legal kernel.

Note: There are two key properties that are required of a kernel function for an application. First, it should capture an appropriate measure of similarity for the given domain, and secondly, *its evaluation should require significant less computation that would be needed in an explicit evaluation of the corresponding mapping ϕ.* There are several ways to shortcut the computation: use a closed form analytic expression, exploit a recursive relation and use dynamic programming.

Another important aspect is that kernel functions are not restricted to vectorial inputs. Kernels can be designed for objects and structures such as graphs, strings, sets, etc.

The polynomial kernel

As mentioned last time, the polynomial kernel is defined as:

$$k_d(x,z) = (< x,z > +\alpha)^d$$

Expanding the polynomial kernel using the binomial theorem we have

$$k_d(x,z) = \sum_{s=0}^{d} \binom{d}{s} \alpha^{d-s} < x,z >^s. \qquad (1)$$

We have $\hat{k}_s(x,z) =< x,z >^s$ is a kernel. A possible feature space is given by all monomials of degree exactly s, $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ where $i_j \in N$ and $\sum_{j=1}^n i_j = s$.

Our discussion last time implies that the features for each component in the sum (1) together form the features for the whole kernel $k_d(x,z)$. So, a possible feature space to the kernel $k_d(x,z)$ are all functions of the form $x_1^{i_1} x_1^{i_2} \cdots x_n^{i_n}$ where $i_j \in N$ and $\sum_{j=1}^n i_j \leq d$.

Fact 1 *The dimension of the feature space of the polynomial kernel $k_d(x,z)$ is $\binom{n+d}{d}$.*

Proof: We prove the result by induction over n and d. For $n = 1$, the number is correctly computed to $d+1$. For $d = 1$ the number is correctly computed to $n+1$. Now consider the general case and divide the monomials into those that contain at least one factor x_1 and those that have $i_1 = 0$. Using the induction hypothesis there are $\binom{n+d-1}{d-1}$ of the first type of monomial, since there is a $1-1$ correspondence between monomials of degree at most d with one factor x_1 and monomials of degree at most $d-1$ involving all base features. The number of monomials of degree at most d satisfying $i_1 = 0$ is on the other hand equal to $\binom{n-1+d}{d}$ since this corresponds to a restriction to one fewer input feature. Hence, the total number of monomials of degree at most d is equal to

$$\binom{n+d-1}{d-1} + \binom{n-1+d}{d} = \binom{n+d}{d},$$

as required. ■

The all subsets kernel

Assume that we have a feature ϕ_A for every $A \subseteq \{1, 2, \ldots, n\}$, defined as $\phi_A(x) = \prod_{i \in A} x_i$. Consider $\phi(x) = (\phi_A(x))_{A \subseteq \{1,2,\ldots,n\}}$ and define $k_\subseteq(x, z) =< \phi(x), \phi(z) >$.

There is a simple computation that evaluates the all-subsets kernel, as the following derivation shows:

$$\begin{aligned} k_\subseteq(x, z) &= < \phi(x), \phi(z) >= \sum_{A \subseteq \{1,2,\ldots,n\}} \phi_A(x)\phi_A(z) = \\ &= \sum_{A \subseteq \{1,2,\ldots,n\}} \prod_{i \in A} x_i z_i = \prod_{i=1}^{n} (1 + x_i z_i). \end{aligned}$$

The ANOVA kernel

The ANOVA (analysis of variance) kernel k_d is like the all-subsets kernel except that is restricted to subsets of the given cardinality d. We have

$$\phi(x) = (\phi_A(x))_{A \subseteq \{1,2,\ldots,n\}, |A|=d}.$$

The dimension of the resulting embedding is clearly $\binom{n}{d}$ since this is the number of such subsets, while the resulting inner product is given by

$$\begin{aligned} k_d(x, z) &= < \phi(x), \phi(z) >= \sum_{|A|=d} \phi_A(x)\phi_A(z) = \\ &= \sum_{1 \le i_1 < i_2 < \ldots i_d \le n} (x_{i_1} z_{i_1})(x_{i_2} z_{i_2}) \ldots (x_{i_d} z_{i_d}). \end{aligned}$$

As we stressed earlier, we aim to be able to evaluate a kernel faster than by an explicit computation of the feature vectors. Here, for an explicit computation the number of operations grows as $\binom{dn}{d}$ since there are $\binom{n}{d}$ features each of which requires $O(d)$ operations to evaluate.

We now show a better recursive method to evaluate the kernel. In order to do so, we introduce a series of intermediate kernels. Let $x_{1:m}$ denote $(x_1, x_2, \ldots, x_m)$. For $m \ge 1$ and $s \ge 0$ we introduce $k_s^m = k_s(x_{1:m}, z_{1:m})$ which is the ANOVA kernel of degree s applied to inputs restricted to the first m coordinates. In order to evaluate $k_s^m(x, z)$ we now argue inductively that its features can be divided into two groups: those that contain x_m and the remainder. There is a $1-1$ correspondence between the first group and the subsets of size $d-1$ restricted to $x_{1:m-1}$, while the second group are subsets of size d restricted to $x_{1:m-1}$. It follows that:

$$k_s^m(x, z) = (x_m z_m) k_{s-1}^{m-1}(x, z) + k_s^{m-1}(x, z).$$

The base of the recursion is $m < s$ or $s = 0$. Clearly, we have, $k_s^m(x, z) = 0$ if $m < s$ (since no subset of size s can be found) and $k_0^m(x, z) = 1$ (since the empty set has a feature value of 1).

The cost of implementing the recursion naively is at least $\binom{n}{d}$. We can however use *dynamic programming* and compute the kernel with $O(nd)$ numerical operations. We save in a dynamic programming table $k_s^m(x,z)$ indexed by s and m as they are computed. If we begin the computation with the first row from left to right and continue down the table taking each row in turn, the evaluation of a particular entry depends on the entry diagonally above to its left and the entry immediately to its left; hence, both values will be in the table. The required value will be bottom rightmost entry in the table.

Diffusion Kernels

Let $G = (S, E)$ be a graph. The vertices are the data points. Let B be a symmetric base similarity matrix of size $|S| \times |S|$ whose entries are the weights of the edges of the graph G. For example, let us consider a biological application. S is a set of proteins, and B is a matrix of 1's and 0's which represent protein-protein interaction. Each location in B with a 1 indicates that the corresponding proteins interact, while a 0 stands for no interaction.

In general, B is not positive semi-definite. Therefore, it cannot be used directly as a kernel. Diffusion kernels convert the similarity rule into a kernel.

Consider $B^2 = BB^T$. If the graph G is unweighted then the (i,j)-th entry of B^2 is the number of common friends between the the i-th and j-th data points (or the number of paths of length 2 between i and j) and it can be thought of as a measure of their similarity. Clearly B^2 is positive semi-definite. Higher powers of B measure higher order similarities. In general, only the even powers are guaranteed to be positive semi-definite. It is natural to consider a weighted sum of the powers of B in which the higher orders are given lower weights. Let us consider

$$\exp(\lambda B) = \sum_{k=0}^{\infty} \frac{1}{k!} \lambda^k B^k,$$

for $\lambda < 1$. If $B = U\Lambda U^T$ is the spectral decomposition of B, then we have

$$B^2 = U\Lambda U^T U\Lambda U^T = U\Lambda^2 U^T.$$

In general we have $B^k = U\Lambda^k U^T$. Therefore,

$$\exp(\lambda B) = U \exp(\lambda\Lambda) U^T.$$

Since $\exp(\lambda\Lambda)$ is a kernel, we get that $\exp(\lambda B)$ is a kernel as well.

This is an example of a diffusion kernel. The term diffusion derives from the connection to random walks and the heat equation in physics.