CS281B/Stat241B: Advanced Topics in Learning & Decision Making

Examples of Kernels I

Lecturer: Michael I. Jordan *Scribes: Vinod Prabhakaran*

In lectures 3 and 4 we encountered the Gaussian and polynomial kernels. In this lecture we will learn a few other examples of widely used kernels.

1 All-subsets kernel (cf. polynomial kernel)

Let $I = \{1, 2, \ldots, m\}$ be the indices of variables x_i, $i \in I$. For every subset A of I, let us define $\phi_A(x) = \prod_{i \in A} x_i$. We will define $\phi_\varnothing(x) = 1$. Let $\phi(x) = (\phi_A(x))_{A \subseteq I}$. The all-subsets kernel is defined as

$$\begin{aligned} K(x, y) &= \langle \phi(x), \phi(y) \rangle \\ &= \sum_{A \subseteq I} \phi_A(x)\phi_B(y) \\ &= \sum_{A \subseteq I} \prod_{i \in A} x_i y_i \\ &= \prod_{i=1}^{m} (1 + x_i y_i). \end{aligned}$$

The last step is easy to see by expanding $(1+x_1y_1)(1+x_2y_2)\ldots(1+x_my_m)$. Another way of checking this is using the graph in Fig. 1. Summing over all paths, the product along the paths, we get the first expression, while the second expression is the product of sum of the parallel paths between adjacent nodes.

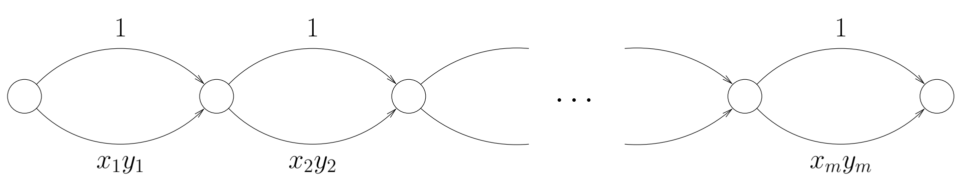

Figure 1: All-subsets kernel as a graph kernel.

This leads to a general technique: for any graph we can try to sum over all paths, the product along the paths. Such kernels are known as *graph kernels*. But they need not necessarily lead to a "nice" formula as in the case above. We are interested in kernels for which fast evaluation algorithms (usually based on dynamic programming) exist.

2 ANOVA kernels

ANOVA (analysis of variance) kernel K_d is like the all-subset kernel, except that it is restricted to subsets of cardinality d.

$$\begin{aligned}\phi_d(x) &= (\phi_A(x))_{|A|=d} \,. \\ K_d(x,y) &= \langle \phi_d(x), \phi_d(y) \rangle \\ &= \sum_{|A|=d} \phi_A(x)\phi_A(y) \\ &= \sum_{1 \le i_1 < i_2 < \ldots < i_d \le m} (x_{i_1} y_{i_1})(x_{i_2} y_{i_2}) \ldots (x_{i_d} y_{i_d}) \,.\end{aligned}$$

Recall that the polynomial kernel is

$$K(x,y) = \sum_{1 \le i_1 \le i_2 \le \ldots \le i_d \le m} (x_{i_1} y_{i_1})(x_{i_2} y_{i_2}) \ldots (x_{i_d} y_{i_d}) \,.$$

To learn how to compute the ANOVA kernel, let us make the following recursive definition

$$K_s^r(x,y) = (x_r y_r) K_{s-1}^{r-1}(x,y) + K_s^{r-1}(x,y).$$

Also let $K_0^0(x,y) = 1$, and $K_{r+1}^r = 1$. The recursion is represented graphically in Fig. 2. It is easy to check that, K_s^r is the ANOVA kernel with cardinality of subsets set to s and involving upto r variables. Therefore, at each intermediate node in the graph, we have a kernel.

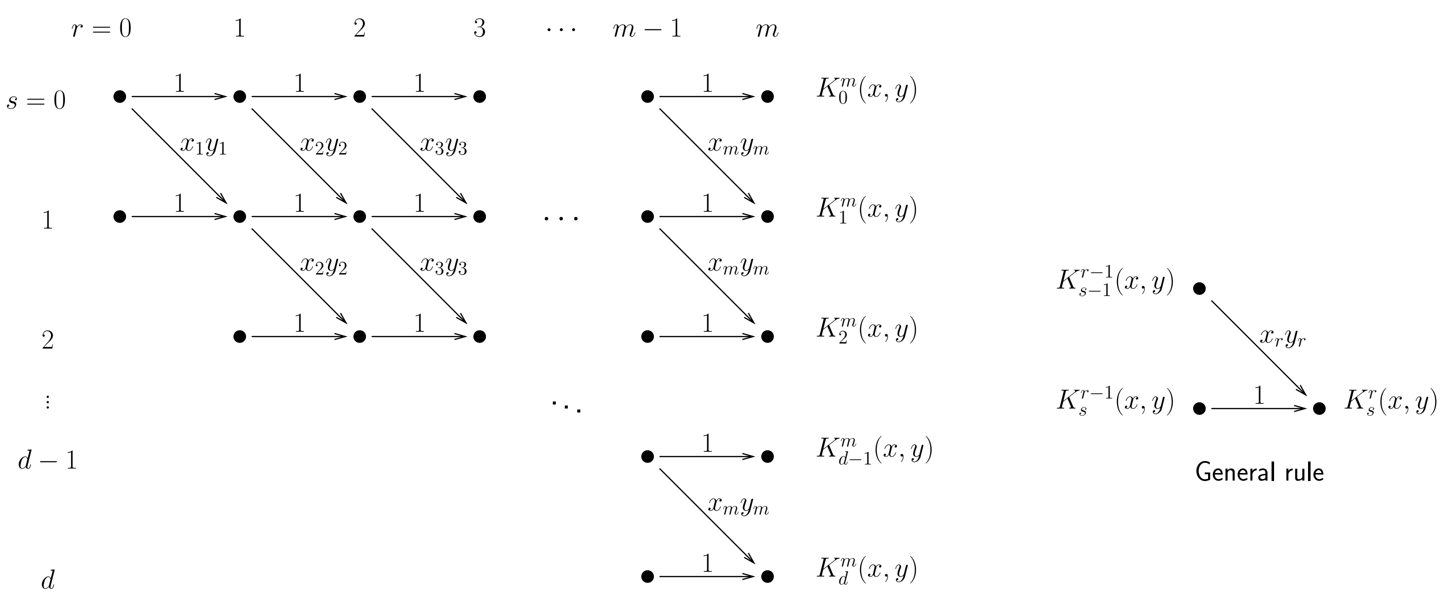

Figure 2: ANOVA kernel as a graph kernel.

For computing the kernel with all degrees less than or equal to d, i.e., $K(x,y) = \sum_{|A| \le d} \phi_A(x)\phi_A(y)$, we add up the last column in Fig. 2,

$$K(x,y) = \sum_{s \le d} K_s^m(x,y).$$

In general, we may have a weighted sum, $K(x,y) = \sum_{i=0}^{d} \alpha_i K_i^m(x,y),\ \alpha_i \ge 0$.

3 Diffusion kernels

Let $G = (S, E)$ be a graph. The vertices are the data points. Let $\mathbf{B}$ be a symmetric *base "similarity" matrix* of size $|S| \times |S|$. The entries in $\mathbf{B}$ 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 $\mathbf{B}$ is a matrix of 1's and 0's which represent protein-protein interaction. Each location in $\mathbf{B}$ with a 1 indicates that the corresponding proteins interact, while a 0 stands for no interaction. Diffusion kernels convert the similarity rule into a kernel. Note that though $\mathbf{B}$ is symmetric, in general, it is not positive semi-definite. Therefore, it cannot be used directly as a kernel.

Consider $\mathbf{B}^2 = \mathbf{BB} = \mathbf{BB}^T$. The (i, j)-th entry of $\mathbf{B}^2$ is the number of "common friends" between the the i-th and j-th data points. Thus it is a measure of their similarity. Higher powers of $\mathbf{B}$ measure higher order similarities, but only the even powers are guaranteed to be positive semi-definite. It is natural to consider a weighted sum of the powers of $\mathbf{B}$ in which the higher orders are given lower weights. Let us consider

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

If $\mathbf{B} = \mathbf{U\Lambda U}^T$ is the spectral decomposition of $\mathbf{B}$, then we have $\mathbf{B}^2 = \mathbf{BB} = \mathbf{U\Lambda U}^T\mathbf{U\Lambda U}^T = \mathbf{U\Lambda}^2\mathbf{U}^T$, and in general $\mathbf{B}^k = \mathbf{U\Lambda}^k\mathbf{U}^T$. Therefore,

$$\exp(\lambda \mathbf{B}) = \mathbf{U} \exp(\lambda \mathbf{\Lambda}) \mathbf{U}^T. \tag{1}$$

Note that this is one of the "19 dubious ways to compute the exponential of a matrix" [1], but it works reasonably well for the case considered here, namely symmetric $\mathbf{B}$. It is also clear from (1) that $\exp(\lambda \mathbf{B})$ is a kernel.

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

Another example of diffusion kernels is the von Neumann kernel

$$\sum_{k=0}^{\infty} \lambda^k \mathbf{B}^k = (\mathbf{I} - \lambda \mathbf{B})^{-1}$$

where we need $|\lambda| \leq \min_k \frac{1}{|\lambda_k|}$, λ_k are the eigenvalues of $\mathbf{B}$, for convergence.

4 String kernels

Strings are sequences defined on some alphabet Σ, say. String kernels usually count the number of "matches" between two strings, where the definition of a "match" varies from kernel to kernel. As an example consider the "p-spectrum" kernel. It is a count of how many contiguous substrings of length p the two strings have in common. In other words,

$$\phi_u^p(s) = |\{(v_1, v_2) : s = v_1 u v_2,\ u \in \Sigma^p\}|$$
$$p\text{-"spectrum" kernel, } K_p(s, t) = \sum_{u \in \Sigma^p} \phi_u^p(s) \phi_u^p(t).$$

We will study more about string kernels in the next lecture.

References

[1] C. Moler and C. Van Loan, “Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later,” *SIAM Rev.*, vol. 45, no. 1, pp. 3–49 (electronic), 2003.