New Trends in Optimization and Computational Algorithms
(NTOC2001)
(December 9-13, 2001, Kyodai-Kaikan, Kyoto, Japan)

Deriving kernels from probabilistic models on discrete objects

Jean-Philippe Vert *
Bioinformatics Center, Institute for Chemical Research, Kyoto University

Several learning methods using kernel functions ([1]) have been developed and shown to be highly efficient for tasks such as classification, regression, density estimation or clustering. One important feature of these methods is there modularity: they can be applied to any kind of object (e.g., images, sound, microarray data, protein sequences...) as soon as a kernel function has been defined for these objects.

Kernels for real-valued vectors, such as polynomial or radial basis function kernels, have been known for a long time and successfully used in many applications where objects are naturally represented as vectors, e.g., images or microarray data. More recently new kernels have been developed for discrete objects such as sequences or trees, which naturally arise in computational biology or in natural language processing ([2, 3, 4]). These kernels open new analysis opportunities in these data-rich fields and are promised a bright future if they are confirmed to outperform "classical" analysis methods.

We have developed a general method to derive a kernel $K(x,y)$ from a probabilistic distribution $p(x)$ for discrete objects $(x,y) \in \mathcal{X}^2$ such as fixed-length strings or trees. This kernel interpolates between two natural kernels (see [3]), the diagonal kernel :

$$\forall (x,y) \in \mathcal{X}^2, \quad K_{\text{diag}}(x,y) = p(x)\delta(x,y), \tag{1}$$

where δ is Kronecker's symbol, and the product kernel:

$$\forall (x,y) \in \mathcal{X}^2, \quad K_{\text{prod}}(x,y) = p(x)p(y). \tag{2}$$

The interpolation is carried out in the following way. We consider composite objects $x = (x_i)_{i \in I}$ where I is a discrete set of indices (e.g., $I = \{1, \dots, N\}$ for sequences of length N) and $x_i \in \mathcal{A}$ for all $i \in I$, where $\mathcal{A}$ is a discrete set called the alphabet. If $J \subset I$ is a subset of indices we can build the following J-based interpolated kernel:

$$K_J(x,y) = K_{\text{prod}}(x_J, y_J) K_{\text{diag}}(x_{I \setminus J}, y_{I \setminus J}) = p(x)p(y)\frac{\delta(x_J, y_J)}{p(x_J)}. \tag{3}$$

In other words the diagonal kernel is applied to elements of x indexed by J, and the product kernel is applied to the other elements. For two sequences x and y to have a non-zero kernel $K(x,y)$ they must have the same value on the elements indexed by J. Moreover (3) shows that

*Uji, Kyoto 611-0011, Japan. e-mail: Jean-Philippe.Vert@mines.org

for given probabilities $p(x)$ and $p(y)$ the value of $K(x,y)$ increases when $p(x_J)$ decreases, that is when the subsequence x_J shared by x and y becomes rare. This is an intuitively satisfactory property as one expects that the rarer a common subpart the stronger the evidence that the two objects x and y are related (for example, if two texts share a very uncommon word, it is likely that they are related; the same is true for two proteins which share a rare combination of amino acids).

Equation (3) easily generalizes to the more interesting situation where one has a list of potentially interesting subset of indices $\mathcal{V} \subset \mathcal{P}(I)$, by the following formula (the notation $\mathcal{P}(I)$ denotes the power set of I, i.e. the set of all subsets of I):

$$K_{\mathcal{V}}(x,y) = \frac{p(x)p(y)}{|\mathcal{V}|} \sum_{J \in \mathcal{V}} \frac{\delta(x_J, y_J)}{p(x_J)}. \tag{4}$$

In particular one easily checks that the diagonal kernel is recovered when $\mathcal{V} = \{\emptyset\}$, and the product kernel is recovered when $\mathcal{V} = \{S\}$.

For a general distribution p and set $\mathcal{V}$, Eq. (4) is usually intractable because of the summation in $J \in \mathcal{V}$ whose explicit computation becomes prohibitive in terms of computational time for large sets $\mathcal{V}$. However various factorization can be obtained for particular choices of p and $\mathcal{V}$, leading to efficient computations (see [5] and a forthcoming paper). As an example if p is a product distribution $p(x) = \prod_{i \in I} p_i(x_i)$ and $\mathcal{V} = \mathcal{P}(I)$ then the following formula holds (involving a product of $|I|$ terms while the size of $\mathcal{V}$ is $2^{|I|}$):

$$K_{\mathcal{V}}(x,y) = \frac{1}{2^{|I|}} \prod_{i \in I} \phi_i(x_i, y_i), \tag{5}$$

with:

$$\phi_i(x_i, y_i) = \begin{cases} p_i(x_i) + p_i(x_i)^2 & \text{if } x_i = y_i, \\ p_i(x_i) p_i(y_i) & \text{if } x_i \neq y_i. \end{cases} \tag{6}$$

This kernel has been implemented and tested with a support vector machine on a problem of cleavage site recognition for protein sequences ([5]), exhibiting good classification performances. More applications are currently under development.

References

[1] V. Vapnik (1998) *Statistical learning theory.* Wiley.

[2] T. Jaakkola, M. Diekhans, and D. Haussler (2000) A discrimitive framework for detecting remote protein homologies. *Journal of Computational Biology*, **7**, 95.

[3] D. Haussler (1999) Convolution Kernels on Discrete Structures. *Technical report UCSC-CRL-99-10.*

[4] C. Watkins (1999) Dynamic alignment kernels. *Technical report CSD-TR-98-11.*

[5] J.-P. Vert (2002) SVM prediction of signal peptide cleavage site using a new class of kernels for strings. *To appear in the Proceedings of the Pacific Symposium on Biocomputing 2002.*