CS281B/Stat241B: Advanced Topics in Learning & Decision Making

Lecture 5: Properties of Kernels and the Gaussian Kernel

Lecturer: Michael I. Jordan *Scribe: Simon Lacoste-Julien*

lecture of 2/04/2004 - notes written on 2/11/2004

Question about last class: for linear regression, how can we express α in terms of the Gram matrix K?

Answer: we don't do linear regression in the feature space since there is the danger of *overfitting* because of the usually high-dimensionality of the feature space. *Ridge regression*, on the other hand, avoids this by penalizing the norm of β.

5.1 Summary of where we are

We have an input space $\mathcal{X}$ and a feature map $\Phi : \mathcal{X} \mapsto \mathcal{H}$, where $\mathcal{H}$ is the feature space (which is usually a Hilbert space, that is, a complete inner product space). To be able to apply the kernel trick, the algorithm must make use of the transformed data $\{\Phi(x_i)\}$ *only* through inner products $K(x_i, x_j) =< \Phi(x_i), \Phi(x_j) >$.

With this said, there are two interesting questions that we can ask:

1. What algorithms can be kernelized? We have seen kernel optimal margin classifier, kernel ridge regression; we will see kernel PCA and a couple of others.
2. Which kernels are interesting? This is a *model selection problem*. We note that one way to find an appropriate kernel for a specific problem is to use a weighted combination of classical kernels.

5.2 Feature space representation for combination of kernels

We now give another interpretation of the closure properties of kernels that we saw last class, now using the feature space representation. So we let $K_1(x, y)$ and $K_2(x, y)$ be some kernels on $\mathbb{R}^N \times \mathbb{R}^N$. To each kernel K_i, there corresponds at least one feature map $\Phi_i : \mathbb{R}^N \mapsto \mathcal{H}_i$ given from Mercer's theorem. Using those feature maps, we can prove that several combinations of K_i's yield a new kernel. The following table lists on the left some combinations of kernels (in kernel space) which give rise to another kernel; on the right, the corresponding feature map Φ which gives rise to this kernel is given in terms of the original feature maps Φ_i.

closure property	**feature space representation**
a) $K_1(x, y) + K_2(x, y)$	$\Phi(x) = (\Phi_1(x), \Phi_2(x))^T$
b) $\alpha K_1(x, y)$ for $\alpha > 0$	$\Phi(x) = \sqrt{\alpha}\Phi_1(x)$
c) $K_1(x, y)K_2(x, y)$	$\Phi(x)_{ij} = \Phi_1(x)_i \Phi_2(x)_j$ (tensor product)
d) $f(x)f(y)$ for any f	$\Phi(x) = f(x)$
e) $x^T Ay$ for $A \succeq 0$ (i.e. psd)	$\Phi(x) = L^T x$ for $A = LL^T$ (Cholesky)

From those properties, we conclude that a polynomial of kernels is still a kernel. Also, it was mentioned that the pointwise limit of kernels is also a kernel.

A few proofs:
a) $\Phi(x) = (\Phi_1(x), \Phi_2(x))^T$, i.e. the concatenation of the Φ_1 vector with the Φ_2 vector[1].

$$\begin{aligned}
K(x,y) &= <\Phi(x), \Phi(y)> \\
&= \left\langle \begin{pmatrix}\Phi_1(x)\\ \Phi_2(x)\end{pmatrix}, \begin{pmatrix}\Phi_1(y)\\ \Phi_2(y)\end{pmatrix} \right\rangle \\
&= <\Phi_1(x), \Phi_1(y)> + <\Phi_2(x), \Phi_2(y)> \\
&= K_1(x,y) + K_2(x,y) \qquad \square
\end{aligned}$$

c) Here the slogan is "Hadamard product in kernel space becomes tensor product in feature space". The new feature vector is defined as a rank 2 tensor (i.e. with two indices): one index for each of the original feature vector. $\Phi(x)_{ij} = \Phi_1(x)_i \Phi_2(x)_j$ means that the (i,j) component of the new feature vector is the product of the i^{th} component of $\Phi_1(x)$ and the j^{th} component of $\Phi_2(x)$[2]. This being said, we have the following derivation, using the natural componentwise inner product for $\Phi(x)$:

$$\begin{aligned}
K(x,y) &= <\Phi(x), \Phi(y)> \\
&= \sum_{i,j} \Phi(x)_{ij} \Phi(y)_{ij} \quad {}^{3} \\
&= \sum_{i,j} \Phi_1(x)_i \Phi_2(x)_j \Phi_1(y)_i \Phi_2(y)_j \\
&= \left(\sum_i \Phi_1(x)_i \Phi_1(y)_i\right)\left(\sum_j \Phi_2(x)_j \Phi_2(y)_j\right) \\
&= <\Phi_1(x), \Phi_1(y)><\Phi_2(x), \Phi_2(y)> \\
&= K_1(x,y) K_2(x,y) \qquad \square
\end{aligned}$$

5.3 Some operations in feature space

We now look at some operations that we can do in the feature space without using Φ explicitly, but by knowing K.

[1]Note from the scribe: a student asked how we could define the concatenation of two infinite vectors. Doing something like $\Phi(x) = (\Phi_1(x)_1, \Phi_1(x)_2, \Phi_1(x)_3, \ldots, \Phi_2(x)_1, \Phi_2(x)_2, \ldots)^T$, where $\Phi_1(x)_i$ denotes the i^{th} component in $\mathcal{H}_1$, seemed to puzzle him because of the intermediate limit in the definition of the vector. We note that this is a perfectly valid operation in mathematics, and the curious reader could find its formalization in the theory of ordinals where the *ordering* of objects like $\{1, 2, 3, \ldots, \omega, \omega + 1, \omega + 2, \ldots, \omega 2, \omega 2 + 1, \omega 2 + 2, \ldots\}$ is formalized. But another way to define the new vector without an intermediate limit is to interlace the two original vectors: $\Phi(x) = (\Phi_1(x)_1, \Phi_2(x)_1, \Phi_1(x)_2, \Phi_2(x)_2, \Phi_1(x)_3, \Phi_2(x)_3, \ldots)^T$, i.e. $\Phi(x)_i = \Phi_1(x)_{(i+1)/2}$ if i is odd; $\Phi_2(x)_{i/2}$ otherwise.

[2]Note from the scribe: Mike has mentioned that he would like us to see the feature space as an abstract vector space rather than as a set of column vectors. This gives us more flexibility to define vectors: they can be matrices, polynomials, functions, etc. And then we can work with vectors as objects rather than having to use their individual components (which depend on the basis chosen). If this sounds gibberish to you, please review your linear algebra!

Aside A **tensor** is the generalization of geometric vectors and their transformation properties (under change of coordinate systems) to objects which need more than 1 index to describe their transformation properties. Their first use was in Physics to describe rigid body mechanics. A tensor of rank 2 (2 indices) can be seen as a matrix. A tensor of rank 1 (1 index) is a vector and a tensor of rank 0 (no index) is a scalar. So if you don't mind working with infinite matrices, you could see the new feature space as the space of $M \times N$ matrices where $M = \dim(\mathcal{H}_1)$, $N = \dim(\mathcal{H}_2)$, and M and N could possibly be ∞ (countable - see note below). Then $\Phi(x) = \Phi_1(x)\Phi_2(x)^T$ (outer matrix product).

[3]For the sum to make sense, we are assuming that the dimension of $\mathcal{H}_i$ is countable. This is the case when Φ_i is constructed from the countable eigenfunction expansion of $K_i(x,y)$ (see [1, p.184]).

5.3.1 Norm of $\Phi(x)$

$$||\Phi(x)|| = \sqrt{< \Phi(x), \Phi(x) >} = \sqrt{K(x,x)}$$

5.3.2 Normalized kernel

Define

$$\tilde{\Phi}(x) \doteq \frac{\Phi(x)}{||\Phi(x)||}$$

We want

$$\tilde{K}(x,y) =< \tilde{\Phi}(x), \tilde{\Phi}(y) >= \frac{K(x,y)}{||\Phi(x)||||\Phi(y)||} = \frac{K(x,y)}{\sqrt{K(x,x)K(y,y)}}$$

and the last equality gives the expression for the normalized kernel (and is a kernel because it was derived from a feature map $\tilde{\Phi}$).

5.3.3 Norm of linear combinations

$$\begin{aligned} ||\sum_i \alpha_i \Phi(x_i)||^2 &= < \sum_i \alpha_i \Phi(x_i), \sum_j \alpha_j \Phi(x_j) >= \sum_{i,j} \alpha_i \alpha_j < \Phi(x_i), \Phi(x_j) > \\ &= \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) = \alpha^T K \alpha \end{aligned}$$

5.4 Example: a simple novelty detection algorithm

The novelty problem is as follows: we watch a sequence of i.i.d. data $\{x_1, \ldots, x_n\}$; for a given new datum x, we want to classify it as "old" or "new" whether it belongs or not (respectively) to the underlying distribution $\mathcal{D}$. A simple algorithm can tackle this task by classifying x as "new" if it appears farther to the centroid of the empirical data than any observed datum. More precisely, we define

$$\phi_c \doteq \frac{1}{n} \sum_i \Phi(x_i)$$

to be the *centroid* (or *sample mean*). The squared distance of x_i to the centroid is $d_i \doteq ||\Phi(x_i) - \phi_c||^2$. Hence, the algorithm classifies x as "new" if

$$||\Phi(x) - \phi_c||^2 > \max_{1 \leq i \leq n} d_i$$

As mentioned in the previous section, we can easily derive an expression for this distance using only the kernel:

$$||\Phi(x) - \phi_c||^2 = K(x,x) - \frac{2}{n} \sum_i K(x, x_i) + \frac{1}{n^2} \sum_{i,j} K(x_i, x_j)$$

What is the probability that I mess up? For the false positive case, what is the probability that I classify x as "new" when it came from the underlying distribution $\mathcal{D}$? The following theorem (which gives a frequentist bound for the probability of having a false positive with a specific fudge factor function of the support radius of $\mathcal{D}$) answers partly this question:

Theorem 1. *For a random i.i.d. training sample picked from $\mathcal{D}^n$, the following bound will hold with probability at least $1-\delta$:*

$$\mathcal{P}_{\mathcal{D}}\left\{ ||\Phi(x)-\phi_c||^2 > \max_{1\leq i\leq n} d_i + 2\sqrt{\frac{2R^2}{n}}(\sqrt{2}+ln\sqrt{\frac{1}{\delta}}) \right\} \leq \frac{1}{n+1}$$

where the support of the distribution $\mathcal{D}$ is assumed to be contained in a ball of radius R.

Note: there are two sources of randomness in this theorem: the training set, and the test element x. The $\mathcal{P}_{\mathcal{D}}$ in the bound refers to the probability distribution for the test element x. For each specific training set, the bound will either be true or false (and won't be random). But for a random training set, the probability that the bound is true is at least $1-\delta$.

5.5 Gaussian kernel

We recall that the Gaussian kernel is defined as

$$K(x,y) = \exp(-\frac{||x-y||^2}{2\sigma^2})$$

There are various proofs that a Gaussian is a kernel. One way is to see the Gaussian as the pointwise limit of polynomials. Another way is using the following theorem of functional analysis:

Theorem 2 (Bochner). *If a kernel K can be written in terms of $||x-y||$, i.e. $K(x,y) = f(||x-y||)$ for some f, then K is a kernel iff the Fourier transform of f is non-negative.*

5.6 Parzen windows (aka "kernel" density estimation)[4]

We now consider the problem of estimating the density function of an underlying distribution $\mathcal{D}$ using i.i.d. data. One method could be to build an histogram of the relative frequency of appearance of the empirical data (using bins). One problem with this method is that the risk (squared error) goes to zero slowly as n goes to infinity.

Another method is to use Gaussian bumps around the data points:

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} K(x,x_i) \tag{1}$$

where $K(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{||x-y||^2}{2\sigma^2})$ and σ is the bandwidth parameter to control the smoothness of the estimate.

The Parzen window estimate is equation (1) with any positive function $K(\cdot,x_i)$ with unit integral and which is usually translation invariant.

We can make the link between the Parzen window estimator and our feature space representation with the following fact:

Fact. *The Parzen window estimator at x using a positive semidefinite "kernel" (local bump function) K can be obtained as the inner product between $\Phi(x)$ and ϕ_c i.e.*

$$f(x) = <\Phi(x), \phi_c>$$

[4]The word "kernel" here has a different meaning than the usual one used in this course; it rather means that some local functions (called "kernel") are used for the density estimation.

where $\phi_c = \frac{1}{n}\sum_i \Phi(x_i)$ *is the centroid and* Φ *is obtained by applying Mercer theorem on* K *(and is thus also a "kernel" in our sense)*

Proof.

$$\begin{aligned} f(x) &= \frac{1}{n}\sum_i K(x, x_i) \\ &= \frac{1}{n}\sum_i < \Phi(x), \Phi(x_i) > \qquad \text{(by Mercer)} \\ &= < \Phi(x), \frac{1}{n}\sum_i \Phi(x_i) > \\ &= < \Phi(x), \phi_c > \end{aligned}$$

□

So in some sense, doing "ball things" in feature space is justified since we can rederive classical results from it.

References

[1] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An Introduction to Kernel-Based Learning Algorithms, in *IEEE Neural Networks*, 12(2):181-201, March 2001. 3