Kernels

Volker Tresp
Summer 2014
Smoothness Assumption

- So far we used prior knowledge to define the right basis functions: the assumption is that $f(x)$ can be approximated by a weighted sum of basis functions.
- Alternatively, it might make sense to have a preference for smooth functions: functional values close in input space should have similar functional values.
- In the figure it might make sense that the functional values at $x_i$ and $x$ are similar (smoothness assumption).
- Thus, one might prefer the smooth (continuous) function in favor of the dashed function.
$f(x)$

$x_i$  $x$
Introduction Kernels

- One can implement smoothness assumptions over kernel functions.
- A kernel function $k(x_i, x)$ determines, how neighboring functional values are influenced when $f(x_i)$ is given.
- Example: Gaussian kernel.
$k(x_i, x)$
Kernels and Basis Functions

• It turns out that there is a close relationship between kernels and basis functions:

\[ k(x_i, x) = \sum_{j=1}^{M_\phi} \phi_j(x_i)\phi_j(x) \]

• Thus: given the basis functions, this equation gives you the corresponding kernel

• We have encountered the kernel already in the discussion on basis functions. Note the kernel is a function that is represented with basis functions \( \phi_j(x) \), that have the weight \( \phi_j(x_i) \).

• For positive definite kernels, we can also go the other way: given the kernels I can give you a corresponding set of basis functions (not unique)
\[
\phi(x) = (0.25, 1.00, 0.25)^T \\
\phi(x) = (0.10, 0.90, 0.50)^T \\
\phi(x) = (0.02, 0.60, 0.90)^T \\
\phi(x) = (0.00, 0.01, 0.30)^T
\]

\[
k(x, x) = \phi^T(x)\phi(x) = 1.12 \\
k(x, x) = \phi^T(x)\phi(x) = 1.05 \\
k(x, x) = \phi^T(x)\phi(x) = 0.83 \\
k(x, x) = \phi^T(x)\phi(x) = 0.08
\]
Kernel Prediction

- Regression

\[ \hat{y}(z) = \sum_{i=1}^{N} v_i k(z, x_i) \]

- Classification

\[ \hat{y}(z) = \text{sign} \left( \sum_{i=1}^{N} v_i k(z, x_i) \right) \]

- The solution contains as many kernels as there are data points \( N \) (independent on the number of underlying basis functions \( M \))

- Thus: I can work with a finite number of kernels, instead of an infinite number of basis functions
One Kernel for Each Data Point
Different Forms of the Cost Function

- We start with the PLS cost function for models with basis functions

- Regularized cost function

\[
\text{cost}^{pen}(\mathbf{w}) = \sum_{i=1}^{N} (y_i - \sum_j w_j \phi_j(x_i))^2 + \lambda \sum_{i=0}^{M} w_i^2
\]

\[
= (\mathbf{y} - \Phi \mathbf{w})^T (\mathbf{y} - \Phi \mathbf{w}) + \lambda \mathbf{w}^T \mathbf{w}
\]

where \( \Phi \) is the design matrix design with \( (\Phi)_{i,j} = \phi_j(x_i) \).
Implicit Solution

- We calculate the first derivatives and set them to zero,

$$\frac{\partial \text{cost}^{pen}(w)}{\partial w} = -2\Phi^T(y - \Phi w) + 2\lambda w = 0$$

It follows that one can write,

$$w_{pen} = \frac{1}{\lambda} \Phi^T(y - \Phi w_{pen})$$
Approach

• This is not an explicit solution ($w_{pen}$ appears on both sides of the equation). But we know now, that we can write the solution as a linear combination of the input vectors

$$w_{pen} = \Phi^T v = \sum_{i=1}^{N} v_i \phi(x_i)$$

• Note that we have a sum over $N$ data points (and not $M$ basis functions)
Kernel Model

- We immediately get,

\[ f(x) = \sum_{j=1}^{M_\phi} w_{j,pen} \phi_j(x_i) = \phi(x)^T w_{pen} \]

\[ = \phi(x)^T \Phi^T v = \sum_{i=1}^{N} v_i k(x, x_i) \]

with \( v = (v_1, \ldots, v_N)^T \) and

\[ k(x, x_i) = \phi(x)^T \phi(x_i) = \sum_{k=1}^{M_\phi} \phi_k(x) \phi_k(x_i) \]
A New Form of the Cost Function

- We can substitute the constraints, and obtain

$$\text{cost}^{pen}(v) = (y - \Phi \Phi^T v)^T (y - \Phi \Phi^T v) + \lambda v^T \Phi \Phi^T v$$

$$= (y - K v)^T (y - K v) + \lambda v^T K v$$

where $K$ is an $N \times N$ matrix with elements

$$k_{i,j} = \phi(x_i)^T \phi(x_j) = \sum_{k=1}^{M_\phi} \phi_k(x_i) \phi_k(x_j)$$

- An important result: We can write the cost function, such that only inner
products of the basis functions appear (i.e., the kernels), but not the basis functions themselves!
Kernel Parameters

• Now we can take the derivative of the cost function with respect to $v$ (note, that $K = K^T$)

$$\frac{\partial \text{cost}^{pen}(v)}{\partial v} = 2K(y - Kv) + 2\lambda K v$$

such that

$$v_{pen} = (K + \lambda I)^{-1}y$$
Kernel Prediction

- A prediction can be written as

$$\hat{f}(z) = \vec{\phi}(z)^T w = \vec{\phi}(z)^T \Phi^T v_{pen} = \sum_{i=1}^{N} v_i k(z, x_i)$$

with

$$k(z, x_i) = \vec{\phi}(z)^T \vec{\phi}(x_i)$$

- Another important result: we can write the solution such that only inner products are used; the solution can be written as a weighted sum of \(N\) kernels.
One Kernel for Each data Point
With only One Training Data Point

- With only one training data point we get

\[ f(z) = v_1 k(z, x_1) \]

- As discussed previously:
Comments and Interpretation of a Kernel

• This is interesting, since there can be more basis functions than data points; in particular this result is valid, even if we work with an infinite number of basis functions!

• It is even possible to start with the kernels, without knowing exactly, what the underlying basis functions are

• Different interpretations of the kernel
  
  – As inner product \( k(x_i, z) = \phi^T(x)\phi(z) \)
  
  – As covariance: how strong is the correlation of the functional values at different inputs \( k(x_i, z) = \text{cov}(f(x_i), f(z)) \)

• When \( N >> M \) it is computationally more efficient to work with basis functions (requiring \( M^3 + M^2N \) operations). When \( M >> N \), the kernel version is more efficient, requiring \( N^3 + N^2M \) operations. If the kernels are known a priori (i.e., if they do not need to be calculated via inner product), the kernel solution requires \( N^3 \) operations.
• Still, not all functions are valid kernel functions. We need the following theorem ...
Mercer’s Theorem

- (From Vapnik: The nature of statistical learning theory. Springer, 2000)

- **Mercer’s Theorem:** To guarantee, that the symmetric functions \( k(x, z) = k(z, x) \) from \( L_2 \) permits an expansion as

\[
k(x, z) = \sum_{h=1}^{\infty} \lambda_h \phi_h^T(x) \phi_h(z)
\]

with positive coefficients \( \lambda_h > 0 \), it is necessary and sufficient, that

\[
\int \int k(x, z) g(x) g(z) dx dz > 0
\]

for all \( g \neq 0 \), for which

\[
\int g^2(x) dx < \infty
\]

- The theorem says, that for so-called positive-definite kernels (“Mercer kernels”), a decomposition in basis functions is possible!
• Each kernel-matrix $K$ is then also positive (semi-) definite
Kernel Design

• Linear Kernel

\[ k(x_i, x_j) = x_i^T x_j \]

Basis functions, as well as kernel functions, are linear

• Polynomial kernel (1)

\[ k(x_i, x_j) = (x_i^T x_j)^d \]

The basis functions are all ordered polynomials of order \( d \)

• Polynomial kernel (2)

\[ k(x_i, x_j) = (x_i^T x_j + R)^d \]

The corresponding basis functions are all polynomials of order \( d \) or smaller

• Gauß-kernels (RBF-kernels)

\[ k(x_i, x_j) = \exp \left( -\frac{1}{2s^2} \| x_i - x_j \|^2 \right) \]
These kernels correspond to infinitely many Gaussian basis functions

- Sigmoid (“neural network”) kernels

\[ k(x_i, x_j) = \text{sig} \left( x_i^T x_j \right) \]
Appendix: Detour on Function Spaces
Rewriting the Cost Function (see lecture on basis functions)

- The cost function
  \[ f(x) = \sum_i w_i \phi_i(x) \]
  can be thought of as an inner product between the function \( f(x') = \sum w_i \phi_i(x') \) and the function \( k(x, x') = \sum \phi_i(x) \phi_i(x') \) in the space of the basis functions, thus
  \[ f(x) = \langle f, k_x \rangle_\phi \]

- \( k(x, x') \) is our kernel and in this context is called a reproducing kernel.

- Also recall that \( w^T w = \langle f, f \rangle_\Phi \)

- With all of this, we can write our cost function as
  \[ \text{cost}^{pen}(w) = \sum_{i=1}^{N} \left( y_i - \langle f, k_{x_i} \rangle_\phi \right)^2 + \lambda \langle f, f \rangle_\Phi \]
Note that only for the minimizer we can write also

\[ \langle f, f \rangle_\Phi = v^T K v \]
Representer Theorem

- **Representer Theorem**: Let $\Omega$ be a strictly monotonously increasing function and let $\text{loss()}$ be an arbitrary loss function, then the minimizer of the loss function

$$\sum_{i=1}^{N} \text{loss}(y_i, f(x_i)) + \Omega(\|f\|_\Phi)$$

can be represented as

$$f(x) = \sum_{i=1}^{N} v_i k(x_i, x)$$

- $\|f\|_\Phi = \sqrt{\langle f, f \rangle_\Phi}$ is a norm in a reproducing kernel Hilbert space (RKHS)
- So kernel solutions are possible for all cost functions we are considering!