TU München
Fakultät für Informatik
PD Dr. Rudolph Triebel
John Chiotellis

Machine Learning for Robotics and Computer Vision
Summer term 2016

Homework Solution 5

Topic 1: Kernels
June 24, 2016

Exercise 1: Constructing kernels

During this solution we assume the feature spaces of k_1 and k_2 to have finite dimensions. Thus they can be written as $k_1(x_1, x_2) = \phi_1(x_1)^T\phi_1(x_2)$, $k_2(x_1, x_2) = \phi_2(x_1)^T\phi_2(x_2)$, where $\phi_1(x) \in \mathbb{R}^{n_1}$, $\phi_2(x) \in \mathbb{R}^{n_2}$. Note however that in general feature spaces can be infinite dimensional (e.g. $\phi(x) \in l^2(\mathbb{R})$, see 4.). We now have to define new kernels via a scalarproduct $k(x_1, x_2) = \langle\phi(x_1), \phi(x_2)\rangle$

a) $k(x_1, x_2) = k_1(x_1, x_2) + k_2(x_1, x_2)$

To warm up:
$\phi(x) = \begin{pmatrix}\phi_1(x)\\ \phi_2(x)\end{pmatrix} \in \mathbb{R}^{n_1+n_2}$

b) $k(x_1, x_2) = k_1(x_1, x_2)k_2(x_1, x_2)$

Note that the matrix-products do not commute, so it is a bit of work:

$$\begin{aligned}
k(x_1, x_2) &= \phi_1(x_1)^T\phi_1(x_2)\phi_2(x_1)^T\phi_2(x_2)\\
&= (\sum_i (\phi_1(x_1))_i(\phi_1(x_2))_i)(\sum_j (\phi_2(x_1))_j(\phi_2(x_2))_j)\\
&= \sum_i\sum_j (\phi_1(x_1))_i(\phi_1(x_2))_i(\phi_2(x_1))_j(\phi_2(x_2))_j\\
&= \underbrace{\sum_i\sum_j}_{\sum_k} \underbrace{(\phi_1(x_1))_i(\phi_2(x_1))_j}_{\phi_k(x_1)}\underbrace{(\phi_1(x_2))_i(\phi_2(x_2))_j}_{\phi_k(x_2)}\\
&\Rightarrow \phi(x) = \begin{pmatrix}(\phi_1(x))_1(\phi_2(x))_1\\ \vdots\\ (\phi_1(x))_1(\phi_2(x))_{n_2}\\ (\phi_1(x))_2(\phi_2(x))_1\\ \vdots\\ (\phi_1(x))_{n_1}(\phi_2(x))_{n_2}\end{pmatrix} \in \mathbb{R}^{n_1\cdot n_2}
\end{aligned}$$

c) $k(x_1, x_2) = f(x_1)k_1(x_1, x_2)f(x_2)$

$\phi(x) = f(x)\phi_1(x)$

d) $k(x,y) = \exp(k_1(x,y))$

Again we write the scalarproduct as a sum:

$$\begin{aligned}\exp((\phi_1(x))^T\phi(y)) &= \exp(\sum(\phi_1(x))_i(\phi_1(y))_i) \\ &= \prod \exp((\phi_1(x))_i(\phi_1(y))_i)\end{aligned}$$

Since we already know that the product of kernels is again a kernel it remains to show, that $\exp((\phi(x))_i(\phi(y))_i)$ is a kernel for a fixed index i. In the following we will omit i and imagine ϕ_1 to be a scalar-valued function. From the Taylor-expansion of the exponential function, we know that

$$\exp(\phi_1(x))(\phi_1(y)) = \sum_{k=0}^{\infty} \frac{1}{k!}(\phi_1(x))^k(\phi_1(y))^k$$

This is an inner product in $l^2(\mathbb{R})$ with

$$\phi(x) = \begin{pmatrix} \phi_1(x) \\ \frac{1}{\sqrt{2}}\phi_1(x)^2 \\ \frac{1}{\sqrt{6}}\phi_1(x)^3 \\ \vdots \\ \frac{1}{\sqrt{k!}}\phi_1(x)^k \\ \vdots \end{pmatrix}$$

e) $k(x_1,x_2) = x_1^T A x_2$

Since A is symmetric positive-definite, it admits a Cholesky decomposition $A = LL^T$. Therefore, we have $x_1^T A x_2 = x_1^T LL^T x_2 = (L^T x_1)^T(L^T x_2)$. So $\phi(x) = L^T x$.

Exercise 2: Polynomial kernel

a) Show (by induction) that $k_d(x_i,x_j) = (x_i^T x_j)^d$ is a kernel for every $d \geq 1$.

$d = 1$: $\phi(x) = x$. Induction step: Exercise 1 a, 1b.

b) Find $\phi_d(x)$ such that $k_d(x_i,x_j) = \phi_d(x_i)^T\phi_d(x_j)$.

Consider first $d = 2$:

$$\begin{aligned}(x_i^T x_j)^2 &= (x_{i1}x_{j1} + x_{i2}x_{j2})^2 \\ &= x_{i1}^2 x_{j1}^2 + 2x_{i1}x_{j1}x_{i2}x_{j2} + x_{i2}^2 x_{j2}^2 \\ \phi(x) &= \begin{pmatrix} x_1^2 & \sqrt{2}x_1x_2 & x_2^2 \end{pmatrix}^T\end{aligned}$$

For larger d the coefficients can be obtained by using the Binomial theorem/Pascal's triangle:

$$\begin{array}{ccccccccc} & & & & 1 & & & & \\ & & & 1 & & 1 & & & \\ & & 1 & & 2 & & 1 & & \\ & 1 & & 3 & & 3 & & 1 & \\ 1 & & 4 & & 6 & & 4 & & 1 \end{array}$$

c) Find $\tilde{\phi}_2(x)$ for $\tilde{k}_2(x, y) = (x^T y + d)^2$ $(d > 0)$.

We can easily construct the kernel using the properties we proved in exercise 1.

i) $x^T y = \phi(x)\phi(y)$ is a valid kernel

ii) $d = \sqrt{d}\sqrt{d}$ is a valid kernel

iii) $x^T y + d$ We proved that a sum of kernels is also a kernel

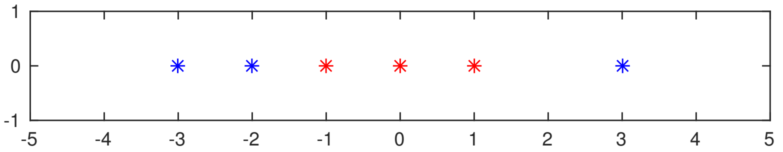
iv) Finally, we proved that the product of two kernels is also a kernel

Exercise 3: Feature Spaces

Consider a dataset with a single feature $x \in \mathbb{R}$ and labels $y \in \{+1, -1\}$. Data points $-3, -2, 3$ have label $+1$ and data points $-1, 0, 1$ have label -1.

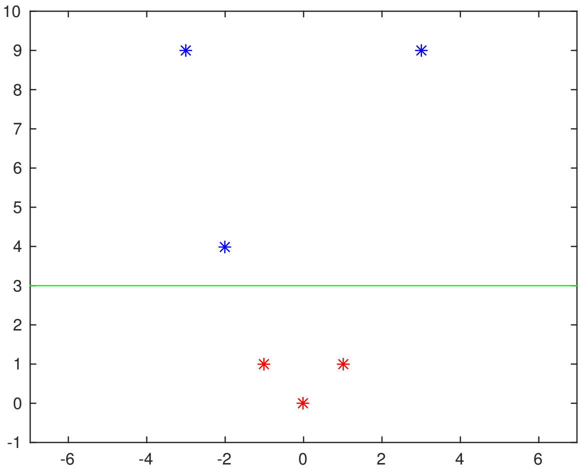
a) Is this dataset linearly separable? Why?

No, the dataset is not linearly separable. This becomes obvious once we plot the data points. There is no single line that can completely separate the two classes.



b) Find a feature map $\phi(x) \in \mathbb{R}^2$ so that the dataset is linearly separable. *(Drawing the data helps.)*

We can choose a feature map $\phi(x) = (x \quad x^2)^T \in \mathbb{R}^2$ so that the dataset becomes linearly separable. Now we can draw a line, actually infinitely many lines, that separate the two classes.



c) Considering the determinant of a 2 × 2 Gram matrix show that a positive definite kernel satisfies the Cauchy-Schwartz inequality.

The 2 × 2 Gram matrix is defined as

$$K = \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) \\ k(x_2, x_1) & k(x_2, x_2) \end{pmatrix} =$$

with

$$k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

. Since the kernel is positive definite, all eigenvalues of K must be positive and therefore the determinant too. This means that

$$\begin{aligned} \det(K) &= k(x_1, x_1)k(x_2, x_2) - k(x_1, x_2)k(x_2, x_1) \\ &= \phi(x_1)^T \phi(x_1)\phi(x_2)^T \phi(x_2) - \phi(x_1)^T \phi(x_2)\phi(x_2)^T \phi(x_1) \\ &= \langle \phi(x_1), \phi(x_1) \rangle \langle \phi(x_2), \phi(x_2) \rangle - |\langle \phi(x_1), \phi(x_2) \rangle|^2 > 0 \end{aligned}$$

$$\Rightarrow \quad |\langle \phi(x_1), \phi(x_2) \rangle|^2 < \langle \phi(x_1), \phi(x_1) \rangle \langle \phi(x_2), \phi(x_2) \rangle$$

Topic 2: Gaussian Processes

Exercise 4: Gaussian Processes Regression

Consider a GP regression model in which the kernel function is defined in terms of a fixed set of nonlinear basis functions. Show that the predictive distribution is identical to the one of the Bayesian linear regression model (see Lecture and Homework Assignment 2).

Hint 1: Both models have Gaussian predictive distributions.
Hint 2: Make use of:

$$(I + AB)^{-1}A = A(I + BA)^{-1}$$

and the Woodburry identity:

$$(A + BD^{-1}C)^{-1} = A^{-1} - A^{-1}B(D + CA^{-1}B)^{-1}CA^{-1}$$

Both the Gaussian process and the linear regression model give rise to Gaussian predictive distributions $p(t_{N+1}|x_{N+1})$ so we simply need to show that these have the same mean and variance.
To do this we make use of the prior over the weights for linear regression:

$$p(w) = \mathcal{N}(w|0, \sigma_w^2 I) \tag{1}$$

which leads to a kernel function defined in terms of the basis functions:

$$k(x_i, x_j) = \sigma_w^2 \phi(x_i)^T \phi(x_j) \tag{2}$$

.

In bayesian linear regression, we assumed Gaussian noise in the data such that

$$p(t_n|y_n) = \mathcal{N}(t_n|y_n, \sigma_n^2) \tag{3}$$

Therefore the elements of the covariance matrix C_N take the form

$$C(x_i, x_j) = k(x_i, x_j) + \sigma_n^2 \delta_{ij} \tag{4}$$

where δ_{ij} is the Kronecker delta.
In matrix notation this is equal to $C_N = \sigma_w^2 \Phi\Phi^T + \sigma_n^2 I_N$ where Φ is the design matrix with $\Phi_{nk} = \phi_k(x_n)$.
We know, a Gaussian process is fully defined by mean and covariance:

$$m(x_{N+1}) = k_*^T C_N^{-1} t \tag{5}$$
$$\sigma^2(x_{N+1}) = (k_{**} + \sigma_n^2) - k_*^T C_N^{-1} k_* \tag{6}$$

.

Combining these results, we get a mean for the predictive distribution

$$m_{N+1} = \sigma_w^2 \phi_{N+1}^T \Phi^T (\sigma_w^2 \Phi\Phi^T + \sigma_n^2 I_N)^{-1} t \tag{7}$$

.

Now we can use the first hint:

$$\begin{aligned}\Phi^T(\sigma_w^2\Phi\Phi^T + \sigma_n^2 I_N)^{-1} &= \Phi^T(\sigma_n^2(\sigma_n^{-2}\sigma_w^2\Phi\Phi^T + I_N))^{-1} \\ &= \Phi^T\sigma_n^{-2}(\sigma_n^{-2}\sigma_w^2\Phi\Phi^T + I_N)^{-1} \\ &= \sigma_n^{-2}(\sigma_n^{-2}\sigma_w^2\Phi^T\Phi + I_M)^{-1}\Phi^T \\ &= \sigma_n^{-2}\sigma_w^{-2}(\sigma_n^{-2}\Phi^T\Phi + \sigma_w^{-2}I_M)^{-1}\Phi^T \\ &= \sigma_n^{-2}\sigma_w^{-2}S_N\Phi^T\end{aligned}$$

Thus the mean becomes

$$m_{N+1} = \sigma_n^{-2}\phi_{N+1}^T S_N \Phi^T t \tag{8}$$

which is exactly the mean of the predictive distribution in bayesian linear regression.

We do the same for the variance:

$$\begin{aligned}\sigma_{N+1}^2(x_{N+1}) &= \sigma_w^2\phi(x_{N+1})^T\phi(x_{N+1}) + \sigma_n^2 - \sigma_w^4\phi(x_{N+1})^T\Phi^T(\sigma_w^2\Phi\Phi^T + \sigma_n^2 I_N)^{-1}\Phi\phi(x_{N+1}) \\ &= \sigma_n^2 + \phi(x_{N+1})^T(\sigma_w^2 I_M - \sigma_w^4(\sigma_w^2\Phi\Phi^T + \sigma_n^2 I_N)^{-1}\Phi)\phi(x_{N+1})\end{aligned}$$

.

Now we set $A = \sigma^{-2}I_M, B = \Phi^T, C = \Phi, D = \sigma_n^2 I_N$ and apply the Woodburry identity:

$$\sigma_w^2 I_M - \sigma_w^4(\sigma_w^2\Phi\Phi^T + \sigma_n^2 I_N)^{-1}\Phi = (\sigma_w^{-2}I_M + \sigma_n^{-2}\Phi^T\Phi)^{-1} = S_N \tag{9}$$

Therefore, we have

$$\sigma_{N+1}^2 = \sigma_n^2 + \phi(x_{N+1})^T S_N \phi(x_{N+1}) \tag{10}$$

Exercise 5: Gaussian Processes Classification (Programming)

See code. Gaussian processes are more accurate than boosting methods, as they inherently estimate a confidence interval for their predictions (variance). On the other hand, boosting methods are easier to implement and much faster to train which makes them an appealing choice when we need to deal with large datasets.