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

Machine Learning for Computer Vision
Summer term 2017

2. Juni 2017
Topic: Regression, Kernels and Gaussian Processes

Exercise 1: Bayesian Update

Consider a linear regression model with basis functions $\phi(x)$ as presented in the lecture. We assume a Gaussian prior distribution for the weights:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|m_0, S_0)$$

Suppose we have already observed N data points, so the posterior distribution is

$$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|m_N, S_N)$$

with

$$m_N = S_N(S_0^{-1}m_0 + \sigma^{-2}\Phi^T\mathbf{t}) \quad \text{and} \quad S_N^{-1} = S_0^{-1} + \sigma^{-2}\Phi^T\Phi.$$

Now, we observe a new data point (x_{N+1}, t_{N+1}). What is the new posterior?

Using Bayes rule, we found out that having a Gaussian prior and a Gaussian likelihood gave us a Gaussian posterior which we can use as the prior for the next iteration (next sample that we observe). Now we want to compute $p(\mathbf{w}|\mathbf{t}, t_{N+1}, x_{N+1})$ which reduces to $p(\mathbf{w}|t_{N+1}, x_{N+1}, m_N, S_N)$.

Our likelihood is

$$p(t_{N+1}|x_{N+1}, \mathbf{w}) = \mathcal{N}(t_{N+1}|y(\mathbf{w}, \phi(x)), \sigma^2)$$

Let $\phi_N = \phi(x_N)$ to simplify notation. Writing the likelihood explicitly we get

$$p(t_{N+1}|x_{N+1}, \mathbf{w}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t_{N+1} - \mathbf{w}^T\phi_{N+1})^2}{2\sigma^2}\right)$$

Our posterior is

$$p(\mathbf{w}|t_{N+1}, x_{N+1}, m_N, S_N) = \frac{p(t_{N+1}|x_{N+1}, \mathbf{w})p(\mathbf{w}|\mathbf{t})}{p(t_{N+1}|x_{N+1}, \mathbf{t})}$$

We want the maximum likelihood of the posterior. The denominator is independent of $\mathbf{w}$ so for we can ignore it.

$$
\begin{aligned}
p(\mathbf{w}|t_{N+1}, x_{N+1}, m_N, S_N) &\propto p(\mathbf{w}|\mathbf{t})p(t_{N+1}|x_{N+1}, \mathbf{w}) \\
&\propto \exp\left(-\frac{1}{2}(\mathbf{w} - m_N)^T S_N^{-1}(\mathbf{w} - m_N) - \frac{(t_{N+1} - \mathbf{w}^T\phi_{N+1})^2}{2\sigma^2}\right)
\end{aligned}
$$

Maximizing the likelihood is equivalent to maximizing the log-likelihood and that is the same as minimizing the negative log-likelihood. Therefore we are left only with the arguments of the exponential, and we can omit the $-\frac{1}{2}$ factors.

$$
\begin{aligned}
&(\mathbf{w} - m_N)^T S_N^{-1}(\mathbf{w} - m_N) + \frac{(t_{N+1} - \mathbf{w}^T\phi_{N+1})^2}{\sigma^2} \\
=&\mathbf{w}^T S_N^{-1}\mathbf{w} - 2\mathbf{w}^T S_N^{-1} m_N - 2\frac{\mathbf{w}^T\phi_{N+1}t_{N+1}}{\sigma^2} + \frac{\mathbf{w}^T\phi_{N+1}\phi_{N+1}^T\mathbf{w}}{\sigma^2} + const. \\
=&\mathbf{w}^T(S_N^{-1} + \frac{\phi_{N+1}\phi_{N+1}^T}{\sigma^2})\mathbf{w} - 2\mathbf{w}^T\left(S_N^{-1}m_N + \frac{\phi_{N+1}t_{N+1}}{\sigma^2}\right) + const.
\end{aligned}
$$

where *const.* denotes remaining terms that are independent of w.

Comparing this expression with the maximum likelihood for the prior, we can see that our posterior is

$$p(\mathbf{w}|t_{N+1}, x_{N+1}, m_N, S_N) = \mathcal{N}(w|m_{N+1}, S_{N+1})$$

with

$$S_{N+1}^{-1} = S_N^{-1} + \frac{1}{\sigma^2}\phi_{N+1}\phi_{N+1}^T \quad \text{and} \quad m_{N+1} = S_{N+1}(S_N^{-1}m_N + \frac{\phi_{N+1}t_{N+1}}{\sigma^2})$$

Exercise 2: 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 $x_1^T A x_2 = x_1^T L L^T x_2 = (L^T x_1)^T (L^T x_2)$. So $\phi(x) = L^T x$.

Exercise 3: 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.

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

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

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

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

Exercise 4: Gaussian Processes Regression (Programming)

See code.