Pattern Classification And Machine Learning - EPFL - Fall 2015

Emtiyaz Khan, Timur Bagautdinov, Carlos Becker, Ilija Bogunovic & Ksenia Konyushkova

6. Gaussian Processes

6.1 Goals

The goal of this exercise is to

- Understand Gaussian Process priors and their relation with the kernel function.
- Apply Gaussian Process regression to a toy example. Fit the model by maximizing the marginal probability $p(\mathbf{y}|X)$.

6.2 Gaussian Process Math Review

Before starting, a quick review of GPs.

The Gaussian Process prior is a multi-variate Normal distribution, or

$$\mathbf{f}|X \sim \mathcal{N}\big(\boldsymbol{\mu}(X),\, K(X,X)\big)\,, \tag{6.1}$$

where $K(X,X) = [k(\mathbf{x}_n, \mathbf{x}_m)]$, also called a *kernel matrix*. Typically, $\boldsymbol{\mu} = \mathbf{0}$, though not strictly necessary.

If measurements of the form $\mathbf{y} = \mathbf{f}(X) + \boldsymbol{\epsilon}$ are available, with $\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \sigma_n^2 I\right)$, then

$$\begin{bmatrix}\mathbf{y}\\ \mathbf{f}_*\end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{\mu}\left(\begin{bmatrix}X\\ X_*\end{bmatrix}\right), \begin{bmatrix}K(X,X)+\sigma_n^2 I & K(X,X_*)\\ K(X_*,X) & K(X_*,X_*)\end{bmatrix}\right)$$

Conditioning on $\mathbf{y}$ (training data), we get

$$\mathbf{f}_* \mid \mathbf{y}, X_*, X \;\sim\; \mathcal{N}(\mu', \,\Sigma')$$

with

$$\mu' \;=\; \boldsymbol{\mu}(X_*) \,+\, K(X_*,X)\left[K(X,X)+\sigma_n^2 I\right]^{-1}\Big(\mathbf{y}-\boldsymbol{\mu}(X)\Big) \tag{6.2}$$

$$\Sigma' \;=\; K(X_*,X_*) \,-\, K(X_*,X)\left[K(X,X)+\sigma_n^2 I\right]^{-1} K(X,X_*) \tag{6.3}$$

Note: unlike the lecture, here, without loss of generality, we allow for the mean of the prior to be a function of X as well.

6.2.1 Marginal Likelihood

Generally the kernel function depends on hyper-parameters θ. For example, in the RBF kernel one hyper-parameter is the length-scale L that defines how smooth our prior is. Setting such parameters is generally not easy. One can use cross-validation. But, as discussed in the class, we can also estimate them by maximizing the marginal log-likelihood $p(\mathbf{y}|X,\theta)$ shown below.

$$\log p(\mathbf{y}|X,\theta) = \; - \; \frac{1}{2}(\mathbf{y}-\boldsymbol{\mu}(X))^T\left(K+\sigma_n^2 I\right)^{-1}(\mathbf{y}-\boldsymbol{\mu}(X)) \\ -\frac{1}{2}\log|K+\sigma_n^2 I| - \frac{n}{2}\log 2\pi\,, \tag{6.4}$$

where n is the number of elements of $\mathbf{y}$, and $\log|W|$ is the notation for the **logarithm of the determinant** of W. Depending on the problem, the noise variance σ_n^2 may also be an unknown, and therefore considered a hyper-parameter, and part of θ.
(If you are interested in the derivation of the equation above, check the book by Rasmussen and Williams on Gaussian Processes)

6.3 Visualizing Gaussian Process Priors

Exercise 6.1 Fill in the necessary code in `ex6_priorplot.m` to show draws of $\mathbf{f}$ from the prior above. Experiment with different kernels, and think about the following questions:

- Constant kernel: $k(\mathbf{x}_n, \mathbf{x}_m) = k_o^2$. *What is the correlation K_{mn} for this kernel? How much information can one point reveal about another one? What does k_o^2 express? What if $\boldsymbol{\mu}$ is constant but not zero?*
- Linear kernel: $k(\mathbf{x}_n, \mathbf{x}_m) = k_o^2 + k_1^2\, \mathbf{x}_n^T\, \mathbf{x}_m$
- RBF kernel: $k(\mathbf{x}_n, \mathbf{x}_m) = k_o^2 + k_1^2 \exp\left(-|\mathbf{x}_n - \mathbf{x}_m|^2/L^2\right)$. *Play with the values of L and observe its effect on the smoothness of the prior.*

To help organize your code, write three functions, one for each kernel type, that compute the kernel matrix given input matrices X_1 and X_2, returning $K(X_1, X_2)$:

- `K = constantKernel(X1, X2, ko)`
- `K = linearKernel(X1, X2, ko, k1)`
- `K = RBFKernel(X1, X2, ko, k1, L)`

Remember that X_1 and X_2 are feature matrices, so that they must have equal number of columns, but may have different number of rows (samples). The size of $K(X_1, X_2)$ is the number of rows of X_1 by the number of rows of X_2.

Note: in this exercise X will have only one column, but you should make your code generic to any number of columns.

k_o, k_1 and L are real-valued, positive hyper-parameters. You can start with $k_o = k_1 = L = 1$. ■

Exercise 6.2 *Bonus:* the mean of the prior $\boldsymbol{\mu}$ can also be a function of X, which allows for introducing more prior knowledge about the problem being solved. Experiment with different values of $\boldsymbol{\mu}(X)$, such as a linear model, sinusoids, etc. ■

6.4 Fitting a Constant

Let's now turn to making predictions, given a prior and measurements. Let's take a constant kernel $k(\mathbf{x}_n, \mathbf{x}_m) = k_o^2$. In our prior we want to encode that we know that f is somewhere around $\mu = 2$, but we know this with high uncertainty, so $k_o^2 = 1$.

Now you are given 4 noisy measurements $y_i = f(x_i) + \epsilon$ (note that x is irrelevant in this case), with $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$:

- $y_1 = 2.5$
- $y_2 = 2.3$
- $y_3 = 2.4$
- $y_4 = 2.2$

The measurements are i.i.d. and the measurement noise variance is $\sigma_n^2 = 0.1$.

Exercise 6.3 Given the prior and measurements above,

- Estimate $p(f|y_1, y_2, y_3, y_4)$, incorporating both the prior and measurements, writing down the Gaussian Process prediction Eq. 6.2 and 6.3 in Matlab.
- Suppose that we were very uncertain about our prior belief, so that k_o^2 is very large, for example $k_o^2 = 1000$. *What predicted mean and variance do you get? Compare it to the above results.*
- Now assume that you are equally uncertain as about your prior belief as your measurement, i.e. we set $k_o^2 = \sigma_n^2$. *What predicted mean and variance do you get? Draw your conclusions.*

■

Note: there is no template code for this part, you have to write the code from scratch yourself.

6.5 Fitting a Gaussian Process to Data: Learning the Hyper-parameters

Now you will tackle a problem close to a real-world scenario.

You are given a set of data points to train a regressor, and you are asked to make predictions with a Gaussian Process with a RBF kernel. You know that the measurements are noisy, but you don't know exactly which σ_n^2 to use, nor the length scale L for the kernel, nor k_1^2. That is, you do not know your hyperparameters.

To make things simple for you, we will fix $k_o^2 = 0$ and also the GP prior to have zero mean.

Exercise 6.4 With the information above, find the hyper-parameters $\theta = \{\sigma_n^2, k_1^2, L\}$ of a Gaussian Process that *maximize* the marginal $p(\mathbf{y}|X, \theta)$. Use the file `ex6_rbf_marginal.m` as your template and fill in the necessary code.

Typically, θ is found through gradient ascent, because Eq. 6.4 is differentiable with respect to θ. However, to make it faster to implement, you can use grid search.

You should re-use the function to compute the RBF kernel matrix you wrote for the first exercise above.

Bonus: Find the optimal θ through gradient descent or gradient ascent. Note that you have to differentiate K with respect to θ. ■

You should get a result similar to that of Figure 6.1.

6.6 Bonus: CO2 Concentration Prediction

You will reproduce the CO2 concentration example from the Rasmussen & Williams book. You need to implement the GP prediction equations, using the four kernels discussed in their book. For this exercise you don't need to do hyperparameter search, you can just use the values reported by Rasmussen & Williams in Chapter 5.4.3 of their book. Use the file `ex6_co2` as your template. You should get something similar to Figure 6.2.

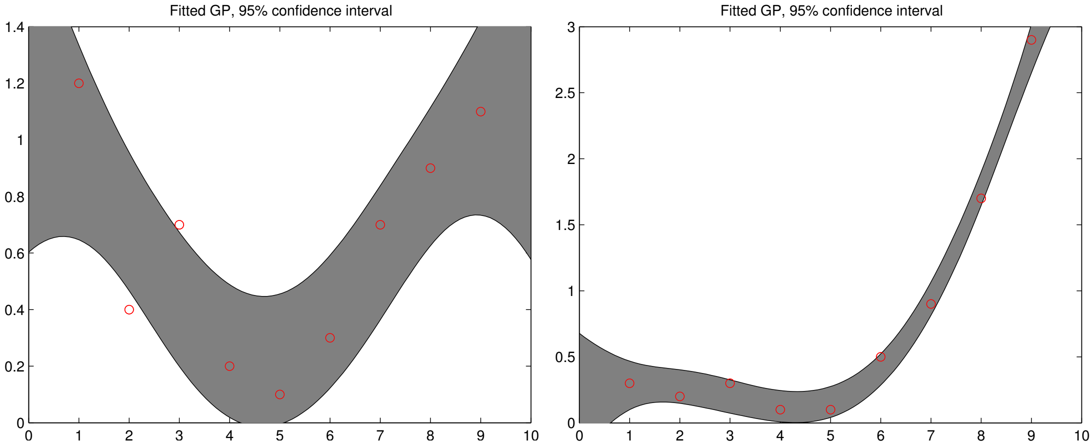

Figure 6.1: GP prediction for the two simulated datasets of Exercise 6.4.

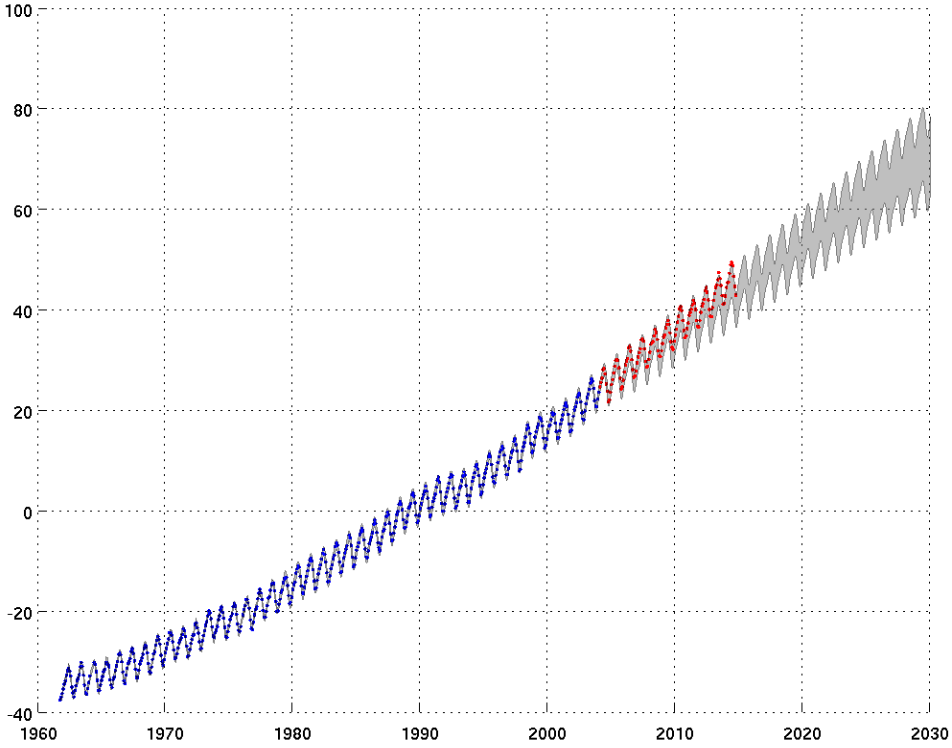

Figure 6.2: CO2 Concentration Data: Training data (blue), unseen data (red), and estimated 95% confidence interval.