Lab 4
Gaussian Processes

In this lab we play with some rather sparse data and try to get a feel for GP. We first have a look at the multivariate Gaussian distribution, and how we can compute the conditional and the marginal distribution of some dimensions given others. We then extend this concept to the Gaussian Process, where we can have infinitely many dimensions, any subset of which has a joint Gaussian distribution.

1 The Gaussian distribution

The conditional distribution of a subset of the dimensions of Gaussian-distributed data is also Gaussian, and we can compute the mean and covariance of this distribution in closed form. If we partition the mean and covariance of the distribution as follows:

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix} \text{ and } \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{bmatrix} \tag{1}$$

then we can define the precision $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$, which we can partition similarly to the covariance

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ba} & \boldsymbol{\Lambda}_{bb} \end{bmatrix} \ . \tag{2}$$

The parameters of the marginal distribution are then given by

$$p(\mathbf{x}_a) = \mathcal{N}(\mathbf{x}_a; \boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa}) \tag{3}$$

while the conditional distribution is parametrised as:

$$p(\mathbf{x}_a|\mathbf{x}_b) = \mathcal{N}(\mathbf{x}_a; \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{aa}^{-1}\boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b), \boldsymbol{\Lambda}_{aa}^{-1}) \tag{4}$$

1.1 Assignment

Consider a Gaussian distribution $\mathcal{N}(\mathbf{x}; \mathbf{0}, \boldsymbol{\Sigma})$ with zero mean and the following covariance matrix:

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \tag{5}$$

1. Plot this distribution using a contour plot

2. What is the marginal distribution over x_1? Use `subplot` to plot this below the contour plot you have just generated.

3. Use `subplot` to plot the distribution over x_1 for the following values of x_2: $x_2 = -3, -2, \ldots, 2, 3$.

Now consider the covariance matrix

$$\mathbf{\Sigma} = \begin{bmatrix} 1 & .7 \\ .7 & 1 \end{bmatrix} \tag{6}$$

1. Plot this distribution using plot3d

2. What is the marginal distribution over x_1?

3. Use `subplot` to plot the distribution over x_1 for the following values of x_2: $x_2 = -3, -2, \ldots, 2, 3$.

2 Gaussian Processes

The point of the above exercise was to emphasise how two things: (1) if we know how two variables co-vary and we observe one, we can compute the distribution of the other and (2), very importantly, if we don't know one of the variables, then the distribution of the other is not affected. These two things allow us to define a Gaussian Process: if we know the value of the output variable for a given input, and we know how that output co-varies with the output for another input, we can compute the distribution over the output for that other output. This distribution is not affected by all the outputs which we do not observe.

I simply give you the implementation of the Gaussian Process in mathematical notation. You are supposed to translate that into matlab, notice how brief the implementation really is, and apply this to a simple dataset. The implementation is given in the following algorithm:

Algorithm 1 Implementation of the Gaussian Process

Input: the training inputs $\mathbf{X}$, the training targets $\mathbf{t}$, the covariance function $k(\mathbf{x}, \mathbf{x}')$, the noise level σ^2, a test input $\mathbf{x}_*$

Output: $\bar{f}_*, \sigma_*^2$, the mean and variance of the predictive distribution, $\log p(\mathbf{t}|\mathbf{X})$ the marginal log-likelihood

$\mathbf{L} \leftarrow \text{cholesky}(\mathbf{K} + \sigma^2\mathbf{I})$

$\boldsymbol{\alpha} \leftarrow \mathbf{L}^\top \backslash (\mathbf{L} \backslash \mathbf{t})$ *A more efficient way of computing* $\mathbf{K}\backslash\mathbf{t}$

$\bar{f}_* \leftarrow \mathbf{k}_*^\top \boldsymbol{\alpha}$ *The most likely prediction for the input* $\mathbf{x}_*$

$\mathbf{v} \leftarrow \mathbf{L} \backslash \mathbf{k}_*$

$\sigma_*^2 \leftarrow k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{v}^\top \mathbf{v}$ *The variance on our prediction*

$\log p(\mathbf{t}|\mathbf{X}) \leftarrow -\frac{1}{2}\mathbf{t}^\top \boldsymbol{\alpha} - \sum_i \log L_{ii} - \frac{n}{2}\log 2\pi$ *How probable is our training data under the model*

The matrix $\mathbf{L}$ is a lower-triangular matrix obtained by Cholesky decomposition of a symmetric square matrix, and is implemented in the `chol` function in Matlab (use the option `'lower'` to force a *lower*-triangular result). The backslash in $\mathbf{a} = \mathbf{M}\backslash\mathbf{b}$ indicates solving the set of linear equations $\mathbf{b} = \mathbf{M}\mathbf{a}$, which is done in Matlab using the backslash operator. In this algorithm, we have defined

the following variables:

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \cdots & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdots & k(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & & & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_2) & \cdots & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \tag{7}$$

$$\mathbf{k}_* = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_*) \\ \vdots \\ k(\mathbf{x}_N, \mathbf{x}_*) \end{bmatrix} \tag{8}$$

The input parameter σ^2 is the noise that we expect to have in our training targets. Make this very small to consider only functions that go exactly through the training datapoints (but don't make it zero, as this results in the kernel matrix being singular).

The kernel function gives the correlation between the outputs corresponding to two inputs. For this lab, use the so-called squared exponential function:

$$k(\mathbf{x}, \mathbf{x}') = \theta \exp -\frac{1}{2l}(\mathbf{x} - \mathbf{x}')^\top (\mathbf{x} - \mathbf{x}') \ , \tag{9}$$

which is parametrised by a scaling parameter θ and the lengthscale l, a parameter which reflects how close together inputs must be to result in a given covariance of the outputs.

2.1 Assignment

For this exercise we will use the "cricket" data set in the file `chirps.mat`, which we saw in the introductory lecture. The first column contains the frequency of the chirps (in Hz), while the second column contains the temperature (in degrees Fahrenheit.) We use GP to perform regression and predict the temperature from the chirps we hear.

First load the data, create training and test sets and plot the data using different colours for both. Then using your implementation of the GP, predict the output and variance on the output for inputs in the range $[12 \dots 22]$. Plot the mean prediction and one standard deviation above and below the mean prediction. Now also make predictions for the inputs corresponding to the test point. Answer the following questions:

1. What is the performance of the GP on the test set
2. How does it depend on the number of training examples
3. Explain the behaviour of the variance around the predictions. How is it affected by the density and the variance of the data?
4. How does the kernel function's parameters affect the result? How about the variance on the targets, σ^2

2.2 Optional Extra

Finally, in blackboard the file "curve.mat" contains the data used in Bishop for the sine-wave regression examples. How well is the GP performing on this data?