a&b) A GP process was trained with a squared exponential covariance function on the prior training data. The central commands were shown below.

**1. hyp.cov=[-1 0];**

**2. hyp.cov=[-10 0];**

**3. z = linspace(-1.9, 1.9, 75)';**

**4. [m s2] = gp(hyp, @infExact, [], covfunc, likfunc, x, y, z);**

|  |  |
| --- | --- |
|  |  |

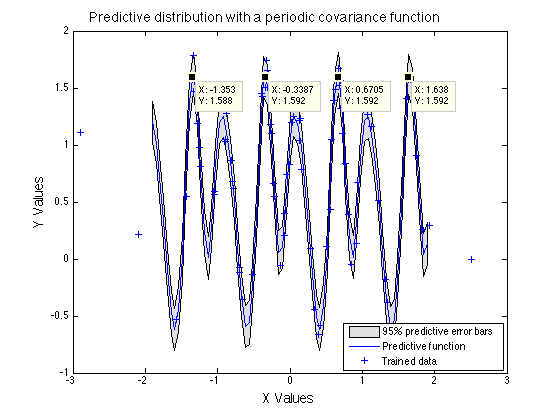
**For all the sections, the following parameters were represented by their respective symbols, M-signal mean, L-length scale, T-signal period, σ-signal variance, N-observation noise, Nlml-negative log marginal likelihood shown in the data table below.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Inferred parameters in part (a), Left | | | | Inferred parameters in part (b), Right | | | |
| L | σ | N | Nlml | L | σ | N | Nlml |
| 0.1282 | 0.897 | 0.1178 | 11.899 | 0 | 0.7064 | 0.7064 | 106.3493 |

The graph shown on the left illustrated the predictive distribution using hyp.cov=[-1 0] while the one on the right demonstrated the distribution using hyp.cov=[-10 0]. The grey shaded regions represented the 95% predictive error bars and blue lines stood for predictive functions. It was obvious that the left-hand data fitting was much better than the right one because the left model fitted data quite well and generated reasonable error bars while the right one had a flat predictive line and it generated huge amounts of error bars regions. It could be verified its inferred length scale of 0 and larger negative log marginal likelihood of 106, indicating the model used had no idea about the locations of predicted data.

c) A GP process was trained with a periodic covariance function on the prior training data.

The central commands were shown below. The central command used was **hyp.cov=[-1 0 0]**.



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Initial parameters | | | | Inferred parameters | | | | |
| L | T | σ | N | Nlml | L | T | σ | N |
| 0.3679 | 1 | 1 | 1 | -29.3368 | 0.7154 | 0.9987 | 0.6902 | 0.0849 |

The data generating mechanism was periodic because the error bars of predictive distribution was smaller than ones in part (a) indicating posterior distribution of predictive function had great confidences on the locations of predictive data. Moreover, the horizontal distance between adjacent labeled points on the graph was roughly equal to 0.997, which was close to the period calculated from inferred parameter, 0.9987. Finally, the small observation noise of 0.0849 indicated the optimal negative log marginal likelihood.

d) To apply the Cholesky decomposition to the covariance matrix, it is necessary to add a small diagonal matrix. This is because covariance function defines elements of a positive definite matrix and the square root of numbers can become negative because of round-off errors in Matlab, in which case the Cholesky decomposition cannot continue. The extra small diagonal matrix into the covariance matrix can promote the positive-definiteness at the expense of lower accuracy. The central commands were shown below.

**1. meanfunc={@meanConst};**

**2. hyp.mean=2;**

**3. meanfunc={@meanSum, {@meanLinear, @meanConst}};**

**4. hyp.mean=[0.5;1];**

**5. y = chol(K)'\*x + m;**

|  |  |
| --- | --- |
|  |  |
| **nlml=77.1381, observation noise=0.2711** | **nlml=60.2954, observation noise=0.1743** |

Different mean sample functions were used to plot their predictive distributions. The left-hand distribution was generated by meanConst and the only effect of this mean function was to change offsets of its distribution. Moreover, the right-hand distribution was generated by the sum of meanLinear and meanConst. As expected, the sum of these two mean function changed linearity of predictive distribution into upwards direction while modifying its offsets. Partial periodic, decreasing and increasing patterns in these two graphs were due to periodic and exponential covariance function used.

e&f) A GP process was trained with a squared exponential with ARD covariance function on the prior 2-D training input and scalar output data. The central commands were shown below.

**1. mesh(reshape(x(:,1),11,11),reshape(x(:,2),11,11),reshape(m,11,11))**

**2. z = x**

**3. [m s2] = gp(hyp, @infExact, meanfunc, covfunc, likfunc, x, y, z)**

|  |  |
| --- | --- |
|  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Initial parameters | | | | | Inferred parameters | | | | | |
| L in D1 | L in D2 | σ | M | N | Nlml | L in D1 | L in D2 | σ | M | N |
| 1 | 1 | 1 | 1 | 0.1 | -20.2214 | 1.4526 | 1.2679 | 0.9971 | 1.7950 | 0.1035 |

The right-hand 3-D graph demonstrated the training data of scalar output versus 2-D input x while the plot on the right represented the posterior distribution of predictive function. It was clear that the predictive function fitted training data very well due to little difference of variance. The observation noise of predictive function is 0.1035

|  |  |
| --- | --- |
|  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Initial parameters | | | | Inferred parameters | | | | |
| L in D1 | σ | M | N | Nlml | L in D1 | σ | M | N |
| 0.3679 | 1 | 1 | 0.1 | -19.5346 | 1.4035 | 1.0148 | 1.9948 | 0.1078 |

Then the previous GP model was trained instead with a squared exponential covariance function on the prior 2-D input and scalar output data. The left-hand graph represented the predictive distribution using covSEard while the right-hand graph stood for the distribution using covSEiso function. Although they did not show huge difference visually, it was inferred that the predictive distribution using covSEard was better due to smaller negative log marginal likelihood (-20.2214<-19.5346) and observation noises. However, the user might also need to consider the limited number of data and over-fitting problems, which could affect us to choose the right model since there would be considerable uncertainty as to which is the better model. The probability density of model in part (e) and (f) were 1.65\*10^-9 and 3.28\*10^-9 respectively so their relative probability was 0.503.

g) A GP process was trained with a sum of two squared exponential with ARD covariance function on the prior 2-D input and output data. The central commands were shown below.

**1. hyp.cov = 0.1\*randn(6,1);**

**2. mesh(reshape(x(:,1),11,11),reshape(x(:,2),11,11),reshape(s2,11,11));**

|  |  |
| --- | --- |
|  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Initial parameters** | | | | | | | | **Inferred parameters** | | | | | | | |
| M | N | 1st covSEard | | | 2st covSEard | | | nlml | N | 1st covSEard | | | 2st covSEard | | |
| L in D1 | L in D2 | σ | L in D1 | L in D2 | σ | L in D1 | L in D2 | σ | L in D1 | L in D2 | σ |
| 1 | 0.1 | 0.796 | 0.9322 | 0.8905 | 0.9352 | 1.0884 | 1.0859 | -66.5233 | 0.0977 | 1.401 | 538.8798 | 1.0152 | 425.1605 | 0.9828 | 0.6988 |

Breaking symmetry was essential because compared with the length scales acquired from part (e), it was clear that for each covSEard function, one of the length scales appeared to be very large (538.88 and 425.16) while the other still remained around 1. It indicated one of the covariance functions was a straight flat line while the other had oscillating strong correlation for adjacent data for each covariance function. Therefore, breaking symmetry could give us two independent strong correlation functions in two different dimensions, promoting more freedom for us to fit data without encountering over-fitting accurately otherwise the model would become the same as in part (e) without breaking symmetry. The strong correlation features were illustrated in the right-hand graph below, which had higher values of covariance and steeper covariance curves compared with the previous model shown on the left.

|  |  |
| --- | --- |
|  |  |

The right-hand graph illustrated the predictive distribution using the sum of covSEard. Furthermore, the predictive distribution generated from new covariance function shown on the right above did not seem to have much difference visually compared with the one obtained from part (e) shown on the left above. However, nlml of -66.5233 was much smaller than both of two previous models, indicating higher probability of using this covariance function to maximize marginal likelihood.

h) A GP process was trained with a linear mean function and a squared exponential covariance function on the monthly prior training CO2 data from 1958 to 2004. Then it was used to predict monthly CO2 data from 2004 to 2011. The central commands were shown below.

**1. hyp = minimize(hyp, @gp, -100, @infExact, meanfunc, covfunc, likfunc, trainyear, trainCO2);**

**2. [m s2] = gp(hyp, @infExact, meanfunc, covfunc, likfunc, trainyear, trainCO2, testyear);**

|  |  |
| --- | --- |
|  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Initial parameters | | | | Inferred parameters | | | | |
| M | L | σ | N | Nlml | M | L | σ | N |
| 1 | 0.3679 | 1 | 1 | 1197.5 | 1.2052 | 31.4088 | 33.7263 | 2.1203 |

The right-hand figure demonstrated the process of using prior training CO2 indicated by blue ‘+’ to predict CO2 from 2004 represented by red line. The zoom-in figure of prediction was shown on the left and the grey shaded region was the predictive error bar with 95% confidence intervals. As can be seen, the predictive GP model used was quite accurate compared with the testCO2 given and the trend from a training model before 2004. The linear patterns of predictive function and error bars were due to the linear covariance function applied. The negative log marginal likelihood of 1197.5 indicated high probabilities of these hyper-parameters to fit the given model with a reasonable observation noise of 2.1203.

i) The GP model in part (h) was modified by substituting a sum covariance function between a period function and a squared exponential covariance function on the monthly training CO2 data from 1958 to 2004. This is because the monthly CO2 data indicated a periodic feature of this training model so the initial value of period hyper-parameter was set to be 0(exp(0)=1). Moreover, the squared exponential covariance function was used due to the advantages of regression using infinitely many Gaussian shaped basis functions placed everywhere, which could give us more freedom to fit model. The central commands were shown below.

**1. covfunc={@covSum,{@covPeriodic, @covSEiso}};**

**2. hyp.cov=[-0.5 0 0 1.85 0];**

|  |  |
| --- | --- |
|  |  |

M-signal mean, L-length scale, T-signal period, σ-signal variance, N-observation noise, Nlml-negative log marginal likelihood represented in the table below.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Initial parameters | | | | | | | Inferred parameters | | | | | | | |
| M | L in covPeriodic | T | σ | L in covSEiso | σ | N | Nlml | M | L in covPeriodic | T | σ | L in covSEiso | σ | N |
| 1 | 0.607 | 1 | 1 | 6.359 | 1 | 1 | 443.699 | 1.185 | 0.643 | 0.999 | 1.049 | 8.928 | 20.315 | 0.511 |

The initial hyper-parameters of mean and covariance matrixes were set to be zero due to avoidance of noises interfered into the GP model. In addition, the initial values of length scale were set relatively small to be 0.607 and 6.359 respectively for better fit to data. The predictive observation noise was 0.511,which was quite optimal since the posterior distribution became nearly peaked around the maximum likelihood. It could be verified by the exponential of negative log marginal likelihood exp(-443.699), which was really small, showing good fit to data. Finally, the significance and principle of two graphs shown above was the same as the ones in part (h). The observation noise of 20.315 was much larger than the previous one due to the periodic covariance function involved, which enlarged the predictive error bars.