**3c) A Practical Example**

To get started, let's consider the simple example of one-dimensional non-linear regression on data corrupted by Gaussian noise. Our data set is

x = gpml\_randn(0.8, 20, 1); % 20 training inputs

y = sin(3\*x) + 0.1\*gpml\_randn(0.9, 20, 1); % 20 noisy training targets

xs = linspace(-3, 3, 61)'; % 61 test inputs

We need specify the mean, covariance and likelihood functions

meanfunc = []; % empty: don't use a mean function

covfunc = @covSEiso; % Squared Exponental covariance function

likfunc = @likGauss; % Gaussian likelihood

Finally we initialize the hyperparameter struct

hyp = struct('mean', [], 'cov', [0 0], 'lik', -1);

Note, that the hyperparameter struct must have the three fields mean, cov and lik. Each field must have the number of elements which corresponds to the functions specified. In our case, the mean function is empty, so takes no parameters. The covariance function is covSEiso, the squared exponential with isotropic distance measure, which takes two parameters (see help covSEiso). As explained in the help for the function, the meaning of the hyperparameters is "log of the length-scale" and the "log of the signal std dev". Initializing both of these to zero, corresponds to length-scale and signal std dev to be initialized to one. The Gaussian likelihood function has a single parameter, which is the log of the noise standard deviation, setting the log to zero corresponds to a standard deviation of exp(-1)=0.37.

A common situation with modeling with GPs is that approprate settings of the hyperparameters are not known a priori. The situation is reflected in the above initialization of the hyperparameters, where the values values are specified without careful justification, perhaps based on some vague notions of the magnitudes likely to be involved. Thus, a common task is to set hyperparameters by optimizing the (log) marginal likelihood. This is done as follows

hyp2 = minimize(hyp, @gp, -100, @infGaussLik, meanfunc, covfunc, likfunc, x, y);

The minimize function minimizes the negative log marginal likelihood, which is returned by the gp function, together with the partial derivatives wrt the hyperparameters. The inference method is specified to be infGaussLik exact inference. The minimize function is allowed a computational budget of 100 function evaluations. The hyperparameters found are

hyp2 =

mean: []

cov: [-0.6352 -0.1045]

lik: -2.3824

showing that the covariance characteristic length-scale is exp(-0.6352)=0.53, the signal std dev is exp(-0.1045)=0.90 and the noise std dev is exp(-2.3824)=0.092 in good agreement with the data generating process.

To make predictions using these hyperparameters

[mu s2] = gp(hyp2, @infGaussLik, meanfunc, covfunc, likfunc, x, y, xs);

To plot the predictive mean at the test points together with the predictive 95% confidence bounds and the training data

f = [mu+2\*sqrt(s2); flipdim(mu-2\*sqrt(s2),1)];

fill([xs; flipdim(xs,1)], f, [7 7 7]/8)

hold on; plot(xs, mu); plot(x, y, '+')