The GPML Matlab Library

This is a short tutorial on the Gaussian Processes for Machine Learning (GPML) Matlab library

- Companion code to Rasmussen and Williams's book
- For documentation and downloading the code, see `http://www.gaussianprocess.org/gpml/code/matlab/doc/`
- I have also posted the code on the LMS

The Exact Function

We will use our favorite polynomial as the exact or "truth" function:

$$f(x) = (x - 3)x^3(x - 6)^4.$$

Open a new script in the same directory as the GPML library, and define this function as follows

```
clear all; close all;
% define the function to be sampled
fe = @(x) (x - 3).*(x.^3).*((x - 6).^4);
```

Sample the Function

Next, we sample the function at the points

$$\{x^{(k)}\}_{k=1}^{5} = \{1, 2, 3.5, 5.25, 7\}$$

Continuing in the script

```
% sample the function
x = [1; 2; 3.5; 5.25; 7];
y = fe(x);
```

Select the Covariance function

We need to tell the library the form of the covariance function $K(x, x') = \sigma^2 \phi(\|x - x'\|)$.

- We will use the squared-exponential covariance
- We will set $\ln(l) = 0.0$ and $\ln(\sigma) = 0.0$.

```
    % set the squared exponential covariance function
    covfunc = @covSEiso;
    hyp.cov = [0; 0]; % hyp.cov= [log(l); log(sigma)]
```

Define the Likelihood Function

We discussed one likelihood function, but there are others

- We need to indicate which one we want
- We will use a Guassian likelihood with zero noise

```
    % set the likelihood function to Gaussian
    likfunc = @likGauss;
    sn = 1e-16; % this is the noise level
    hyp.lik = log(sn);
```

Find the Hyperparameters

Next, we maximize the likelihood to find suitable hyperparameters

```
    % minimize the negative log likelihood function
    hyp = minimize(hyp, @gp, -100, @infExact, ...
                   [], covfunc, likfunc, x, y);
```

Sample and Plot

Finally, we can sample the resulting GP and plot it.

```
    % now sample the GP
    z = linspace(0, 7.5, 125)';
    [m s2] = gp(hyp, @infExact, [], covfunc, ...
                likfunc, x, y, z);
    % ...and plot
    f = [m+1.96*sqrt(s2); ...
         flipdim(m-1.96*sqrt(s2),1)];
    fill([z; flipdim(z,1)], f, [7 7 7]/8)
    hold on;
    plot(z, m);
    plot(x, y, '+');
```