Gaussian process regression

Bernád Emőke

2007

Gaussian processes

Definition *A Gaussian Process is a collection of random variables, any finite number of which have (consistent) joint Gaussian distributions.*

A Gaussian process is fully specified by its mean function m(x) and covariance function k(x,x').

$$f \sim GP(m,k)$$

Generalization from distribution to process

- Consider the Gaussian process given by:

f ~ GP(m,k), $m(x) = \frac{1}{4}x^2$ *and* $k(x,x') = e^{-\frac{(x-x')^2}{2}}$

We can draw samples from the function *f* (vector x).

$$\mu(x_i) = \frac{1}{4}x_i^2 \qquad \Sigma(x_i, x_j) = e^{-\frac{(x_i - x_j)^2}{2}}, \quad i,j = 1,..,n$$

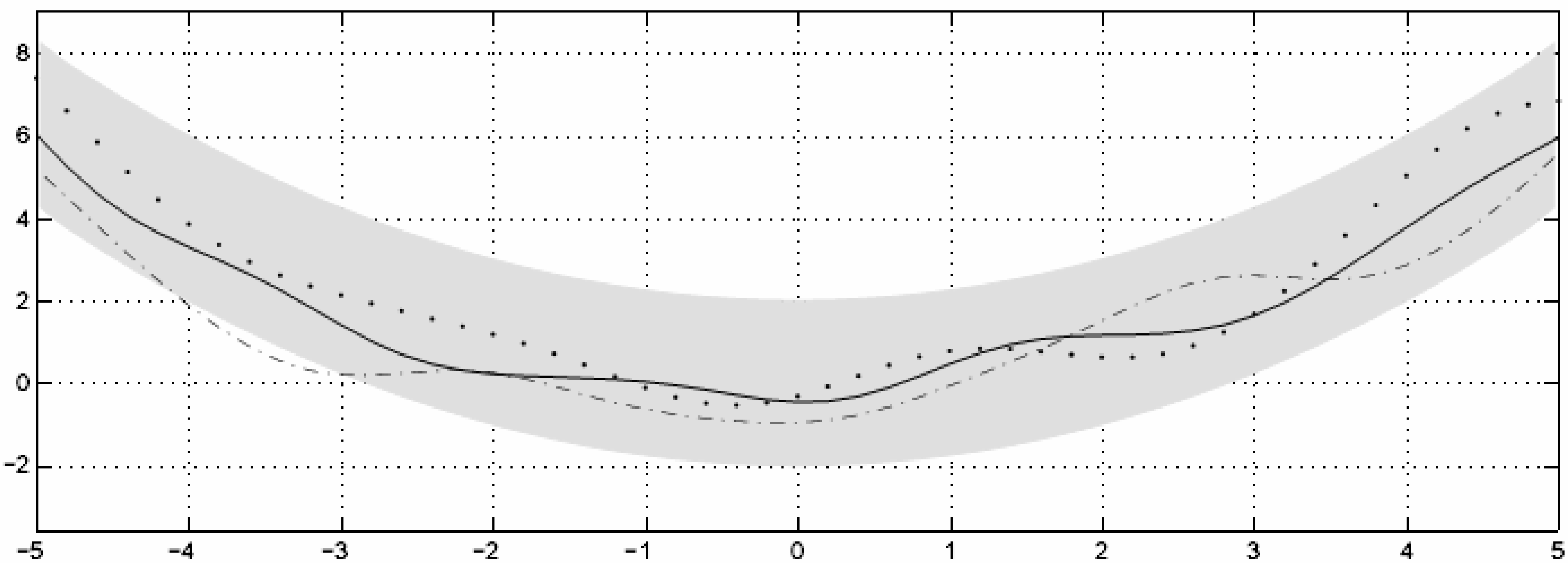
The algorithm

…

```
xs = (-5:0.2:5)';
ns = size(xs,1); keps = 1e-9;
% the mean function
m = inline('0.25*x.^2');
% the covariance function
K = inline('exp(-0.5*(repmat(p'',size(q))-repmat(q,size(p'')).^2)');
% the distribution function
fs = m(xs) + chol(K(xs,xs)+keps*eye(ns))'*randn(ns,1);
plot(xs,fs,'.')
```

…

The result



The dots are the values generated with algorithm, the two other curves have (less correctly) been drawn by connecting sampled points.

Posterior Gaussian Process

- The GP will be used as a prior for Bayesian inference.
- The primary goals computing the posterior is that it can be used to make predictions for unseen test cases.
- This is useful *if we have enough prior information* about a dataset at hand to confidently specify prior mean and covariance functions.
- Notations:

f : function values of training cases (x)

f* : function values of the test set (x')

$\mu = m(x_i)$: training means (m(x))

$\mu*$: test means

$\sum$: covariance (k(x,x'))

$\sum*$: training set covariance

$\sum**$: training-test set covariance

$$\begin{bmatrix} f \\ f^* \end{bmatrix} = N\left(\begin{bmatrix} \mu \\ \mu^* \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma^* \\ \Sigma^{*T} & \Sigma^{**} \end{bmatrix} \right)$$

Posterior Gaussian Process

- The formula for conditioning a joint Gaussian distribution is:

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}, \begin{bmatrix}A & C\\ C^\top & B\end{bmatrix}\right) \implies \mathbf{x}|\mathbf{y} \sim \mathcal{N}(a + CB^{-1}(\mathbf{y}-b),\ A - CB^{-1}C^\top).$$

- The conditional distribution:

$$f^* \mid f \sim N(\mu^* + \Sigma^{*T}\Sigma^{-1}(f-\mu), \Sigma^{**} - \Sigma^{*T}\Sigma^{-1}\Sigma^*)$$

- This is the posterior distribution for a specific set of test cases. It is easy to verify that the corresponding posterior process

$$F \mid D \sim GP(m_D, k_D) \quad m_D(x) = m(x) + \Sigma(X,x)^T\Sigma^{-1}(f-m)$$

$$k_D(x,x') = k(x,x') + \Sigma(X,x)^T\Sigma^{-1}\Sigma(X,x')$$

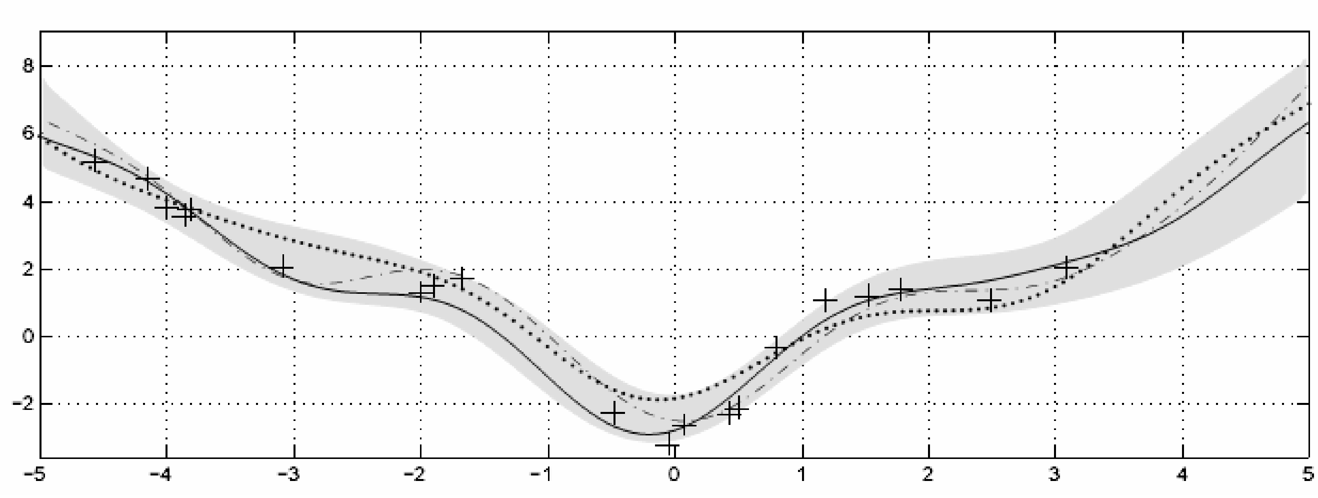
Where ∑(X,x) is a vector of covariances between every training case and x.

Gaussian noise in the training outputs

- Every *f(x)* has a extra covariance with itself only, with a magnitude equal to the noise variance:

$$y(x) = f(x) + \varepsilon\,, \qquad \varepsilon \sim N(0, \sigma_n^2)$$

$$f \sim GP(m, k)\,, \qquad y \sim GP(m, k + \sigma_n^2 \delta_{ii'})$$



20 training data
GP posterior
noise level 0,7

Training a Gaussian Process

- The mean and covariance functions are parameterized in terms of hyperparameters.
- For example: *f ~ GP(m,k),*

$$m(x) = ax^2 + bx + c$$

$$k(x,x') = \sigma_y^2 e^{-\frac{(x-x')^2}{2l^2}} + \sigma_n^2 \delta_{ii'}$$

- The hyperparameters: $\theta = \{a, b, c, \sigma_y, \sigma_n, l\}$
- The log marginal likelihood:

$$L = \log p(y \mid x, \theta) = -\frac{1}{2}\log|\Sigma| - \frac{1}{2}(y-\mu)^T \Sigma^{-1}(y-\mu) - \frac{n}{2}\log(2\pi)$$

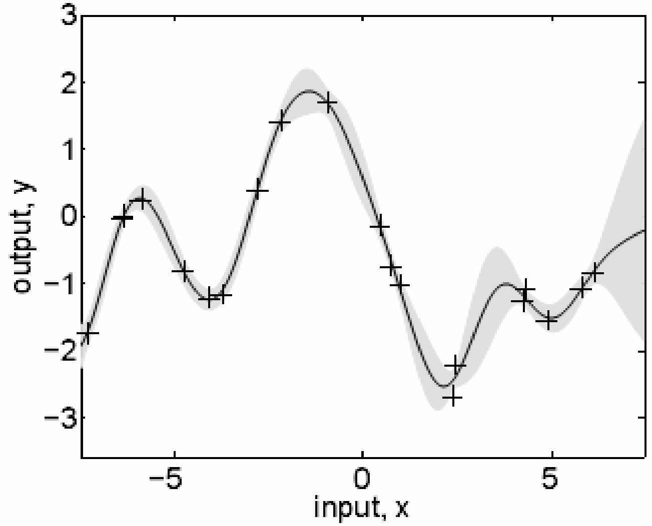
Optimizing the marginal likelihood

- Calculating the partial derivatives:

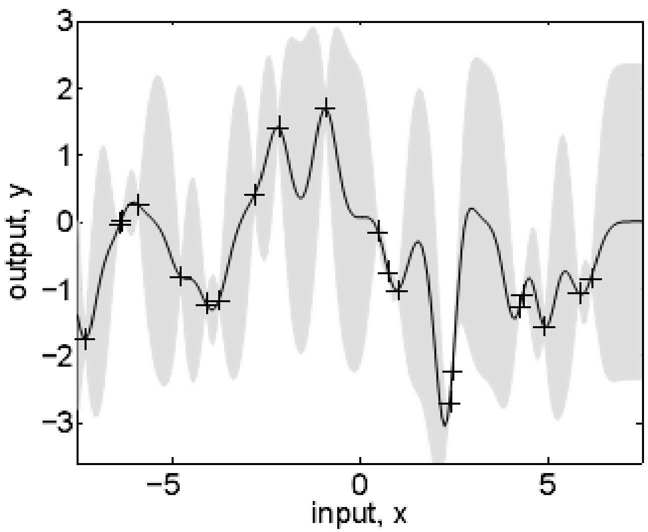
$$\frac{\delta L}{\delta \theta_m} = -(y-\mu)^T \Sigma^{-1} \frac{\delta m}{\delta \theta_m}$$

$$\frac{\delta L}{\delta \theta_k} = \frac{1}{2} trace\,(\Sigma^{-1} \frac{\delta \Sigma}{\delta \theta_k}) + \frac{1}{2}(y-\mu)^T \frac{\delta \Sigma}{\delta \theta_k} \Sigma^{-1} \frac{\delta \Sigma}{\delta \theta_k}(y-\mu)$$

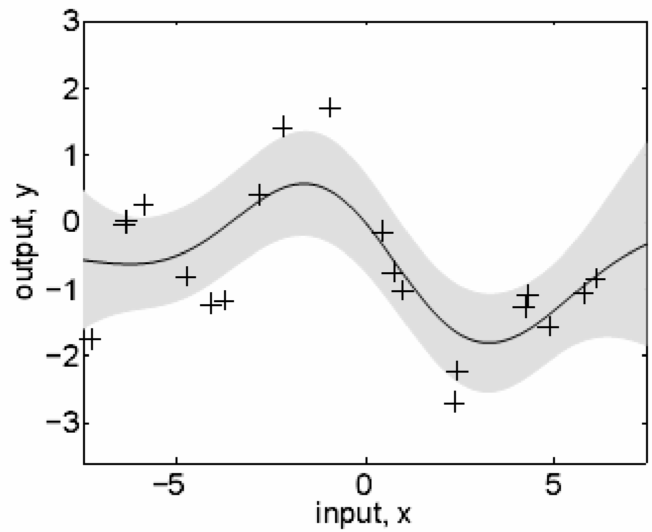
- With a numerical optimization routine conjugate gradients to find good hyperparameter settings.



(a), $\ell = 1$

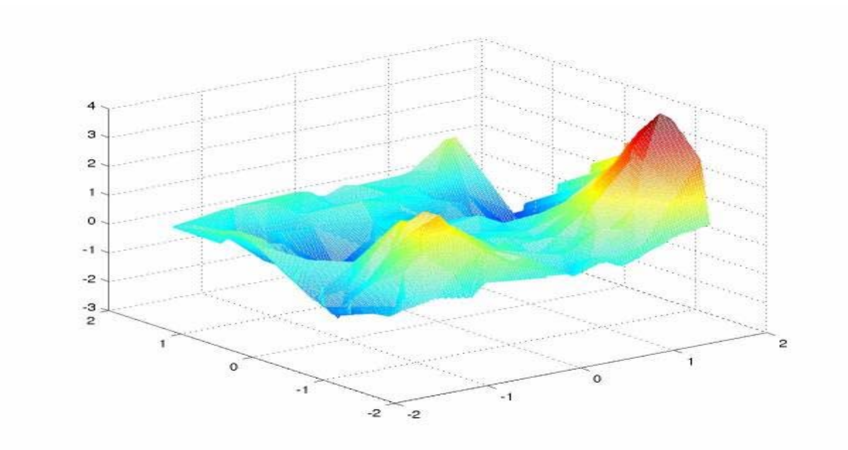


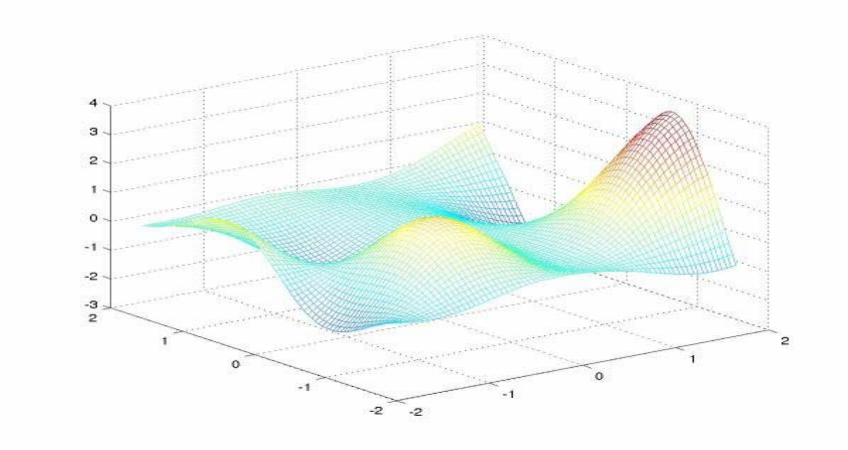
(b), $\ell = 0.3$



(c), $\ell = 3$

2-dimensional regression





- The training data has an unknown Gaussian noise and can be seen in the figure 1.
- in MLP network with Bayesian learning we needed 2500 samples
- With Gaussian Processes we needed only 350 samples to reach the "right" distribution
- The CPU time needed to sample the 350 samples on a 2400MHz Intel Pentium workstation was approximately 30 minutes.

References

- Carl Edward Rasmussen: Gaussian Processes in Machine Learning
- Carl Edward Rasmussen and Christopher K. I. Williams: Gaussian Processes for Machine Learning

  http://www.gaussianprocess.org/gpml/
- http://www.lce.hut.fi/research/mm/mcmcstuff/demo_2ingp.shtml