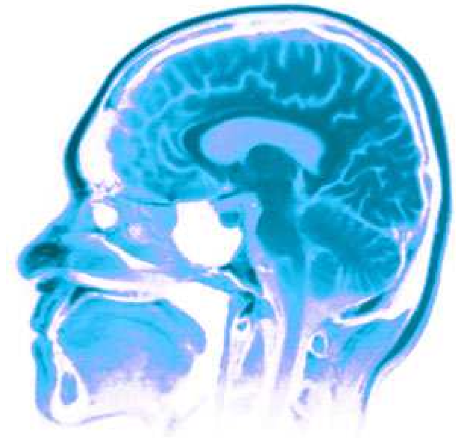
CPSC540



Gaussian Processes

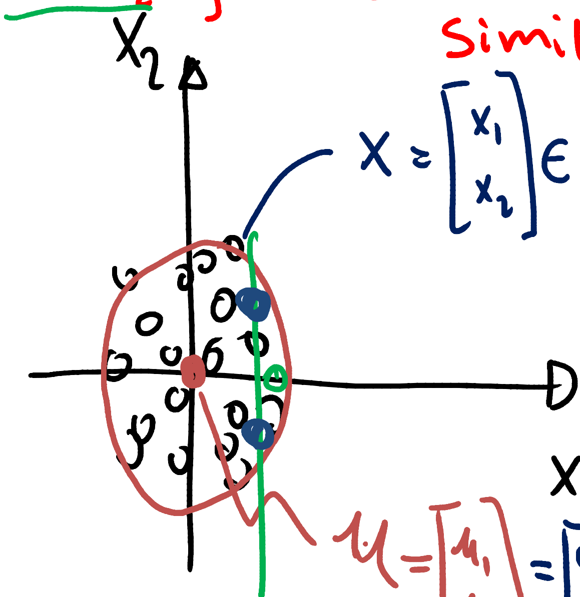
Nando de Freitas

January 2013

KPM Book Sections 4.3 and 15.2.

Gaussian basics

$$\left.\begin{array}{l}[1\ \ 0]\begin{bmatrix}1\\0\end{bmatrix} = 1\\[1\ \ 0]\begin{bmatrix}0\\1\end{bmatrix} = 0\end{array}\right\}\ \text{dot products measure similarity}$$

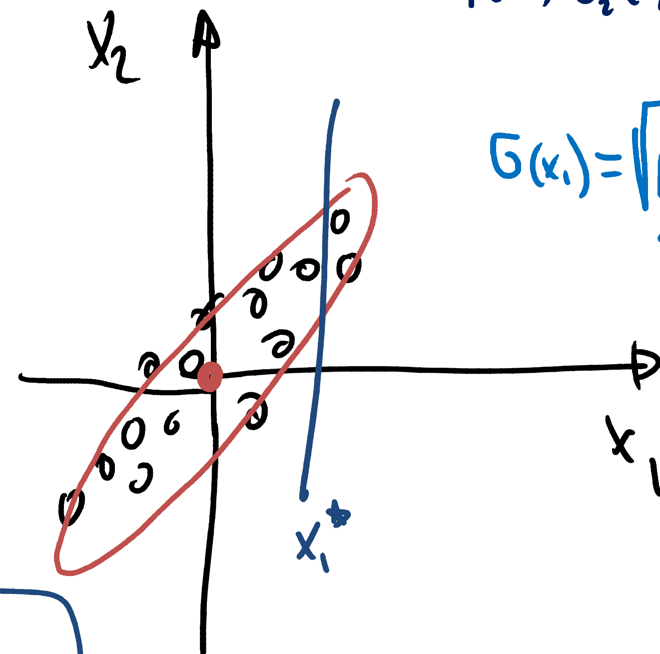


$$x = \begin{bmatrix}x_1\\x_2\end{bmatrix} \in \mathbb{R}^2$$

$$\mu = \begin{bmatrix}\mu_1\\\mu_2\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix}$$

$$\rho_{x_1x_2} = \frac{cov(x_1, x_2)}{\sigma_1(x_1)\,\sigma_2(x_2)}$$

$$\sigma(x_1) = \sqrt{\mathbb{E}(x_1^2)}$$



$\mathbb{E}(x_1x_2)$

$$\begin{bmatrix}x_1\\x_2\end{bmatrix} \sim N\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & \boxed{0}\\0 & 1\end{bmatrix}\right)$$

$$\begin{bmatrix}x_1\\x_2\end{bmatrix} \sim N\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0.5\\0.5 & 1\end{bmatrix}\right)$$

Gaussian basics

$x^T \Sigma x$

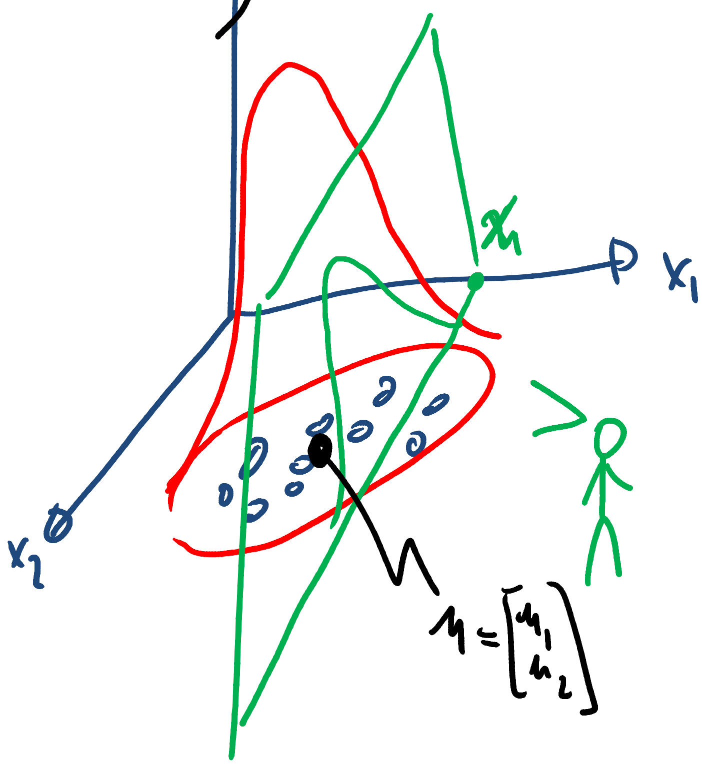
conditional

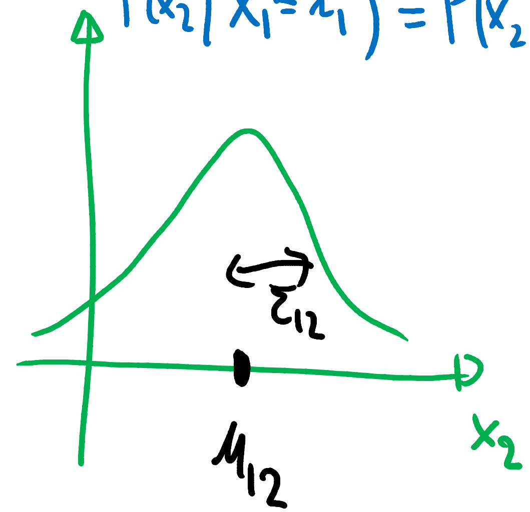
given

$$\begin{bmatrix}x_1\\x_2\end{bmatrix} \sim N\left(\begin{bmatrix}\mu_1\\\mu_2\end{bmatrix}, \begin{bmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{bmatrix}\right)$$

$P(x_1, x_2)$ joint

$P(x_2 \mid X_1 = x_1) = P(x_2 \mid x_1)$





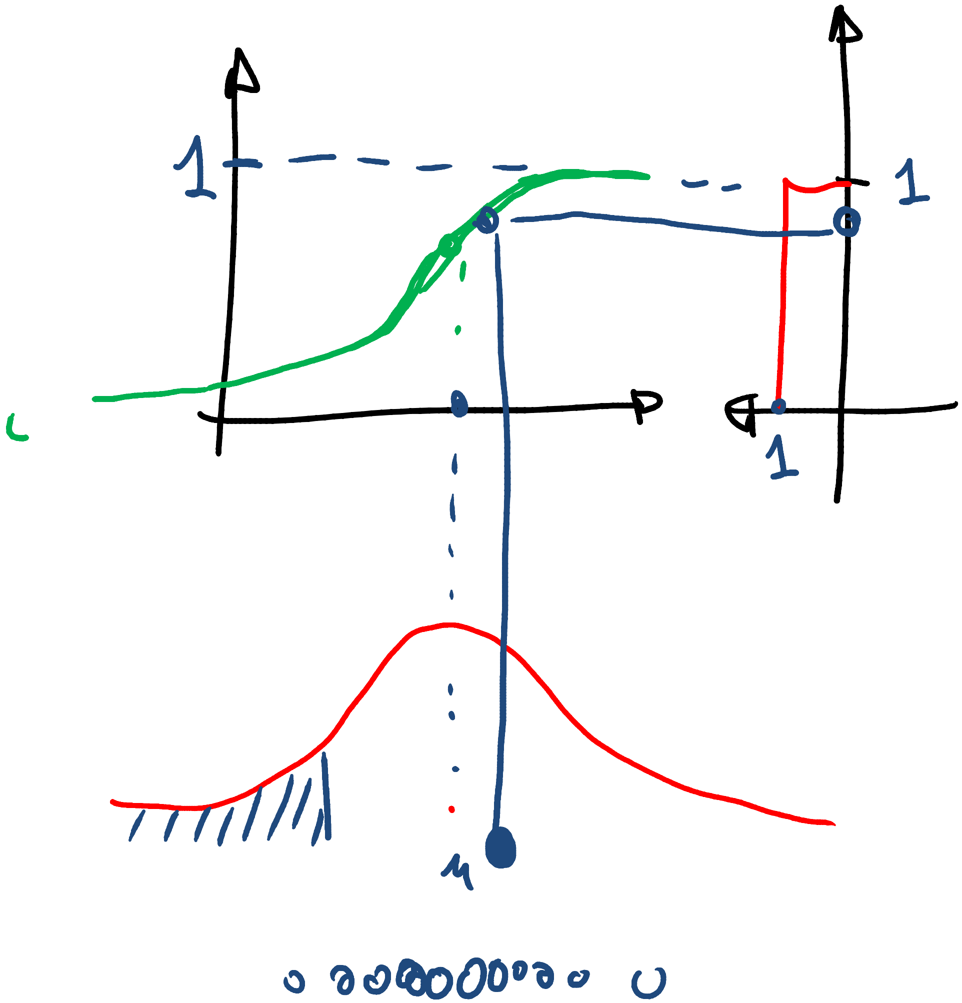
$$\mu_{12} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2)$$

$$\Sigma_{12} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

Gaussian basics

Cholesky

$\Sigma = LL^T$



$N(0,1)$

$$X_i \sim N(0,1)$$

$$X_i \sim N(\mu, \sigma^2)$$

$$\sim \mu + \sigma N(0,1)$$

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix}_i \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)$$

$$\underbrace{\begin{bmatrix} X_1 \\ X_2 \end{bmatrix}_i}_{\vec{x}} \sim N\left(\underbrace{\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}}_{\mu}, \underbrace{\begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}}_{\Sigma} \right)$$

$X \sim N(\mu, \Sigma)$ $\quad$ $X \sim \mu + L\,N(0, I)$

Multivariate Gaussian Theorem (see KPM)

Theorem 4.2.1 (Marginals and conditionals of an MVN)**.** *Suppose* $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ *is jointly Gaussian with parameters*

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}, \quad \boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1} = \begin{pmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{pmatrix} \tag{4.12}$$

Then the marginals are given by

$$\begin{aligned} p(\mathbf{x}_1) &= \mathcal{N}(\mathbf{x}_1 | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11}) \\ p(\mathbf{x}_2) &= \mathcal{N}(\mathbf{x}_2 | \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22}) \end{aligned}$$

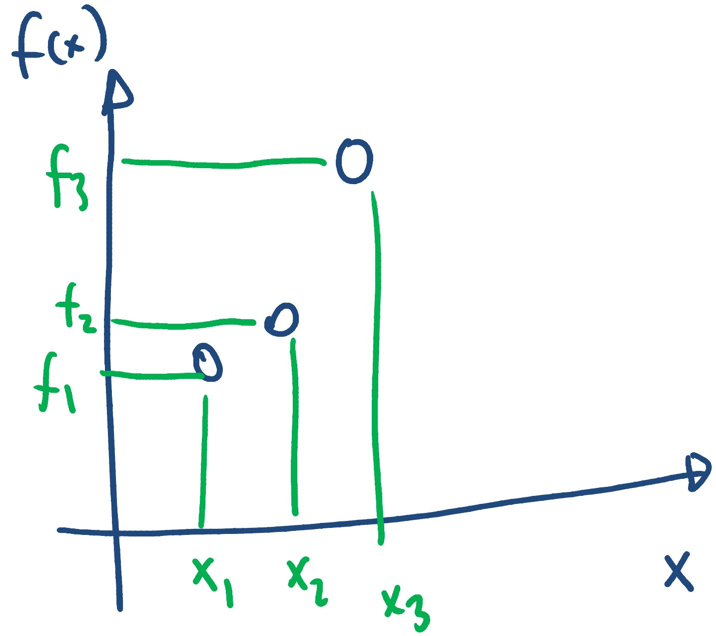
and the posterior conditional is given by

$$\boxed{\begin{aligned} p(\mathbf{x}_1|\mathbf{x}_2) &= \mathcal{N}(\mathbf{x}_1|\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \\ \boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \\ &= \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\Lambda}_{12}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \\ &= \boldsymbol{\Sigma}_{1|2}\left(\boldsymbol{\Lambda}_{11}\boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{12}(\mathbf{x}_2 - \boldsymbol{\mu}_2)\right) \\ \boldsymbol{\Sigma}_{1|2} &= \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} = \boldsymbol{\Lambda}_{11}^{-1} \end{aligned}}$$

Gaussian basics

x's given

want to model f's



$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \sim \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \right)$$

$$\sim \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \underbrace{\begin{bmatrix} 1 & 0.7 & 0.2 \\ 0.7 & 1 & 0.6 \\ 0.2 & 0.6 & 1 \end{bmatrix}}_{K} \right)$$

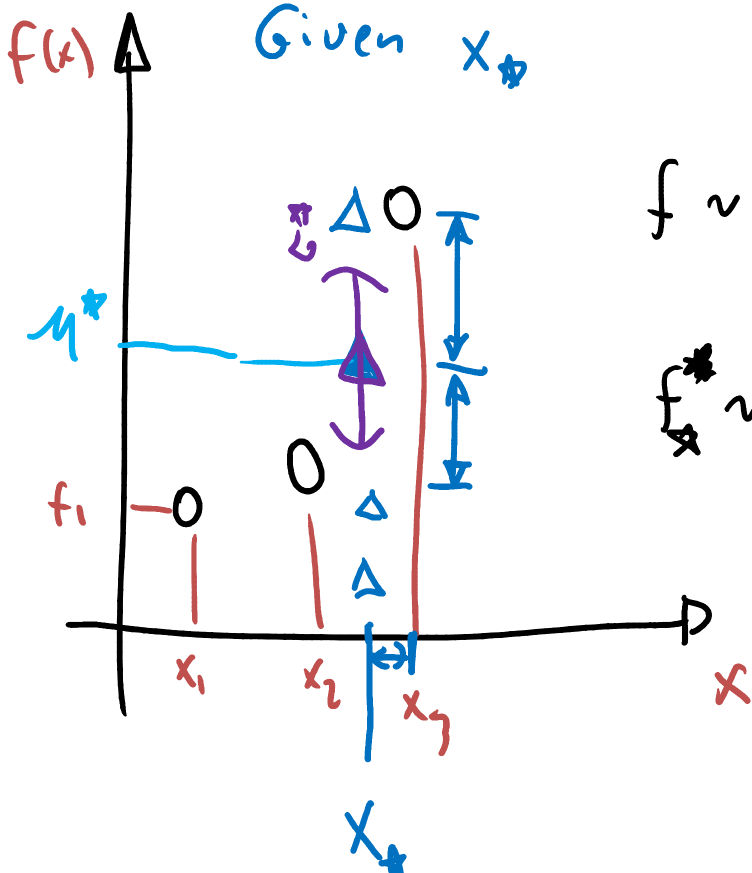
$$K_{ij} = e^{-\lambda \|x_i - x_j\|^2} = \begin{cases} 0 & \|x_i - x_j\| \to \infty \\ 1 & x_i = x_j \end{cases}$$

$$f \sim N(0, K)$$

Gaussian basics

Given Data $D = \{(x_1, f_1), (x_2, f_2), (x_3, f_3)\} \Rightarrow f_* = ?$

Given x_*



$$K(x_*, x_*) = e^{-\|x_* - x_*\|^2} = 1$$

$$f \sim N(0, K)$$

$$f^* \sim N(0, \underbrace{K(x_*, x_*)}_{K_{**}})$$

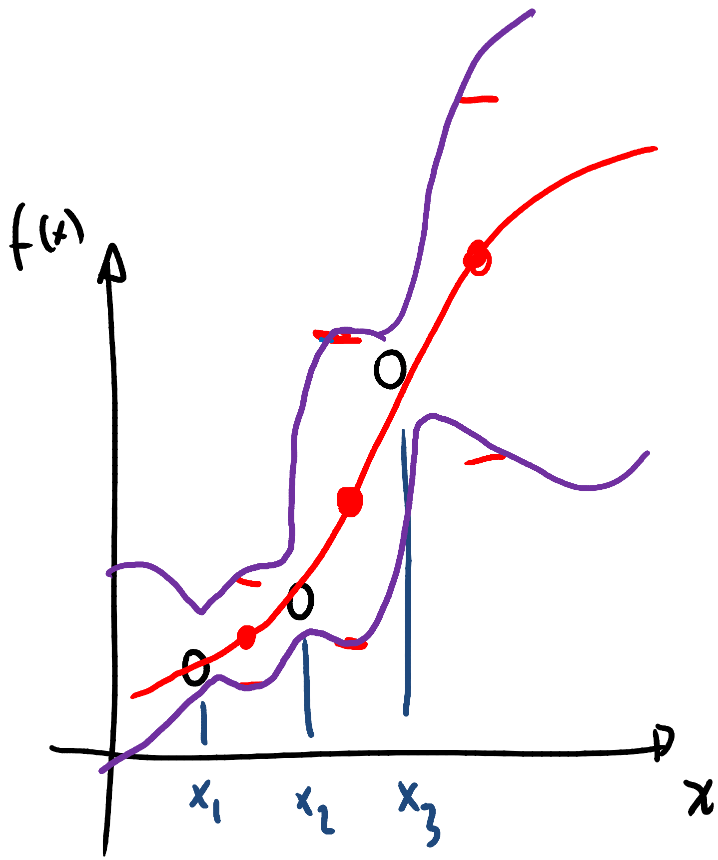
$$K_{1*} = K(x_1, x_*)$$

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} \overbrace{\begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix}}^{K} & \overbrace{\begin{bmatrix} K_{1*} \\ K_{2*} \\ K_{3*} \end{bmatrix}}^{K_*} \\ K_{*1} \quad K_{*2} \quad K_{*3} & \underbrace{K_{**}}_{K_{**}} \end{bmatrix}\right)$$

$$\mu^* = E(f^*) = K_*^T K^{-1} f$$

$$\sigma^* = -K_*^T K^{-1} K_* + K_{**}$$

Gaussian basics



GP: a distribution over functions

A GP is a Gaussian distribution over functions:

$$f(\mathbf{x}) \sim GP(\underline{m(\mathbf{x})}, \underline{\kappa(\mathbf{x}, \mathbf{x}')})$$

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}\left[f(\mathbf{x})\right] \\ \kappa(\mathbf{x}, \mathbf{x}') &= \mathbb{E}\left[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))^T\right] \end{aligned}$$

$$\boxed{k(x, x') = \exp(-\tfrac{1}{2}(x - x')^2)}$$

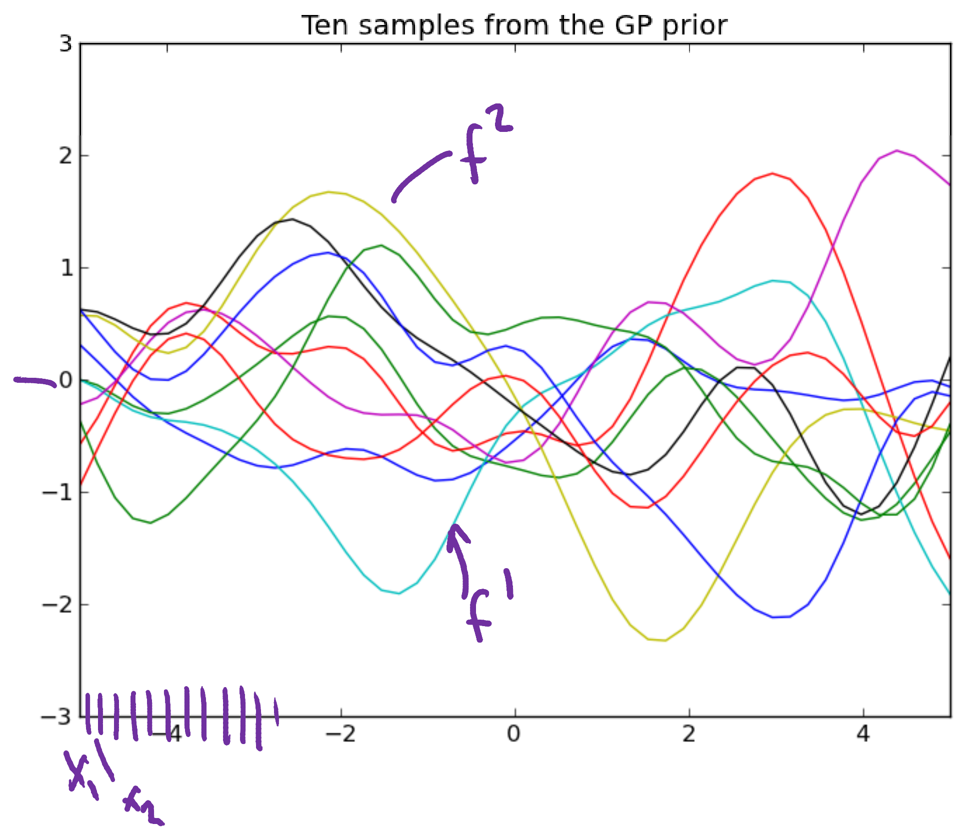
Create $x_{1:N}$

Create $\mu = 0_N$, $K_{N \times N}$

$K = L L^T$

$f^i \sim N(0_N, K)$

$\sim N(0, I) L$



Sampling from $P(f)$

```
from __future__ import division
import numpy as np
import matplotlib.pyplot as pl

def kernel(a, b):
    """ GP squared exponential kernel """
    sqdist = np.sum(a**2,1).reshape(-1,1) + np.sum(b**2,1) - 2*np.dot(a, b.T)
    return np.exp(-.5 * sqdist)

n = 50                                              # number of test points.
Xtest = np.linspace(-5, 5, n).reshape(-1,1)  # Test points.
K_ = kernel(Xtest, Xtest)                           # Kernel at test points.

# draw samples from the prior at our test points.
L = np.linalg.cholesky(K_ + 1e-6*np.eye(n))
f_prior = np.dot(L, np.random.normal(size=(n,10)))

pl.plot(Xtest, f_prior)
```

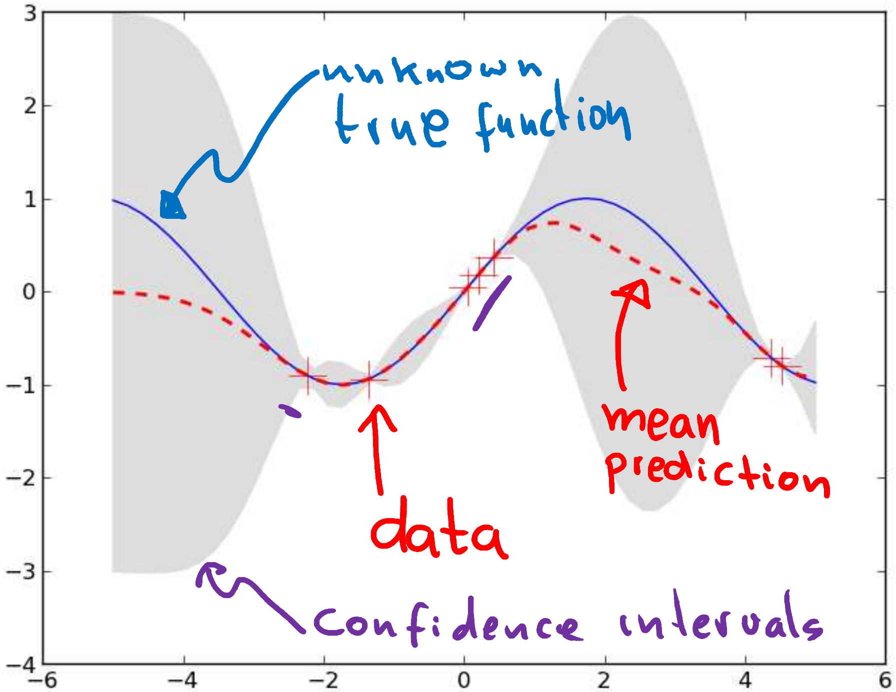
$K = LL^T$

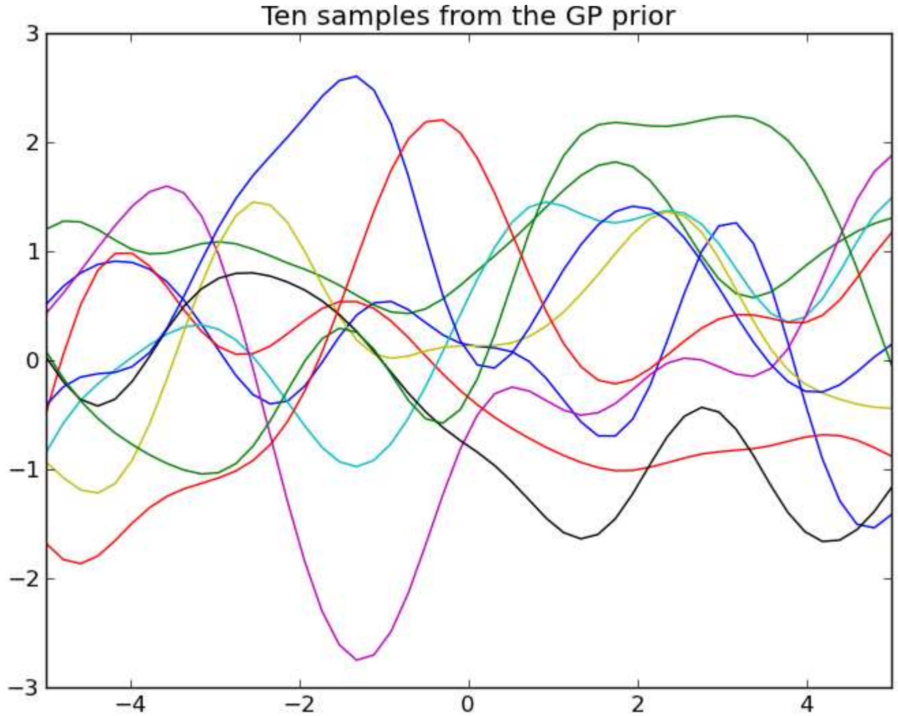
$L\mathcal{N}(0, I)$

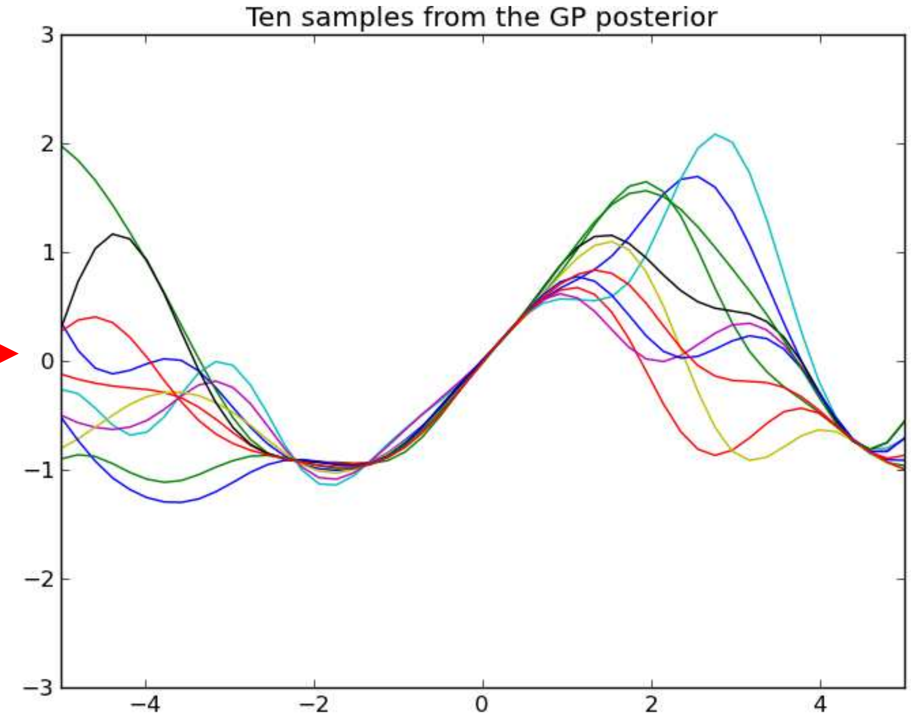
GP posterior

$$\mathcal{D} = \{(\mathbf{x}_i, f_i), i = 1 : N\}$$

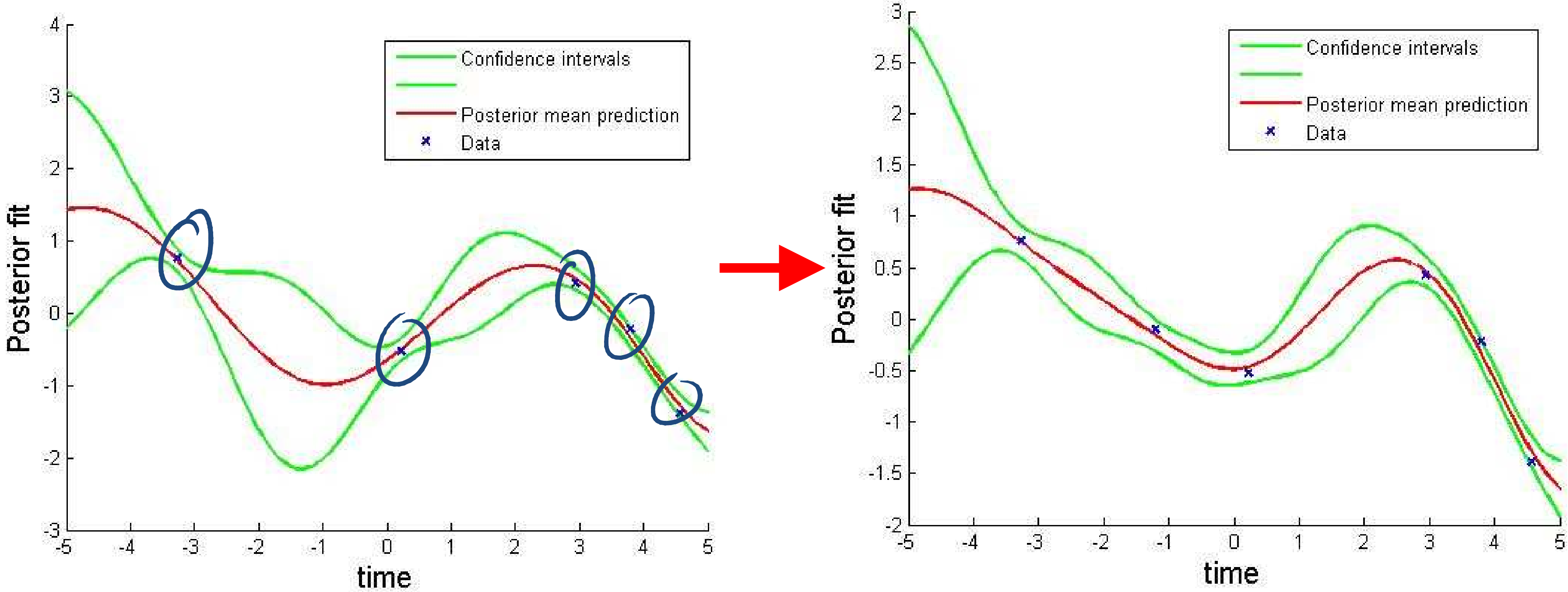
$$p(f|\mathcal{D}) = \frac{p(\mathcal{D}|f)p(f)}{p(\mathcal{D})}$$







Active learning with GPs



Noiseless GP regression

we observe a training set $\mathcal{D} = \{(\mathbf{x}_i, f_i), i = 1 : N\}$, where $f_i = f(\mathbf{x}_i)$

Given a test set $\mathbf{X}_*$ of size $N_* \times D$, we want to predict the function outputs $\mathbf{f}_*$.

$$\begin{pmatrix} \mathbf{f} \\ \mathbf{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{pmatrix}\right)$$

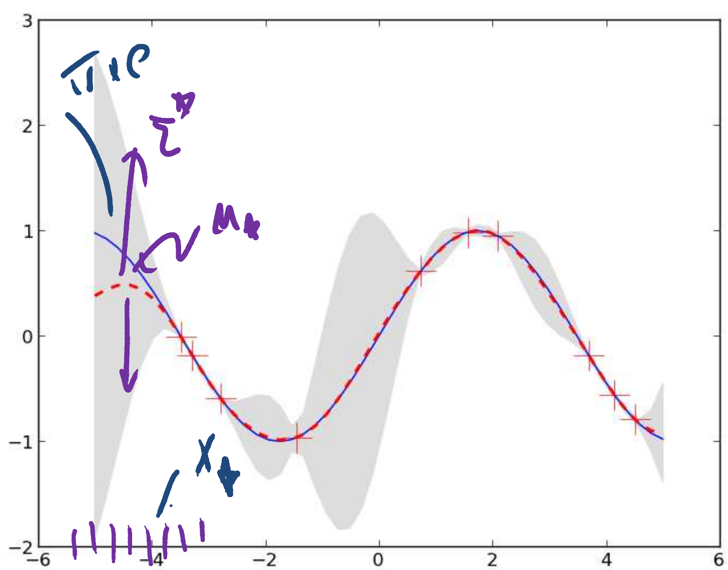
where $\mathbf{K} = \kappa(\mathbf{X}, \mathbf{X})$ is $N \times N$, $\mathbf{K}_* = \kappa(\mathbf{X}, \mathbf{X}_*)$ is $N \times N_*$, and $\mathbf{K}_{**} = \kappa(\mathbf{X}_*, \mathbf{X}_*)$ is $N_* \times N_*$.

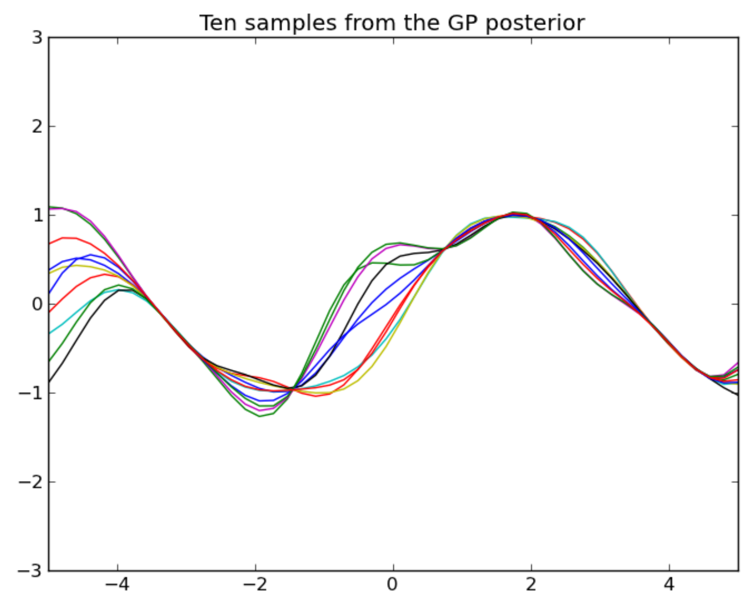
$$\boxed{\kappa(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}(x - x')^2\right)}$$

Noiseless GP regression

$$\begin{pmatrix} \mathbf{f} \\ \mathbf{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{pmatrix} \right)$$

$$\begin{aligned} p(\mathbf{f}_* | \mathbf{X}_*, \mathbf{X}, \mathbf{f}) &= \mathcal{N}(\mathbf{f}_* | \boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*) \\ \boldsymbol{\mu}_* &= \boldsymbol{\mu}(\mathbf{X}_*) + \mathbf{K}_*^T \mathbf{K}^{-1}(\mathbf{f} - \boldsymbol{\mu}(\mathbf{X})) \\ \boldsymbol{\Sigma}_* &= \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{K}_* \end{aligned}$$



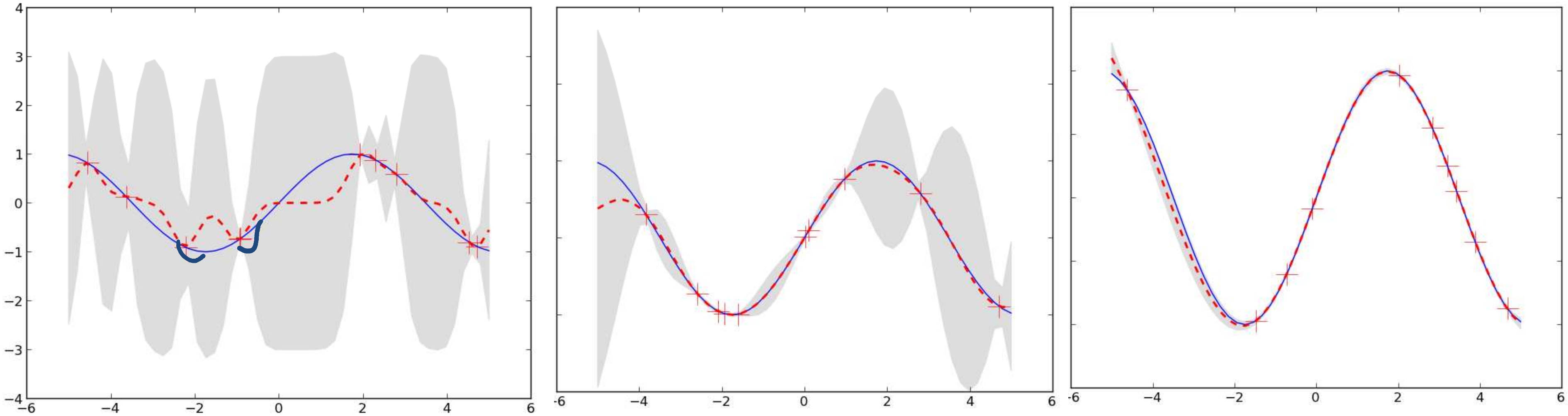


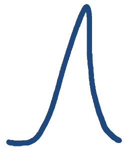
Effect of kernel width parameter

$$\kappa(x, x') = \sigma_f^2 \exp(-\frac{1}{2\ell^2}(x - x')^2)$$

Let $\sigma_f = 1$

$\ell^2 = 0.1$ $\quad$ $\ell^2 = 1$ $\quad$ $\ell^2 = 10$





Noisy GP regression

noisy

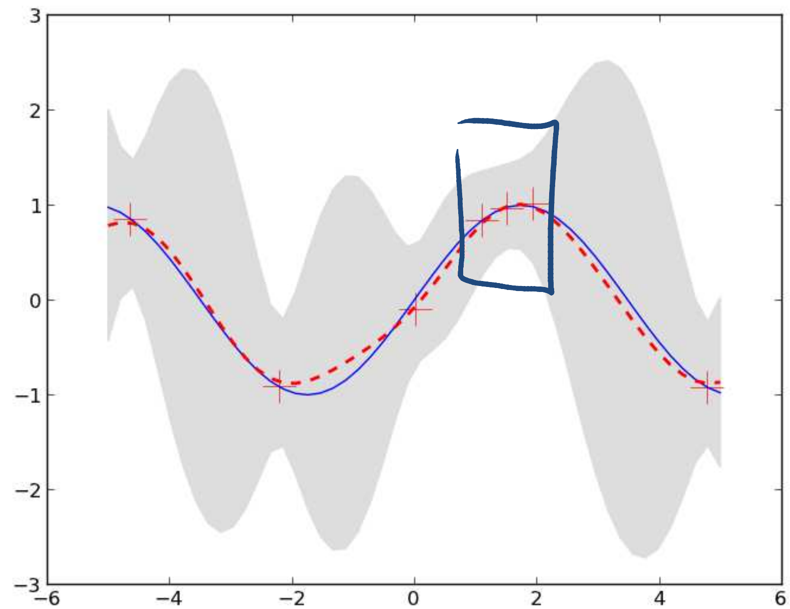
$$y = f(\mathbf{x}) + \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, \sigma_y^2)$$

$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|\mathbf{f}, \mathbf{X}) p(\mathbf{f}|\mathbf{X}) d\mathbf{f}$$

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})$$

$$p(\mathbf{y}|\mathbf{f}) = \prod_i \mathcal{N}(y_i|f_i, \sigma_y^2)$$

$$\text{cov}\left[\mathbf{y}|\mathbf{X}\right] = \mathbf{K} + \boxed{\sigma_y^2 \mathbf{I}_N} \triangleq \mathbf{K}_y$$



$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}_* \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} \mathbf{K}_y & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{pmatrix}\right) \xrightarrow{\text{thm}}$$

$$\begin{aligned} p(\mathbf{f}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y}) &= \mathcal{N}(\mathbf{f}_*|\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*) \\ \boldsymbol{\mu}_* &= \mathbf{K}_*^T \mathbf{K}_y^{-1} \mathbf{y} \\ \boldsymbol{\Sigma}_* &= \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}_y^{-1} \mathbf{K}_* \end{aligned}$$

Noisy GP regression

In the case of a single test input, $\mathbf{x}_*$ this simplifies as follows

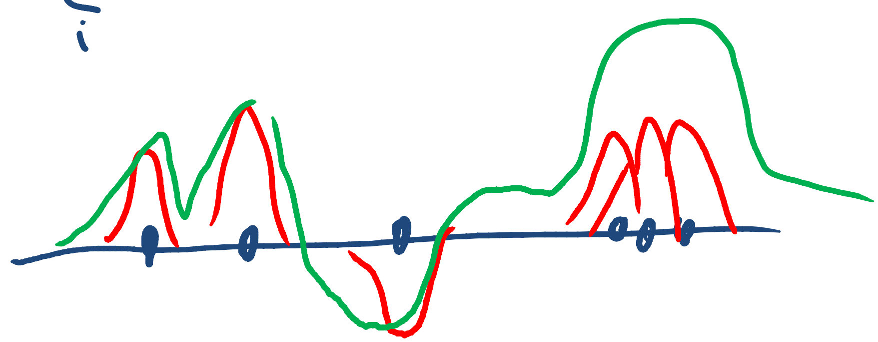
$$p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \boxed{\mathcal{N}(f_*|\mathbf{k}_*^T \mathbf{K}_y^{-1}\mathbf{y}, k_{**} - \mathbf{k}_*^T \mathbf{K}_y^{-1}\mathbf{k}_*)}$$

where $\mathbf{k}_* = [\kappa(\mathbf{x}_*, \mathbf{x}_1), \ldots, \kappa(\mathbf{x}_*, \mathbf{x}_N)]$ and $\underline{k_{**} = \kappa(\mathbf{x}_*, \mathbf{x}_*)} = 1$.

$$\overline{f}_* = \underbrace{\overset{1\times N}{\mathbf{k}_*^T}\,\overbrace{\overset{N\times N}{\mathbf{K}_y^{-1}}\,\overset{N\times 1}{\mathbf{y}}}^{\alpha\ (N\times 1)}}_{\substack{\text{mean}\\ 1\times 1}} = \sum_{i=1}^{N} \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}_*) \qquad \boldsymbol{\alpha} = \underbrace{\mathbf{K}_y^{-1}\mathbf{y}}_{\text{training}}$$

$$= k_*^T \alpha$$

$$= \sum_i \alpha_i e^{-\frac{1}{\lambda}\|x_i - x_*\|^2}$$



Noisy GP regression and Ridge

$$\mathbf{X} \in \mathbb{R}^{n\times d}$$

$$\mathbf{y} \in \mathbb{R}^{n}$$

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \delta^2\|\boldsymbol{\theta}\|_2^2$$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

$$x_i \in \mathbb{R}^{1\times d}$$

$$\underline{(\mathbf{X}^T\mathbf{X} + \delta^2\mathbf{I}_d)\boldsymbol{\theta} = \mathbf{X}^T\mathbf{y}}$$

solution can be written as $\boxed{\boldsymbol{\theta} = \mathbf{X}^T\boldsymbol{\alpha}}$, where $\underline{\boldsymbol{\alpha} = \delta^{-2}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})}$

$$X^TX\theta + \delta^2\theta = X^Ty$$

$$\delta^2\theta = X^T(y - X\theta)$$

$$\theta = X^T\delta^{-2}(y - X\theta) = X^T\alpha$$

Noisy GP regression and Ridge

$$(\underbrace{\mathbf{X}^T\mathbf{X}}_{d\times d} + \delta^2\mathbf{I}_d)\underbrace{\boldsymbol{\theta}}_{d\times 1} = \mathbf{X}^T\mathbf{y}$$

solution can be written as $\boldsymbol{\theta} = \mathbf{X}^T\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = \delta^{-2}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})$

$\boldsymbol{\alpha}$ can also be written as follows: $\underbrace{\boldsymbol{\alpha}}_{n\times 1} = (\underbrace{\mathbf{X}\mathbf{X}^T}_{n\times n} + \delta^2\mathbf{I}_n)^{-1}\mathbf{y}$

$$\delta^2\alpha = y - X\theta$$

$$\delta^2\alpha = y - XX^T\alpha$$

$$XX^T\alpha + \delta^2 I_n\alpha = y$$

$$\alpha = \boxed{(XX^T + \delta^2 I_n)^{-1} y}$$

$$y^* = x^*\theta = x^* X^T \alpha$$

Noisy GP regression and Ridge

$$\overline{f}_* = \mathbf{k}_*^T \mathbf{K}_y^{-1} \mathbf{y}$$

$$\underset{1\times1}{Y^*} = \underset{1\times d}{x^*} \underset{d\times1}{\Theta}$$

$$= x^* X^T \alpha$$

$$= \underbrace{x^* X^T}_{K_*^T} \underbrace{\left[XX^T + \overset{G_y}{\delta^2} I_n\right]^{-1}}_{K_y^{-1}} Y$$

$$= K_*^T K_y^{-1} Y$$

$$X = \begin{bmatrix} \overset{1\times d}{x_1} \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$K_*^T = \underbrace{\begin{bmatrix} x^* x_1^T & x^* x_2^T & \cdots & x^* x_n^T \end{bmatrix}}_{1\times n}$$

$$K_y = \underbrace{\underset{n\times d}{X}\underset{d\times n}{X^T}}_{n\times n} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \begin{bmatrix} x_1^T & \cdots & x_n^T \end{bmatrix} = \begin{bmatrix} x_1 x_1^T & \cdots \\ & \ddots \\ x_n x_1^T & \cdots \end{bmatrix}$$

d | d

Learning the kernel parameters

marginal likelihood

$$p(\mathbf{y}|\mathbf{X},\ell^2) = \int p(\mathbf{y}|\mathbf{f},\mathbf{X})p(\mathbf{f}|\mathbf{X})d\mathbf{f}$$

$$p(\mathbf{f}|\mathbf{X},\ell^2) = \mathcal{N}(\mathbf{f}|\mathbf{0},\mathbf{K})$$

$\theta = \ell$

$$p(\mathbf{y}|\mathbf{f},\ell^2) = \prod_i \mathcal{N}(y_i|f_i,\sigma_y^2)$$

$$\log p(\mathbf{y}|\mathbf{X}) = \log\mathcal{N}(\mathbf{y}|\mathbf{0},\mathbf{K}_y) = -\frac{1}{2}\mathbf{y}\mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_y| - \frac{N}{2}\log(2\pi)$$

$$\frac{\partial}{\partial\theta_j}\log p(\mathbf{y}|\mathbf{X}) = \frac{1}{2}\mathbf{y}^T\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j}\mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\mathrm{tr}(\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j})$$

Numerical computation considerations

$$m_* = \overline{f_*} = \mathbf{k}_*^T \underbrace{\mathbf{K}_y^{-1}\mathbf{y}}_{\alpha}$$

$$\mathbf{K}_y = \mathbf{L}\mathbf{L}^T$$

$$\boldsymbol{\alpha} = \mathbf{K}_y^{-1}\mathbf{y} = \mathbf{L}^{-T}\underbrace{\mathbf{L}^{-1}\mathbf{y}}_{m}$$

$m = L^{-1} y$

$Lm = y$

$\underbrace{L^{-T} m}_{\alpha}$

$L^T \alpha = m$

Algorithm 15.1: GP regression

1. $\mathbf{L} = \text{cholesky}(\mathbf{K} + \sigma_y^2 \mathbf{I})$;
2. $\boldsymbol{\alpha} = \mathbf{L}^T \setminus \underbrace{(\mathbf{L} \setminus \mathbf{y})}_{m}$;
3. $\mathbb{E}[f_*] = \mathbf{k}_*^T \boldsymbol{\alpha}$;
4. $\mathbf{v} = \mathbf{L} \setminus \mathbf{k}_*$;
5. $\text{var}[f_*] = \kappa(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{v}^T \mathbf{v}$;
6. $\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^T\boldsymbol{\alpha} - \sum_i \log L_{ii} - \frac{N}{2}\log(2\pi)$

Next lecture

In the next lecture, we capitalize on GPs to introduce active learning, Bayesian optimization and GP bandits.

For a recent article on bandits and Thompson sampling at work at Google, see:

http://analytics.blogspot.ca/2013/01/multi-armed-bandit-experiments.html

For an article on Bayesian optimization, see:

http://arxiv.org/abs/1012.2599