Gaussian Processes

Instructor: Hemanth Venkateswara
Computer Science & Engineering

Arizona State University

Overview

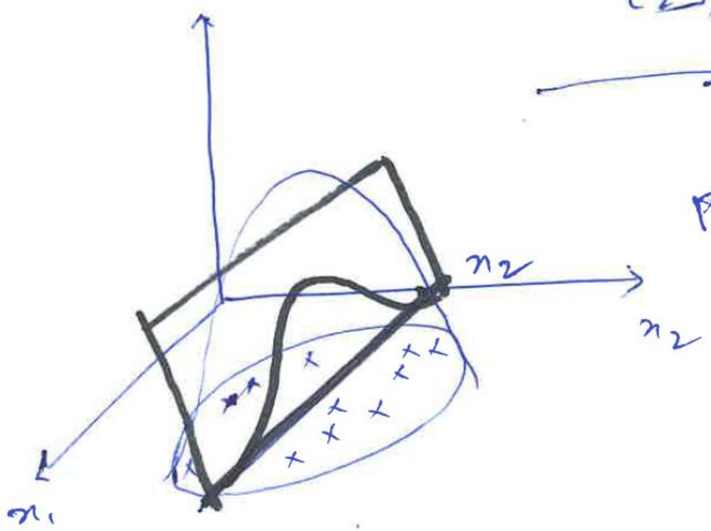
Introduction to Gaussian Processes

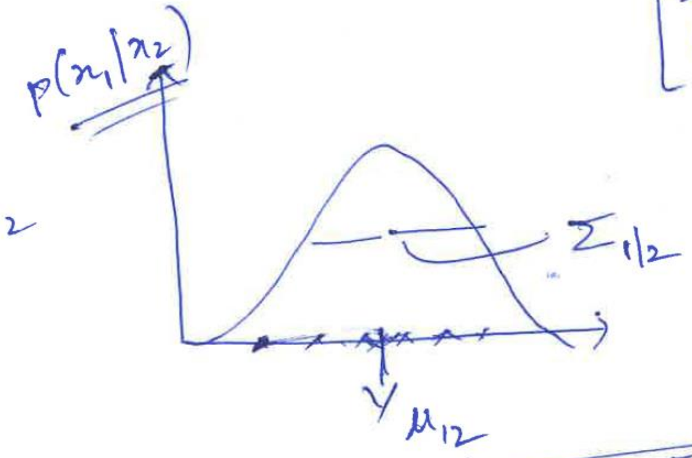
- Gaussian Conditionals
- Sampling from a Gaussian Distribution
- Gaussian Processes for Regression

Gaussian Conditionals

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

$$\begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix}\begin{array}{l}\to k\\ \to d-k\end{array} \qquad \begin{array}{c c c} & k & d-k \\ k & \Sigma_{11} & \Sigma_{12} \\ d-k & \Sigma_{21} & \Sigma_{22} \end{array}$$





$$\boxed{\begin{aligned}\mu_{1|2} &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2-\mu_2)\\ \Sigma_{1|2} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\end{aligned}}$$

Gaussian Conditionals

Theorem 4.3.1 (Marginals and conditionals of an MVN). *Suppose* $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ *is jointly Gaussian with parameters*

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}, \quad \boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1} = \begin{pmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{pmatrix} \tag{4.67}$$

Then the marginals are given by

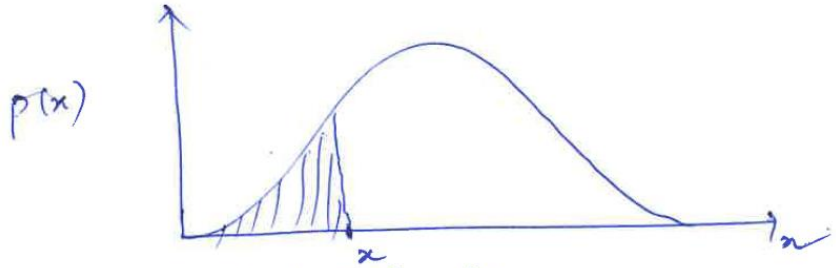
$$\begin{aligned} p(\mathbf{x}_1) &= \mathcal{N}(\mathbf{x}_1 | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11}) \\ p(\mathbf{x}_2) &= \mathcal{N}(\mathbf{x}_2 | \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22}) \end{aligned} \tag{4.68}$$

and the posterior conditional is given by

$$\boxed{\begin{aligned} p(\mathbf{x}_1 | \mathbf{x}_2) &= \mathcal{N}(\mathbf{x}_1 | \boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \\ \boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12} \boldsymbol{\Sigma}_{22}^{-1} (\mathbf{x}_2 - \boldsymbol{\mu}_2) \\ &= \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{11}^{-1} \boldsymbol{\Lambda}_{12} (\mathbf{x}_2 - \boldsymbol{\mu}_2) \\ &= \boldsymbol{\Sigma}_{1|2} \left(\boldsymbol{\Lambda}_{11} \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{12} (\mathbf{x}_2 - \boldsymbol{\mu}_2) \right) \\ \boldsymbol{\Sigma}_{1|2} &= \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12} \boldsymbol{\Sigma}_{22}^{-1} \boldsymbol{\Sigma}_{21} = \boldsymbol{\Lambda}_{11}^{-1} \end{aligned}} \tag{4.69}$$

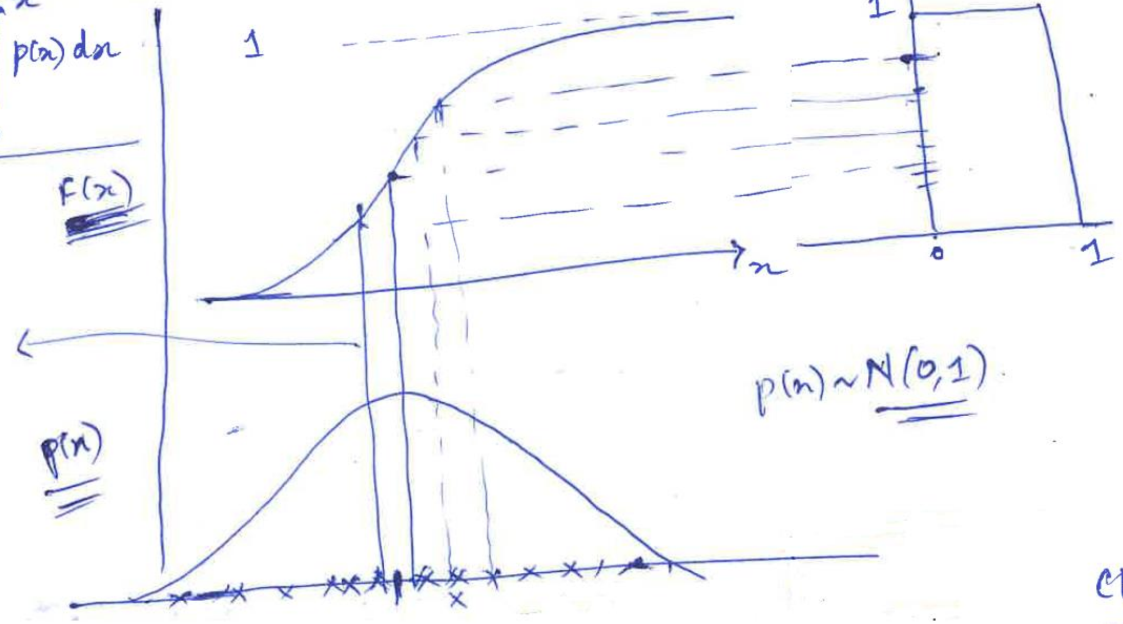
Chap 4: Kevin Murphy: Machine Learning – a Probabilistic Perspective

Sampling from a Gaussian Distribution



Cumulative Distribution Function.

$$F(x) = \int_{-\infty}^{x} p(x)\,dx$$



$F^{-1}(x)$

inverse CDF

$x \sim N(\mu, \sigma^2)$.

$$\boxed{p(x) \quad \underline{\mu} + \sigma N(0,1) \sim N(\mu, \sigma^2)}$$

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

$p(x_1) \sim N(0,1)$ $\quad p(x_2) \sim N(0,1)$

$$p\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}\right) = N\left(\begin{pmatrix}\mu_1\\\mu_2\end{pmatrix}, \begin{bmatrix}\sigma_1^2&0\\0&\sigma_2^2\end{bmatrix}\right)$$

$p(x_1) = N(\mu_1, \sigma_1^2)$ $\quad p(x_2) \sim N(\mu_2, \sigma_2^2)$

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

sq root

$$p\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}\right) \sim \mu + \underbrace{L}_{\text{sqroot}(\Sigma)} N\left(\begin{bmatrix}0\\0\end{bmatrix}, I\right)$$

Cholesky Decomposition

$$\boxed{LL^T = \Sigma}$$

Additional Sources – Gaussian Processes

Richard Turner, Univ. of Cambridge:
https://www.youtube.com/watch?v=92-98SYOdlY

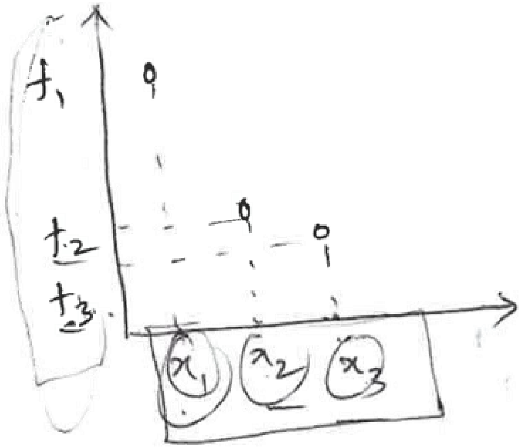
Slides: http://cbl.eng.cam.ac.uk/pub/Public/Turner/News/imperial-gp-tutorial.pdf

Nando de Freitas, Univ. of British Columbia

https://www.youtube.com/watch?v=4vGiHC35j9s

https://www.youtube.com/watch?v=MfHKW5z-OOA

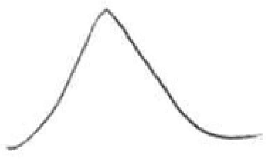
Gaussian Processes



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

$$\begin{bmatrix} 1 & 0.7 & 0.5 \\ 0.7 & 1 & 0.4 \\ 0.5 & 0.4 & 1 \end{bmatrix} \rightarrow k(x_1, x_3).$$

RBF



$l_2 \rightarrow$ horizontal width

$\sigma_f \rightarrow$ vertical width

Kernel function

$$\boxed{K_{ij} = \sigma_f^2 \exp \left\| \frac{-1}{2l^2} \; x - x' \right\|_2^2}$$

$K_{ij} = 0$ if $\|x - x'\|^2 \rightarrow \infty$.

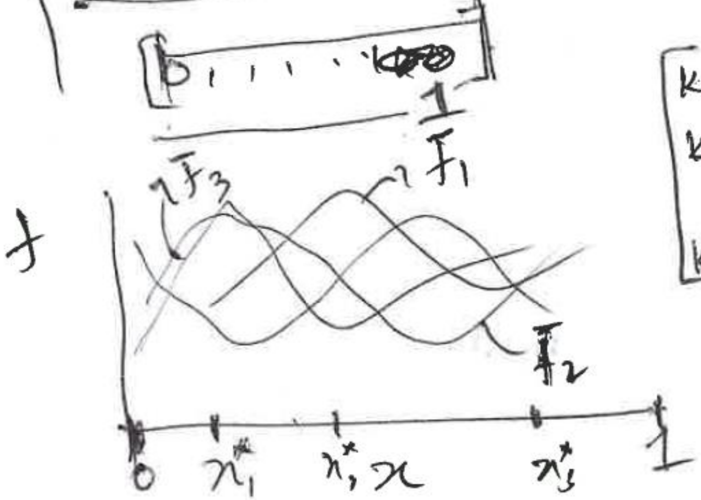
$K_{ij} = 1$ if $\|x - x'\|^2 \rightarrow 0$

Kernel

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

Gaussian Processes Prior

$\underline{X_{1:N}}$ k_{ij} for every pair $k(x_i, x_j)$



$$\begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \cdots & k(x_1,x_n) \\ k(x_2,x_1) & \cdots & & k(x_2,x_n) \\ \vdots & & & \vdots \\ k(x_n,x_1) & \cdots & & k(x_n,x_n) \end{bmatrix}$$

$\bar{K}$

$$\exp\left\{-\tfrac{1}{2}|x - x'|^2\right\}$$

$$\begin{array}{cc} 100 & \begin{bmatrix} 1 & 0.9 & 0.8 & \cdots \\ 0.9 & 1 & & \vdots \\ \vdots & & 1 & \vdots \\ 0.01 & \cdots & \cdots & 1 \end{bmatrix} \\ & 100 \end{array}$$

$$f_{1:10} = \mathcal{N}\left([0]_{100}, [\bar{K}]\right).$$

$X_{1:N}$

$$\begin{bmatrix} 0 \\ 0.01 \\ 0.02 \\ 0.1 \\ \vdots \\ 1.0 \end{bmatrix}$$

$k_{ij} = \exp\{|x-x'|^2\}$

Ker function

$k(0, 0.01)$

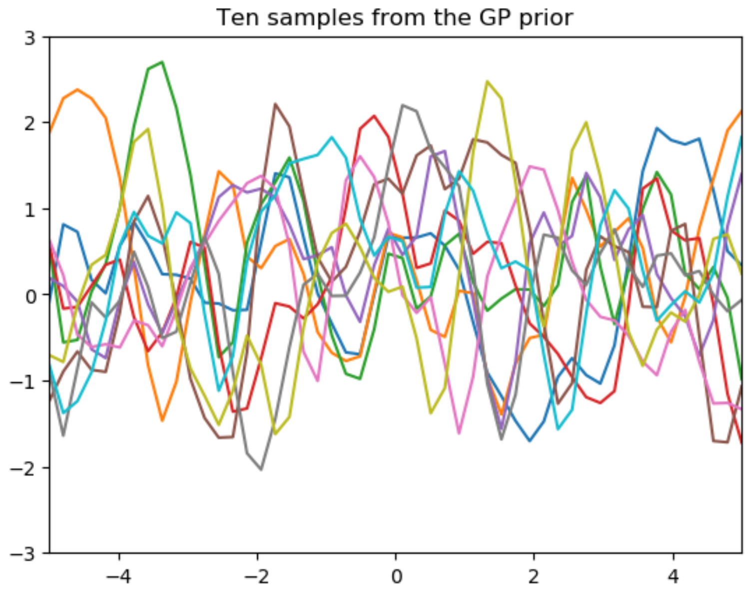
$k(0, 0.02)$

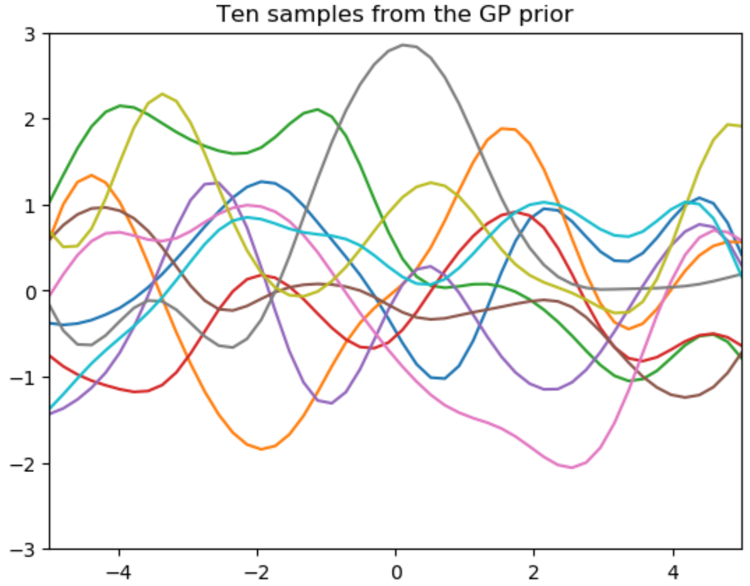
$k(0, 0.03)$

$K_{1,100} = k(0, 1)$

Gaussian Processes Prior

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

def kernel(a, b):
kernelParameter = 0.1
sqdist = np.sum(a**2,1).reshape(-1,1) + np.sum(b**2,1) - 2*np.dot(a, b.T)
return np.exp(-.5 * (1/kernelParameter) * sqdist)


n = 50
Xtest = np.linspace(-5, 5, n).reshape(-1,1)
K_ = kernel(Xtest, Xtest)


# 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)
```





Code Source: Nando de Freitas: https://www.youtube.com/watch?v=4vGiHC35j9s

Gaussian Processes: A Distribution Over Functions

In this section, we discuss GPs for regression. Let the prior on the regression function be a GP, denoted by

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}')) \tag{15.2}$$

where $m(\mathbf{x})$ is the mean function and $\kappa(\mathbf{x}, \mathbf{x}')$ is the kernel or covariance function, i.e.,

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] & (15.3) \\ \kappa(\mathbf{x}, \mathbf{x}') &= \mathbb{E}\left[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))^T\right] & (15.4) \end{aligned}$$

We obviously require that $\kappa()$ be a positive definite kernel. For any finite set of points, this process defines a joint Gaussian:

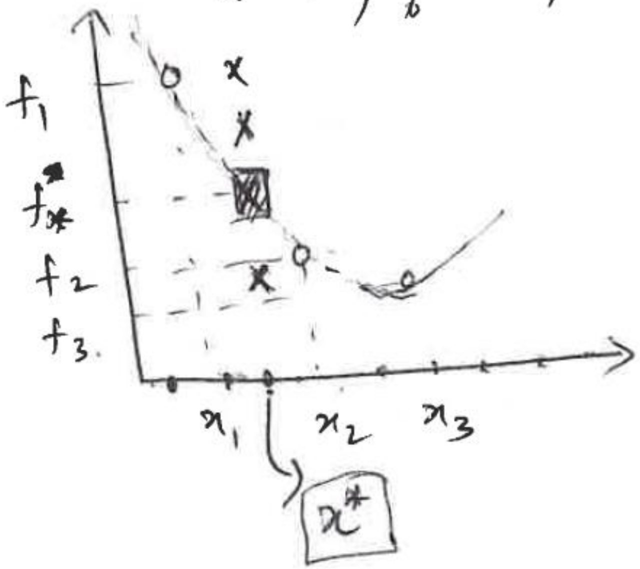
$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}, \mathbf{K}) \tag{15.5}$$

where $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ and $\boldsymbol{\mu} = (m(\mathbf{x}_1), \ldots, m(\mathbf{x}_N))$.

Chap 15: Kevin Murphy: Machine Learning – a Probabilistic Perspective

Prediction of f_* for sample x_*

$x \quad (x_1, f_1) \quad (x_2, f_2) \quad (x_3, f_3)$



$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_* \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{1*} \\ k_{21} & k_{22} & k_{23} & k_{2*} \\ k_{31} & k_{32} & k_{33} & k_{3*} \\ k_{*1} & k_{*2} & k_{*3} & k_{**} \end{bmatrix} \right)$$

$K \qquad K_* \qquad K_*^T \qquad K_{**}$

$$\begin{bmatrix} K_{1*} & K_{2*} & K_{3*} \end{bmatrix}$$

$$p\left(f_* \mid [f_1, f_2, f_3]\right)$$

Gaussian Prior with zero mean

The joint density of the observed data and the latent, noise-free function on the test points is given by

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

where we are assuming the mean is zero, for notational simplicity. Hence the posterior predictive density is

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

In the case of a single test input, this simplifies as follows

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

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

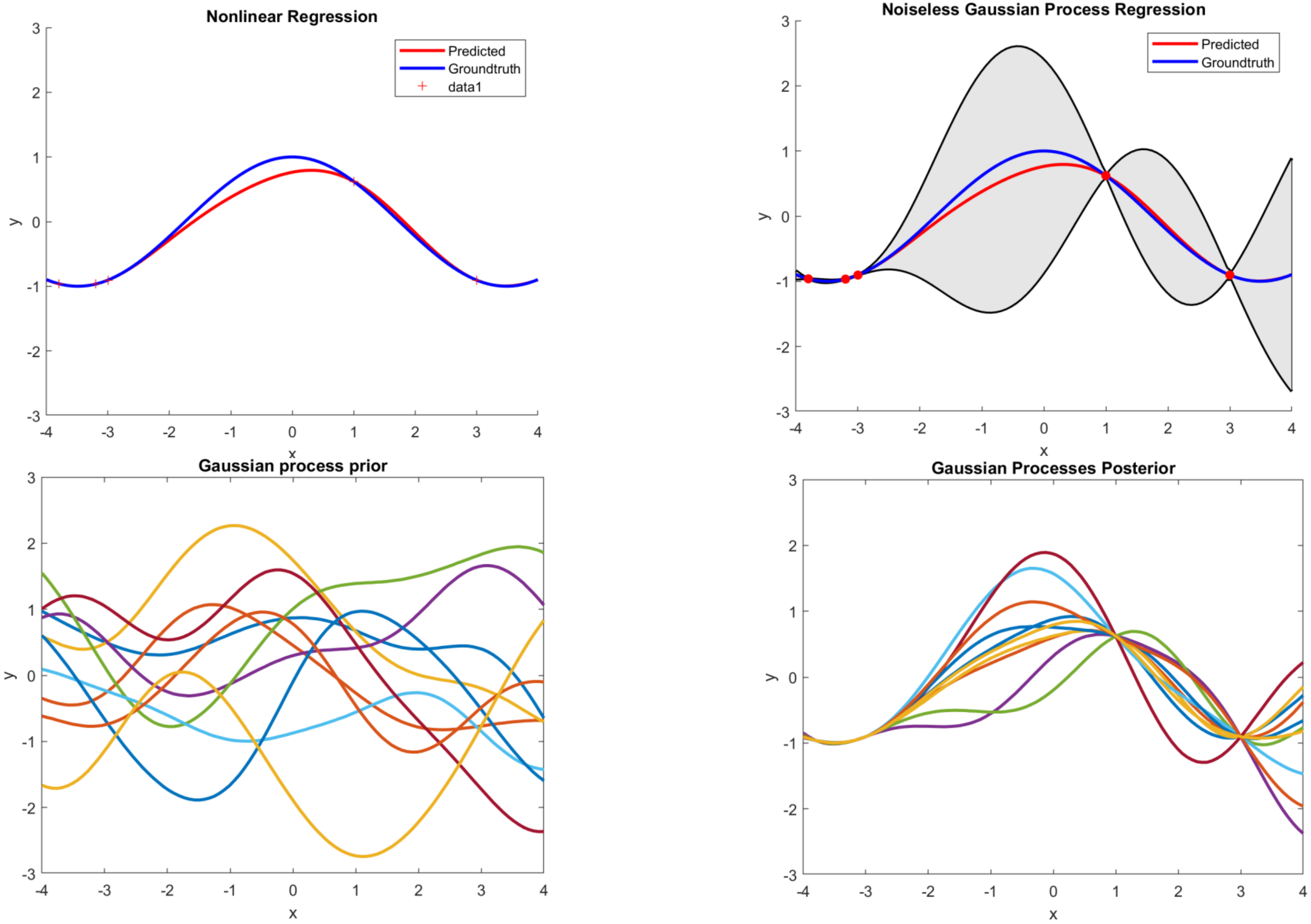
Chap 15: Kevin Murphy: Machine Learning – a Probabilistic Perspective

Gaussian Processes Nonlinear Regression

Gaussian Processes Regression:
Probabilistic Interpretation

$$D = \{(x_i, f_i), i = 1{:}N\}$$

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



GP Algorithm – Zero Mean Prior

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

$$\boldsymbol{\alpha} = \mathbf{K}^{-1}\mathbf{y} = \mathbf{L}^{-T}\mathbf{L}^{-1}\mathbf{y}$$

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

Solve for m in System of Linear Equations $Lm = y$
This is more stable than estimating $L^{-1}y$
Similarly, solve for α in system of linear equations $L^T\alpha = m$
This is more stable than estimating $L^{-T}m$

Algorithm 15.1: GP regression

1 $\mathbf{L} = \text{cholesky}(\mathbf{K} + \sigma_y^2\mathbf{I})$;
2 $\boldsymbol{\alpha} = \mathbf{L}^T \setminus (\mathbf{L} \setminus \mathbf{y})$;
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 $\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^T\boldsymbol{\alpha} - \sum_i \log L_{ii} - \frac{N}{2}\log(2\pi)$

Chap 15: Kevin Murphy: Machine Learning – a Probabilistic Perspective