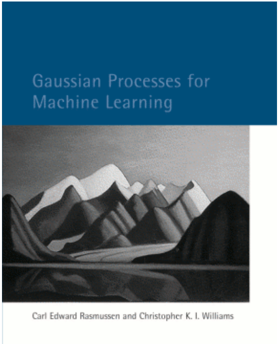
Learning with Gaussian Processes using GPy

Nishanth Koganti

April 3, 2017

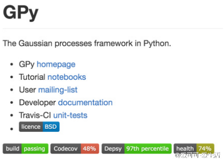
Outline

Resources



Gaussian Processes for Machine Learning [1]

Gaussian Process Summer School [2]



GPy Library [3]

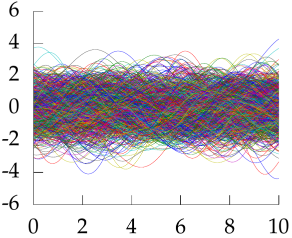
[1] Gaussian Processes for Machine Learning, C. Williams, C. Rasmussen

[2] Gaussian Process Summer Schools, `http://gpss.cc/`

[3] GPy Library, `https://github.com/SheffieldML/GPy`

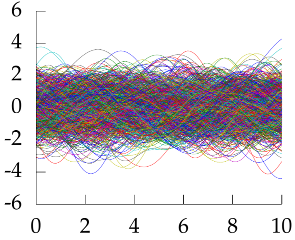
Gaussian Processes: Extremely Short Overview

Generate functions

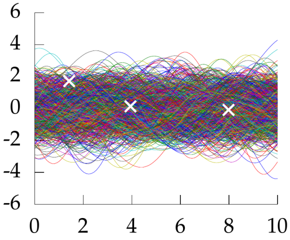


Gaussian Processes: Extremely Short Overview

Generate functions

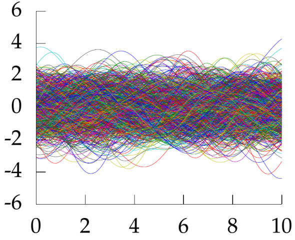


Observe Data

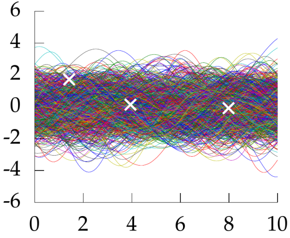


Gaussian Processes: Extremely Short Overview

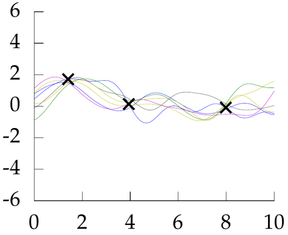
Generate functions



Observe Data



Remove invalid functions



What is a Gaussian Process?

Generalization of a multivariate Gaussian to infinitely many variables.

Definition: *Gaussian Process is a collection of random variables, any finite collection of which are Gaussian Distributed.*

Gaussian distribution: mean vector, $\boldsymbol{\mu}$, and covariance matrix $\boldsymbol{\Sigma}$:

$$\mathbf{f} = (f_1, \ldots, f_n)^T \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \text{indices } i = 1, \ldots, n$$

Gaussian process: mean function, $m(x)$, and covariance function $k(x, x')$:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')), \quad \text{indices: } x$$

Marginalization Property

How can we represent infinite mean vector and infinite covariance matrix?

...luckily saved by *marginalization property*:

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{y}) d\mathbf{y}$$

For Gaussians:

$$p(\mathbf{x}, \mathbf{y}) = \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}, \begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}^T & \mathbf{C}\end{bmatrix}\right)$$

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{a}, \mathbf{A})$$

Random sampling from Gaussian Process

Considering one dimensional Gaussian process:

$$p(f(x)) \sim \mathcal{GP}\left(m(x) = 0, k(x, x') = \exp\left(-\frac{1}{2}(x - x')^2\right)\right)$$

Sampling is done by focusing on subset $\mathbf{f} = (f(x_1), f(x_2), \ldots, f(x_n))^T$:

$$\mathbf{f} \sim \mathcal{N}(0, \mathbf{\Sigma}), \text{ where } \mathbf{\Sigma}_{ij} = k(x_i, x_j)$$

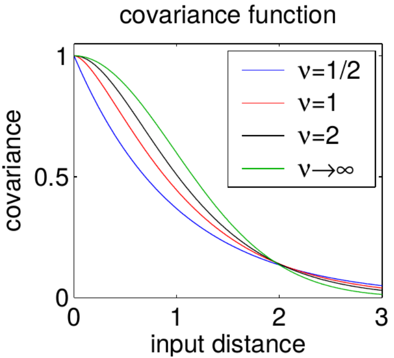
Coordinates of $\mathbf{f}$ are plot as a function of corresponding x

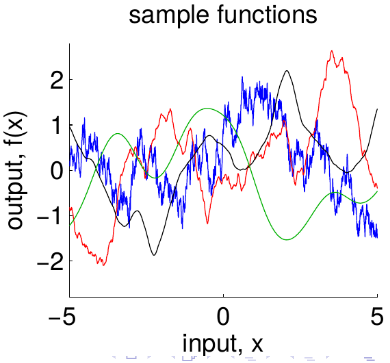
Matern Covariance Function

$$k(x, x') = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left[\frac{\sqrt{2\nu}}{l} |x - x'| \right]^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu}l}{|} x - x'| \right)$$

where K_ν is a Bessel function of order ν, and l is the length scale.

Samples of Matern forms are $\lfloor \nu - 1 \rfloor$ times differentiable.

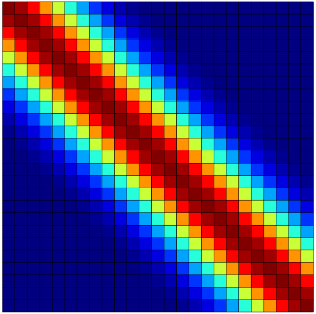


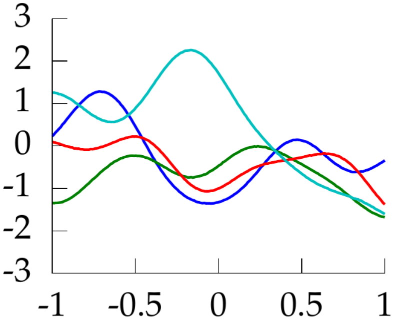


Squared Exponential Covariance Function

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2}\right)$$

where α is the variance and l is the length scale of the covariance function





Gaussian Process Regression

Parameters are replaced by "function" itself!
Gaussian Likelihood:

$$\mathbf{y}|\mathbf{x}, f(x), M \sim \mathcal{N}(\mathbf{f}, \sigma^2_{noise} I)$$

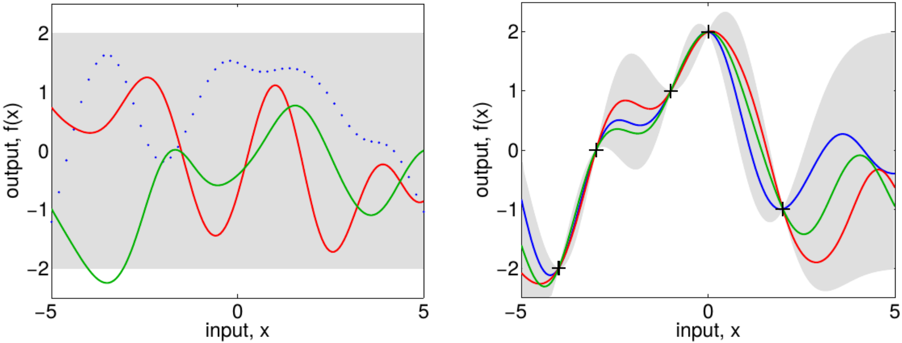
Gaussian Process Prior:

$$f(x)|M \sim \mathcal{GP}(m(x) = 0, k(x, x'))$$

Leading to Gaussian Process Posterior:

$$f(x)|\mathbf{x}, \mathbf{y}, M \sim \mathcal{GP}(m_{\text{post}}(x) = k(x, \mathbf{x})[K(\mathbf{x}, \mathbf{x}) + \sigma^2_{noise} I]^{-1}\mathbf{y},$$
$$k_{\text{post}}(x, x') = k(x, x') - k(x, \mathbf{x})[K(\mathbf{x}, \mathbf{x}) + \sigma^2_{noise} I]^{-1} k(\mathbf{x}, x'))$$

Prior and Posterior for $\mathcal{GP}$ Learning



Gaussian Process Predictive Distribution:

$$p(y^*|x^*, \mathbf{x}, \mathbf{y}) \sim \mathcal{N}(k(x^*, \mathbf{x})[K + \sigma^2_{noise}]^{-1}\mathbf{y},$$
$$k(x*, x*) - k(x^*, \mathbf{x})[K + \sigma^2_{noise}I]^{-1}k(\mathbf{x}, x^*))$$

Non-linear Dimensionality Reduction

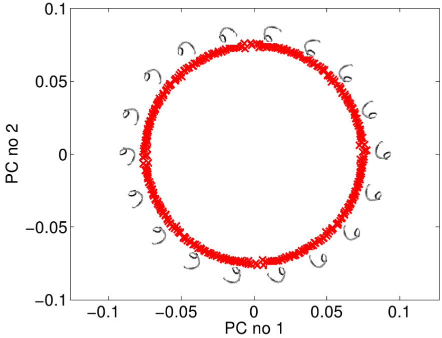
UPSC Handwritten Digit Dataset

3648 dimensional space

Digit 6 Image

Random Image

Low-dimensional manifold for digit rotation



Probabilistic Generative Model

- Observed (high-dimensional) data: $\mathbf{Y} = [y_1 \ y_2 \ \cdots \ y_N]^T \in \mathbb{R}^{N \times D}$
- Latent (low-dimensional) data: $\mathbf{X} = [x_1 \ x_2 \ \cdots \ x_N]^T \in \mathbb{R}^{N \times Q}, \ Q << D$
- Assume a relationship/mapping of the form:

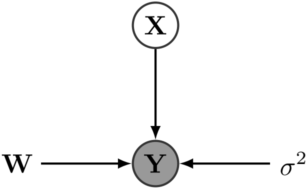
$$
\begin{gathered}
y_i = \mathbf{W} x_i + \epsilon_i, \ \epsilon_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}) \\
y_i = f(x_i) = \epsilon_i
\end{gathered} \tag{1}
$$

- Resultant likelihood on the data:

$$
p(\mathbf{Y} | \mathbf{X}, \mathbf{W}) = \prod_{i=1}^{N} \mathcal{N}(y_i | \mathbf{W} x_i, \sigma^2 \mathbf{I}) \tag{2}
$$

Probabilistic Generative Model

Probabilistic PCA

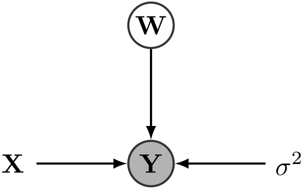


Places prior on latent space **X** and optimises linear mapping **W**

$$p(\mathbf{X}) = \prod_{i=1}^{N} \mathcal{N}(x_i|\mathbf{0}, \mathbf{I})$$

$$p(\mathbf{Y}|\mathbf{W}, \sigma^2) = \int p(\mathbf{Y}|\mathbf{W}, \mathbf{X}, \sigma^2)\ p(\mathbf{X})$$

Dual Probabilistic PCA



Places prior on linear mapping **W** and optimises latent space **X**

$$p(\mathbf{W}) = \prod_{i=1}^{D} \mathcal{N}(w_i|\mathbf{0}, \mathbf{I})$$

$$p(\mathbf{Y}|\mathbf{X}, \sigma^2) = \int p(\mathbf{Y}|\mathbf{W}, \mathbf{X}, \sigma^2)\ p(\mathbf{W}) \quad (3)$$

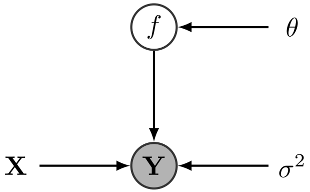
From Dual PPCA to GP-LVM

PPCA and Dual PPCA are equivalent eigenvalue problems with same Maximum Likelihood solution

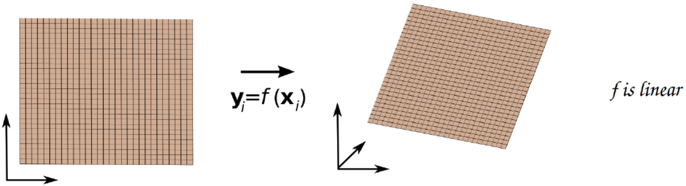
- GP-LVM: Instead of placing prior $p(\mathbf{W})$ on the function parameters in Dual PPCA, we can place a prior $p(f)$ directly on the mapping function i.e. $\mathcal{GP}$ Prior

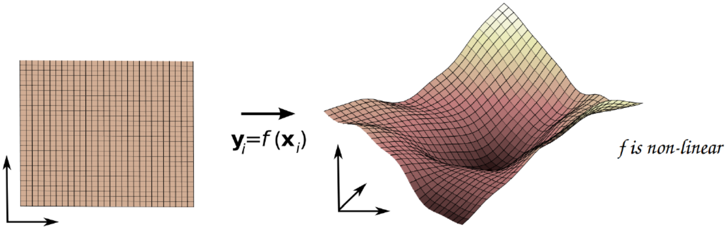
- A $\mathcal{GP}$ Prior allows for non-linear mappings if the covariance function is non-linear. For example, the SE Covariance Function:

$$k(x, x') = \alpha \exp\left(-\frac{\gamma}{2}(x - x')^T (x - x')\right) \tag{4}$$



Linear vs. Non-linear Dimensionality Reduction





Extensions of GP-LVM

Back Constrained GP-LVM: Ensures points close in the observation space (Y) will be close in latent space by constraining back mapping $f' : Y \rightarrow X$

GP-LVM with Dynamics Model: Computes latent space assuming that the latent positions $(\mathbf{X})$ are sequential:

$$x_t = h(x_{t-1}) + \epsilon_{dyn}, \epsilon_{dyn} \sim \mathcal{N}(\mathbf{0}, \sigma_{dyn}^2 \mathbf{I}) \tag{5}$$

A $\mathcal{GP}$ Prior is placed on the function $h(x)$. The resultant optimization becomes:

$$\mathbf{X}, \theta, \theta_{dyn} = \text{argmax}_{\mathbf{X}, \theta, \theta_{dyn}} \; p(\mathbf{Y}|\mathbf{X}, \theta) \; p(\mathbf{X}|\theta_{dyn}) \tag{6}$$