Machine Learning

Gaussian Processes II

Carl Henrik Ek - carlhenrik.ek@bristol.ac.uk

October 10, 2017

http://www.carlhenrik.com

Introduction

Gaussian Processes

- We have uncertainty in our observed outputs

Gaussian Processes

- We have uncertainty in our observed outputs
- We have no uncertainty in our mapping

Gaussian Processes

- We have uncertainty in our observed outputs
- We have no uncertainty in our mapping
    - Linear, it is a line

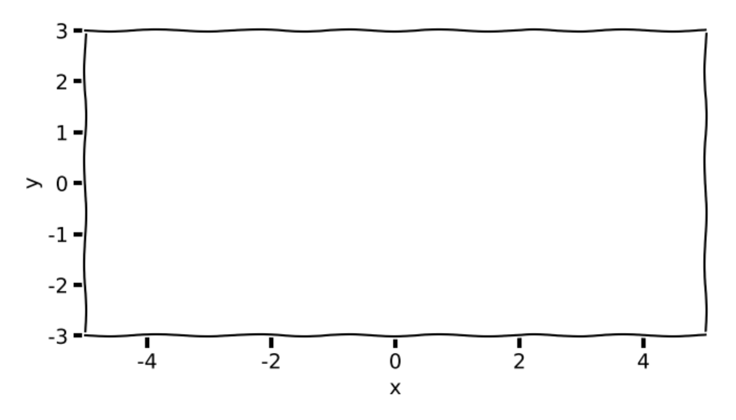
Gaussian Processes

- We have uncertainty in our observed outputs
- We have no uncertainty in our mapping
    - Linear, it is a line
    - Kernels, it is this specific basis function

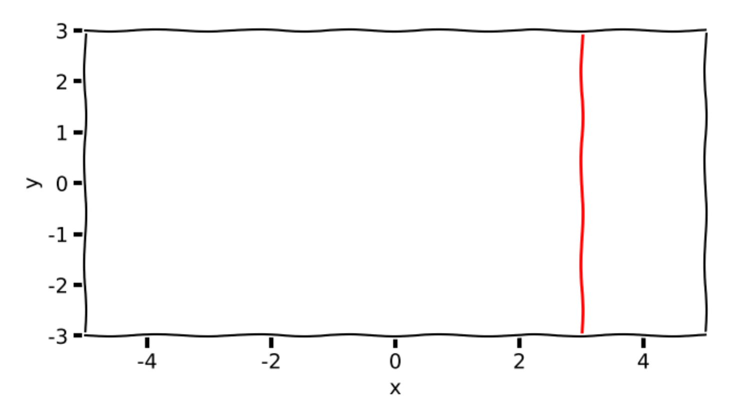
Gaussian Processes

- We have uncertainty in our observed outputs
- We have no uncertainty in our mapping
    - Linear, it is a line
    - Kernels, it is this specific basis function
- *need a prior assumption over the space of functions*

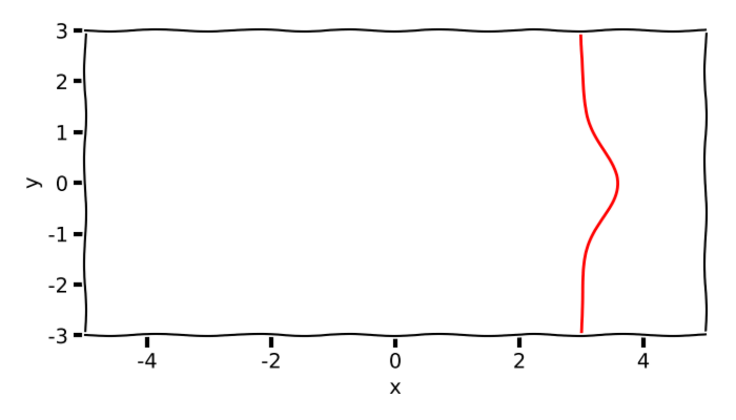
Gaussian Processes



Gaussian Processes

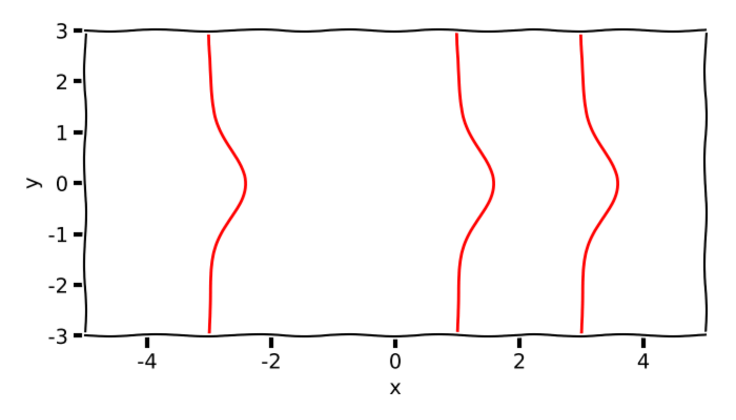


Gaussian Processes



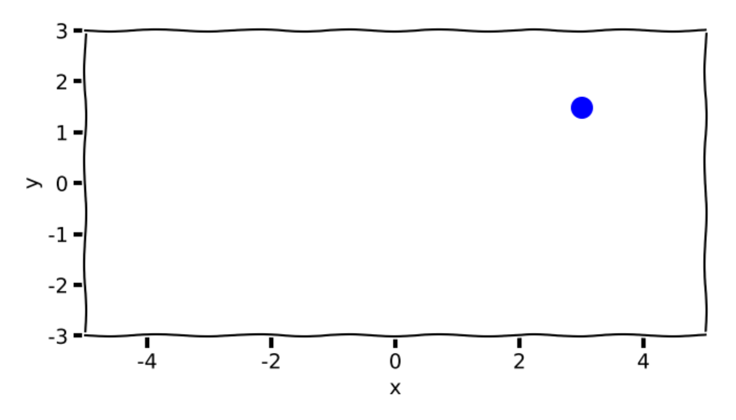
$$p(y|x) = \mathcal{N}(\mu(x), \Sigma(x))$$

Gaussian Processes

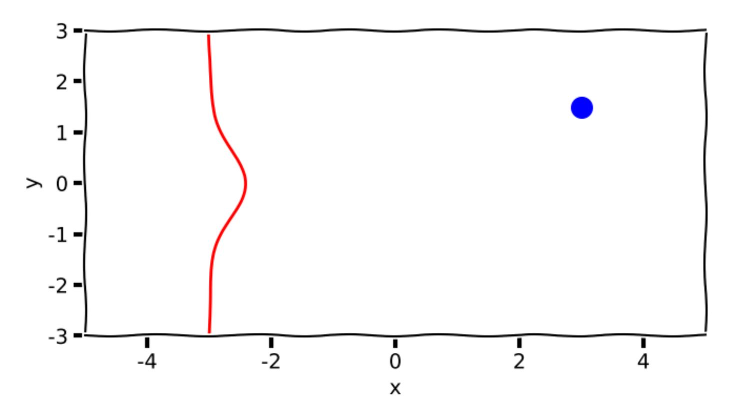


$$p(y_1, y_2, y_3 | x_1, x_2, x_3)$$

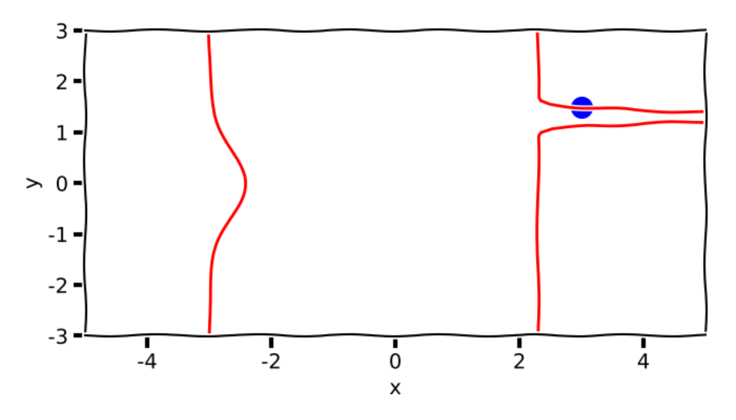
Gaussian Processes



$$p(y_2|x_2, y_1, x_2) = \mathcal{N}(\mu(x_2, x_1, y_1), \Sigma(x_2, x_1, y_1))$$

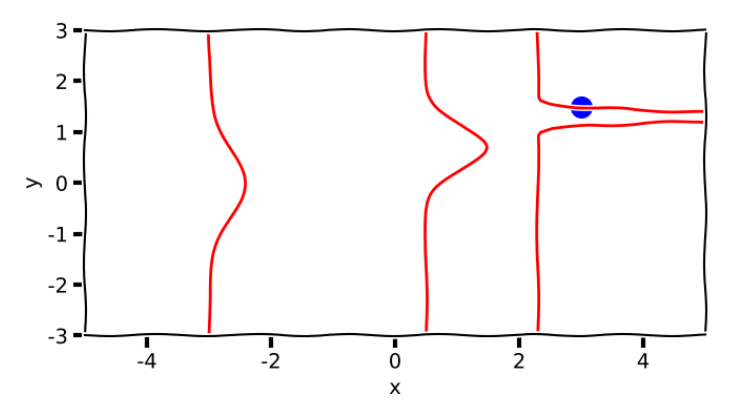


$$p(y_2|x_2, y_1, x_2) = \mathcal{N}(\mu(x_2, x_1, y_1), \Sigma(x_2, x_1, y_1))$$



$$p(y_2|x_2, y_1, x_2) = \mathcal{N}(\mu(x_2, x_1, y_1), \Sigma(x_2, x_1, y_1))$$

Gaussian Processes



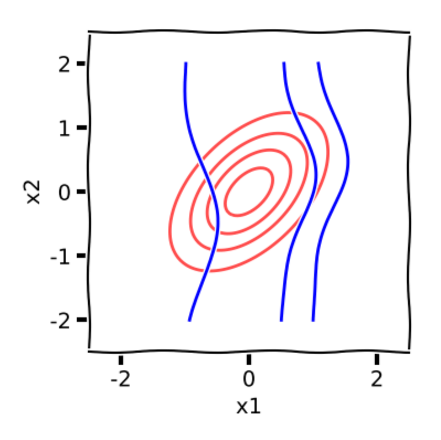
$$p(y_2|x_2, y_1, x_2) = \mathcal{N}(\mu(x_2, x_1, y_1), \Sigma(x_2, x_1, y_1))$$

Conditional Gaussians

$$p(y_1, y_2, y_3 | x_1, x_2, x_3) =$$

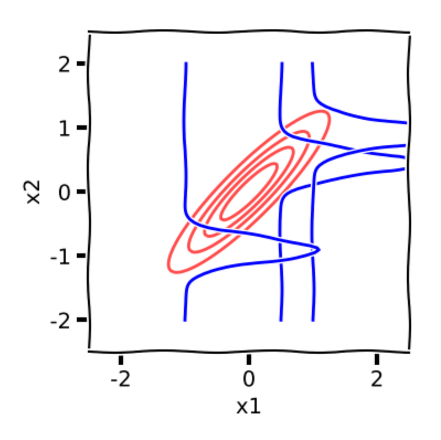
$$\mathcal{N}\left(\begin{bmatrix} \mu(x_1) \\ \mu(x_2) \\ \mu(x_3), \end{bmatrix}, \begin{bmatrix} \Sigma(x_1, x_1) & \Sigma(x_1, x_2) & \Sigma(x_1, x_3) \\ \Sigma(x_2, x_1) & \Sigma(x_2, x_2) & \Sigma(x_2, x_3) \\ \Sigma(x_3, x_1) & \Sigma(x_3, x_2) & \Sigma(x_3, x_3) \end{bmatrix}\right)$$

Conditional Gaussians



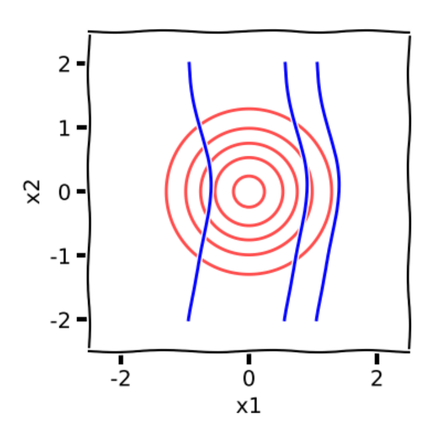
$$\mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}\right)$$

Conditional Gaussians



$$\mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}\right)$$

Conditional Gaussians



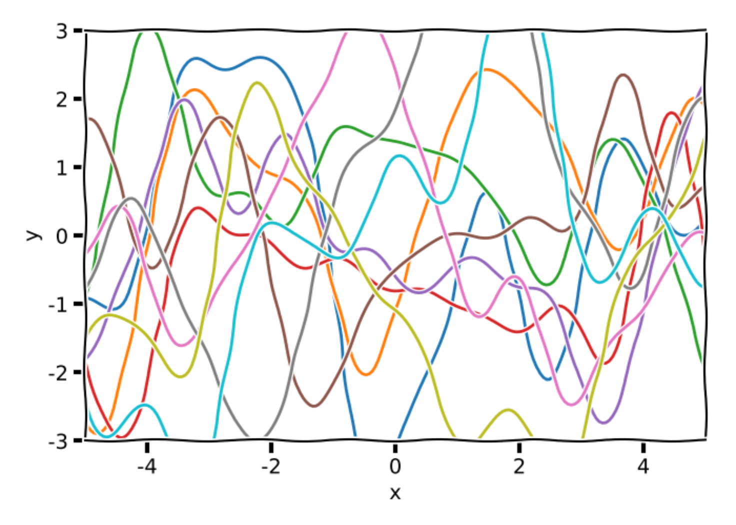
$$\mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$$

Gaussian Process

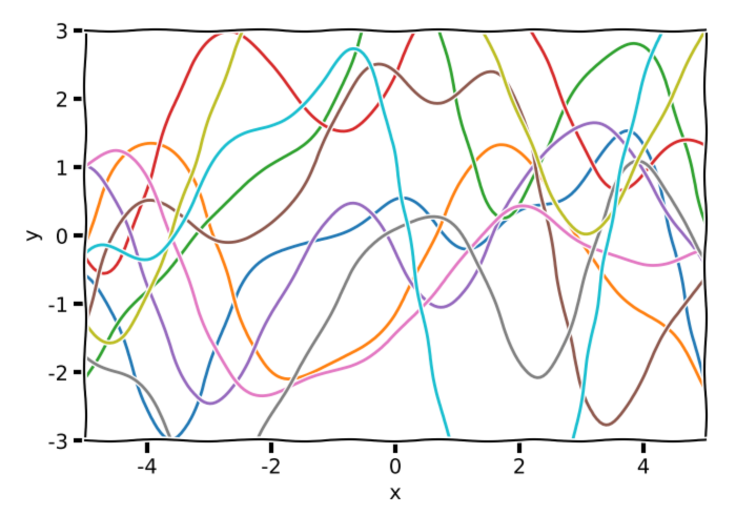
$$p(y_1, \ldots, y_N | x_1, \ldots, x_N) = \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \vdots \\ \mu_N, \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \cdots & \Sigma_{1N} \\ \vdots & \ddots & \vdots \\ \Sigma_{N1} & \cdots & \Sigma_{NN} \end{bmatrix}\right)$$

- $x \in \mathcal{X}$ our infinite input domain
- $\mu_i = \mu(x_i)$ a function from $\mu(x) : \mathcal{X} \to \mathbb{R}$
- $\Sigma_{ij} = k(x_i, x_j)$ a function $k(x_i, x_j) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
- these two functions completely specifies a Gaussian process

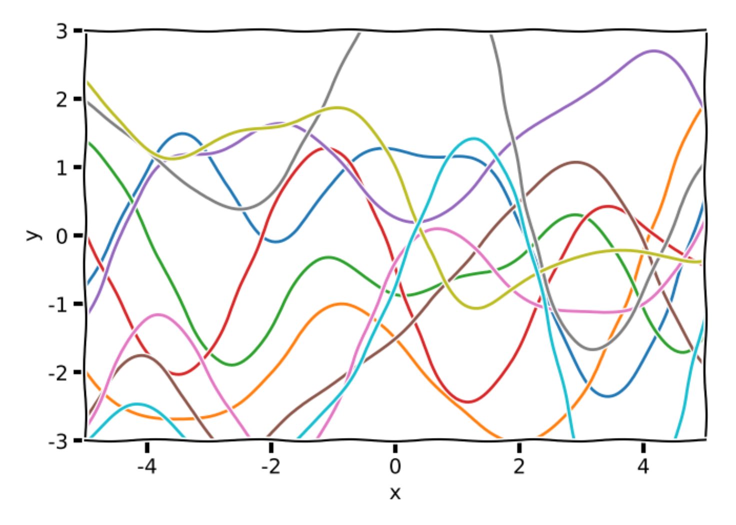
Sampling



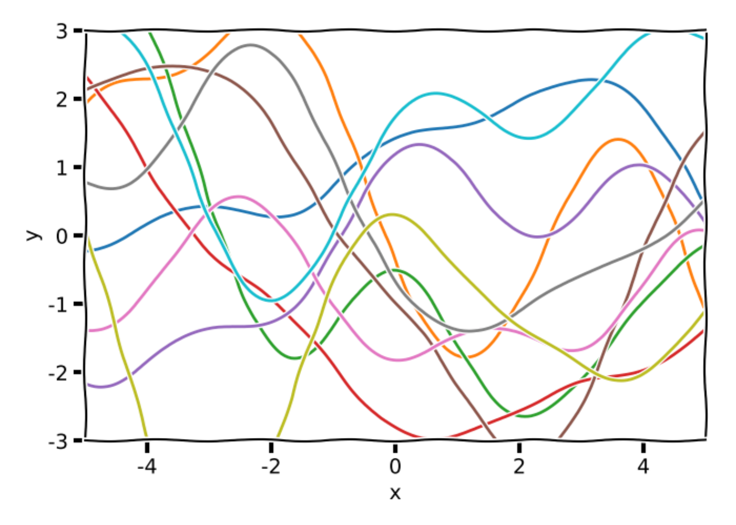
Sampling



Sampling



Sampling



Gaussian Process

$$p(y_1, \ldots, y_N | x_1, \ldots, x_N) = \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \vdots \\ \mu_N, \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \cdots & \Sigma_{1N} \\ \vdots & \ddots & \vdots \\ \Sigma_{N1} & \cdots & \Sigma_{NN} \end{bmatrix} \right)$$

- $x \in \mathcal{X}$ our infinite input domain
- $\mu_i = \mu(x_i)$ a function from $\mu(x) : \mathcal{X} \to \mathbb{R}$
- $\Sigma_{ij} = k(x_i, x_j)$ a function $k(x_i, x_j) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
- these two functions completely specifies a Gaussian process

Posterior

$$\begin{bmatrix} \mathsf{f} \\ \mathsf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} k(\mathsf{X}, \mathsf{X}) & k(\mathsf{X}, \mathsf{x}_*) \\ k(\mathsf{x}_*, \mathsf{X}) & k(\mathsf{x}_*, \mathsf{x}_*) \end{bmatrix} \right)$$

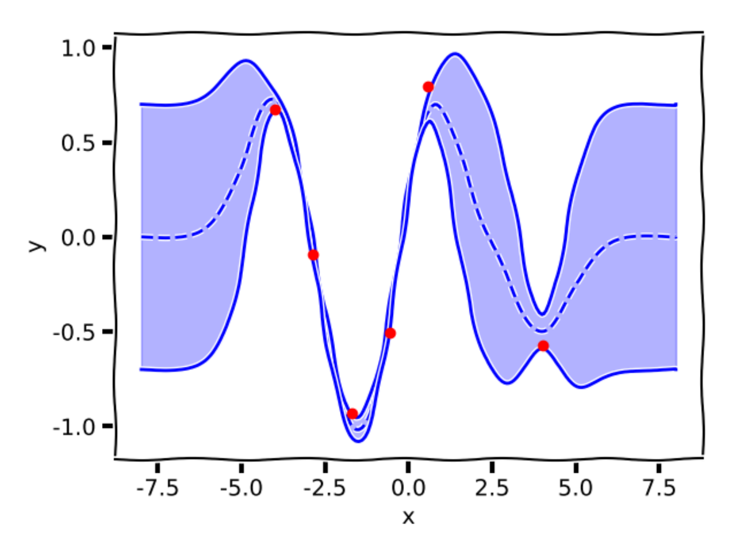
$$\begin{aligned} p(f_* | \mathsf{x}_*, \mathsf{X}, \mathsf{f}, \theta) = \mathcal{N}(& k(\mathsf{x}_*, \mathsf{X})^{\mathrm{T}} K(\mathsf{X}, \mathsf{X})^{-1} \mathsf{f}, \\ & k(\mathsf{x}_*, \mathsf{x}_*) - k(\mathsf{x}_*, \mathsf{X})^{\mathrm{T}} K(\mathsf{X}, \mathsf{X})^{-1} K(\mathsf{X}, \mathsf{x}_*)) \end{aligned}$$

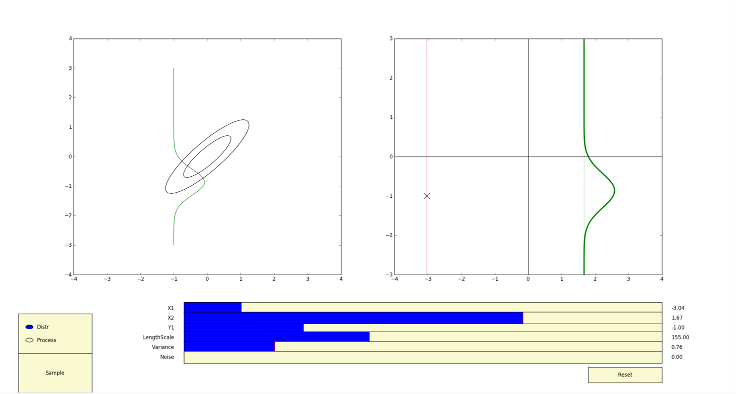
Posterior

$$p(y_1, y_2|x_1, x_2) = p(y_1|y_2, x_1, x_2)p(y_2|x_2)$$

- As soon as we evalute the GP at a finite number of locations its a simple Gaussian distribution
- We used the relationship above to derive the posterior over y_1

Posterior




Distr
Process
Sample
X1
X2
Y1
LengthScale
Variance
Noise
-3.04
1.67
-1.00
155.00
0.76
0.00
Reset

Learning

Gaussian Processes

$$p(f|\mathsf{x}) = \mathcal{N}(\mu(\mathsf{x}), k(\mathsf{x}, \mathsf{x}'))$$

$$k(\mathsf{x}_i, \mathsf{x}_j) = \sigma^2 e^{-\frac{2}{\ell^2}\sin^2\left(\pi\frac{|\mathsf{x}_i - \mathsf{x}_j|}{p}\right)}$$

$$k(\mathsf{x}_i, \mathsf{x}_j) = (\mathsf{x}_i^{\mathrm{T}}\mathsf{x}_j)$$

$$k(\mathsf{x}_i, \mathsf{x}_j) = \frac{2}{\pi}\sin^{-1}\left(\frac{2\mathsf{x}_i^{\mathrm{T}}\Sigma\mathsf{x}_j}{\sqrt{(1 + 2\mathsf{x}_i^{\mathrm{T}}\Sigma\mathsf{x}_i)(1 + 2\mathsf{x}_j^{\mathrm{T}}\Sigma\mathsf{x}_j)}}\right)$$

$$k(\mathsf{x}_i, \mathsf{x}_j) = \sigma^2 e^{||\mathsf{x}_i - \mathsf{x}_j||^2/l^2}$$

- how do we set the parameters of the co-variance function?

Marginal Likelihood

$$p(\mathbf{Y}|\mathbf{X}, \theta) = \int p(\mathbf{Y}|\mathbf{f})p(\mathbf{f}|\mathbf{X}, \theta)\mathrm{d}f$$

- We are not interested in f directly
- Marginalise out f
- Gaussian likelihood and Gaussian prior $\rightarrow$ Gaussian marginal

Marginalisation

- Deterministic world

$$\mathbb{E}[y] = \int y p(y) \mathrm{d}y$$

Marginalisation

- Deterministic world

$$\mathbb{E}[y] = \int y p(y) \mathrm{d}y$$

- Stochastic world

$$\mathbb{E}[p(y)] = \int p(y|x) p(x) \mathrm{d}x$$

Learning

$$\hat{\theta} = \mathrm{argmax}_{\theta} p(\mathbf{Y}|\mathbf{X}, \theta)$$

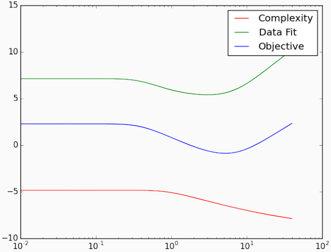
- Type-II Maximum likelihood [1] 3.5.0
- minimise logarithm of marginal likelihood

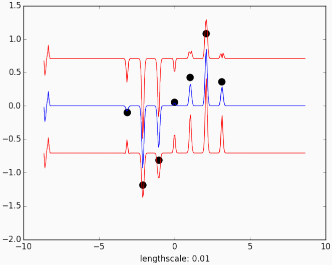
Learning

$$\text{argmax}_{\boldsymbol{\theta}} p(\mathsf{Y}|\mathsf{X}, \boldsymbol{\theta}) = \text{argmin}_{\boldsymbol{\theta}} - \log\left(p(\mathsf{Y}|\mathsf{X}, \boldsymbol{\theta})\right) = \text{argmin}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$$

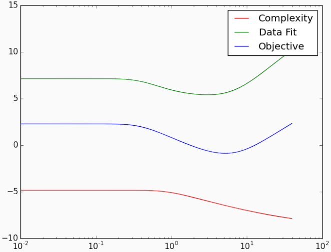
$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{N}{2}\log(2\pi)$$

Learning

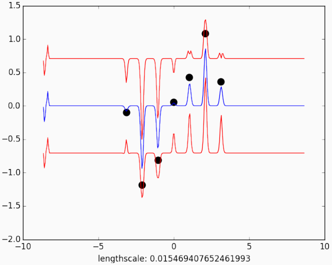

Complexity
Data Fit
Objective
15
10
5
0
−5
−10
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−10
−5
0
5
10
lengthscale: 0.01

Learning

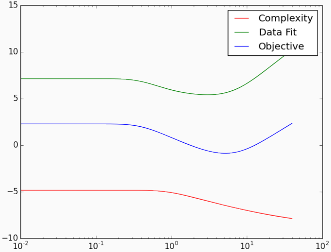


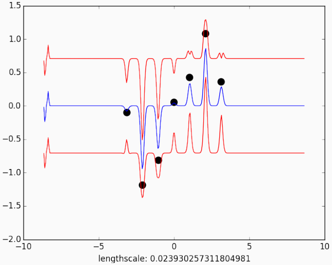
Complexity
Data Fit
Objective
15
10
5
0
−5
−10
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}



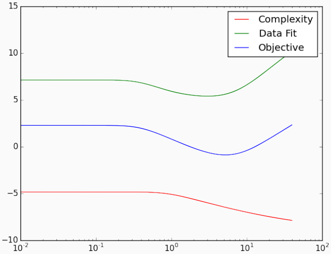
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−10
−5
0
5
10
lengthscale: 0.015469407652461993

Learning

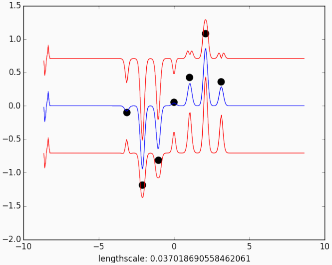

Complexity
Data Fit
Objective
15
10
5
0
−5
−10
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−10
−5
0
5
10
lengthscale: 0.023930257311804981

Learning

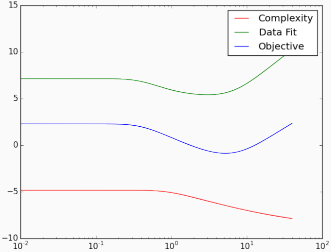


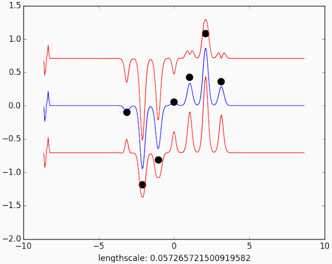
Complexity
Data Fit
Objective



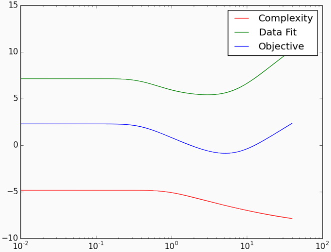
lengthscale: 0.037018690558462061

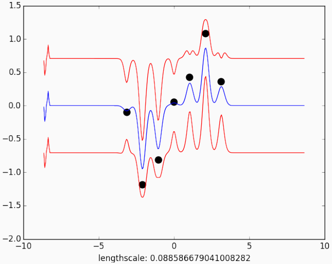
Learning


Complexity
Data Fit
Objective

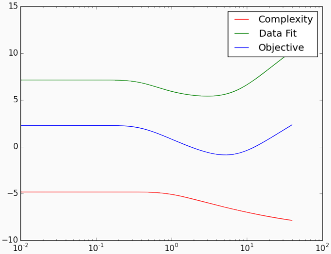

lengthscale: 0.057265721500919582

Learning

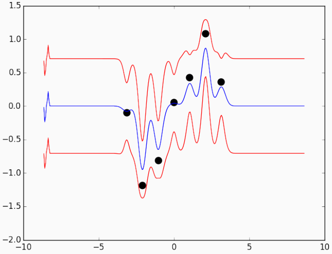

Complexity
Data Fit
Objective


lengthscale: 0.088586679041008282

Learning

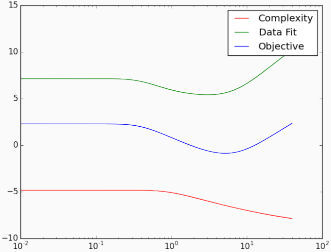


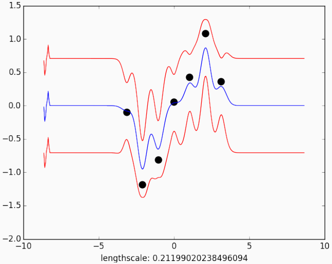
Complexity
Data Fit
Objective



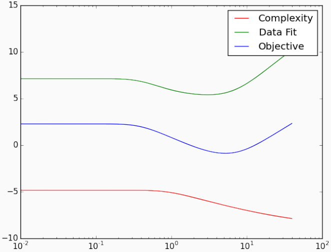
lengthscale: 0.13703834506631679

Learning

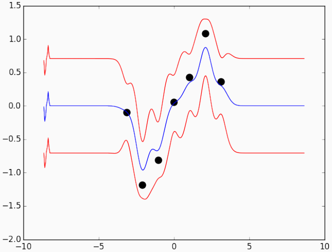

Complexity
Data Fit
Objective


lengthscale: 0.21199020238496094

Learning

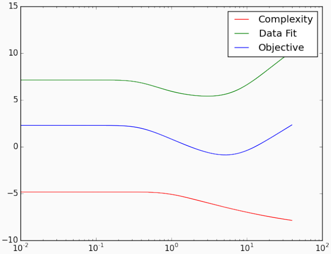


Complexity
Data Fit
Objective

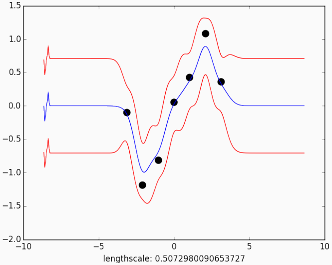


lengthscale: 0.32793628590208818

Learning

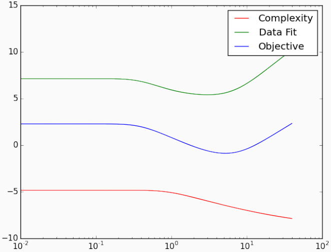


Complexity
Data Fit
Objective
15
10
5
0
−5
−10
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}

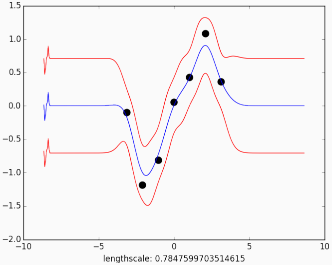


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−10
−5
0
5
10
lengthscale: 0.5072980090653727

Learning

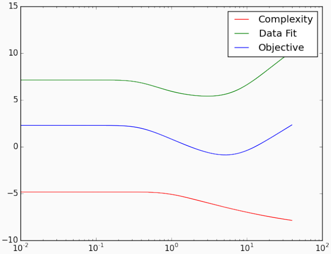


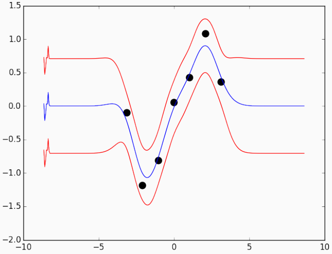
Complexity
Data Fit
Objective
15
10
5
0
−5
−10
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}



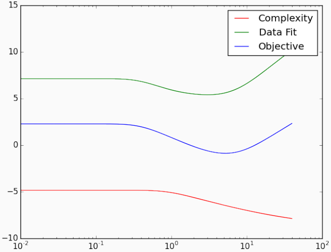
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−10
−5
0
5
10
lengthscale: 0.7847599703514615

Learning

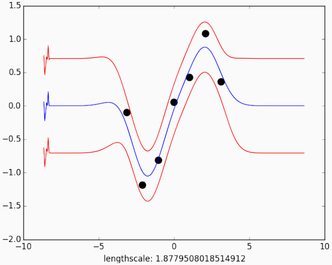

Complexity
Data Fit
Objective


lengthscale: 1.2139771890700752

Learning

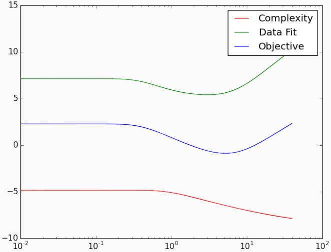


Complexity
Data Fit
Objective
15
10
5
0
−5
−10
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}

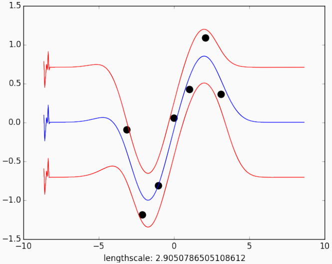


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−10
−5
0
5
10
lengthscale: 1.8779508018514912

Learning

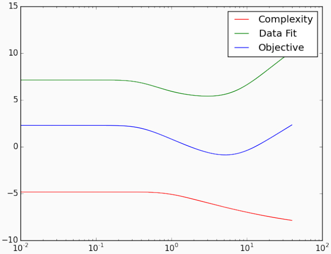


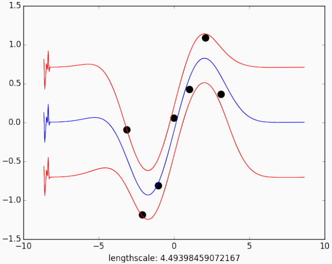
Complexity
Data Fit
Objective
15
10
5
0
−5
−10
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}



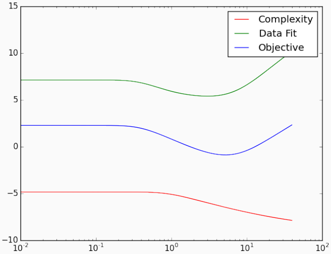
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10
−5
0
5
10
lengthscale: 2.9050786505108612

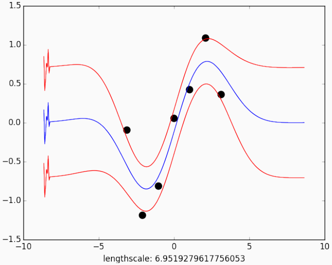
Learning


Complexity
Data Fit
Objective

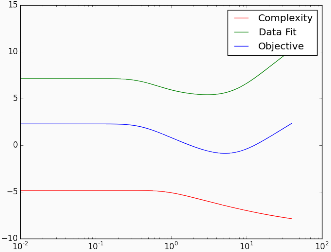

lengthscale: 4.49398459072167

Learning

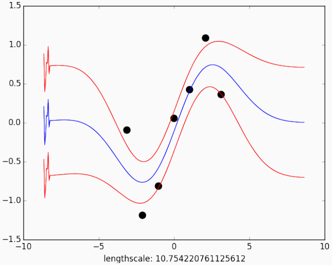

15
10
5
0
−5
−10
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
Complexity
Data Fit
Objective


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10
−5
0
5
10
lengthscale: 6.9519279617756053

Learning

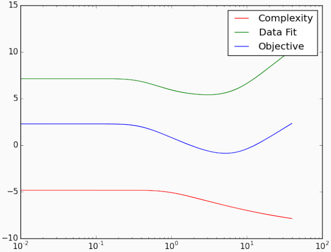


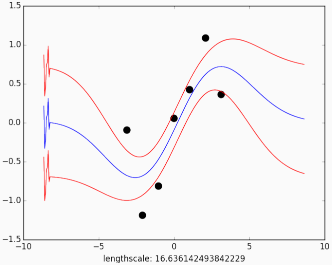
Complexity
Data Fit
Objective
15
10
5
0
−5
−10
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}



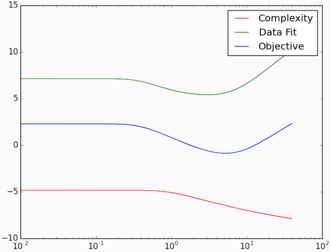
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10
−5
0
5
10
lengthscale: 10.754220761125612

Learning

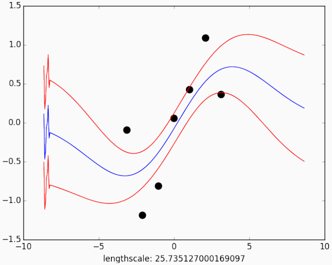

Complexity
Data Fit
Objective


lengthscale: 16.636142493842229

Learning

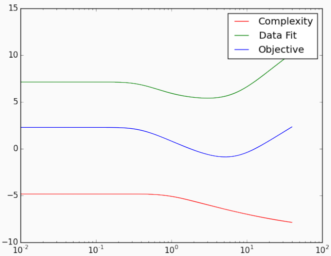


Complexity
Data Fit
Objective
15
10
5
0
−5
−10
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}

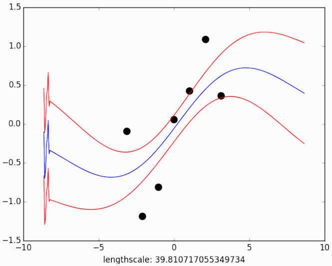


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10
−5
0
5
10
lengthscale: 25.735127000169097

Learning

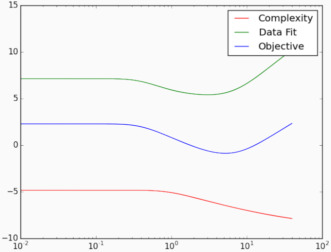


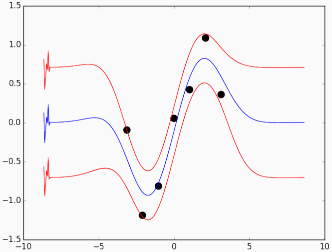
Complexity
Data Fit
Objective



lengthscale: 39.810717055349734

Learning


Complexity
Data Fit
Objective


lengthscale: 4.49398459072167

Gaussian Processes

- completely specified by mean and covariance function
- mean and covariance takes input variable
- every instantiation of the function is jointly Gaussian
  - conditional and marginal distribution trivial
- very flexible
  - covariance function can encode any behaviour

Unsupervised Learning

Strength of Priors

$$y = f(x)$$

- given input output pairs we have made assumptions about f
- from data we can update our assumption
- can we push this further?

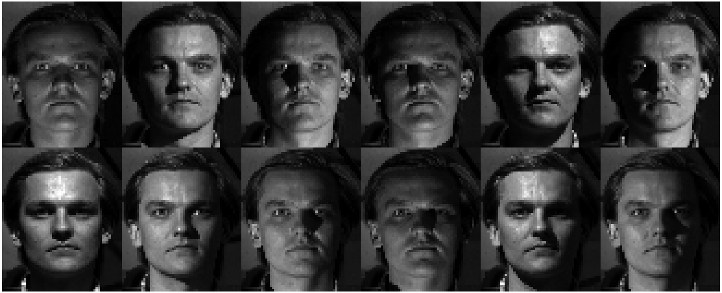
Unsupervised learning

$$y = f(x)$$

- In unsupervised learning we are given only output
- Input is *latent*
- Task: recover both f and x

Latent Variable Models

Latent Variable Models

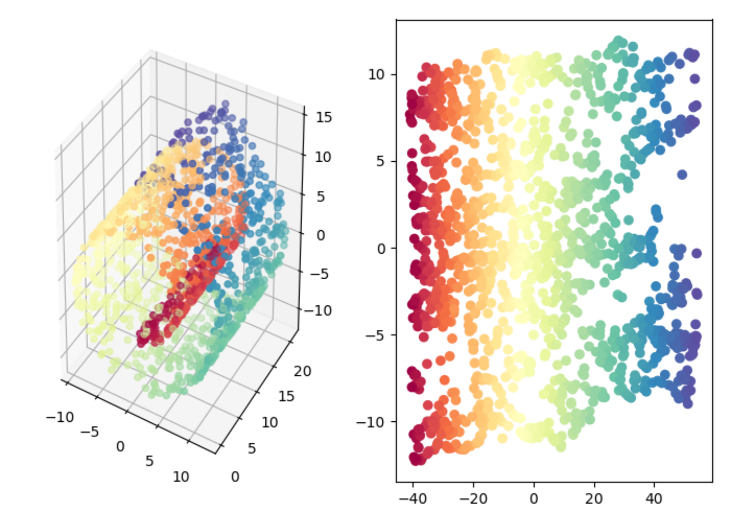


output data $\mathbf{y} \in \mathbb{R}^{256 \times 256} \rightarrow 65536$ dimensions

input location on sphere $\rightarrow 3$ dimensions

manifold images lie on a 3 dimensional surface in 65536 dimensions

Manifold



Linear Latent Variable Models [1] 12.2

- Observed data

$$\mathbf{x} \in \mathbb{R}^D$$

- Latent variable

$$\mathbf{z} \in \mathbb{R}^M$$

- Mapping

$$\mathbf{x}_i = \mathbf{W}\mathbf{z}_i + \epsilon$$

- Likelihood: make noise assumption $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$

$$p(\mathbf{x}|\mathbf{z}, \mathbf{W}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I})$$

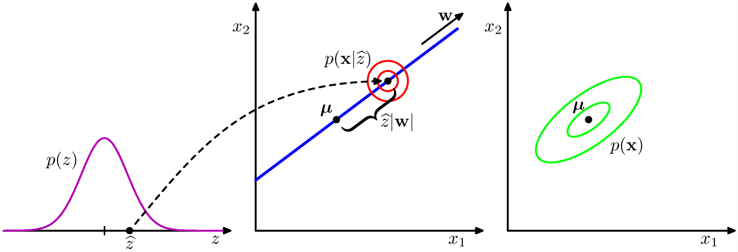
- Prior ?

Linear Latent Variable Models

$$\mathbf{x}_i = \mathbf{W}\mathbf{z}_i + \epsilon$$

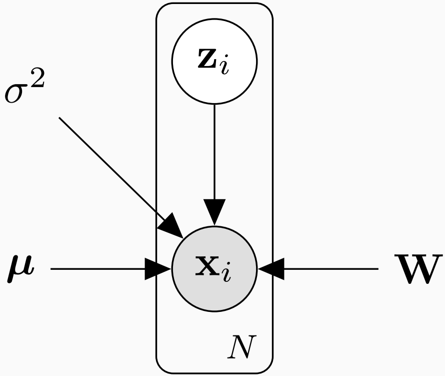
- marginalise out both $\mathbf{W}$ and $\mathbf{z}$ is intractable
- marginalise out one and infer the other
- $\mathbf{W} \in \mathbb{R}^{D \times M}$ and $\mathbf{z} \in \mathbb{R}^{M \times N}$
- N commonly larger than $D \Rightarrow$ integrate out $\mathbf{z}$

Principal Component Analysis [1] Figure 12.9



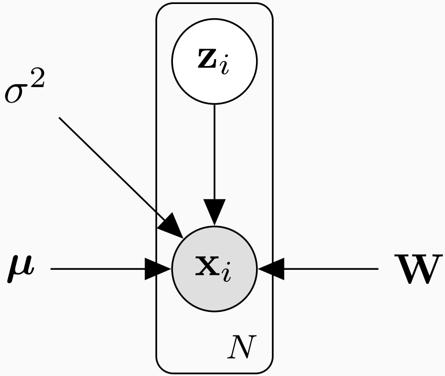
$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \boldsymbol{I})$$

Graphical Model



$$p(\mathsf{x}, \mathsf{z}) = p(\mathsf{x}|\mathsf{z})p(\mathsf{z})$$

Graphical Model



$$p(\mathsf{x}, \mathsf{z} | \mathsf{W}, \boldsymbol{\mu}, \sigma^2) = p(\mathsf{x} | \mathsf{z}, \mathsf{W}, \boldsymbol{\mu}, \sigma^2) p(\mathsf{z})$$

Marginal distribution

$$p(\mathsf{x}|\mathsf{W}) = \int p(\mathsf{x}|\mathsf{z}, \mathsf{W}) p(\mathsf{z}) \mathrm{d}\mathsf{z} = \mathcal{N}(\mathsf{x}|\boldsymbol{\mu}, \mathsf{C})$$

- Gaussian distribution closed under linear transformation (interesting proof)

Marginal distribution

$$
\begin{aligned}
p(\mathsf{x}|\mathsf{W}) &= \int p(\mathsf{x}|\mathsf{z},\mathsf{W})p(\mathsf{z})\mathrm{d}\mathsf{z} = \mathcal{N}(\mathsf{x}|\boldsymbol{\mu},\mathsf{C}) \\
&= \mathcal{N}(\mathsf{x}|\boldsymbol{\mu},\mathsf{W}\mathsf{W}^{\mathrm{T}}+\sigma^2\mathsf{I})
\end{aligned}
$$

- Gaussian distribution closed under linear transformation (interesting proof)

Maximum Likelihood [1] Ch. 12.2.1

$$\log p(\mathbf{X}|\boldsymbol{\mu}, \mathbf{W}, \sigma^2)$$

- find stationary with respect to each variable gives Maximum likelihood solution to $\mathbf{W}$, $\boldsymbol{\mu}$ and σ^2
- *In the assignment we make it easier and take derivatives instead and optimise*

Maximum Likelihood [1] Ch. 12.2.1

$$\log p(\mathbf{X}|\boldsymbol{\mu}, \mathbf{W}, \sigma^2) = \sum_{n=1}^{N} \log p(\mathbf{x}_n|\mathbf{W}, \boldsymbol{\mu}, \sigma^2)$$

- find stationary with respect to each variable gives Maximum likelihood solution to $\mathbf{W}$, $\boldsymbol{\mu}$ and σ^2
- *In the assignment we make it easier and take derivatives instead and optimise*

Maximum Likelihood [1] Ch. 12.2.1

$$
\begin{aligned}
\log p(\mathbf{X}|\boldsymbol{\mu}, \mathbf{W}, \sigma^2) &= \sum_{n=1}^{N} \log p(\mathbf{x}_n|\mathbf{W}, \boldsymbol{\mu}, \sigma^2) \\
&= -\frac{ND}{2}\log(2\pi) - \frac{N}{2}\log|\mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}| \\
&- \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu})^{\mathrm{T}}(\mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I})^{-1}(\mathbf{x}_n - \boldsymbol{\mu})
\end{aligned}
$$

- find stationary with respect to each variable gives Maximum likelihood solution to $\mathbf{W}$, $\boldsymbol{\mu}$ and σ^2
- *In the assignment we make it easier and take derivatives instead and optimise*

Posterior

$$p(\mathbf{z}|\mathbf{x}) \propto p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$$

$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|(\mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I})^{-1}\mathbf{W}^{\mathrm{T}}(\mathbf{x} - \boldsymbol{\mu}), \sigma^2(\mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I})^{-1})$$

- Gaussian likelihood and Gaussian prior $\rightarrow$ Gaussian posterior

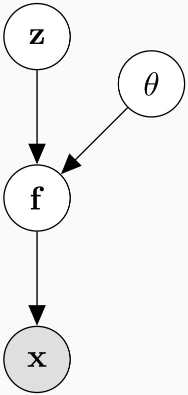
Principal Component Analysis

- You might have seen this explained in a different way
  - *Retain variance*
  - *Error minimisation*
- These provides the same solution as the maximum likelihood but solved by an eigenvalue problem
- Do not provide intuition as it doesn't state assumptions

Question 15-21

You now have all the material to finish the assignment!

Non-linear Latent variable model



$$p(\mathsf{x}|\mathsf{z}, \theta) = \int p(\mathsf{x}|\mathsf{f})p(\mathsf{f}|\mathsf{z}, \theta)\mathrm{d}\mathsf{f}$$

Demo

Font Demo

Summary

Summary

- Type II Maximum likelihood
- As long as I make assumptions I can learn from data
- Unsupervised learning, just the same, just a prior instead of observations
- Next 3 lectures

eof

References

Christopher M. Bishop.
Pattern Recognition and Machine Learning (Information Science and Statistics).
Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.