Gaussian Processes for Regression

COMP9418 — Advanced Topics in Statistical Machine Learning

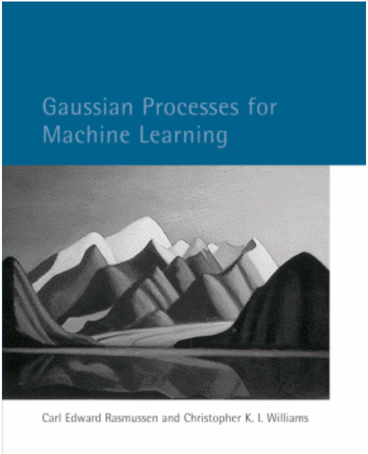
Edwin V. Bonilla

School of Computer Science and Engineering
UNSW Sydney

September 20th, 2017

(Last Update: Tuesday 19th September, 2017 at 18:15)

Acknowledgements



Carl Edward Rasmussen and Christopher K. I. Williams

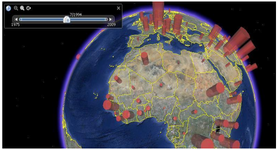
All chapters available online along with software and datasets:
http://www.gaussianprocess.org/gpml

Aims

This lecture will allow you to understand Gaussian processes as priors over functions and apply them to regression problems. Following it you should be to:

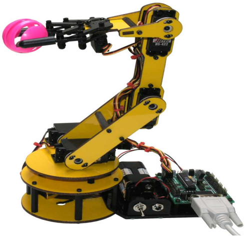
- Understand and apply Bayesian approaches to linear regression.
- Understand and apply Bayesian linear-in-the-parameters models to non-linear regression problems.
- Understand the connection between Bayesian regression with non-linear feature spaces and Gaussian process regression.
- Derive and apply the function-space view of Gaussian process regression.
- Carry out model selection in Gaussian process regression models.

Some Applications of Gaussian Process (GP) Models (1)

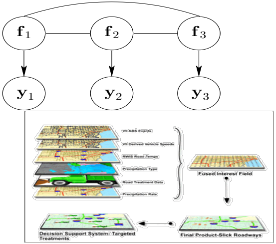


Spatio-temporal modelling

Classification

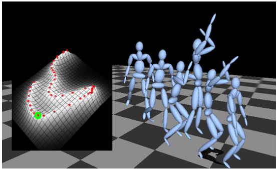


Robot inverse dynamics



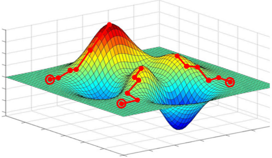
Data fusion / multi-task learning

Some Applications of GP Models (2)

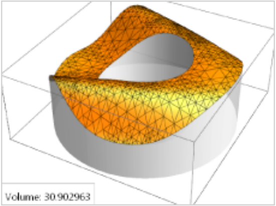


Style-based inverse kinematics

Preference learning



Bayesian optimisation

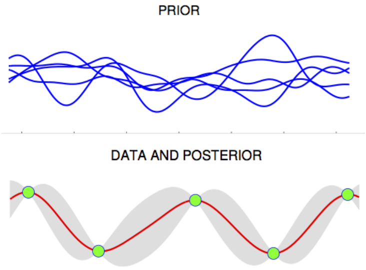


Bayesian quadrature

How can We ‘Solve’ All these Problems with the Humble Gaussian Distribution?

Key components of GP models:

- Non-parametric prior
- Bayesian
- Kernels (covariance functions)



Bayesian non-linear regression

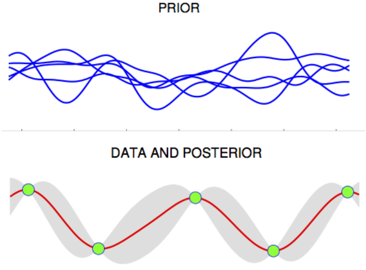
How can We 'Solve' All these Problems with the Humble Gaussian Distribution?

Key components of GP models:

- Non-parametric prior
- Bayesian
- Kernels (covariance functions)

Tasks

- Prediction (posterior inference)
- Hyperparameter learning



Bayesian non-linear regression

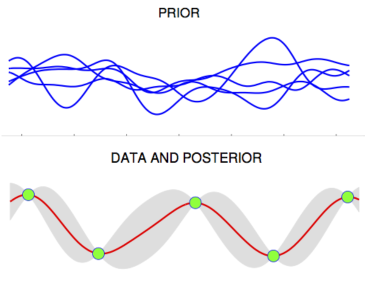
How can We 'Solve' All these Problems with the Humble Gaussian Distribution?

Key components of GP models:

- Non-parametric prior
- Bayesian
- Kernels (covariance functions)

Tasks

- Prediction (posterior inference)
- Hyperparameter learning



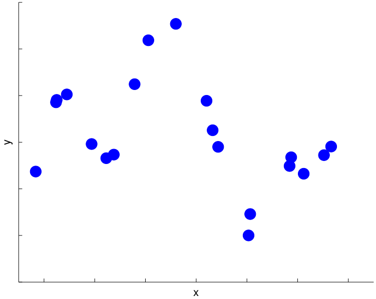
Bayesian non-linear regression

Challenges

- Intractability for non-Gaussian likelihoods
  - E.g. a sigmoid likelihood for classification
- High computational cost (in time and memory) with # datapoints

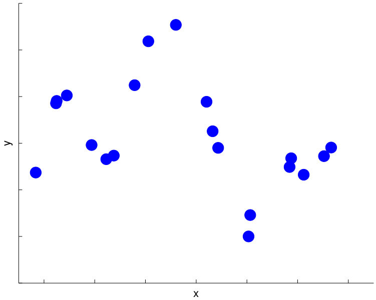
The Prediction Problem

Learn mapping $\mathbf{x} \rightarrow f(\mathbf{x})$ from observations $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$.



The Prediction Problem

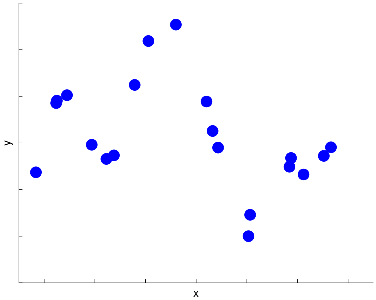
Learn mapping $\mathbf{x} \rightarrow f(\mathbf{x})$ from observations $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$.



- What parameterization?

The Prediction Problem

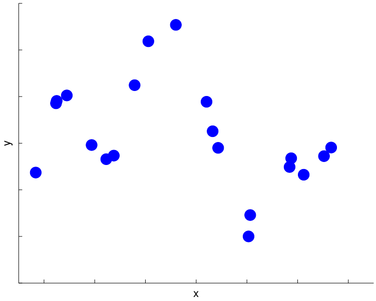
Learn mapping $\mathbf{x} \to f(\mathbf{x})$ from observations $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$.



- What parameterization?
- $f(\mathbf{x}) = \sum_j w_j x_j$

The Prediction Problem

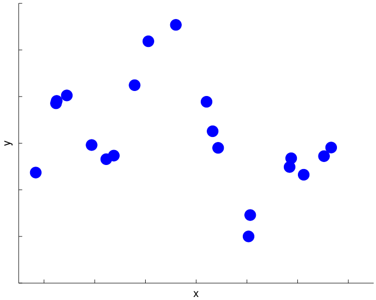
Learn mapping $\mathbf{x} \rightarrow f(\mathbf{x})$ from observations $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$.



- What parameterization?
- $f(\mathbf{x}) = \sum_j w_j \phi_j(\mathbf{x})$

The Prediction Problem

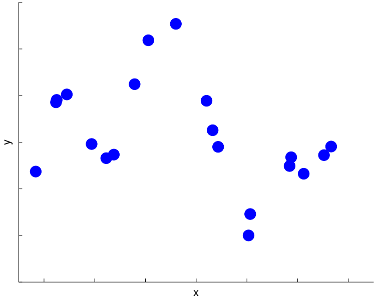
Learn mapping $\mathbf{x} \to f(\mathbf{x})$ from observations $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$.



- What parameterization?
- $f(\mathbf{x}) = \sum_j w_j \phi_j(\mathbf{x})$
- Flexibility v generalization

The Prediction Problem

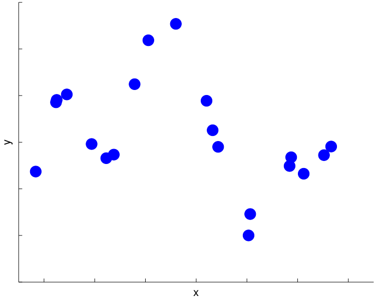
Learn mapping $\mathbf{x} \to f(\mathbf{x})$ from observations $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$.



- What parameterization?
- $f(\mathbf{x}) = \sum_j w_j \phi_j(\mathbf{x})$
- Flexibility v generalization
- What basis functions?
  How many?

The Prediction Problem

Learn mapping $\mathbf{x} \to f(\mathbf{x})$ from observations $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$.

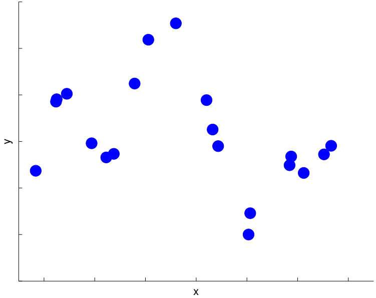


- What parameterization?
- $f(\mathbf{x}) = \sum_j w_j \phi_j(\mathbf{x})$
- Flexibility v generalization
- What basis functions? How many?

- What about Neural nets?

The Prediction Problem

Learn mapping $\mathbf{x} \rightarrow f(\mathbf{x})$ from observations $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$.

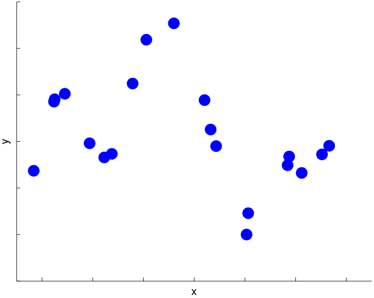


- What parameterization?
- $f(\mathbf{x}) = \sum_j w_j \phi_j(\mathbf{x})$
- Flexibility v generalization
- What basis functions? How many?

- What about Neural nets?
- How to avoid overfitting? (cf regularization)

The Prediction Problem

Learn mapping $\mathbf{x} \rightarrow f(\mathbf{x})$ from observations $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$.

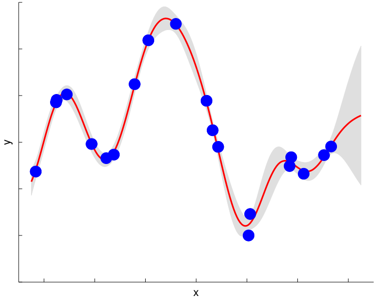


- What parameterization?
- $f(\mathbf{x}) = \sum_j w_j \phi_j(\mathbf{x})$
- Flexibility v generalization
- What basis functions? How many?

- What about Neural nets?
- How to avoid overfitting? (cf regularization)
- Confidence on our predictions?

The Prediction Problem

Learn mapping $\mathbf{x} \to f(\mathbf{x})$ from observations $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$.

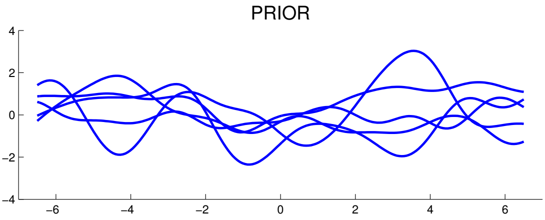


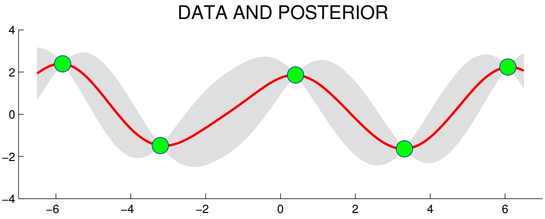
- What parameterization?
- $f(\mathbf{x}) = \sum_j w_j \phi_j(\mathbf{x})$
- Flexibility v generalization
- What basis functions? How many?

- What about Neural nets?
- How to avoid overfitting? (cf regularization)
- Confidence on our predictions?

We can address these issues in a principled way with Gaussian process models

Demo





- Smooth functions
- Closeness in input space $\rightarrow$ closeness in output space

Why Gaussian Processes

- Parametric models constrain the class of functions we consider
- Flexibility (no underfitting) due to non-parametric nature

Why Gaussian Processes

- Parametric models constrain the class of functions we consider
- Flexibility (no underfitting) due to non-parametric nature
- Generalization (no overfitting)
- Bayesian, distribution over functions: prior, likelihood, posterior

Why Gaussian Processes

- Parametric models constrain the class of functions we consider
- Flexibility (no underfitting) due to non-parametric nature
- Generalization (no overfitting)
- Bayesian, distribution over functions: prior, likelihood, posterior
- How can we do computations with infinite vectors?
  - "Efficient" Inference due to consistency (Gaussian distributions)

Why Gaussian Processes

- Parametric models constrain the class of functions we consider
- Flexibility (no underfitting) due to non-parametric nature
- Generalization (no overfitting)
- Bayesian, distribution over functions: prior, likelihood, posterior
- How can we do computations with infinite vectors?
  - "Efficient" Inference due to consistency (Gaussian distributions)
- Characteristics of the functions can be learned from data
  - Covariance function: smoothness, stationarity, length-scale
  - Hyperparameter learning

Why Gaussian Processes

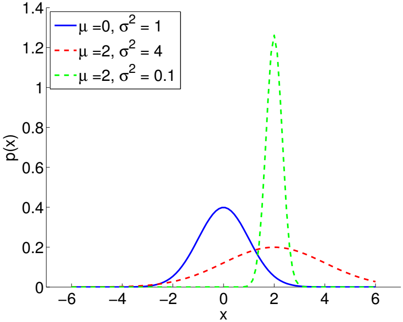
- Parametric models constrain the class of functions we consider
- Flexibility (no underfitting) due to non-parametric nature
- Generalization (no overfitting)
- Bayesian, distribution over functions: prior, likelihood, posterior
- How can we do computations with infinite vectors?
  - "Efficient" Inference due to consistency (Gaussian distributions)
- Characteristics of the functions can be learned from data
  - Covariance function: smoothness, stationarity, length-scale
  - Hyperparameter learning
- Many standard regression models are special cases of GPs

Why Gaussian Processes

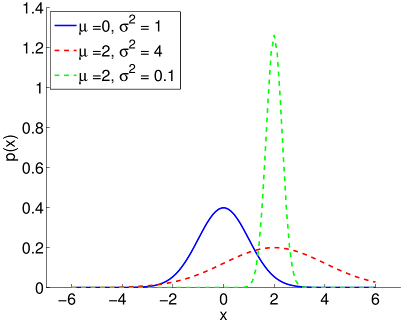
- Parametric models constrain the class of functions we consider
- Flexibility (no underfitting) due to non-parametric nature
- Generalization (no overfitting)
- Bayesian, distribution over functions: prior, likelihood, posterior
- How can we do computations with infinite vectors?
  - "Efficient" Inference due to consistency (Gaussian distributions)
- Characteristics of the functions can be learned from data
  - Covariance function: smoothness, stationarity, length-scale
  - Hyperparameter learning
- Many standard regression models are special cases of GPs
- GP models also applicable to non-regression settings

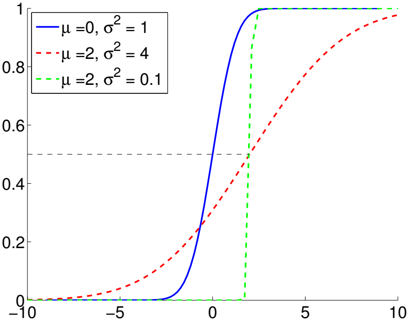
Outline

1. The Gaussian Distribution Revisited
2. Standard Bayesian Linear Regression
3. Bayesian Regression with Non-linear Feature Spaces
4. Gaussian Processes for Regression
   - Function-space View
   - Predictions
   - Model Selection
   - Other Covariance Functions



$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$



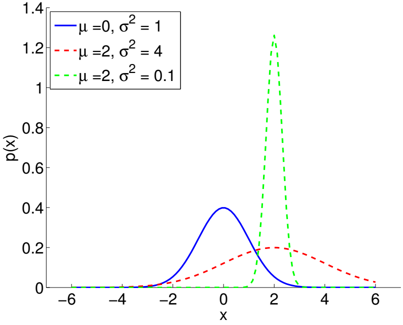


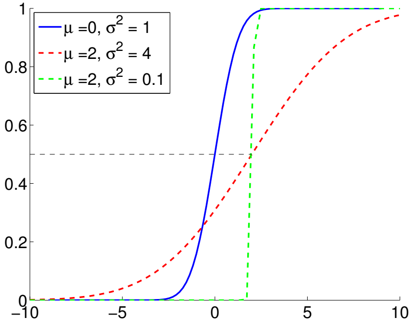
$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$

$$F(x) = \int_{-\infty}^{x} \mathcal{N}(z|\mu,\sigma^2)dz$$

The Gaussian Distribution

1D Example

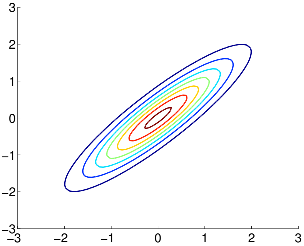




$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x - \mu)^2\right)$$

$$F(x) = \int_{-\infty}^{x} \mathcal{N}(z|\mu, \sigma^2)dz$$

In general: $$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{|2\pi\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$$

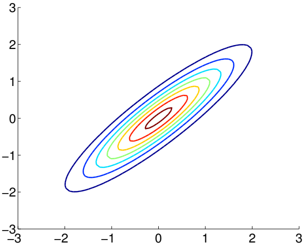


$$p(x_1, x_2) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

Joint

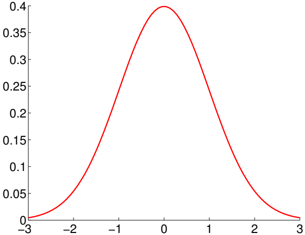
The Gaussian Distribution

2D Example



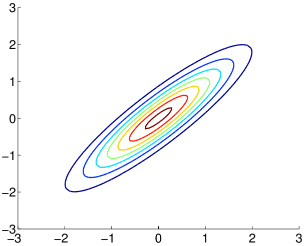
$p(x_1, x_2) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

Joint



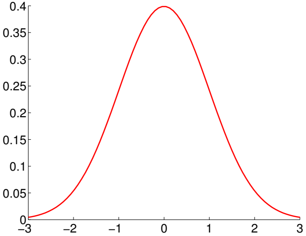
$p(x_1)$

Marginal



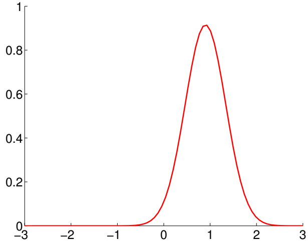
$p(x_1, x_2) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

Joint



$p(x_1)$

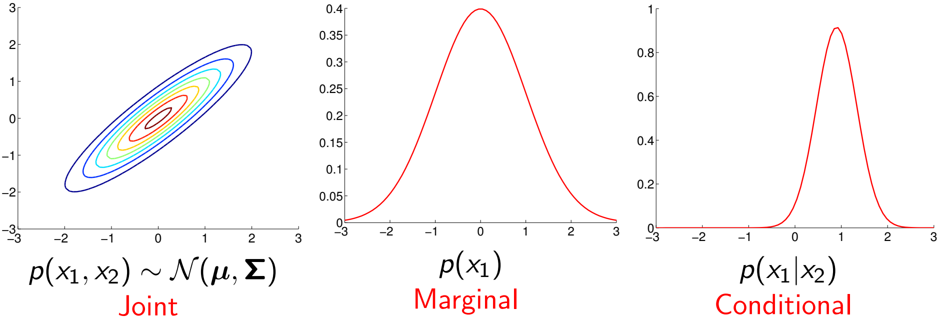
Marginal



$p(x_1 | x_2)$

Conditional

The Gaussian Distribution
2D Example



The marginal and the conditional distributions are also Gaussians

Partitioned Gaussians

For general Gaussian random vectors $\mathbf{x}$, we can partition:

$$\mathbf{x} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}^T & \boldsymbol{\Sigma}_{22} \end{bmatrix} \right),$$

Partitioned Gaussians

For general Gaussian random vectors $\mathbf{x}$, we can partition:

$$\mathbf{x} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}^T & \boldsymbol{\Sigma}_{22} \end{bmatrix} \right),$$

hence, the marginal $\mathbf{x}_1 \sim \mathcal{N}(\mathbf{x}_1 | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11})$

Partitioned Gaussians

For general Gaussian random vectors $\mathbf{x}$, we can partition:

$$\mathbf{x} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}^T & \boldsymbol{\Sigma}_{22} \end{bmatrix} \right),$$

hence, the marginal $\mathbf{x}_1 \sim \mathcal{N}(\mathbf{x}_1 | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11})$

and the conditional $\mathbf{x}_1 | \mathbf{x}_2 \sim \mathcal{N}(\mathbf{x}_1 | \boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2})$

Partitioned Gaussians

For general Gaussian random vectors $\mathbf{x}$, we can partition:

$$\mathbf{x} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}^T & \boldsymbol{\Sigma}_{22} \end{bmatrix} \right),$$

hence, the marginal $\mathbf{x}_1 \sim \mathcal{N}(\mathbf{x}_1 | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11})$

and the conditional $\mathbf{x}_1 | \mathbf{x}_2 \sim \mathcal{N}(\mathbf{x}_1 | \boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2})$

where $\boldsymbol{\mu}_{1|2} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2)$,

and $\boldsymbol{\Sigma}_{1|2} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{12}^T$

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{|2\pi\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

$\boldsymbol{\Sigma}$: is the covariance matrix

$\boldsymbol{\Sigma}^{-1}$: is the precision matrix

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{|2\pi\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

$\boldsymbol{\Sigma}$: is the covariance matrix

$\boldsymbol{\Sigma}^{-1}$: is the precision matrix

- An entry $\boldsymbol{\Sigma}^{-1}_{ij} = 0$ indicates that the variables i and j are conditionally independent given all the other variables.

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{|2\pi\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

$\boldsymbol{\Sigma}$: is the covariance matrix

$\boldsymbol{\Sigma}^{-1}$: is the precision matrix

- An entry $\boldsymbol{\Sigma}_{ij}^{-1} = 0$ indicates that the variables i and j are conditionally independent given all the other variables.
- An entry $\boldsymbol{\Sigma}_{ij} = 0$ indicates that the variables i and j are marginally independent given all the other variables.

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{|2\pi\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

$\boldsymbol{\Sigma}$: is the covariance matrix

$\boldsymbol{\Sigma}^{-1}$: is the precision matrix

- An entry $\boldsymbol{\Sigma}^{-1}_{ij} = 0$ indicates that the variables i and j are conditionally independent given all the other variables.
- An entry $\boldsymbol{\Sigma}_{ij} = 0$ indicates that the variables i and j are marginally independent given all the other variables.
- Marginalizing out a variable leaves $\boldsymbol{\Sigma}$ unchanged but changes $\boldsymbol{\Sigma}^{-1}$.
  - This is crucial when parameterizing a Gaussian process.

Gaussian Quiz

Data : $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$, $\mathbf{x} \in \mathbb{R}^D$, $y \in \mathbb{R}$

Input : $(\mathbf{X})_{D \times N}$, Targets: $(\mathbf{y})_{N \times 1}$

Goal : $\mathbf{x} \stackrel{f(\mathbf{x})}{\rightarrow} \mathbf{y}$

The Standard Linear Regression Model

Notation and Settings

Data : $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$, $\mathbf{x} \in \mathbb{R}^D$, $y \in \mathbb{R}$

Input : $(\mathbf{X})_{D \times N}$, Targets: $(\mathbf{y})_{N \times 1}$

Goal : $\mathbf{x} \overset{f(\mathbf{x})}{\rightarrow} \mathbf{y}$

Model	$f(\mathbf{x}) = \sum_{j=1}^{D} w_j x_j$	$= \mathbf{w}^T \mathbf{x}$
Noise	$y = f(\mathbf{x}) + \eta$	with $\eta \sim \mathcal{N}(\eta \vert 0, \sigma^2)$
Likelihood	$y \vert f(\mathbf{x}) \sim \mathcal{N}(y \vert f(\mathbf{x}), \sigma^2)$	$= \mathcal{N}(y \vert \mathbf{w}^T \mathbf{x}, \sigma^2)$

The Standard Linear Regression Model

Notation and Settings

$$\text{Data} : \mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}, \ \mathbf{x} \in \mathbb{R}^D, \ y \in \mathbb{R}$$

$$\text{Input} : (\mathbf{X})_{D \times N}, \ \text{Targets: } (\mathbf{y})_{N \times 1}$$

$$\text{Goal} : \mathbf{x} \overset{f(\mathbf{x})}{\rightarrow} \mathbf{y}$$

$$
\begin{aligned}
\text{Model} \quad & f(\mathbf{x}) = \textstyle\sum_{j=1}^{D} w_j x_j && = \mathbf{w}^T \mathbf{x} \\
\text{Noise} \quad & y = f(\mathbf{x}) + \eta && \text{with } \eta \sim \mathcal{N}(\eta | 0, \sigma^2) \\
\text{Likelihood} \quad & y | f(\mathbf{x}) \sim \mathcal{N}(y | f(\mathbf{x}), \sigma^2) && = \mathcal{N}(y | \mathbf{w}^T \mathbf{x}, \sigma^2)
\end{aligned}
$$

Thus, the data-likelihood is given by:

$$
\begin{aligned}
p(\mathbf{y}|\mathbf{X}, \mathbf{w}) &= \prod_{n=1}^{N} p(y^{(n)} | \mathbf{x}^{(n)}, \mathbf{w}) = \prod_{n=1}^{N} \mathcal{N}(y^{(n)} | \mathbf{w}^T \mathbf{x}^{(n)}, \sigma^2) \\
&= \mathcal{N}(\mathbf{y} | \mathbf{X}^T \mathbf{w}, \sigma^2 \mathbf{I})
\end{aligned}
$$

$$\text{Data} : \mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N},\ \mathbf{x} \in \mathbb{R}^D,\ y \in \mathbb{R}$$

$$\text{Input} : (\mathbf{X})_{D \times N},\ \text{Targets: } (\mathbf{y})_{N \times 1}$$

$$\text{Goal} : \mathbf{x} \stackrel{f(\mathbf{x})}{\rightarrow} \mathbf{y}$$

$$\begin{aligned}
&\text{Model} & f(\mathbf{x}) &= \textstyle\sum_{j=1}^{D} w_j x_j & &= \mathbf{w}^T\mathbf{x} \\
&\text{Noise} & y &= f(\mathbf{x}) + \eta & &\text{with } \eta \sim \mathcal{N}(\eta|0, \sigma^2) \\
&\text{Likelihood} & y|f(\mathbf{x}) &\sim \mathcal{N}(y|f(\mathbf{x}), \sigma^2) & &= \mathcal{N}(y|\mathbf{w}^T\mathbf{x}, \sigma^2)
\end{aligned}$$

Thus, the data-likelihood is given by:

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} p(y^{(n)}|\mathbf{x}^{(n)}, \mathbf{w}) = \prod_{n=1}^{N} \mathcal{N}(y^{(n)}|\mathbf{w}^T\mathbf{x}^{(n)}, \sigma^2)$$
$$= \mathcal{N}(\mathbf{y}|\mathbf{X}^T\mathbf{w}, \sigma^2\mathbf{I})$$

We need do to inference on $\mathbf{w}$.

Consider a zero-mean Gaussian prior over the weights:

$$\mathbf{w} \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{\Sigma}_w)$$

Consider a zero-mean Gaussian prior over the weights:

$$\mathbf{w} \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \boldsymbol{\Sigma}_w)$$

Then the posterior distribution over the weights is given by:

$$\begin{aligned} p(\mathbf{w}|\mathbf{X}, \mathbf{y}) &= \frac{p(\mathbf{w})\ p(\mathbf{y}|\mathbf{X}, \mathbf{w})}{p(\mathbf{y}|\mathbf{X})} \\ &= \mathcal{N}(\mathbf{w}|\bar{\mathbf{w}}, \mathbf{A}^{-1}) \end{aligned}$$

where $\bar{\mathbf{w}} = \frac{1}{\sigma^2}\mathbf{A}^{-1}\mathbf{X}\mathbf{y}$, and $\mathbf{A} = (\frac{1}{\sigma^2}\mathbf{X}\mathbf{X}^T + \boldsymbol{\Sigma}_w^{-1})$.

Consider a zero-mean Gaussian prior over the weights:

$$\mathbf{w} \sim \mathcal{N}(\mathbf{w} | \mathbf{0}, \mathbf{\Sigma}_w)$$

Then the posterior distribution over the weights is given by:

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{w}) \; p(\mathbf{y}|\mathbf{X}, \mathbf{w})}{p(\mathbf{y}|\mathbf{X})} = \mathcal{N}(\mathbf{w}|\bar{\mathbf{w}}, \mathbf{A}^{-1})$$

where $\bar{\mathbf{w}} = \frac{1}{\sigma^2}\mathbf{A}^{-1}\mathbf{X}\mathbf{y}$, and $\mathbf{A} = (\frac{1}{\sigma^2}\mathbf{X}\mathbf{X}^T + \mathbf{\Sigma}_w^{-1})$.

- Mean of posterior is equal to its mode

Consider a zero-mean Gaussian prior over the weights:

$$\mathbf{w} \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{\Sigma}_w)$$

Then the posterior distribution over the weights is given by:

$$\begin{aligned} p(\mathbf{w}|\mathbf{X}, \mathbf{y}) &= \frac{p(\mathbf{w}) \; p(\mathbf{y}|\mathbf{X}, \mathbf{w})}{p(\mathbf{y}|\mathbf{X})} \\ &= \mathcal{N}(\mathbf{w}|\bar{\mathbf{w}}, \mathbf{A}^{-1}) \end{aligned}$$

where $\bar{\mathbf{w}} = \frac{1}{\sigma^2}\mathbf{A}^{-1}\mathbf{X}\mathbf{y}$, and $\mathbf{A} = (\frac{1}{\sigma^2}\mathbf{X}\mathbf{X}^T + \mathbf{\Sigma}_w^{-1})$.

- Mean of posterior is equal to its mode
- MAP solution (non-Bayesian): negative log prior as penalty term

Consider a zero-mean Gaussian prior over the weights:

$$\mathbf{w} \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{\Sigma}_w)$$

Then the posterior distribution over the weights is given by:

$$p(\mathbf{w}|\mathbf{X},\mathbf{y}) = \frac{p(\mathbf{w}) \; p(\mathbf{y}|\mathbf{X},\mathbf{w})}{p(\mathbf{y}|\mathbf{X})} = \mathcal{N}(\mathbf{w}|\bar{\mathbf{w}}, \mathbf{A}^{-1})$$

where $\bar{\mathbf{w}} = \frac{1}{\sigma^2}\mathbf{A}^{-1}\mathbf{X}\mathbf{y}$, and $\mathbf{A} = (\frac{1}{\sigma^2}\mathbf{X}\mathbf{X}^T + \mathbf{\Sigma}_w^{-1})$.

- Mean of posterior is equal to its mode
- MAP solution (non-Bayesian): negative log prior as penalty term
- This penalized maximum likelihood is known as **ridge regression**
  - Consider $\mathbf{\Sigma}_w = \lambda\mathbf{I}$ Then :

$$\bar{\mathbf{w}} = (\mathbf{X}\mathbf{X}^T + \frac{1}{\lambda}\sigma^2\mathbf{I})^{-1}\mathbf{X}\mathbf{y}$$

We are interested in making predictions at a new test point $\mathbf{x}_*$

- In fact we obtain the predictive distribution by averaging over all possible parameter values (weighted by their posterior probabilities):

$$p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \int p(f_*|\mathbf{x}_*, \mathbf{w}) p(\mathbf{w}|\mathbf{X}, \mathbf{y}) \, d\mathbf{w} = \mathcal{N}(f_*|\mathbf{x}_*^T \bar{\mathbf{w}}, \mathbf{x}_*^T \mathbf{A}^{-1} \mathbf{x}_*)$$

  - Predictive mean: linear combination of weights' posterior mean
  - Predictive variance: grows with the magnitude of the test point

We are interested in making predictions at a new test point $\mathbf{x}_*$

- In fact we obtain the predictive distribution by averaging over all possible parameter values (weighted by their posterior probabilities):

$$p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \int p(f_*|\mathbf{x}_*, \mathbf{w})p(\mathbf{w}|\mathbf{X}, \mathbf{y}) \; d\mathbf{w} = \mathcal{N}(f_*|\mathbf{x}_*^T \bar{\mathbf{w}}, \mathbf{x}_*^T \mathbf{A}^{-1} \mathbf{x}_*)$$

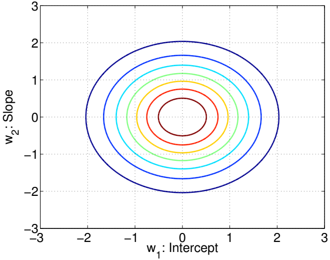
  - Predictive mean: linear combination of weights' posterior mean
  - Predictive variance: grows with the magnitude of the test point

- Point predictions: Need to consider the expected loss (or **risk**):

$$y_{\text{opt}} = \underset{y_{\text{pred}}}{\text{argmin}} \int \mathcal{L}(f_*, y_{\text{pred}}) p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) df_*$$

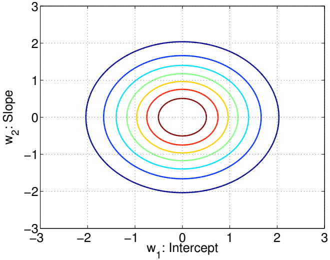
  - e.g. Square loss $\mathcal{L} = (y_{\text{pred}} - f_*)^2$
  - c.f. Empirical risk minimization (ERM)

Bayesian Linear Regression Example

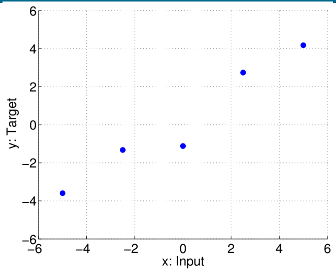


Prior Weights

Bayesian Linear Regression Example

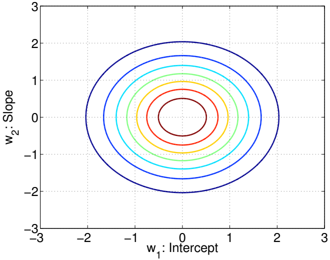


Prior Weights

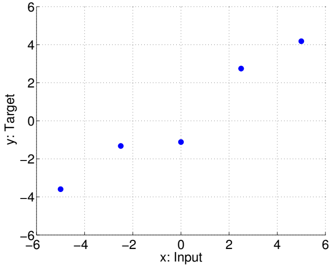


Observed Data

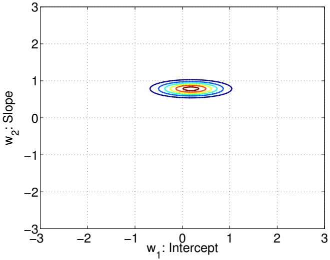
Bayesian Linear Regression Example



Prior Weights

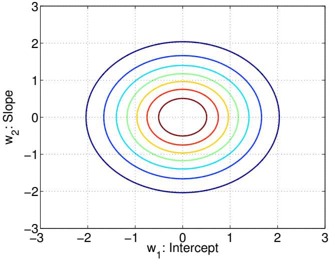


Observed Data

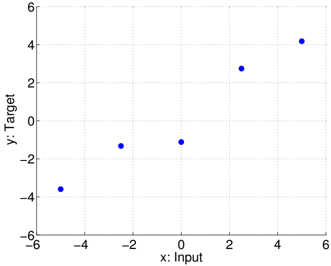


Likelihood

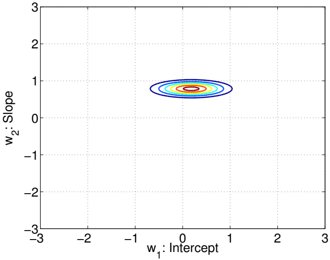
Bayesian Linear Regression Example



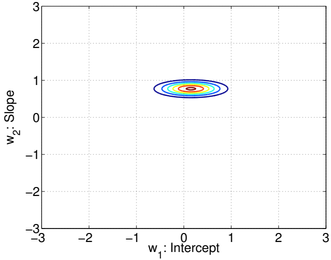
Prior Weights



Observed Data

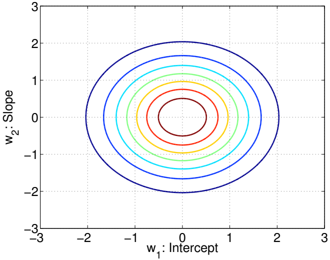


Likelihood

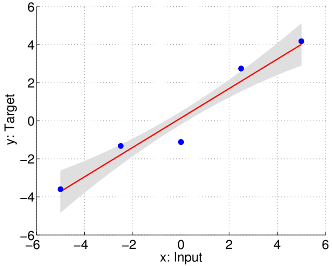


Posterior Weights

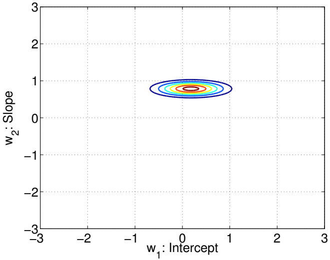
Bayesian Linear Regression Example



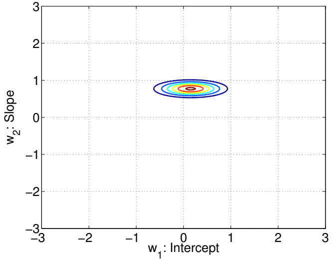
Prior Weights



Predictive Distribution



Likelihood



Posterior Weights

Non-linear Feature Spaces

- Consider the model $f(\mathbf{x}) = \sum_{i=1}^{D'} w_i \phi_i(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$
  - Each $\phi_i(\mathbf{x})$ is a (non-linear) feature on $\mathbf{x}$, e.g. $x_1, x_2, x_1^2, x_2^2, x_1 x_2 \ldots$
  - We have a non-linear mapping but a linear-in-the-parameters model
  - The number of these features can be very large, i.e. $D' \gg D$

Non-linear Feature Spaces

- Consider the model $f(\mathbf{x}) = \sum_{i=1}^{D'} w_i \phi_i(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$
  - Each $\phi_i(\mathbf{x})$ is a (non-linear) feature on $\mathbf{x}$, e.g. $x_1, x_2, x_1^2, x_2^2, x_1 x_2 \ldots$
  - We have a non-linear mapping but a linear-in-the-parameters model
  - The number of these features can be very large, i.e. $D' \gg D$
- All the Bayesian analysis is similar to the standard linear model:

$$p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(f_* | \sigma^{-2} \phi_*^T \mathbf{A}^{-1} \mathbf{\Phi} \mathbf{y}, \phi_*^T \mathbf{A}^{-1} \phi_*)$$

where: $\phi_* = \phi(\mathbf{x}_*)$, $\mathbf{\Phi} = \mathbf{\Phi}(\mathbf{X})$, and $\mathbf{A} = (\frac{1}{\sigma^2} \mathbf{\Phi} \mathbf{\Phi}^T + \mathbf{\Sigma}_w^{-1})$

Non-linear Feature Spaces

- Consider the model $f(\mathbf{x}) = \sum_{i=1}^{D'} w_i \phi_i(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$
  - Each $\phi_i(\mathbf{x})$ is a (non-linear) feature on $\mathbf{x}$, e.g. $x_1, x_2, x_1^2, x_2^2, x_1 x_2 \ldots$
  - We have a non-linear mapping but a linear-in-the-parameters model
  - The number of these features can be very large, i.e. $D' \gg D$
- All the Bayesian analysis is similar to the standard linear model:

$$p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(f_* | \sigma^{-2} \phi_*^T \mathbf{A}^{-1} \mathbf{\Phi} \mathbf{y}, \phi_*^T \mathbf{A}^{-1} \phi_*)$$

where: $\phi_* = \phi(\mathbf{x}_*)$, $\mathbf{\Phi} = \mathbf{\Phi}(\mathbf{X})$, and $\mathbf{A} = (\frac{1}{\sigma^2} \mathbf{\Phi} \mathbf{\Phi}^T + \mathbf{\Sigma}_w^{-1})$

- Note we need to invert $\mathbf{A}$ of ? dimensions.

- Consider the model $f(\mathbf{x}) = \sum_{i=1}^{D'} w_i \phi_i(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$
  - Each $\phi_i(\mathbf{x})$ is a (non-linear) feature on $\mathbf{x}$, e.g. $x_1, x_2, x_1^2, x_2^2, x_1 x_2 \ldots$
  - We have a non-linear mapping but a linear-in-the-parameters model
  - The number of these features can be very large, i.e. $D' \gg D$
- All the Bayesian analysis is similar to the standard linear model:

$$p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(f_* | \sigma^{-2} \phi_*^T \mathbf{A}^{-1} \mathbf{\Phi} \mathbf{y}, \phi_*^T \mathbf{A}^{-1} \phi_*)$$

where: $\phi_* = \phi(\mathbf{x}_*)$, $\mathbf{\Phi} = \mathbf{\Phi}(\mathbf{X})$, and $\mathbf{A} = (\frac{1}{\sigma^2} \mathbf{\Phi}\mathbf{\Phi}^T + \mathbf{\Sigma}_w^{-1})$

  - Note we need to invert $\mathbf{A}$ of ? dimensions.
- We can rewrite the predictive distribution as:

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

where $\mathbf{k}_* = \mathbf{\Phi}^T \mathbf{\Sigma}_w \phi_*$, $k_{\star\star} = \phi_*^T \mathbf{\Sigma}_w \phi_*$, and $\widetilde{\mathbf{K}} = \mathbf{\Phi}^T \mathbf{\Sigma}_w \mathbf{\Phi} + \sigma^2 \mathbf{I}$

Non-linear Feature Spaces

- Consider the model $f(\mathbf{x}) = \sum_{i=1}^{D'} w_i \phi_i(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$
  - Each $\phi_i(\mathbf{x})$ is a (non-linear) feature on $\mathbf{x}$, e.g. $x_1, x_2, x_1^2, x_2^2, x_1 x_2 \ldots$
  - We have a non-linear mapping but a linear-in-the-parameters model
  - The number of these features can be very large, i.e. $D' \gg D$
- All the Bayesian analysis is similar to the standard linear model:

$$p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(f_* | \sigma^{-2} \phi_*^T \mathbf{A}^{-1} \mathbf{\Phi} \mathbf{y}, \phi_*^T \mathbf{A}^{-1} \phi_*)$$

where: $\phi_* = \phi(\mathbf{x}_*)$, $\mathbf{\Phi} = \mathbf{\Phi}(\mathbf{X})$, and $\mathbf{A} = (\frac{1}{\sigma^2} \mathbf{\Phi} \mathbf{\Phi}^T + \mathbf{\Sigma}_w^{-1})$

  - Note we need to invert $\mathbf{A}$ of ? dimensions.
- We can rewrite the predictive distribution as:

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

where $\mathbf{k}_* = \mathbf{\Phi}^T \mathbf{\Sigma}_w \phi_*$, $k_{\star\star} = \phi_*^T \mathbf{\Sigma}_w \phi_*$, and $\widetilde{\mathbf{K}} = \mathbf{\Phi}^T \mathbf{\Sigma}_w \mathbf{\Phi} + \sigma^2 \mathbf{I}$

  - Now we need to invert $\widetilde{\mathbf{K}}$ of ? dimensions ▸ GP prediction

The Kernel Trick

- Note that in:

$$\mathbf{k}_* = \boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \phi_*, \ k_{\star\star} = \phi_*^T \boldsymbol{\Sigma}_w \phi_* \text{ and } \widetilde{\mathbf{K}} = \boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi} + \sigma^2 \mathbf{I}$$

  the features always enter in the form $\phi(\mathbf{x})^T \boldsymbol{\Sigma}_w \phi(\mathbf{x}')$

The Kernel Trick

- Note that in:

$$\mathbf{k}_* = \boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \phi_*, \; k_{\star\star} = \phi_*^T \boldsymbol{\Sigma}_w \phi_* \text{ and } \widetilde{\mathbf{K}} = \boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi} + \sigma^2 \mathbf{I}$$

  the features always enter in the form $\phi(\mathbf{x})^T \boldsymbol{\Sigma}_w \phi(\mathbf{x}')$
- This is an inner product wrt $\boldsymbol{\Sigma}_w$

The Kernel Trick

- Note that in:

$$\mathbf{k}_* = \boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \phi_*, \; k_{\star\star} = \phi_*^T \boldsymbol{\Sigma}_w \phi_* \text{ and } \widetilde{\mathbf{K}} = \boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi} + \sigma^2 \mathbf{I}$$

  the features always enter in the form $\phi(\mathbf{x})^T \boldsymbol{\Sigma}_w \phi(\mathbf{x}')$
- This is an inner product wrt $\boldsymbol{\Sigma}_w$
- As $\boldsymbol{\Sigma}_w$ is PD we can rewrite:

$$\begin{aligned}
\phi(\mathbf{x})^T \boldsymbol{\Sigma}_w \phi(\mathbf{x}') &= \phi(\mathbf{x})^T \boldsymbol{\Sigma}_w^{1/2} \boldsymbol{\Sigma}_w^{1/2} \phi(\mathbf{x}') \\
&= (\underbrace{\boldsymbol{\Sigma}_w^{1/2} \phi(\mathbf{x})}_{\psi(\mathbf{x})})^T (\underbrace{\boldsymbol{\Sigma}_w^{1/2} \phi(\mathbf{x}')}_{\psi(\mathbf{x}')}) \\
\kappa(\mathbf{x}, \mathbf{x}') &= \boldsymbol{\psi}(\mathbf{x}) \cdot \boldsymbol{\psi}(\mathbf{x}')
\end{aligned}$$

The Kernel Trick

- Note that in:

$$\mathbf{k}_* = \boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \phi_*, \; k_{\star\star} = \phi_*^T \boldsymbol{\Sigma}_w \phi_* \text{ and } \widetilde{\mathbf{K}} = \boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi} + \sigma^2 \mathbf{I}$$

  the features always enter in the form $\phi(\mathbf{x})^T \boldsymbol{\Sigma}_w \phi(\mathbf{x}')$
- This is an inner product wrt $\boldsymbol{\Sigma}_w$
- As $\boldsymbol{\Sigma}_w$ is PD we can rewrite:

$$\begin{aligned}
\phi(\mathbf{x})^T \boldsymbol{\Sigma}_w \phi(\mathbf{x}') &= \phi(\mathbf{x})^T \boldsymbol{\Sigma}_w^{1/2} \boldsymbol{\Sigma}_w^{1/2} \phi(\mathbf{x}') \\
&= (\underbrace{\boldsymbol{\Sigma}_w^{1/2} \phi(\mathbf{x})}_{\psi(\mathbf{x})})^T (\underbrace{\boldsymbol{\Sigma}_w^{1/2} \phi(\mathbf{x}')}_{\psi(\mathbf{x}')}) \\
\kappa(\mathbf{x}, \mathbf{x}') &= \boldsymbol{\psi}(\mathbf{x}) \cdot \boldsymbol{\psi}(\mathbf{x}')
\end{aligned}$$

- $\kappa(\cdot, \cdot)$ is called a kernel or covariance function

The Kernel Trick

- Note that in:

$$\mathbf{k}_* = \mathbf{\Phi}^T \mathbf{\Sigma}_w \phi_*, \; k_{\star\star} = \phi_*^T \mathbf{\Sigma}_w \phi_* \text{ and } \widetilde{\mathbf{K}} = \mathbf{\Phi}^T \mathbf{\Sigma}_w \mathbf{\Phi} + \sigma^2 \mathbf{I}$$

  the features always enter in the form $\phi(\mathbf{x})^T \mathbf{\Sigma}_w \phi(\mathbf{x}')$
- This is an inner product wrt $\mathbf{\Sigma}_w$
- As $\mathbf{\Sigma}_w$ is PD we can rewrite:

$$\begin{aligned}
\phi(\mathbf{x})^T \mathbf{\Sigma}_w \phi(\mathbf{x}') &= \phi(\mathbf{x})^T \mathbf{\Sigma}_w^{1/2} \mathbf{\Sigma}_w^{1/2} \phi(\mathbf{x}') \\
&= (\underbrace{\mathbf{\Sigma}_w^{1/2} \phi(\mathbf{x})}_{\psi(\mathbf{x})})^T (\underbrace{\mathbf{\Sigma}_w^{1/2} \phi(\mathbf{x}')}_{\psi(\mathbf{x}')}) \\
\kappa(\mathbf{x}, \mathbf{x}') &= \boldsymbol{\psi}(\mathbf{x}) \cdot \boldsymbol{\psi}(\mathbf{x}')
\end{aligned}$$

- $\kappa(\cdot, \cdot)$ is called a kernel or covariance function
- We can replace all occurrences of inner products by $\kappa(\cdot, \cdot)$

The Kernel Trick

- Note that in:

$$\mathbf{k}_* = \mathbf{\Phi}^T \mathbf{\Sigma}_w \phi_*, \; k_{\star\star} = \phi_*^T \mathbf{\Sigma}_w \phi_* \text{ and } \widetilde{\mathbf{K}} = \mathbf{\Phi}^T \mathbf{\Sigma}_w \mathbf{\Phi} + \sigma^2 \mathbf{I}$$

  the features always enter in the form $\phi(\mathbf{x})^T \mathbf{\Sigma}_w \phi(\mathbf{x}')$
- This is an inner product wrt $\mathbf{\Sigma}_w$
- As $\mathbf{\Sigma}_w$ is PD we can rewrite:

$$\begin{aligned}
\phi(\mathbf{x})^T \mathbf{\Sigma}_w \phi(\mathbf{x}') &= \phi(\mathbf{x})^T \mathbf{\Sigma}_w^{1/2} \mathbf{\Sigma}_w^{1/2} \phi(\mathbf{x}') \\
&= (\underbrace{\mathbf{\Sigma}_w^{1/2} \phi(\mathbf{x})}_{\psi(\mathbf{x})})^T (\underbrace{\mathbf{\Sigma}_w^{1/2} \phi(\mathbf{x}')}_{\psi(\mathbf{x}')}) \\
\kappa(\mathbf{x}, \mathbf{x}') &= \boldsymbol{\psi}(\mathbf{x}) \cdot \boldsymbol{\psi}(\mathbf{x}')
\end{aligned}$$

- $\kappa(\cdot, \cdot)$ is called a kernel or covariance function
- We can replace all occurrences of inner products by $\kappa(\cdot, \cdot)$
- We do not need to compute the feature vectors explicitly

From a Prior over Weights to a Prior over Functions

- Consider the kinds of functions that can be generated from a set of basis functions with random weights.

From a Prior over Weights to a Prior over Functions

- Consider the kinds of functions that can be generated from a set of basis functions with random weights.
- Then $f(\mathbf{x})$ at a particular point is a random variable:

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) \text{ with } \mathbf{w} \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{\Sigma}_w)$$

defined as a linear combination of Gaussian random variables.

From a Prior over Weights to a Prior over Functions

- Consider the kinds of functions that can be generated from a set of basis functions with random weights.
- Then $f(\mathbf{x})$ at a particular point is a random variable:

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) \text{ with } \mathbf{w} \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{\Sigma}_w)$$

  defined as a linear combination of Gaussian random variables.
- A collection of these random variables indexed by $\mathbf{x}$: $f(\mathbf{x}^{(1)}), \ldots, f(\mathbf{x}^{(N)})$, define a stochastic process in a consistent way.

From a Prior over Weights to a Prior over Functions

- Consider the kinds of functions that can be generated from a set of basis functions with random weights.
- Then $f(\mathbf{x})$ at a particular point is a random variable:

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) \text{ with } \mathbf{w} \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{\Sigma}_w)$$

  defined as a linear combination of Gaussian random variables.
- A collection of these random variables indexed by $\mathbf{x}$: $f(\mathbf{x}^{(1)}), \ldots, f(\mathbf{x}^{(N)})$, define a stochastic process in a consistent way.
- The mean and the covariance function for this stochastic process is given by:

$$\mathbb{E}_{\mathbf{w}}[f(\mathbf{x})] = 0$$
$$\mathbb{E}_{\mathbf{w}}\left[f(\mathbf{x})f(\mathbf{x}')\right] = \phi^T(\mathbf{x})\mathbf{\Sigma}_w\phi(\mathbf{x}')$$

From a Prior over Weights to a Prior over Functions

- Consider the kinds of functions that can be generated from a set of basis functions with random weights.
- Then $f(\mathbf{x})$ at a particular point is a random variable:

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) \text{ with } \mathbf{w} \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{\Sigma}_w)$$

  defined as a linear combination of Gaussian random variables.
- A collection of these random variables indexed by $\mathbf{x}$: $f(\mathbf{x}^{(1)}), \ldots, f(\mathbf{x}^{(N)})$, define a stochastic process in a consistent way.
- The mean and the covariance function for this stochastic process is given by:

$$\mathbb{E}_{\mathbf{w}}[f(\mathbf{x})] = 0$$
$$\mathbb{E}_{\mathbf{w}}\left[f(\mathbf{x})f(\mathbf{x}')\right] = \phi^T(\mathbf{x})\mathbf{\Sigma}_w\phi(\mathbf{x}')$$

- The Bayesian linear model is a Gaussian process

From a Prior over Weights to a Prior over Functions

- Consider the kinds of functions that can be generated from a set of basis functions with random weights.
- Then $f(\mathbf{x})$ at a particular point is a random variable:

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) \text{ with } \mathbf{w} \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{\Sigma}_w)$$

  defined as a linear combination of Gaussian random variables.
- A collection of these random variables indexed by $\mathbf{x}$: $f(\mathbf{x}^{(1)}), \ldots, f(\mathbf{x}^{(N)})$, define a stochastic process in a consistent way.
- The mean and the covariance function for this stochastic process is given by:

$$\mathbb{E}_{\mathbf{w}}[f(\mathbf{x})] = 0$$

$$\mathbb{E}_{\mathbf{w}}\left[f(\mathbf{x})f(\mathbf{x}')\right] = \phi^T(\mathbf{x})\mathbf{\Sigma}_w\phi(\mathbf{x}')$$

- The Bayesian linear model is a Gaussian process
  - The Function values corresponding to any number of inputs have a joint Gaussian distribution.

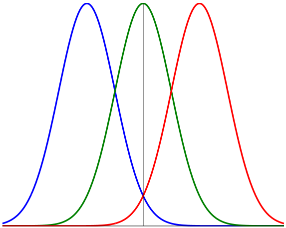
Sample Functions from the Linear Model

1. Define $\phi_i(x) = \exp(-\frac{1}{2}(x - \mu_i)^2)$, for $i = 1, 2, 3$
2. Construct $\Phi(i, j) = \phi_i(x^{(j)})$, for $i = 1, 2, 3$
3. Draw $\mathbf{w} \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{I})$
4. Draw $\mathbf{f} = \mathbf{\Phi}^T\mathbf{w}$

Sample Functions from the Linear Model

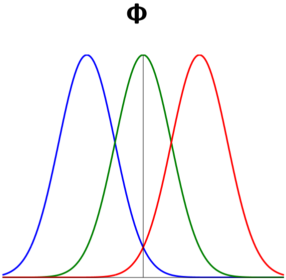
1. Define $\phi_i(x) = \exp(-\frac{1}{2}(x - \mu_i)^2)$, for $i = 1, 2, 3$
2. Construct $\Phi(i, j) = \phi_i(x^{(j)})$, for $i = 1, 2, 3$
3. Draw $\mathbf{w} \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{I})$
4. Draw $\mathbf{f} = \mathbf{\Phi}^T\mathbf{w}$

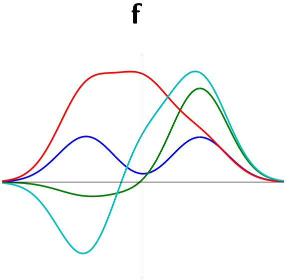
$\mathbf{\Phi}$



Sample Functions from the Linear Model

1. Define $\phi_i(x) = \exp(-\frac{1}{2}(x - \mu_i)^2)$, for $i = 1, 2, 3$
2. Construct $\Phi(i, j) = \phi_i(x^{(j)})$, for $i = 1, 2, 3$
3. Draw $\mathbf{w} \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{I})$
4. Draw $\mathbf{f} = \mathbf{\Phi}^T\mathbf{w}$





Function-space View

Gaussian Process (GP)

$f(\mathbf{x})$ is a Gaussian process if for any finite subset of points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$, the function values $f(\mathbf{x}^{(1)}), \ldots, f(\mathbf{x}^{(N)})$ follow a Gaussian distribution.

$$
\begin{aligned}
f(\mathbf{x}) &\sim \mathcal{GP}(\mu(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})), \\
\mu(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})], \\
\kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) &= \mathbb{E}\big[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))\big],
\end{aligned}
$$

$\mu(\mathbf{x})$: mean function, consider $\mu(\mathbf{x}) \equiv \mathbf{0}$

$\kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$: parameterized covariance function, notion of similarity

Function-space View

Gaussian Process (GP)

$f(\mathbf{x})$ is a Gaussian process if for any finite subset of points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$, the function values $f(\mathbf{x}^{(1)}), \ldots, f(\mathbf{x}^{(N)})$ follow a Gaussian distribution.

$$
\begin{aligned}
f(\mathbf{x}) &\sim \mathcal{GP}(\mu(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})), \\
\mu(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})], \\
\kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) &= \mathbb{E}\left[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))\right],
\end{aligned}
$$

$\mu(\mathbf{x})$: mean function, consider $\mu(\mathbf{x}) \equiv \mathbf{0}$

$\kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$: parameterized covariance function, notion of similarity

- Stochastic process: collection of random variables

Function-space View

Gaussian Process (GP)

$f(\mathbf{x})$ is a Gaussian process if for any finite subset of points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$, the function values $f(\mathbf{x}^{(1)}), \ldots, f(\mathbf{x}^{(N)})$ follow a Gaussian distribution.

$$\begin{aligned} f(\mathbf{x}) &\sim \mathcal{GP}(\mu(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})), \\ \mu(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})], \\ \kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) &= \mathbb{E}\left[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))\right], \end{aligned}$$

$\mu(\mathbf{x})$: mean function, consider $\mu(\mathbf{x}) \equiv \mathbf{0}$

$\kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$: parameterized covariance function, notion of similarity

- Stochastic process: collection of random variables
- These variables are the values of the function $f(\mathbf{x})$ **indexed** by the set of all possible input

Function-space View

Gaussian Process (GP)

$f(\mathbf{x})$ is a Gaussian process if for any finite subset of points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$, the function values $f(\mathbf{x}^{(1)}), \ldots, f(\mathbf{x}^{(N)})$ follow a Gaussian distribution.

$$
\begin{aligned}
f(\mathbf{x}) &\sim \mathcal{GP}(\mu(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})), \\
\mu(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})], \\
\kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) &= \mathbb{E}\left[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))\right],
\end{aligned}
$$

$\mu(\mathbf{x})$: mean function, consider $\mu(\mathbf{x}) \equiv \mathbf{0}$

$\kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$: parameterized covariance function, notion of similarity

- Stochastic process: collection of random variables
- These variables are the values of the function $f(\mathbf{x})$ **indexed** by the set of all possible input
- Consistency: marginalization property
  $(f_1, f_2) \sim \mathcal{N}(\mathbf{f} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) \rightarrow f_1 \sim \mathcal{N}(f_1 | \mu_1, \Sigma_{11})$

The Covariance Function

- It specifies the covariance between pairs of random variables:

$$\mathbb{C}\text{ov}(f(\mathbf{x}^{(p)}), f(\mathbf{x}^{(q)})) = \kappa(\mathbf{x}^{(p)}, \mathbf{x}^{(q)})$$

The Covariance Function

- It specifies the covariance between pairs of random variables:

$$\mathbb{C}\text{ov}(f(\mathbf{x}^{(p)}), f(\mathbf{x}^{(q)})) = \kappa(\mathbf{x}^{(p)}, \mathbf{x}^{(q)})$$

  - Covariance between outputs as a function of the inputs

The Covariance Function

- It specifies the covariance between pairs of random variables:

$$\mathbb{C}\mathrm{ov}(f(\mathbf{x}^{(p)}), f(\mathbf{x}^{(q)})) = \kappa(\mathbf{x}^{(p)}, \mathbf{x}^{(q)})$$

  - Covariance between outputs as a function of the inputs
  - A crucial component in GPs

The Covariance Function

- It specifies the covariance between pairs of random variables:

$$\mathbb{C}\text{ov}(f(\mathbf{x}^{(p)}), f(\mathbf{x}^{(q)})) = \kappa(\mathbf{x}^{(p)}, \mathbf{x}^{(q)})$$

  - Covariance between outputs as a function of the inputs
  - A crucial component in GPs
  - Intuitively, it describes the notion of similarity

The Covariance Function

- It specifies the covariance between pairs of random variables:

$$\mathbb{C}\text{ov}(f(\mathbf{x}^{(p)}), f(\mathbf{x}^{(q)})) = \kappa(\mathbf{x}^{(p)}, \mathbf{x}^{(q)})$$

  - Covariance between outputs as a function of the inputs
  - A crucial component in GPs
  - Intuitively, it describes the notion of similarity
  - It can be parametrized and we can learn its hyperparameters from data

The Covariance Function

- It specifies the covariance between pairs of random variables:

$$\mathbb{C}\text{ov}(f(\mathbf{x}^{(p)}), f(\mathbf{x}^{(q)})) = \kappa(\mathbf{x}^{(p)}, \mathbf{x}^{(q)})$$

  - Covariance between outputs as a function of the inputs
  - A crucial component in GPs
  - Intuitively, it describes the notion of similarity
  - It can be parametrized and we can learn its hyperparameters from data
- The matrix $\mathbf{K}$ such that $K_{i,j} = \kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ at all pairwise input points is known as the covariance matrix or Gram matrix.

The Covariance Function

- It specifies the covariance between pairs of random variables:

$$\mathbb{C}\text{ov}(f(\mathbf{x}^{(p)}), f(\mathbf{x}^{(q)})) = \kappa(\mathbf{x}^{(p)}, \mathbf{x}^{(q)})$$

  - Covariance between outputs as a function of the inputs
  - A crucial component in GPs
  - Intuitively, it describes the notion of similarity
  - It can be parametrized and we can learn its hyperparameters from data
- The matrix $\mathbf{K}$ such that $K_{i,j} = \kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ at all pairwise input points is known as the covariance matrix or Gram matrix.
- it must generate a positive semidefinite (PSD) matrix at any subset of points, i.e. $\mathbf{b}^T\mathbf{K}\mathbf{b} \geq 0$, $\forall \mathbf{b} \in \mathbb{R}^N$

The Covariance Function

- It specifies the covariance between pairs of random variables:

$$\mathbb{C}\text{ov}(f(\mathbf{x}^{(p)}), f(\mathbf{x}^{(q)})) = \kappa(\mathbf{x}^{(p)}, \mathbf{x}^{(q)})$$

  - Covariance between outputs as a function of the inputs
  - A crucial component in GPs
  - Intuitively, it describes the notion of similarity
  - It can be parametrized and we can learn its hyperparameters from data
- The matrix $\mathbf{K}$ such that $K_{i,j} = \kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ at all pairwise input points is known as the covariance matrix or Gram matrix.
- it must generate a positive semidefinite (PSD) matrix at any subset of points, i.e. $\mathbf{b}^T\mathbf{K}\mathbf{b} \geq 0$, $\forall \mathbf{b} \in \mathbb{R}^N$
- Stationary: $\varphi(\mathbf{x} - \mathbf{x}')$ - translation invariant

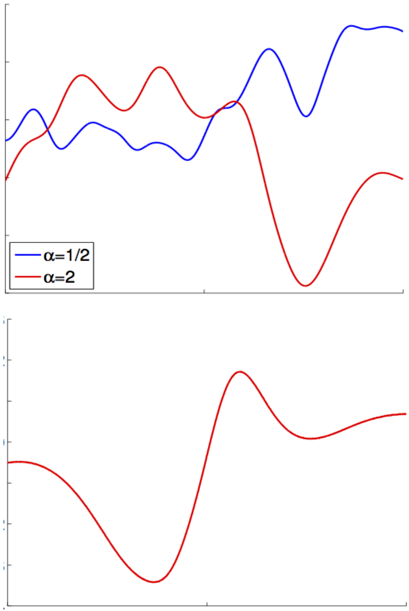
The Covariance Function

- It specifies the covariance between pairs of random variables:

$$\mathbb{C}\mathrm{ov}(f(\mathbf{x}^{(p)}), f(\mathbf{x}^{(q)})) = \kappa(\mathbf{x}^{(p)}, \mathbf{x}^{(q)})$$

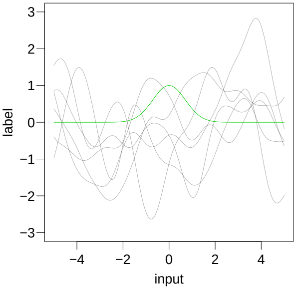
  - Covariance between outputs as a function of the inputs
  - A crucial component in GPs
  - Intuitively, it describes the notion of similarity
  - It can be parametrized and we can learn its hyperparameters from data
- The matrix $\mathbf{K}$ such that $K_{i,j} = \kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ at all pairwise input points is known as the covariance matrix or Gram matrix.
- it must generate a positive semidefinite (PSD) matrix at any subset of points, i.e. $\mathbf{b}^T\mathbf{K}\mathbf{b} \geq 0$, $\forall \mathbf{b} \in \mathbb{R}^N$
- Stationary: $\varphi(\mathbf{x} - \mathbf{x}')$ - translation invariant
- Isotropic: $\varphi(\|\mathbf{x} - \mathbf{x}'\|)$

Samples from a Gaussian Process

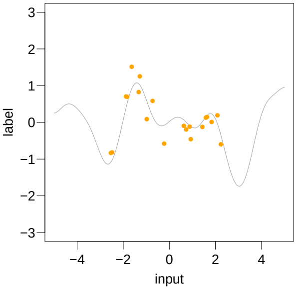


Computing with Infinite Vectors

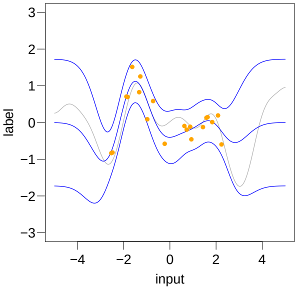
GP prior

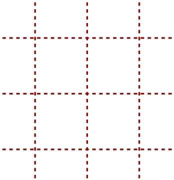


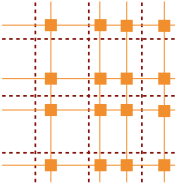
GP regression example

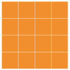


Inference result



$K_\infty =$ 

$K_\infty =$ 

$K_{\mathbf{y}} =$  $+$ 

The Squared Exponential (SE) Covariance Function

$$\kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) = \sigma_s^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \mathbf{C}(\mathbf{x} - \mathbf{x}')\right)$$

- σ_s^2 is the signal variance

The Squared Exponential (SE) Covariance Function

$$\kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) = \sigma_s^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \mathbf{C} (\mathbf{x} - \mathbf{x}')\right)$$

- σ_s^2 is the signal variance
- $\mathbf{C}$ is a symmetric matrix that can have different parameterizations

The Squared Exponential (SE) Covariance Function

$$\kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) = \sigma_s^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \mathbf{C} (\mathbf{x} - \mathbf{x}')\right)$$

- σ_s^2 is the signal variance
- $\mathbf{C}$ is a symmetric matrix that can have different parameterizations
- $\mathbf{C} = \ell^{-2}\mathbf{I}$: isotropic SE

The Squared Exponential (SE) Covariance Function

$$\kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) = \sigma_s^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \mathbf{C} (\mathbf{x} - \mathbf{x}')\right)$$

- σ_s^2 is the signal variance
- $\mathbf{C}$ is a symmetric matrix that can have different parameterizations
- $\mathbf{C} = \ell^{-2}\mathbf{I}$: isotropic SE
- $\mathbf{C} = \mathrm{diag}(\boldsymbol{\ell})^{-2}$ with $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_D)$: Automatic Relevance Determination (ARD)

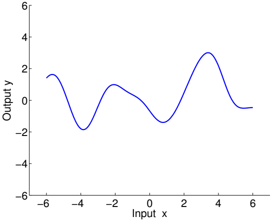
The Squared Exponential (SE) Covariance Function

$$\kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) = \sigma_s^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \mathbf{C}(\mathbf{x} - \mathbf{x}')\right)$$

- σ_s^2 is the signal variance
- $\mathbf{C}$ is a symmetric matrix that can have different parameterizations
- $\mathbf{C} = \ell^{-2}\mathbf{I}$: isotropic SE
- $\mathbf{C} = \text{diag}(\boldsymbol{\ell})^{-2}$ with $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_D)$: Automatic Relevance Determination (ARD)
- Each ℓ_j is known as the characteristic length-scale: distance for which the function values are expected to vary significantly

The Squared Exponential (SE) Covariance Function

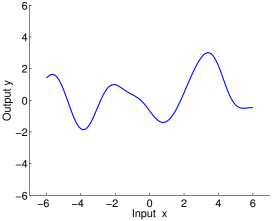
Example



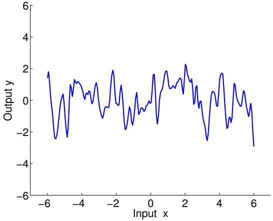
$$\ell = 1,\ \sigma_s^2 = 1$$

The Squared Exponential (SE) Covariance Function

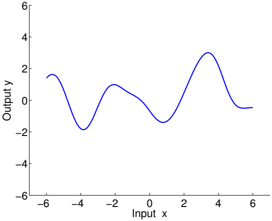
Example



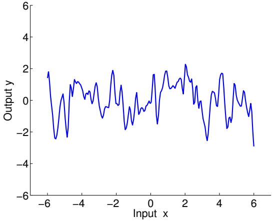
$\ell = 1,\ \sigma_s^2 = 1$



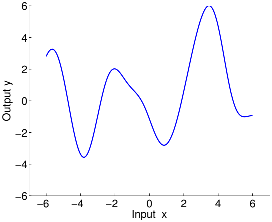
$\ell = 0.1,\ \sigma_s^2 = 1$



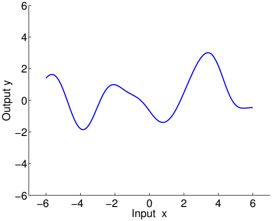
$\ell = 1,\ \sigma_s^2 = 1$



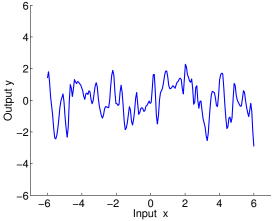
$\ell = 0.1,\ \sigma_s^2 = 1$



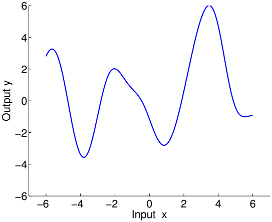
$\ell = 1,\ \sigma_s^2 = 4$



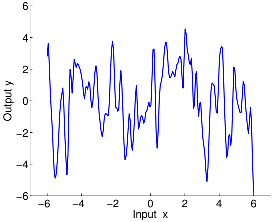
$\ell = 1,\ \sigma_s^2 = 1$



$\ell = 0.1,\ \sigma_s^2 = 1$



$\ell = 1,\ \sigma_s^2 = 4$



$\ell = 0.1,\ \sigma_s^2 = 4$

Standard GP Regression Model: Predictions (1)

Data : $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$, $\mathbf{x} \in \mathbb{R}^D$, $y \in \mathbb{R}$

Input : $(\mathbf{X})_{D \times N}$, Targets: $(\mathbf{y})_{N \times 1}$

Goal : Make predictions $f_* = f(\mathbf{x}_*)$ at $\mathbf{x}_*$

Standard GP Regression Model: Predictions (1)

Data : $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$, $\mathbf{x} \in \mathbb{R}^D$, $y \in \mathbb{R}$

Input : $(\mathbf{X})_{D \times N}$, Targets: $(\mathbf{y})_{N \times 1}$

Goal : Make predictions $f_* = f(\mathbf{x}_*)$ at $\mathbf{x}_*$

$$\text{Prior} \quad f(\mathbf{x}) \quad \sim \quad \mathcal{GP}(\mathbf{0}, \kappa(\mathbf{x}, \mathbf{x}'))$$

Standard GP Regression Model: Predictions (1)

Data : $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$, $\mathbf{x} \in \mathbb{R}^D$, $y \in \mathbb{R}$

Input : $(\mathbf{X})_{D \times N}$, Targets: $(\mathbf{y})_{N \times 1}$

Goal : Make predictions $f_* = f(\mathbf{x}_*)$ at $\mathbf{x}_*$

$$
\begin{array}{llcll}
\text{Prior} & f(\mathbf{x}) & \sim & \mathcal{GP}(\mathbf{0}, \kappa(\mathbf{x}, \mathbf{x}')) & \\
\text{Noise} & y & = & f(\mathbf{x}) + \eta & \eta \sim \mathcal{N}(0, \sigma_n^2)
\end{array}
$$

Standard GP Regression Model: Predictions (1)

Data : $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$, $\mathbf{x} \in \mathbb{R}^D$, $y \in \mathbb{R}$

Input : $(\mathbf{X})_{D \times N}$, Targets: $(\mathbf{y})_{N \times 1}$

Goal : Make predictions $f_* = f(\mathbf{x}_*)$ at $\mathbf{x}_*$

$$
\begin{array}{llcll}
\text{Prior} & f(\mathbf{x}) & \sim & \mathcal{GP}(\mathbf{0}, \kappa(\mathbf{x}, \mathbf{x}')) & \\
\text{Noise} & y & = & f(\mathbf{x}) + \eta & \eta \sim \mathcal{N}(0, \sigma_n^2)
\end{array}
$$

- The joint distribution of $\mathbf{y}$ and f_* is a Gaussian

Standard GP Regression Model: Predictions (1)

Data $: \mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$, $\mathbf{x} \in \mathbb{R}^D$, $y \in \mathbb{R}$

Input $: (\mathbf{X})_{D \times N}$, Targets: $(\mathbf{y})_{N \times 1}$

Goal : Make predictions $f_* = f(\mathbf{x}_*)$ at $\mathbf{x}_*$

$$
\begin{array}{llcl}
\text{Prior} & f(\mathbf{x}) & \sim & \mathcal{GP}(\mathbf{0}, \kappa(\mathbf{x}, \mathbf{x}')) \\
\text{Noise} & y & = & f(\mathbf{x}) + \eta \qquad \eta \sim \mathcal{N}(0, \sigma_n^2)
\end{array}
$$

- The joint distribution of $\mathbf{y}$ and f_* is a Gaussian
- We simply need to figure out the covariance structure:
  $\mathbb{C}\mathrm{ov}(y^{(p)}, y^{(q)}) = \kappa(\mathbf{x}^{(p)}, \mathbf{x}^{(q)}) + \sigma_n^2 \delta_{pq} \rightarrow \mathbb{C}\mathrm{ov}(\mathbf{y}) = \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}$

Standard GP Regression Model: Predictions (1)

Data : $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$, $\mathbf{x} \in \mathbb{R}^D$, $y \in \mathbb{R}$

Input : $(\mathbf{X})_{D \times N}$, Targets: $(\mathbf{y})_{N \times 1}$

Goal : Make predictions $f_* = f(\mathbf{x}_*)$ at $\mathbf{x}_*$

$$
\begin{array}{llcll}
\text{Prior} & f(\mathbf{x}) & \sim & \mathcal{GP}(\mathbf{0}, \kappa(\mathbf{x}, \mathbf{x}')) & \\
\text{Noise} & y & = & f(\mathbf{x}) + \eta & \eta \sim \mathcal{N}(0, \sigma_n^2)
\end{array}
$$

- The joint distribution of $\mathbf{y}$ and f_* is a Gaussian
- We simply need to figure out the covariance structure: $\mathbb{C}\mathrm{ov}(y^{(p)}, y^{(q)}) = \kappa(\mathbf{x}^{(p)}, \mathbf{x}^{(q)}) + \sigma_n^2 \delta_{pq} \rightarrow \mathbb{C}\mathrm{ov}(\mathbf{y}) = \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}$
- To get the posterior on f_* we need to constrain this distribution to agree with the observed data $(\mathbf{X}, \mathbf{y})$

Standard GP Regression Model: Predictions (1)

Data : $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$, $\mathbf{x} \in \mathbb{R}^D$, $y \in \mathbb{R}$

Input : $(\mathbf{X})_{D \times N}$, Targets: $(\mathbf{y})_{N \times 1}$

Goal : Make predictions $f_* = f(\mathbf{x}_*)$ at $\mathbf{x}_*$

$$
\begin{array}{llcll}
\text{Prior} & f(\mathbf{x}) & \sim & \mathcal{GP}(\mathbf{0}, \kappa(\mathbf{x}, \mathbf{x}')) & \\
\text{Noise} & y & = & f(\mathbf{x}) + \eta & \eta \sim \mathcal{N}(0, \sigma_n^2)
\end{array}
$$

- The joint distribution of $\mathbf{y}$ and f_* is a Gaussian
- We simply need to figure out the covariance structure: $\mathbb{C}\text{ov}(y^{(p)}, y^{(q)}) = \kappa(\mathbf{x}^{(p)}, \mathbf{x}^{(q)}) + \sigma_n^2 \delta_{pq} \rightarrow \mathbb{C}\text{ov}(\mathbf{y}) = \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}$
- To get the posterior on f_* we need to constrain this distribution to agree with the observed data $(\mathbf{X}, \mathbf{y})$
- This is achieved simply by conditioning: $p(f_* | \mathbf{X}, \mathbf{y}, \mathbf{x}_*)$

Standard GP Regression Model: Predictions (2)

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{array}{cc} \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2\mathbf{I} & \mathbf{k}(\mathbf{X},\mathbf{x}_*) \\ \mathbf{k}(\mathbf{x}_*,\mathbf{X}) & \kappa(\mathbf{x}_*\mathbf{x}_*) \end{array}\right)$$

Standard GP Regression Model: Predictions (2)

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{matrix} \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2\mathbf{I} & \mathbf{k}(\mathbf{X},\mathbf{x}_*) \\ \mathbf{k}(\mathbf{x}_*,\mathbf{X}) & \kappa(\mathbf{x}_*\mathbf{x}_*) \end{matrix}\right)$$

Denoting $\mathbf{k}_* = \mathbf{K}(\mathbf{X}, \mathbf{x}_*)$ and $\widetilde{\mathbf{K}} = \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2\mathbf{I}$

Standard GP Regression Model: Predictions (2)

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2\mathbf{I} & \mathbf{k}(\mathbf{X},\mathbf{x}_*) \\ \mathbf{k}(\mathbf{x}_*,\mathbf{X}) & \kappa(\mathbf{x}_*\mathbf{x}_*) \end{pmatrix}\right)$$

Denoting $\mathbf{k}_* = \mathbf{K}(\mathbf{X},\mathbf{x}_*)$ and $\widetilde{\mathbf{K}} = \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2\mathbf{I}$ then:

$$\begin{aligned} f_*|\mathbf{X},\mathbf{y},\mathbf{x}_* &\sim \mathcal{N}(\mathbb{E}[f_*],\mathbb{V}[f_*]), \\ \mathbb{E}[f_*] &= \mathbf{k}_*^T\widetilde{\mathbf{K}}^{-1}\mathbf{y}, \\ \mathbb{V}[f_*] &= \kappa(\mathbf{x}_*,\mathbf{x}_*) - \mathbf{k}_*^T\widetilde{\mathbf{K}}^{-1}\mathbf{k}_*. \end{aligned}$$

Standard GP Regression Model: Predictions (2)

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2\mathbf{I} & \mathbf{k}(\mathbf{X},\mathbf{x}_*) \\ \mathbf{k}(\mathbf{x}_*,\mathbf{X}) & \kappa(\mathbf{x}_*\mathbf{x}_*) \end{pmatrix}\right)$$

Denoting $\mathbf{k}_* = \mathbf{K}(\mathbf{X}, \mathbf{x}_*)$ and $\widetilde{\mathbf{K}} = \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2\mathbf{I}$ then:

$$\begin{aligned} f_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_* &\sim \mathcal{N}(\mathbb{E}[f_*], \mathbb{V}[f_*]), \\ \mathbb{E}[f_*] &= \mathbf{k}_*^T \widetilde{\mathbf{K}}^{-1}\mathbf{y}, \\ \mathbb{V}[f_*] &= \kappa(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T \widetilde{\mathbf{K}}^{-1}\mathbf{k}_*. \end{aligned}$$

- $\mathbb{E}[f_*]$: Linear combination of N observations, i.e. linear predictor

Standard GP Regression Model: Predictions (2)

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{matrix} \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2 \mathbf{I} & \mathbf{k}(\mathbf{X},\mathbf{x}_*) \\ \mathbf{k}(\mathbf{x}_*,\mathbf{X}) & \kappa(\mathbf{x}_* \mathbf{x}_*) \end{matrix}\right)$$

Denoting $\mathbf{k}_* = \mathbf{K}(\mathbf{X}, \mathbf{x}_*)$ and $\widetilde{\mathbf{K}} = \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2 \mathbf{I}$ then:

$$\begin{aligned} f_* | \mathbf{X}, \mathbf{y}, \mathbf{x}_* &\sim \mathcal{N}(\mathbb{E}[f_*], \mathbb{V}[f_*]), \\ \mathbb{E}[f_*] &= \mathbf{k}_*^T \widetilde{\mathbf{K}}^{-1} \mathbf{y}, \\ \mathbb{V}[f_*] &= \kappa(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T \widetilde{\mathbf{K}}^{-1} \mathbf{k}_*. \end{aligned}$$

- $\mathbb{E}[f_*]$: Linear combination of N observations, i.e. linear predictor
- Say $\boldsymbol{\alpha} = (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}$ then: $\mathbb{E}[f_*] = \sum_{n=1}^{N} \alpha_i \kappa(\mathbf{x}^{(n)}, \mathbf{x}_*)$ is a linear combination of N kernel functions: Representer theorem

Standard GP Regression Model: Predictions (2)

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{array}{cc} \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2\mathbf{I} & \mathbf{k}(\mathbf{X},\mathbf{x}_*) \\ \mathbf{k}(\mathbf{x}_*,\mathbf{X}) & \kappa(\mathbf{x}_*\mathbf{x}_*) \end{array}\right)$$

Denoting $\mathbf{k}_* = \mathbf{K}(\mathbf{X},\mathbf{x}_*)$ and $\widetilde{\mathbf{K}} = \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2\mathbf{I}$ then:

$$\begin{aligned} f_*|\mathbf{X},\mathbf{y},\mathbf{x}_* &\sim \mathcal{N}(\mathbb{E}[f_*], \mathbb{V}[f_*]), \\ \mathbb{E}[f_*] &= \mathbf{k}_*^T \widetilde{\mathbf{K}}^{-1}\mathbf{y}, \\ \mathbb{V}[f_*] &= \kappa(\mathbf{x}_*,\mathbf{x}_*) - \mathbf{k}_*^T\widetilde{\mathbf{K}}^{-1}\mathbf{k}_*. \end{aligned}$$

- $\mathbb{E}[f_*]$: Linear combination of N observations, i.e. linear predictor
- Say $\boldsymbol{\alpha} = (\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}$ then: $\mathbb{E}[f_*] = \sum_{n=1}^{N} \alpha_i \kappa(\mathbf{x}^{(n)}, \mathbf{x}_*)$ is a linear combination of N kernel functions: Representer theorem
- We encountered this predictive distribution before ▸ Go to Linear Model

Standard GP Regression Model: Predictions (2)

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{matrix} \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2 \mathbf{I} & \mathbf{k}(\mathbf{X}, \mathbf{x}_*) \\ \mathbf{k}(\mathbf{x}_*, \mathbf{X}) & \kappa(\mathbf{x}_* \mathbf{x}_*) \end{matrix} \right)$$

Denoting $\mathbf{k}_* = \mathbf{K}(\mathbf{X}, \mathbf{x}_*)$ and $\widetilde{\mathbf{K}} = \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2 \mathbf{I}$ then:

$$\begin{aligned} f_* | \mathbf{X}, \mathbf{y}, \mathbf{x}_* &\sim \mathcal{N}(\mathbb{E}[f_*], \mathbb{V}[f_*]), \\ \mathbb{E}[f_*] &= \mathbf{k}_*^T \widetilde{\mathbf{K}}^{-1} \mathbf{y}, \\ \mathbb{V}[f_*] &= \kappa(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T \widetilde{\mathbf{K}}^{-1} \mathbf{k}_*. \end{aligned}$$

- $\mathbb{E}[f_*]$: Linear combination of N observations, i.e. linear predictor
- Say $\boldsymbol{\alpha} = (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}$ then: $\mathbb{E}[f_*] = \sum_{n=1}^{N} \alpha_i \kappa(\mathbf{x}^{(n)}, \mathbf{x}_*)$ is a linear combination of N kernel functions: Representer theorem
- We encountered this predictive distribution before Go to Linear Model
- $\mathbb{V}[f_*]$ does not depend on $\mathbf{y}$

Standard GP Regression Model: Predictions (2)

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{array}{cc} \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2\mathbf{I} & \mathbf{k}(\mathbf{X},\mathbf{x}_*) \\ \mathbf{k}(\mathbf{x}_*,\mathbf{X}) & \kappa(\mathbf{x}_*\mathbf{x}_*) \end{array}\right)$$

Denoting $\mathbf{k}_* = \mathbf{K}(\mathbf{X},\mathbf{x}_*)$ and $\widetilde{\mathbf{K}} = \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2\mathbf{I}$ then:

$$\begin{aligned} f_*|\mathbf{X},\mathbf{y},\mathbf{x}_* &\sim \mathcal{N}(\mathbb{E}[f_*], \mathbb{V}[f_*]), \\ \mathbb{E}[f_*] &= \mathbf{k}_*^T \widetilde{\mathbf{K}}^{-1}\mathbf{y}, \\ \mathbb{V}[f_*] &= \kappa(\mathbf{x}_*,\mathbf{x}_*) - \mathbf{k}_*^T \widetilde{\mathbf{K}}^{-1}\mathbf{k}_*. \end{aligned}$$

- $\mathbb{E}[f_*]$: Linear combination of N observations, i.e. linear predictor
- Say $\boldsymbol{\alpha} = (\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}$ then: $\mathbb{E}[f_*] = \sum_{n=1}^{N} \alpha_i \kappa(\mathbf{x}^{(n)}, \mathbf{x}_*)$ is a linear combination of N kernel functions: Representer theorem
- We encountered this predictive distribution before Go to Linear Model
- $\mathbb{V}[f_*]$ does not depend on $\mathbf{y}$
- In fact we have a Gaussian posterior process

The Graphical Model for GPs

The Graphical Model for GPs

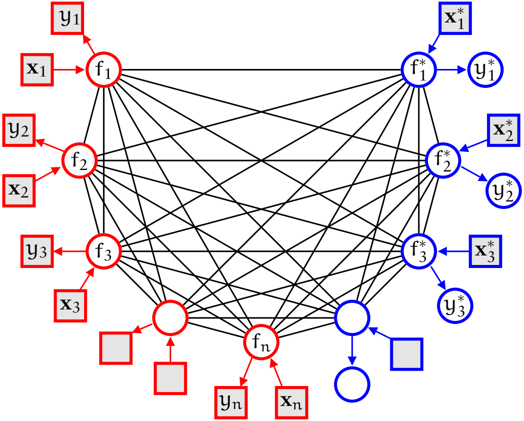


Figure from Carl Rasmussen's slides

The Graphical Model for GPs

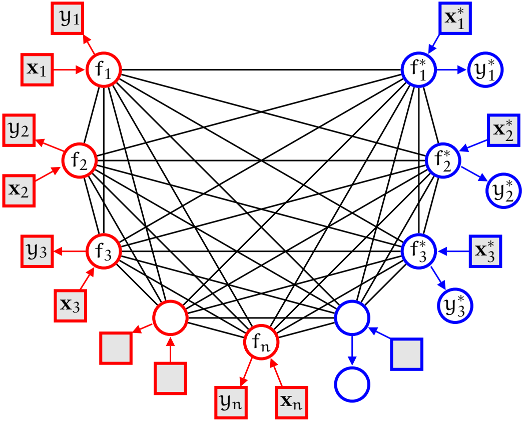


Figure from Carl Rasmussen's slides

- Observations y depend on their corresponding latent function f

The Graphical Model for GPs

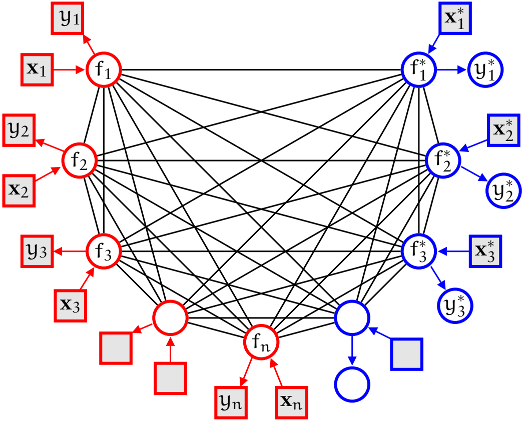


Figure from Carl Rasmussen's slides

- Observations y depend on their corresponding latent function f
- The marginalization property implies that adding a new $\mathbf{x}_i^*$, f_i^*, y_i^* does not affect the distribution

1. The Gaussian Distribution Revisited
2. Standard Bayesian Linear Regression
3. Bayesian Regression with Non-linear Feature Spaces
4. Gaussian Processes for Regression
   - Function-space View
   - Predictions
   - **Model Selection**
   - Other Covariance Functions

Model Selection

- It includes the discrete choice of the functional form for the covariance function and the values for the hyper-parameters.

Model Selection

- It includes the discrete choice of the functional form for the covariance function and the values for the hyper-parameters.
- E.g. for the SE: $\kappa(\mathbf{x}, \mathbf{x}') = \sigma_s^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \mathbf{C} (\mathbf{x} - \mathbf{x}')^T\right)$ the parameters are σ_s^2 and the parameters of $\mathbf{C}$

Model Selection

- It includes the discrete choice of the functional form for the covariance function and the values for the hyper-parameters.
- E.g. for the SE: $\kappa(\mathbf{x}, \mathbf{x}') = \sigma_s^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \mathbf{C} (\mathbf{x} - \mathbf{x}')^T\right)$ the parameters are σ_s^2 and the parameters of $\mathbf{C}$
- However, we will refer to the set of hyper-parameters $\boldsymbol{\theta}$ as the parameters of the covariance and the noise variance σ_n^2

Model Selection

- It includes the discrete choice of the functional form for the covariance function and the values for the hyper-parameters.
- E.g. for the SE: $\kappa(\mathbf{x}, \mathbf{x}') = \sigma_s^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \mathbf{C} (\mathbf{x} - \mathbf{x}')^T\right)$ the parameters are σ_s^2 and the parameters of $\mathbf{C}$
- However, we will refer to the set of hyper-parameters $\boldsymbol{\theta}$ as the parameters of the covariance and the noise variance σ_n^2
- We can do cross-validation (potential problems?)

Model Selection

- It includes the discrete choice of the functional form for the covariance function and the values for the hyper-parameters.
- E.g. for the SE: $\kappa(\mathbf{x}, \mathbf{x}') = \sigma_s^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \mathbf{C} (\mathbf{x} - \mathbf{x}')^T\right)$ the parameters are σ_s^2 and the parameters of $\mathbf{C}$
- However, we will refer to the set of hyper-parameters $\boldsymbol{\theta}$ as the parameters of the covariance and the noise variance σ_n^2
- We can do cross-validation (potential problems?)
- We focus here on the so-called type II maximum likelihood, i.e. we want to maximize the marginal likelihood.

Model Selection

- It includes the discrete choice of the functional form for the covariance function and the values for the hyper-parameters.
- E.g. for the SE: $\kappa(\mathbf{x}, \mathbf{x}') = \sigma_s^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \mathbf{C} (\mathbf{x} - \mathbf{x}')^T\right)$ the parameters are σ_s^2 and the parameters of $\mathbf{C}$
- However, we will refer to the set of hyper-parameters $\boldsymbol{\theta}$ as the parameters of the covariance and the noise variance σ_n^2
- We can do cross-validation (potential problems?)
- We focus here on the so-called type II maximum likelihood, i.e. we want to maximize the marginal likelihood.
- Integrate out the "parameters" of the GP: (which parameters?)

Model Selection

- It includes the discrete choice of the functional form for the covariance function and the values for the hyper-parameters.
- E.g. for the SE: $\kappa(\mathbf{x}, \mathbf{x}') = \sigma_s^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \mathbf{C} (\mathbf{x} - \mathbf{x}')^T\right)$ the parameters are σ_s^2 and the parameters of $\mathbf{C}$
- However, we will refer to the set of hyper-parameters $\boldsymbol{\theta}$ as the parameters of the covariance and the noise variance σ_n^2
- We can do cross-validation (potential problems?)
- We focus here on the so-called type II maximum likelihood, i.e. we want to maximize the marginal likelihood.
- Integrate out the "parameters" of the GP: (which parameters?)

$$
\begin{aligned}
p(\mathbf{y}|\mathbf{X}, \theta) &= \int p(\mathbf{y}|\mathbf{f}, \mathbf{X}, \theta) p(\mathbf{f}|\mathbf{X}, \theta) d\mathbf{f} \\
&= \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K} + \sigma^2 \mathbf{I})
\end{aligned}
$$

Log Marginal Likelihood

$$\begin{aligned}\mathcal{L} &= \log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) \\ &= \underbrace{-\frac{1}{2}\mathbf{y}^T(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}}_{\text{data-fit}} - \underbrace{\frac{1}{2}\log|\mathbf{K} + \sigma_n^2\mathbf{I}|}_{\text{complexity}} - \underbrace{\frac{N}{2}\log 2\pi}_{normaliz.}\end{aligned}$$

Log Marginal Likelihood

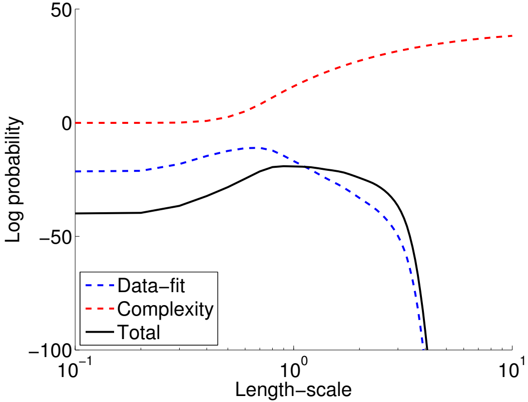
$$\begin{aligned}\mathcal{L} &= \log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) \\ &= \underbrace{-\frac{1}{2}\mathbf{y}^T(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}}_{\text{data-fit}} - \underbrace{\frac{1}{2}\log|\mathbf{K} + \sigma_n^2\mathbf{I}|}_{\text{complexity}} - \underbrace{\frac{N}{2}\log 2\pi}_{\textit{normaliz.}}\end{aligned}$$

- Isotropic SE
- $\sigma_s^2 = 1$, $\sigma_n^2 = 0.01$
- $\ell = 1$
- $N = 20$

Log Marginal Likelihood

$$\mathcal{L} = \log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$$
$$= \underbrace{-\frac{1}{2}\mathbf{y}^T(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}}_{\text{data-fit}} - \underbrace{\frac{1}{2}\log|\mathbf{K} + \sigma_n^2\mathbf{I}|}_{\text{complexity}} - \underbrace{\frac{N}{2}\log 2\pi}_{\textit{normaliz.}}$$

- Isotropic SE
- $\sigma_s^2 = 1$, $\sigma_n^2 = 0.01$
- $\ell = 1$
- $N = 20$



Hyper-parameter Learning

Let $\widetilde{\mathbf{K}} = \mathbf{K} + \sigma_n^2 \mathbf{I}$:

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \frac{1}{2} \mathbf{y}^T \widetilde{\mathbf{K}}^{-1} \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \widetilde{\mathbf{K}}^{-1} \mathbf{y} - \frac{1}{2} \operatorname{tr} \left(\widetilde{\mathbf{K}}^{-1} \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \right)$$
$$= \frac{1}{2} \operatorname{tr} \left((\boldsymbol{\alpha} \boldsymbol{\alpha}^T - \widetilde{\mathbf{K}}^{-1}) \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \right)$$

where $\boldsymbol{\alpha} = \widetilde{\mathbf{K}}^{-1} \mathbf{y}$.

Hyper-parameter Learning

Let $\widetilde{\mathbf{K}} = \mathbf{K} + \sigma_n^2 \mathbf{I}$:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \theta_i} &= \frac{1}{2} \mathbf{y}^T \widetilde{\mathbf{K}}^{-1} \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \widetilde{\mathbf{K}}^{-1} \mathbf{y} - \frac{1}{2} \operatorname{tr} \left(\widetilde{\mathbf{K}}^{-1} \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \right) \\
&= \frac{1}{2} \operatorname{tr} \left((\boldsymbol{\alpha} \boldsymbol{\alpha}^T - \widetilde{\mathbf{K}}^{-1}) \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \right)
\end{aligned}
$$

where $\boldsymbol{\alpha} = \widetilde{\mathbf{K}}^{-1} \mathbf{y}$.

- Can use gradient-based optimization

Hyper-parameter Learning

Let $\widetilde{\mathbf{K}} = \mathbf{K} + \sigma_n^2 \mathbf{I}$:

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \frac{1}{2}\mathbf{y}^T \widetilde{\mathbf{K}}^{-1} \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \widetilde{\mathbf{K}}^{-1}\mathbf{y} - \frac{1}{2} \operatorname{tr} \left(\widetilde{\mathbf{K}}^{-1} \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \right)$$

$$= \frac{1}{2} \operatorname{tr} \left((\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \widetilde{\mathbf{K}}^{-1}) \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \right)$$

where $\boldsymbol{\alpha} = \widetilde{\mathbf{K}}^{-1}\mathbf{y}$.

- Can use gradient-based optimization
- General approach and only needs derivatives of the covariance

Hyper-parameter Learning

Let $\widetilde{\mathbf{K}} = \mathbf{K} + \sigma_n^2 \mathbf{I}$:

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \frac{1}{2} \mathbf{y}^T \widetilde{\mathbf{K}}^{-1} \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \widetilde{\mathbf{K}}^{-1} \mathbf{y} - \frac{1}{2} \operatorname{tr} \left(\widetilde{\mathbf{K}}^{-1} \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \right)$$

$$= \frac{1}{2} \operatorname{tr} \left((\boldsymbol{\alpha} \boldsymbol{\alpha}^T - \widetilde{\mathbf{K}}^{-1}) \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \right)$$

where $\boldsymbol{\alpha} = \widetilde{\mathbf{K}}^{-1} \mathbf{y}$.

- Can use gradient-based optimization
- General approach and only needs derivatives of the covariance
- Such principled “kernel” learning does not exist in standard SVM

Hyper-parameter Learning

Let $\widetilde{\mathbf{K}} = \mathbf{K} + \sigma_n^2 \mathbf{I}$:

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \frac{1}{2} \mathbf{y}^T \widetilde{\mathbf{K}}^{-1} \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \widetilde{\mathbf{K}}^{-1} \mathbf{y} - \frac{1}{2} \operatorname{tr} \left(\widetilde{\mathbf{K}}^{-1} \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \right)$$

$$= \frac{1}{2} \operatorname{tr} \left((\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \widetilde{\mathbf{K}}^{-1}) \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \right)$$

where $\boldsymbol{\alpha} = \widetilde{\mathbf{K}}^{-1} \mathbf{y}$.

- Can use gradient-based optimization
- General approach and only needs derivatives of the covariance
- Such principled “kernel” learning does not exist in standard SVM
- Non-convex optimization

Hyper-parameter Learning

Let $\widetilde{\mathbf{K}} = \mathbf{K} + \sigma_n^2 \mathbf{I}$:

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \frac{1}{2} \mathbf{y}^T \widetilde{\mathbf{K}}^{-1} \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \widetilde{\mathbf{K}}^{-1} \mathbf{y} - \frac{1}{2} \operatorname{tr} \left(\widetilde{\mathbf{K}}^{-1} \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \right)$$

$$= \frac{1}{2} \operatorname{tr} \left((\boldsymbol{\alpha} \boldsymbol{\alpha}^T - \widetilde{\mathbf{K}}^{-1}) \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \right)$$

where $\boldsymbol{\alpha} = \widetilde{\mathbf{K}}^{-1} \mathbf{y}$.

- Can use gradient-based optimization
- General approach and only needs derivatives of the covariance
- Such principled "kernel" learning does not exist in standard SVM
- Non-convex optimization
- Multiple local optima correspond to different explanations of the data

Hyper-parameter Learning

Let $\widetilde{\mathbf{K}} = \mathbf{K} + \sigma_n^2 \mathbf{I}$:

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \frac{1}{2} \mathbf{y}^T \widetilde{\mathbf{K}}^{-1} \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \widetilde{\mathbf{K}}^{-1} \mathbf{y} - \frac{1}{2} \operatorname{tr} \left(\widetilde{\mathbf{K}}^{-1} \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \right)$$

$$= \frac{1}{2} \operatorname{tr} \left((\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \widetilde{\mathbf{K}}^{-1}) \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta_i} \right)$$

where $\boldsymbol{\alpha} = \widetilde{\mathbf{K}}^{-1} \mathbf{y}$.

- Can use gradient-based optimization
- General approach and only needs derivatives of the covariance
- Such principled "kernel" learning does not exist in standard SVM
- Non-convex optimization
- Multiple local optima correspond to different explanations of the data
- Computational Requirements?

Automatic Relevance Determination (ARD)

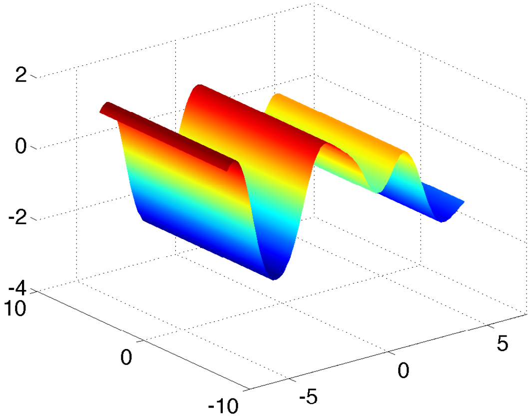
- Inverse of the length-scale determines the relevance of the dimension.

Automatic Relevance Determination (ARD)

- Inverse of the length-scale determines the relevance of the dimension.
- The larger the length-scale the more irrelevant the corresponding input is.

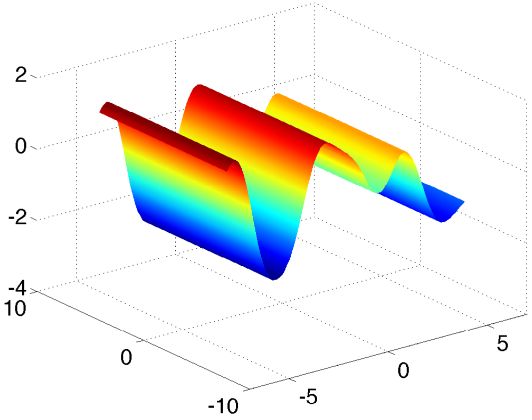
Automatic Relevance Determination (ARD)

- Inverse of the length-scale determines the relevance of the dimension.
- The larger the length-scale the more irrelevant the corresponding input is.



Automatic Relevance Determination (ARD)

- Inverse of the length-scale determines the relevance of the dimension.
- The larger the length-scale the more irrelevant the corresponding input is.



Learned lengh-scale for irrelevant dimension: 1.0557×10^5

1. The Gaussian Distribution Revisited

2. Standard Bayesian Linear Regression

3. Bayesian Regression with Non-linear Feature Spaces

4. Gaussian Processes for Regression
   - Function-space View
   - Predictions
   - Model Selection
   - Other Covariance Functions

Other Covariance Functions: Matérn Covariance

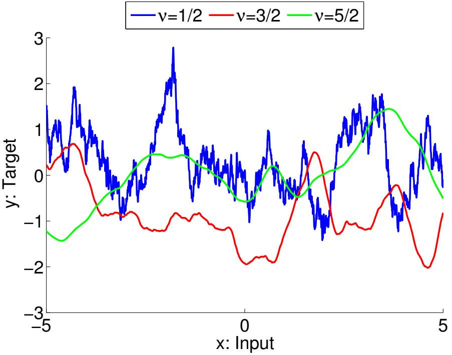
$$\kappa(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|\mathbf{x} - \mathbf{x}'\|}{\ell} \right)^{\nu} \mathcal{K}_{\nu} \left(\frac{\sqrt{2\nu}\|\mathbf{x} - \mathbf{x}'\|}{\ell} \right)$$

where $\mathcal{K}_{\nu}$ is a modified Bessel function and $\nu > 0$, $\ell > 0$.

Other Covariance Functions: Matérn Covariance

$$\kappa(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|\mathbf{x} - \mathbf{x}'\|}{\ell} \right)^{\nu} \mathcal{K}_{\nu} \left(\frac{\sqrt{2\nu}\|\mathbf{x} - \mathbf{x}'\|}{\ell} \right)$$

where $\mathcal{K}_{\nu}$ is a modified Bessel function and $\nu > 0$, $\ell > 0$.

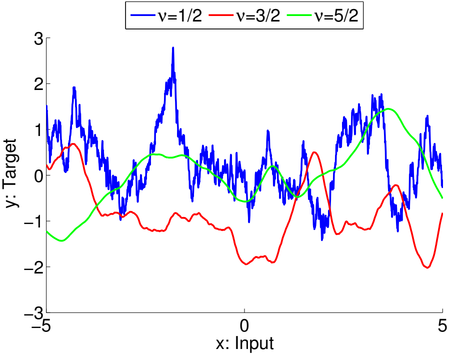


$\ell = 1$

Other Covariance Functions: Matérn Covariance

$$\kappa(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|\mathbf{x} - \mathbf{x}'\|}{\ell} \right)^{\nu} \mathcal{K}_{\nu} \left(\frac{\sqrt{2\nu}\|\mathbf{x} - \mathbf{x}'\|}{\ell} \right)$$

where $\mathcal{K}_{\nu}$ is a modified Bessel function and $\nu > 0$, $\ell > 0$.



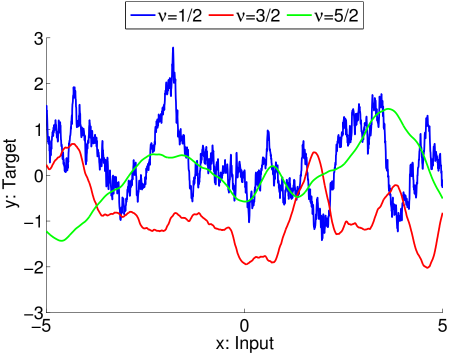
$\ell = 1$

- Stationary, Isotropic

Other Covariance Functions: Matérn Covariance

$$\kappa(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|\mathbf{x} - \mathbf{x}'\|}{\ell} \right)^{\nu} \mathcal{K}_{\nu} \left(\frac{\sqrt{2\nu}\|\mathbf{x} - \mathbf{x}'\|}{\ell} \right)$$

where $\mathcal{K}_{\nu}$ is a modified Bessel function and $\nu > 0$, $\ell > 0$.



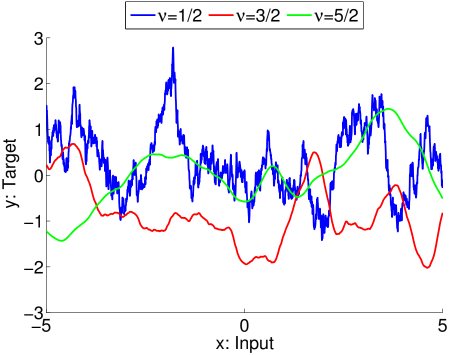
$\ell = 1$

- Stationary, Isotropic
- $\nu = 1/2$:
  $\kappa(\mathbf{x}, \mathbf{x}') = \exp(-\frac{|\mathbf{x}-\mathbf{x}'|}{\ell})$
  - Very rough process
  - Brownian motion
  - Ornstein-Uhlenbeck (D=1)

Other Covariance Functions: Matérn Covariance

$$\kappa(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|\mathbf{x} - \mathbf{x}'\|}{\ell} \right)^{\nu} \mathcal{K}_{\nu} \left(\frac{\sqrt{2\nu}\|\mathbf{x} - \mathbf{x}'\|}{\ell} \right)$$

where $\mathcal{K}_{\nu}$ is a modified Bessel function and $\nu > 0$, $\ell > 0$.



$\ell = 1$

- Stationary, Isotropic
- $\nu = 1/2$:
  $\kappa(\mathbf{x}, \mathbf{x}') = \exp(-\frac{|\mathbf{x}-\mathbf{x}'|}{\ell})$
  - Very rough process
  - Brownian motion
  - Ornstein-Uhlenbeck (D=1)
- $\nu \to \infty$: SE covariance

Other Covariance Functions: Rational Quadratic

$$\kappa(\mathbf{x}, \mathbf{x}') = \left(1 + \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\alpha\ell^2}\right)^{-\alpha}$$

with $\alpha > 0$, $\ell > 0$.

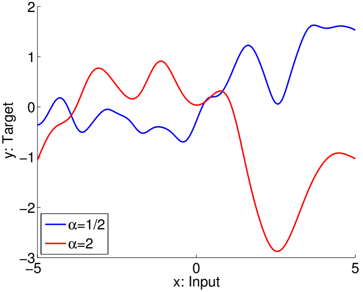
can be seen as an infinite sum of squared exponential (SE) covariance functions with different characteristic length-scales.

Other Covariance Functions: Rational Quadratic

$$\kappa(\mathbf{x}, \mathbf{x}') = \left(1 + \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\alpha\ell^2}\right)^{-\alpha}$$

with $\alpha > 0$, $\ell > 0$.
can be seen as an infinite sum of squared exponential (SE) covariance functions with different characteristic length-scales.

$\ell = 1$

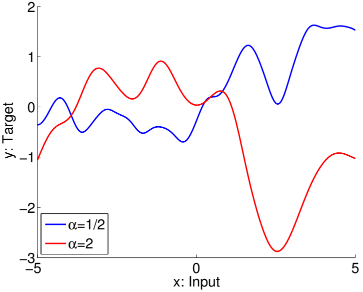


Other Covariance Functions: Rational Quadratic

$$\kappa(\mathbf{x}, \mathbf{x}') = \left(1 + \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\alpha\ell^2}\right)^{-\alpha}$$

with $\alpha > 0$, $\ell > 0$.

can be seen as an infinite sum of squared exponential (SE) covariance functions with different characteristic length-scales.



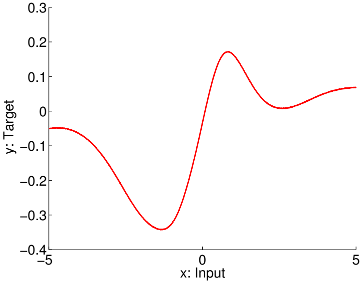
$\ell = 1$

with $\alpha \to \infty$ is the SE covariance with length-scale ℓ.

Other Covariance Functions: Neural Network Covariance

- Consider a neural network with one hidden layer and N_H hidden units.
- Under certain assumptions the corresponding stochastic process will converge to a Gaussian Process as $N_H \to \infty$.
- For a specific settings of the transfer function of the neural net:

$$\kappa(\mathbf{x}, \mathbf{x}') = \frac{2}{\pi} \sin^{-1} \left(\frac{2\tilde{\mathbf{x}}^T \boldsymbol{\Sigma} \tilde{\mathbf{x}}'}{\sqrt{(1 + 2\tilde{\mathbf{x}}^T \boldsymbol{\Sigma} \tilde{\mathbf{x}})(1 + 2\tilde{\mathbf{x}}'^T \boldsymbol{\Sigma} \tilde{\mathbf{x}}')}} \right)$$



Other Covariance Functions: Periodic, Smooth Functions

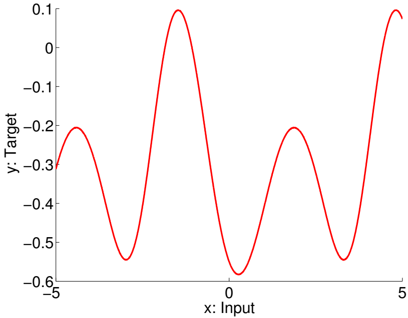
We can create a distribution over periodic functions of x by using the mapping $\mathbf{u}(x) = (cos(x), sin(x))$ and then use the SE covariance on $\mathbf{u}$ space. This gives rise to:

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

Other Covariance Functions: Periodic, Smooth Functions

We can create a distribution over periodic functions of x by using the mapping $\mathbf{u}(x) = (cos(x), sin(x))$ and then use the SE covariance on $\mathbf{u}$ space. This gives rise to:

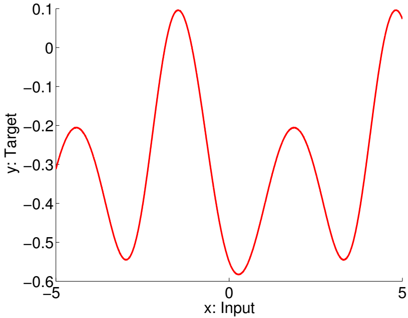
$$\kappa(x, x') = \exp\left(-\frac{2\sin^2(\frac{x-x'}{2})}{\ell^2}\right)$$



Other Covariance Functions: Periodic, Smooth Functions

We can create a distribution over periodic functions of x by using the mapping $\mathbf{u}(x) = (cos(x), sin(x))$ and then use the SE covariance on $\mathbf{u}$ space. This gives rise to:

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



This is called warping and can also be used to introduce non-stationarity.

Conclusions

- Models based on Gaussian process (GP) priors are flexible non-parametric Bayesian approaches to non-linear regression.
- GP regression can be seen as a generalisation of Bayesian regression with non-linear feature spaces and infinite-dimensional feature maps.
- Inference in GP regression models is analytically tractable and is computable thanks to the marginalisation property underlying GPs.
- Hyper-parameter learning carried out via non-covex optimisation of the marginal likelihood.
- High computational cost in time and memory, $\mathcal{O}(N^3)$ and $\mathcal{O}(N^2)$
- Reading: Rasmussen & Williams (GPML, 2006): Ch. 1, 2, 4 (except Sec. 4.3, 4.4), Sec. 5.1, 5.2, 5.4.