

Machine Learning I

Raquel Urtasun

TTI-Chicago

October 12, 2009

Definition

$k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a kernel if there exist a feature map $\psi : \mathcal{X} \rightarrow l_2$ such that $k(\mathbf{x}, \mathbf{x}') = \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle = \psi(\mathbf{x})^T \psi(\mathbf{x}')$.

We saw some some properties of a kernel

- Symmetry

$$k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$$

- Positiveness

$$k(\mathbf{x}, \mathbf{x}) = \|\psi(\mathbf{x})\|^2 \geq 0$$

We also saw mercer's condition.

Definition

$k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a kernel if there exist a feature map $\psi : \mathcal{X} \rightarrow l_2$ such that $k(\mathbf{x}, \mathbf{x}') = \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle = \psi(\mathbf{x})^T \psi(\mathbf{x}')$.

We saw also properties to construct kernels

- If k_1 is a kernel, then $k = \alpha k_1$ with $\alpha \geq 0$ is also a kernel
- If k_1 and k_2 are kernels, then $k = k_1 + k_2$ is also a kernel
- If k_1 and k_2 are kernels, then $k = k_1 \cdot k_2$ is also a kernel
- No matter how complicated a function is, I can construct a kernel

$$k(\mathbf{x}, \mathbf{x}') = f(\mathbf{x}) \cdot f(\mathbf{x}')$$

- ...

Random weights and random functions

- Lets consider a feature map of the form

$$\psi(x) = \left[1, x, \frac{1}{\sqrt{2}}x^2, \dots, \frac{x^n}{\sqrt{n!}} \right]^T$$

and a weight vector of the form

$$\mathbf{w}_f = \left[f(0), f'(0), \frac{1}{\sqrt{2}}f''(0), \dots, \frac{f^n(0)}{\sqrt{n!}} \right]^T$$

Random weights and random functions

- Lets consider a feature map of the form

$$\psi(x) = \left[1, x, \frac{1}{\sqrt{2}}x^2, \dots, \frac{x^n}{\sqrt{n!}} \right]^T$$

and a weight vector of the form

$$\mathbf{w}_f = \left[f(0), f'(0), \frac{1}{\sqrt{2}}f''(0), \dots, \frac{f^n(0)}{\sqrt{n!}} \right]^T$$

- I can then represent any function since

$$\mathbf{w}_f^T \psi(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \dots + \frac{f^n(0)}{n!}x^n = f(x)$$

- Since the features and weights can be infinite dimensional, I can represent my function with infinitesimally small error.

Random weights and random functions

- Lets consider a feature map of the form

$$\psi(x) = \left[1, x, \frac{1}{\sqrt{2}}x^2, \dots, \frac{x^n}{\sqrt{n!}} \right]^T$$

and a weight vector of the form

$$\mathbf{w}_f = \left[f(0), f'(0), \frac{1}{\sqrt{2}}f''(0), \dots, \frac{f^n(0)}{\sqrt{n!}} \right]^T$$

- I can then represent any function since

$$\mathbf{w}_f^T \psi(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \dots + \frac{f^n(0)}{n!}x^n = f(x)$$

- Since the features and weights can be infinite dimensional, I can represent my function with infinitesimally small error.

Functions of multiple variables

- Let's $\mathbf{x} \in \mathbb{R}^2$, with $\mathbf{x} = [x_1, x_2]^T$, and let's consider the feature map

$$\psi(\mathbf{x}) = \left[1, x_1, x_2, \frac{1}{\sqrt{2}}x_1^2, \sqrt{2}x_1x_2, \frac{1}{\sqrt{2}}x_2^2, \dots \right]^T$$

and let's consider the weights to be

$$\mathbf{w}_f = \left[1, f_{x_1}(0), f_{x_2}(0), \frac{1}{\sqrt{2}}f_{x_1x_1}(0), \sqrt{2}f_{x_1x_2}(0), \frac{1}{\sqrt{2}}f_{x_2x_2}(0), \dots \right]^T$$

Functions of multiple variables

- Let's $\mathbf{x} \in \mathbb{R}^2$, with $\mathbf{x} = [x_1, x_2]^T$, and let's consider the feature map

$$\psi(\mathbf{x}) = \left[1, x_1, x_2, \frac{1}{\sqrt{2}}x_1^2, \sqrt{2}x_1x_2, \frac{1}{\sqrt{2}}x_2^2, \dots \right]^T$$

and let's consider the weights to be

$$\mathbf{w}_f = \left[1, f_{x_1}(0), f_{x_2}(0), \frac{1}{\sqrt{2}}f_{x_1x_1}(0), \sqrt{2}f_{x_1x_2}(0), \frac{1}{\sqrt{2}}f_{x_2x_2}(0), \dots \right]^T$$

- Then we can represent any function

$$\begin{aligned} \mathbf{w}_f^T \psi(\mathbf{x}) &= 1 + x_1 f_{x_1}(0) + x_2 f_{x_2}(0) + \frac{1}{2!} (f_{x_1x_1}(0)x_1^2 + 2f_{x_1x_2}(0)x_1x_2 + f_{x_2x_2}(0)x_2^2) \\ &\quad + \dots \\ &= f(\mathbf{x}) \end{aligned}$$

- Before distribution over the weights, now distribution over functions!

Functions of multiple variables

- Let's $\mathbf{x} \in \mathbb{R}^2$, with $\mathbf{x} = [x_1, x_2]^T$, and let's consider the feature map

$$\psi(\mathbf{x}) = \left[1, x_1, x_2, \frac{1}{\sqrt{2}}x_1^2, \sqrt{2}x_1x_2, \frac{1}{\sqrt{2}}x_2^2, \dots \right]^T$$

and let's consider the weights to be

$$\mathbf{w}_f = \left[1, f_{x_1}(0), f_{x_2}(0), \frac{1}{\sqrt{2}}f_{x_1x_1}(0), \sqrt{2}f_{x_1x_2}(0), \frac{1}{\sqrt{2}}f_{x_2x_2}(0), \dots \right]^T$$

- Then we can represent any function

$$\begin{aligned} \mathbf{w}_f^T \psi(\mathbf{x}) &= 1 + x_1 f_{x_1}(0) + x_2 f_{x_2}(0) + \frac{1}{2!} (f_{x_1x_1}(0)x_1^2 + 2f_{x_1x_2}(0)x_1x_2 + f_{x_2x_2}(0)x_2^2) \\ &\quad + \dots \\ &= f(\mathbf{x}) \end{aligned}$$

- Before distribution over the weights, now distribution over functions!

Statistical view of regression

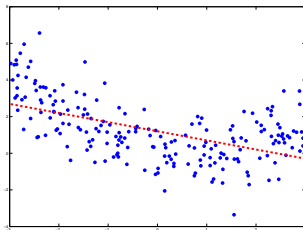
- We will now explicitly model the randomness in the data:

$$y = f(\mathbf{x}; \mathbf{w}) + \nu, \quad f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T \psi(\mathbf{x})$$

where the *noise* ν accounts for everything not captured by the linear mapping.

- We have model the noise as Gaussian

$$p(\nu) = \mathcal{N}(\nu; \mu, \sigma^2)$$



Statistical view of regression

- We can compute the posterior

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})}$$

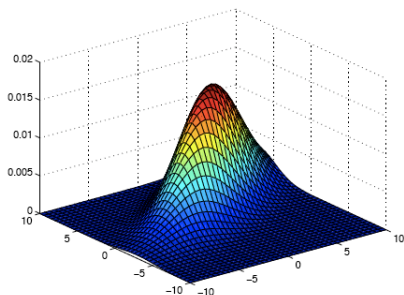
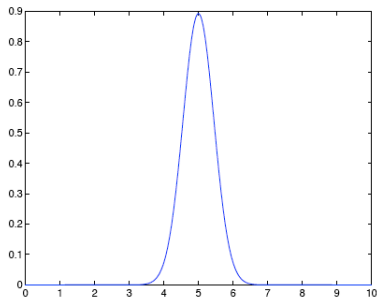
where $p(\mathbf{y}|\mathbf{X})$ is the normalization constant independent of the weights

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

Review in multivariate Gaussians

- A random variable $\mathbf{x} \in \mathbb{R}^d$ is said to have a multivariate Gaussian distribution with mean $\mu \in \mathbb{R}^n$ and covariance $\Sigma \in \mathbf{S}_{++}^d$ if its probability density function is given by

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



Statistical view of regression

- In case of linear regression and Gaussian noise the likelihood is

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \prod_{i=1}^N \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - \mathbf{w}^T \psi(\mathbf{x}_i))^2}{2\sigma^2}\right)$$

- We specify a prior over the weights

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \Sigma_p)$$

- Incorporating this prior the posterior becomes

$$\begin{aligned} p(\mathbf{w}|\mathbf{X}, \mathbf{y}) &\propto p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w}) \\ &\propto \prod_{i=1}^N \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - \mathbf{w}^T \psi(\mathbf{x}_i))^2}{2\sigma^2}\right) \exp\left(-\frac{1}{2}\mathbf{w}^T \Sigma_p^{-1} \mathbf{w}\right) \end{aligned}$$

- Obtain the best prediction by ML or MAP estimation

Statistical view of regression

- Writing only the terms that depend on the weights and completing the square we have

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}) \propto \exp\left(-\frac{1}{2}(\mathbf{w} - \bar{\mathbf{w}})^T \left(\frac{1}{\sigma^2} \Psi^T \Psi + \Sigma_p^{-1}\right) (\mathbf{w} - \bar{\mathbf{w}})\right)$$

- You can identify that the posterior is distributed as a Gaussian

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

with

$$\mathbf{A} = \sigma^{-2} \Psi^T \Psi + \Sigma_p^{-1} \quad \text{and} \quad \bar{\mathbf{w}} = \frac{1}{\sigma^2} \mathbf{A}^{-1} \Psi^T \mathbf{y}$$

- Nothing new: maximizing this posterior is Ridge Regression

Statistical view of regression

- Writing only the terms that depend on the weights and completing the square we have

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}) \propto \exp\left(-\frac{1}{2}(\mathbf{w} - \bar{\mathbf{w}})^T \left(\frac{1}{\sigma^2}\Psi^T\Psi + \Sigma_p^{-1}\right) (\mathbf{w} - \bar{\mathbf{w}})\right)$$

- You can identify that the posterior is distributed as a Gaussian

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

with

$$\mathbf{A} = \sigma^{-2}\Psi^T\Psi + \Sigma_p^{-1} \quad \text{and} \quad \bar{\mathbf{w}} = \frac{1}{\sigma^2}\mathbf{A}^{-1}\Psi^T\mathbf{y}$$

- Nothing new: maximizing this posterior is Ridge Regression

Gaussian process: weight view

- Try to be a bit more Bayesian...
- To make predictions instead of using the MAP estimate, we average over all possible parameter values, weighted by the posterior distribution

$$\begin{aligned} p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) &= \int p(f_* | \mathbf{x}_*, \mathbf{w}) p(\mathbf{w} | \mathbf{X}, \mathbf{y}) d\mathbf{w} \\ &= \mathcal{N} \left(f_*; \frac{1}{\sigma^2} \psi(\mathbf{x}_*) \mathbf{A}^{-1} \Psi(\mathbf{X})^T \mathbf{y}, \psi(\mathbf{x}_*) \mathbf{A}^{-1} \psi(\mathbf{x}_*)^T \right) \end{aligned}$$

where \mathbf{x}_* is the input of the test data that I would like to make predictions from.

Gaussian process: weight view

- Try to be a bit more Bayesian...
- To make predictions instead of using the MAP estimate, we average over all possible parameter values, weighted by the posterior distribution

$$\begin{aligned} p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) &= \int p(f_* | \mathbf{x}_*, \mathbf{w}) p(\mathbf{w} | \mathbf{X}, \mathbf{y}) d\mathbf{w} \\ &= \mathcal{N} \left(f_*; \frac{1}{\sigma^2} \psi(\mathbf{x}_*) \mathbf{A}^{-1} \Psi(\mathbf{X})^T \mathbf{y}, \psi(\mathbf{x}_*) \mathbf{A}^{-1} \psi(\mathbf{x}_*)^T \right) \end{aligned}$$

where \mathbf{x}_* is the input of the test data that I would like to make predictions from.

- An easy way to proof this is to compute the expectations from $f_* = \mathbf{w}^T \psi(\mathbf{x}_*)$.

$$E_{p(\mathbf{w} | \mathbf{X}, \mathbf{y})} [\psi(\mathbf{x}_*) \mathbf{w}^T] = \psi(\mathbf{x}_*) E_{p(\mathbf{w} | \mathbf{X}, \mathbf{y})} [\mathbf{w}^T] = \frac{1}{\sigma^2} \psi(\mathbf{x}_*) \mathbf{A}^{-1} \Psi(\mathbf{X})^T \mathbf{y}$$

and

$$\begin{aligned} E_{p(\mathbf{w} | \mathbf{X}, \mathbf{y})} [(\mathbf{w} \psi(\mathbf{x}_*)^T)^T (\mathbf{w} \psi(\mathbf{x}_*)^T)] &= \psi(\mathbf{x}_*) E_{p(\mathbf{w} | \mathbf{X}, \mathbf{y})} [\mathbf{w} \mathbf{w}^T] \psi(\mathbf{x}_*)^T \\ &= \psi(\mathbf{x}_*) \mathbf{A}^{-1} \psi(\mathbf{x}_*)^T \end{aligned}$$

Gaussian process: weight view

- Try to be a bit more Bayesian...
- To make predictions instead of using the MAP estimate, we average over all possible parameter values, weighted by the posterior distribution

$$\begin{aligned} p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) &= \int p(f_* | \mathbf{x}_*, \mathbf{w}) p(\mathbf{w} | \mathbf{X}, \mathbf{y}) d\mathbf{w} \\ &= \mathcal{N} \left(f_*; \frac{1}{\sigma^2} \psi(\mathbf{x}_*) \mathbf{A}^{-1} \Psi(\mathbf{X})^T \mathbf{y}, \psi(\mathbf{x}_*) \mathbf{A}^{-1} \psi(\mathbf{x}_*)^T \right) \end{aligned}$$

where \mathbf{x}_* is the input of the test data that I would like to make predictions from.

- An easy way to proof this is to compute the expectations from $f_* = \mathbf{w}^T \psi(\mathbf{x}_*)$.

$$E_{p(\mathbf{w} | \mathbf{X}, \mathbf{y})} [\psi(\mathbf{x}_*) \mathbf{w}^T] = \psi(\mathbf{x}_*) E_{p(\mathbf{w} | \mathbf{X}, \mathbf{y})} [\mathbf{w}^T] = \frac{1}{\sigma^2} \psi(\mathbf{x}_*) \mathbf{A}^{-1} \Psi(\mathbf{X})^T \mathbf{y}$$

and

$$\begin{aligned} E_{p(\mathbf{w} | \mathbf{X}, \mathbf{y})} [(\mathbf{w} \psi(\mathbf{x}_*)^T)^T (\mathbf{w} \psi(\mathbf{x}_*)^T)] &= \psi(\mathbf{x}_*) E_{p(\mathbf{w} | \mathbf{X}, \mathbf{y})} [\mathbf{w} \mathbf{w}^T] \psi(\mathbf{x}_*)^T \\ &= \psi(\mathbf{x}_*) \mathbf{A}^{-1} \psi(\mathbf{x}_*)^T \end{aligned}$$

- The predictive distribution is

$$\begin{aligned} p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) &= \int p(f_* | \mathbf{x}_*, \mathbf{w}) p(\mathbf{w} | \mathbf{X}, \mathbf{y}) d\mathbf{w} \\ &= \mathcal{N} \left(f_*; \frac{1}{\sigma^2} \psi(\mathbf{x}_*) \mathbf{A}^{-1} \Psi(\mathbf{X})^T \mathbf{y}, \psi(\mathbf{x}_*) \mathbf{A}^{-1} \psi(\mathbf{x}_*)^T \right) \end{aligned}$$

where \mathbf{x}_* is the input of the test data that I would like to make predictions from.

- The mean is the same as for ridge regression, but now we have a confidence value.

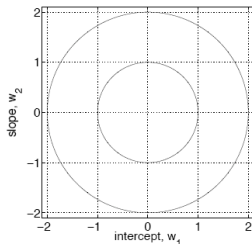
- The predictive distribution is

$$\begin{aligned} p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) &= \int p(f_* | \mathbf{x}_*, \mathbf{w}) p(\mathbf{w} | \mathbf{X}, \mathbf{y}) d\mathbf{w} \\ &= \mathcal{N} \left(f_*; \frac{1}{\sigma^2} \psi(\mathbf{x}_*) \mathbf{A}^{-1} \Psi(\mathbf{X})^T \mathbf{y}, \psi(\mathbf{x}_*) \mathbf{A}^{-1} \psi(\mathbf{x}_*)^T \right) \end{aligned}$$

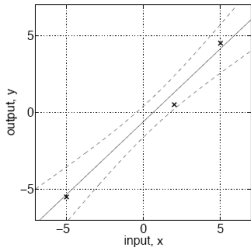
where \mathbf{x}_* is the input of the test data that I would like to make predictions from.

- The mean is the same as for ridge regression, but now we have a confidence value.

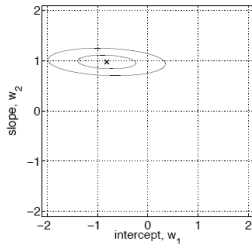
Example $f(x, \mathbf{w}) = w_1 + w_2\phi(x)$



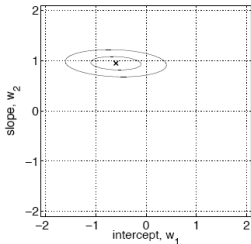
(a)



(b)



(c)



(d)

- (a) $p(\mathbf{w})$
- (b) $p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y})$
- (c) $p(\mathbf{y} | \mathbf{X}, \mathbf{w})$
- (d) $p(\mathbf{w} | \mathbf{X}, \mathbf{y})$

- The predictive distribution is

$$p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N} \left(f_*; \frac{1}{\sigma^2} \psi(\mathbf{x}_*) \mathbf{A}^{-1} \Psi(\mathbf{X})^T \mathbf{y}, \psi(\mathbf{x}_*) \mathbf{A}^{-1} \psi(\mathbf{x}_*)^T \right)$$

where \mathbf{x}_* is the input of the test data that I would like to make predictions from.

- We have to invert a matrix of size $(d + 1) \times (d + 1)$. If $(d + 1)$ is large (potentially infinite) this is not convenient.

Gaussian process: weight view

- The predictive distribution is

$$p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N} \left(f_*; \frac{1}{\sigma^2} \psi(\mathbf{x}_*) \mathbf{A}^{-1} \Psi(\mathbf{X})^T \mathbf{y}, \psi(\mathbf{x}_*) \mathbf{A}^{-1} \psi(\mathbf{x}_*)^T \right)$$

where \mathbf{x}_* is the input of the test data that I would like to make predictions from.

- We have to invert a matrix of size $(d + 1) \times (d + 1)$. If $(d + 1)$ is large (potentially infinite) this is not convenient.
- A more efficient way when d is large to write the predictive distribution is

$$p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N} (f_*; \mathbf{k}_* (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}, k_{**} - \mathbf{k}_* (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_*)$$

where we have defined $\mathbf{K} = \Psi \Sigma_p \Psi^T$, $k_{**} = \psi_* \Sigma_p \psi_*^T$, and $\mathbf{k}_* = \psi_* \Sigma_p \Psi^T$.

Gaussian process: weight view

- The predictive distribution is

$$p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N} \left(f_*; \frac{1}{\sigma^2} \psi(\mathbf{x}_*) \mathbf{A}^{-1} \Psi(\mathbf{X})^T \mathbf{y}, \psi(\mathbf{x}_*) \mathbf{A}^{-1} \psi(\mathbf{x}_*)^T \right)$$

where \mathbf{x}_* is the input of the test data that I would like to make predictions from.

- We have to invert a matrix of size $(d + 1) \times (d + 1)$. If $(d + 1)$ is large (potentially infinite) this is not convenient.
- A more efficient way when d is large to write the predictive distribution is

$$p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N} (f_*; \mathbf{k}_* (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}, k_{**} - \mathbf{k}_* (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_*)$$

where we have defined $\mathbf{K} = \Psi \Sigma_p \Psi^T$, $k_{**} = \psi_* \Sigma_p \psi_*^T$, and $\mathbf{k}_* = \psi_* \Sigma_p \Psi^T$.

- The matrix to invert is $N \times N$, with N the number of training points.

Gaussian process: weight view

- The predictive distribution is

$$p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N} \left(f_*; \frac{1}{\sigma^2} \psi(\mathbf{x}_*) \mathbf{A}^{-1} \Psi(\mathbf{X})^T \mathbf{y}, \psi(\mathbf{x}_*) \mathbf{A}^{-1} \psi(\mathbf{x}_*)^T \right)$$

where \mathbf{x}_* is the input of the test data that I would like to make predictions from.

- We have to invert a matrix of size $(d + 1) \times (d + 1)$. If $(d + 1)$ is large (potentially infinite) this is not convenient.
- A more efficient way when d is large to write the predictive distribution is

$$p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N} (f_*; \mathbf{k}_* (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}, k_{**} - \mathbf{k}_* (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_*)$$

where we have defined $\mathbf{K} = \Psi \Sigma_p \Psi^T$, $k_{**} = \psi_* \Sigma_p \psi_*^T$, and $\mathbf{k}_* = \psi_* \Sigma_p \Psi^T$.

- The matrix to invert is $N \times N$, with N the number of training points.

Sketch of the proof: mean

- We have to proof that the means are equal

$$\frac{1}{\sigma^2} \psi(\mathbf{x}_*)^T \mathbf{A}^{-1} \Psi(\mathbf{X}) \mathbf{y} = \mathbf{k}_* (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$

- First note that

$$\frac{1}{\sigma^2} \Psi (\mathbf{K} + \sigma^2 \mathbf{I}) = \frac{1}{\sigma^2} \Psi (\Psi^T \Sigma_p \Psi + \sigma^2 \mathbf{I}) = \left(\frac{1}{\sigma^2} \Psi \Psi^T \Sigma_p + \Sigma_p^{-1} \Sigma_p \right) \Psi = \mathbf{A} \Sigma_p \Psi$$

Sketch of the proof: mean

- We have to proof that the means are equal

$$\frac{1}{\sigma^2} \psi(\mathbf{x}_*)^T \mathbf{A}^{-1} \Psi(\mathbf{X}) \mathbf{y} = \mathbf{k}_* (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$

- First note that

$$\frac{1}{\sigma^2} \Psi(\mathbf{K} + \sigma^2 \mathbf{I}) = \frac{1}{\sigma^2} \Psi(\Psi^T \Sigma_p \Psi + \sigma^2 \mathbf{I}) = \left(\frac{1}{\sigma^2} \Psi \Psi^T \Sigma_p + \Sigma_p^{-1} \Sigma_p \right) \Psi = \mathbf{A} \Sigma_p \Psi$$

- Multiplying \mathbf{A}^{-1} from left and $(\mathbf{K} + \sigma^2 \mathbf{I})^{-1}$ from right gives

$$\frac{1}{\sigma^2} \mathbf{A}^{-1} \Psi = \Sigma_p \Psi (\mathbf{K} + \sigma^2 \mathbf{I})^{-1}$$

Sketch of the proof: mean

- We have to proof that the means are equal

$$\frac{1}{\sigma^2} \psi(\mathbf{x}_*)^T \mathbf{A}^{-1} \Psi(\mathbf{X}) \mathbf{y} = \mathbf{k}_* (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$

- First note that

$$\frac{1}{\sigma^2} \Psi (\mathbf{K} + \sigma^2 \mathbf{I}) = \frac{1}{\sigma^2} \Psi (\Psi^T \Sigma_p \Psi + \sigma^2 \mathbf{I}) = \left(\frac{1}{\sigma^2} \Psi \Psi^T \Sigma_p + \Sigma_p^{-1} \Sigma_p \right) \Psi = \mathbf{A} \Sigma_p \Psi$$

- Multiplying \mathbf{A}^{-1} from left and $(\mathbf{K} + \sigma^2 \mathbf{I})^{-1}$ from right gives

$$\frac{1}{\sigma^2} \mathbf{A}^{-1} \Psi = \Sigma_p \Psi (\mathbf{K} + \sigma^2 \mathbf{I})^{-1}$$

- Finally the equivalence between the means is obtain by multiplying left by $\psi(\mathbf{x}_*)$ and right by \mathbf{y} .

Sketch of the proof: mean

- We have to proof that the means are equal

$$\frac{1}{\sigma^2} \psi(\mathbf{x}_*)^T \mathbf{A}^{-1} \Psi(\mathbf{X}) \mathbf{y} = \mathbf{k}_* (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$

- First note that

$$\frac{1}{\sigma^2} \Psi (\mathbf{K} + \sigma^2 \mathbf{I}) = \frac{1}{\sigma^2} \Psi (\Psi^T \Sigma_p \Psi + \sigma^2 \mathbf{I}) = \left(\frac{1}{\sigma^2} \Psi \Psi^T \Sigma_p + \Sigma_p^{-1} \Sigma_p \right) \Psi = \mathbf{A} \Sigma_p \Psi$$

- Multiplying \mathbf{A}^{-1} from left and $(\mathbf{K} + \sigma^2 \mathbf{I})^{-1}$ from right gives

$$\frac{1}{\sigma^2} \mathbf{A}^{-1} \Psi = \Sigma_p \Psi (\mathbf{K} + \sigma^2 \mathbf{I})^{-1}$$

- Finally the equivalence between the means is obtain by multiplying left by $\psi(\mathbf{x}_*)$ and right by \mathbf{y} .

Sketch of the proof: mean

- We have to proof that the means are equal

$$\frac{1}{\sigma^2} \psi(\mathbf{x}_*)^T \mathbf{A}^{-1} \Psi(\mathbf{X}) \mathbf{y} = \mathbf{k}_* (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$

- First note that

$$\frac{1}{\sigma^2} \Psi (\mathbf{K} + \sigma^2 \mathbf{I}) = \frac{1}{\sigma^2} \Psi (\Psi^T \Sigma_p \Psi + \sigma^2 \mathbf{I}) = \left(\frac{1}{\sigma^2} \Psi \Psi^T \Sigma_p + \Sigma_p^{-1} \Sigma_p \right) \Psi = \mathbf{A} \Sigma_p \Psi$$

- Multiplying \mathbf{A}^{-1} from left and $(\mathbf{K} + \sigma^2 \mathbf{I})^{-1}$ from right gives

$$\frac{1}{\sigma^2} \mathbf{A}^{-1} \Psi = \Sigma_p \Psi (\mathbf{K} + \sigma^2 \mathbf{I})^{-1}$$

- Finally the equivalence between the means is obtain by multiplying left by $\psi(\mathbf{x}_*)$ and right by \mathbf{y} .

Sketch of the proof: covariance

- The equivalence between the variances

$$\psi(\mathbf{x}_*)^T \mathbf{A}^{-1} \psi(\mathbf{x}_*) = k_{**} - \mathbf{k}_* (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_*$$

- can be obtained using the matrix inversion lemma

$$(\mathbf{Z} + \mathbf{U}\mathbf{W}\mathbf{V}^T)^{-1} = \mathbf{Z}^{-1} - \mathbf{Z}^{-1}\mathbf{U}(\mathbf{W}^{-1} + \mathbf{V}^T\mathbf{Z}^{-1}\mathbf{U})^{-1}\mathbf{V}^T\mathbf{Z}^{-1}$$

with $\mathbf{Z}^{-1} = \Sigma_p$, $\mathbf{W}^{-1} = \sigma^2 \mathbf{I}$ and $\mathbf{U} = \mathbf{V} = \Psi$

We can now use kernels...

... and infinite dimensional features

$$p(f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(f_*; \mathbf{k}_*(\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}, k_{**} - \mathbf{k}_*(\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_*)$$

Definition

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

- Probability Distribution over Functions
- Functions are infinite dimensional.
 - ▶ Prior distribution over *instantiations* of the function: finite dimensional objects.

Definition

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

- Probability Distribution over Functions
- Functions are infinite dimensional.
 - ▶ Prior distribution over *instantiations* of the function: finite dimensional objects.
- GPs are consistent

$$\text{if } p\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right) \text{ then } p(y_1) = \mathcal{N}(\mu_1, \Sigma_{11})$$

with Σ_{11} the relevant submatrix of Σ .

Definition

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

- Probability Distribution over Functions
- Functions are infinite dimensional.
 - ▶ Prior distribution over *instantiations* of the function: finite dimensional objects.
- GPs are consistent

$$\text{if } p\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right) \text{ then } p(y_1) = \mathcal{N}(\mu_1, \Sigma_{11})$$

with Σ_{11} the relevant submatrix of Σ .

Definition

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

- A Gaussian process is completely identified by its mean and covariance

$$y = f(\mathbf{x}; \mathbf{w}) + \nu, \quad f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T \psi(\mathbf{x})$$

Definition

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

- A Gaussian process is completely identified by its mean and covariance

$$y = f(\mathbf{x}; \mathbf{w}) + \nu, \quad f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T \psi(\mathbf{x})$$

- We can compute the mean

$$E_{p(\mathbf{w})} [f(\mathbf{x}; \mathbf{w})] = \psi(x)^T E_{p(\mathbf{w})} [\mathbf{w}] = 0$$

Definition

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

- A Gaussian process is completely identified by its mean and covariance

$$y = f(\mathbf{x}; \mathbf{w}) + \nu, \quad f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T \psi(\mathbf{x})$$

- We can compute the mean

$$E_{p(\mathbf{w})} [\mathbf{f}(\mathbf{x}; \mathbf{w})] = \psi(\mathbf{x})^T E_{p(\mathbf{w})} [\mathbf{w}] = 0$$

and the covariance

$$E_{p(\mathbf{w})} [\mathbf{f}(\mathbf{x}; \mathbf{w})\mathbf{f}(\mathbf{x}'; \mathbf{w})] = \psi(\mathbf{x})^T E_{p(\mathbf{w})} [\mathbf{w}\mathbf{w}^T] \psi(\mathbf{x}') = \psi(\mathbf{x})^T \Sigma_p \psi(\mathbf{x}') = \mathbf{k}(\mathbf{x}, \mathbf{x}')$$

Definition

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

- A Gaussian process is completely identified by its mean and covariance

$$y = f(\mathbf{x}; \mathbf{w}) + \nu, \quad f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T \psi(\mathbf{x})$$

- We can compute the mean

$$E_{p(\mathbf{w})} [\mathbf{f}(\mathbf{x}; \mathbf{w})] = \psi(\mathbf{x})^T E_{p(\mathbf{w})} [\mathbf{w}] = 0$$

and the covariance

$$E_{p(\mathbf{w})} [\mathbf{f}(\mathbf{x}; \mathbf{w})\mathbf{f}(\mathbf{x}'; \mathbf{w})] = \psi(\mathbf{x})^T E_{p(\mathbf{w})} [\mathbf{w}\mathbf{w}^T] \psi(\mathbf{x}') = \psi(\mathbf{x})^T \Sigma_p \psi(\mathbf{x}') = \mathbf{k}(\mathbf{x}, \mathbf{x}')$$

Definition

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

- A Gaussian process is completely identified by its mean and covariance

$$y = f(\mathbf{x}; \mathbf{w}) + \nu, \quad f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T \psi(\mathbf{x})$$

- We can compute the mean

$$E_{p(\mathbf{w})} [\mathbf{f}(\mathbf{x}; \mathbf{w})] = \psi(\mathbf{x})^T E_{p(\mathbf{w})} [\mathbf{w}] = 0$$

and the covariance

$$E_{p(\mathbf{w})} [\mathbf{f}(\mathbf{x}; \mathbf{w})\mathbf{f}(\mathbf{x}'; \mathbf{w})] = \psi(\mathbf{x})^T E_{p(\mathbf{w})} [\mathbf{w}\mathbf{w}^T] \psi(\mathbf{x}') = \psi(\mathbf{x})^T \Sigma_p \psi(\mathbf{x}') = \mathbf{k}(\mathbf{x}, \mathbf{x}')$$

- A (zero mean) Gaussian process likelihood is of the form

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

- The specification of the covariance function implies a distribution over functions.
- To see this we can draw samples from the distribution of functions evaluated at any point, by first selecting the points \mathbf{X}_* , and then sampling from a multivariate Gaussian with

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

- A (zero mean) Gaussian process likelihood is of the form

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

- The specification of the covariance function implies a distribution over functions.
- To see this we can draw samples from the distribution of functions evaluated at any point, by first selecting the points \mathbf{X}_* , and then sampling from a multivariate Gaussian with

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

- How to draw samples from a multivariate distribution?

- A (zero mean) Gaussian process likelihood is of the form

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

- The specification of the covariance function implies a distribution over functions.
- To see this we can draw samples from the distribution of functions evaluated at any point, by first selecting the points \mathbf{X}_* , and then sampling from a multivariate Gaussian with

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

- How to draw samples from a multivariate distribution?

Gaussian process prior

- A (zero mean) Gaussian process likelihood is of the form

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

- Generating samples

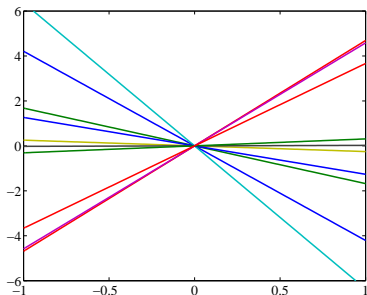


Figure: linear kernel, $\mathbf{K} = \Psi(\mathbf{X})\Psi(\mathbf{X})^T$

Gaussian process prior

- A (zero mean) Gaussian process likelihood is of the form

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

- Generating samples

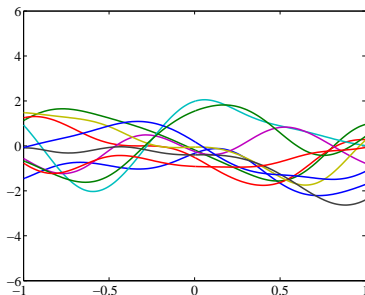


Figure: RBF kernel, $k_{i,j} = \alpha \exp\left(-\frac{1}{2l} \|\psi(\mathbf{x}_i) - \psi(\mathbf{x}_j)\|^2\right)$, with $l = 0.32$, $\alpha = 1$

Gaussian process prior

- A (zero mean) Gaussian process likelihood is of the form

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

- Generating samples

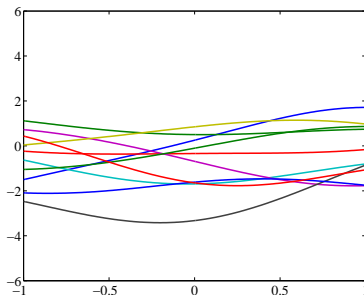


Figure: RBF kernel, $k_{i,j} = \alpha \exp\left(-\frac{1}{2l} \|\psi(\mathbf{x}_i) - \psi(\mathbf{x}_j)\|^2\right)$, with $l = 1$, $\alpha = 1$

Gaussian process prior

- A (zero mean) Gaussian process likelihood is of the form

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

- Generating samples

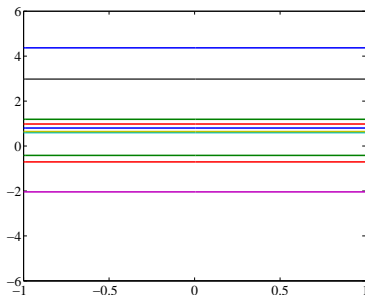


Figure: bias 'kernel', $k_{i,j} = \alpha$, with $\alpha = 1$ and

Gaussian process prior

- A (zero mean) Gaussian process likelihood is of the form

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

- Generating samples

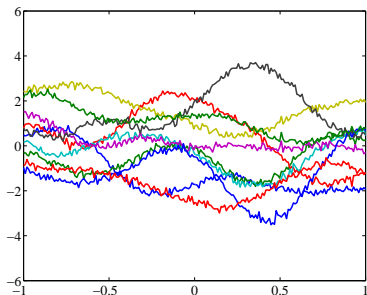


Figure: summed combination of: RBF kernel, $\alpha = 1$, $l = 0.3$; bias kernel, $\alpha = 1$; and white noise kernel, $\beta = 100$

Prediction with noise-free observations

- The probability over joint training and testing outputs is

$$p\left(\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix}\right) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) & \mathbf{K}(\mathbf{X}, \mathbf{X}_*) \\ \mathbf{K}(\mathbf{X}_*, \mathbf{X}) & \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right)$$

- We can then compute the predictive distribution as

$$p(\mathbf{f}_* | \mathbf{X}, \mathbf{f}, \mathbf{X}_*) = \mathcal{N}(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*))$$

where

$$\bar{\mathbf{f}}_* = \mathbf{K}(\mathbf{X}_*, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{f}$$

and

$$\text{cov}(\mathbf{f}_*) = \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{K}(\mathbf{X}, \mathbf{X}_*)$$

Prediction with noise-free observations

- The probability over joint training and testing outputs is

$$p\left(\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix}\right) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) & \mathbf{K}(\mathbf{X}, \mathbf{X}_*) \\ \mathbf{K}(\mathbf{X}_*, \mathbf{X}) & \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right)$$

- We can then compute the predictive distribution as

$$p(\mathbf{f}_* | \mathbf{X}, \mathbf{f}, \mathbf{X}_*) = \mathcal{N}(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*))$$

where

$$\bar{\mathbf{f}}_* = \mathbf{K}(\mathbf{X}_*, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{f}$$

and

$$\text{cov}(\mathbf{f}_*) = \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{K}(\mathbf{X}, \mathbf{X}_*)$$

Prediction with noisy observations

- The probability over joint training and testing outputs is

$$p\left(\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix}\right) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I} & \mathbf{K}(\mathbf{X}, \mathbf{X}_*) \\ \mathbf{K}(\mathbf{X}_*, \mathbf{X}) & \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right)$$

- We can then compute the predictive distribution as

$$p(\mathbf{f}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}_*) = \mathcal{N}(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*))$$

where

$$\bar{\mathbf{f}}_* = \mathbf{K}(\mathbf{X}_*, \mathbf{X})(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$

and

$$\text{cov}(\mathbf{f}_*) = \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*, \mathbf{X})(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I})^{-1} \mathbf{K}(\mathbf{X}, \mathbf{X}_*)$$

Sampling the predictive distribution

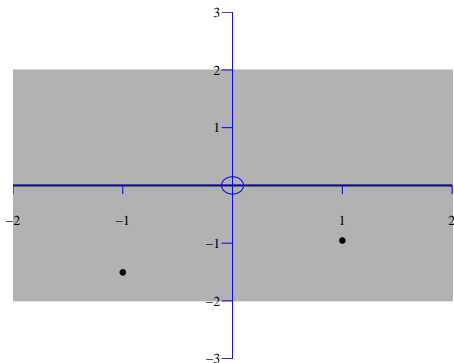


Figure: Examples include WiFi localization, C14 calibration curve.

Sampling the predictive distribution

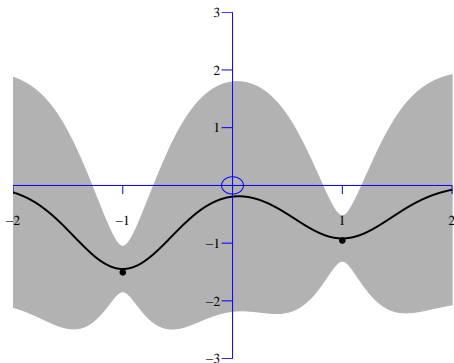


Figure: Examples include WiFi localization, C14 calibration curve.

Sampling the predictive distribution

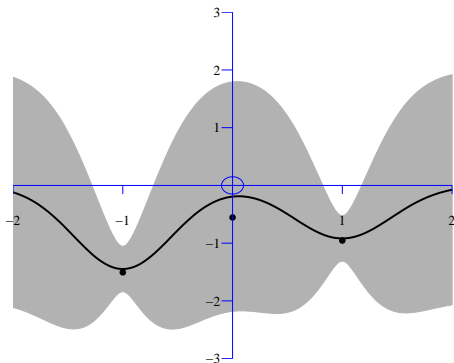


Figure: Examples include WiFi localization, C14 calibration curve.

Sampling the predictive distribution

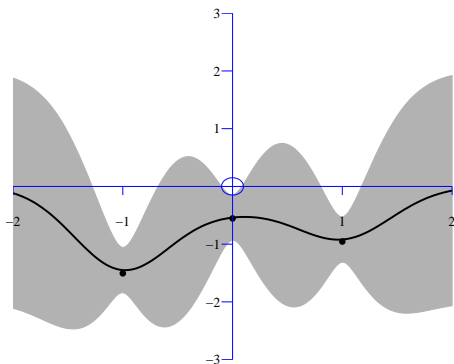


Figure: Examples include WiFi localization, C14 calibration curve.

Sampling the predictive distribution

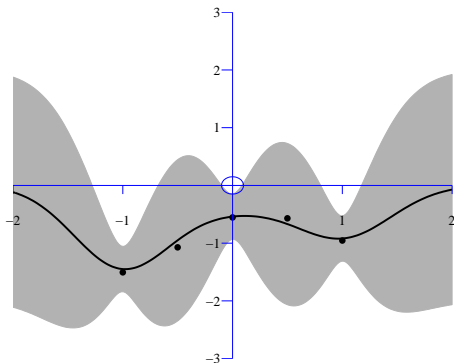


Figure: Examples include WiFi localization, C14 calibration curve.

Sampling the predictive distribution

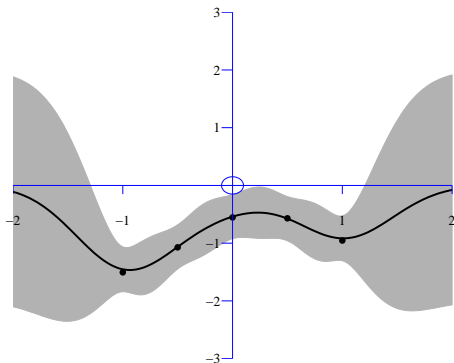


Figure: Examples include WiFi localization, C14 calibration curve.

Sampling the predictive distribution

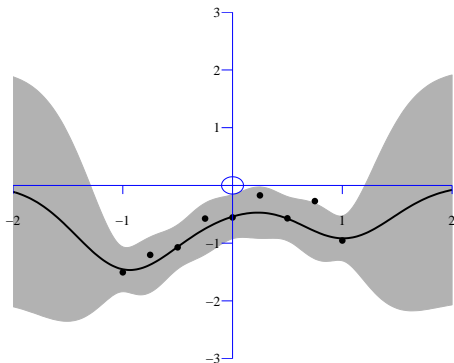


Figure: Examples include WiFi localization, C14 calibration curve.

Sampling the predictive distribution

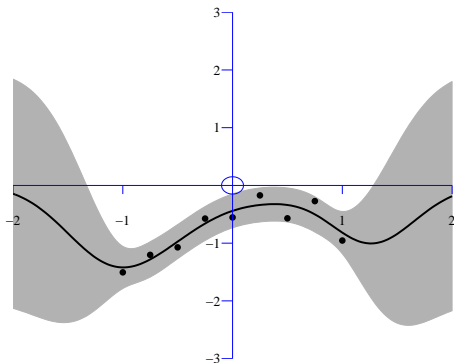


Figure: Examples include WiFi localization, C14 calibration curve.

Predictive distribution and representer theorem

- Using compact notation

$$\bar{\mathbf{f}}_* = \mathbf{k}_*^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$

- The mean prediction is a linear combination of the observations \mathbf{y} .

Predictive distribution and representer theorem

- Using compact notation

$$\bar{\mathbf{f}}_* = \mathbf{k}_*^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$

- The mean prediction is a linear combination of the observations \mathbf{y} .
- It is also a linear combination of n kernel functions, each center at a training point

$$\bar{\mathbf{f}}_* = \sum_{i=1}^n \alpha_i k(\mathbf{x}_i, \mathbf{x}_*)$$

where the $\alpha = (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$.

Predictive distribution and representer theorem

- Using compact notation

$$\bar{\mathbf{f}}_* = \mathbf{k}_*^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$

- The mean prediction is a linear combination of the observations \mathbf{y} .
- It is also a linear combination of n kernel functions, each center at a training point

$$\bar{\mathbf{f}}_* = \sum_{i=1}^n \alpha_i k(\mathbf{x}_i, \mathbf{x}_*)$$

where the $\alpha = (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$.

- This is the representer theorem!

Predictive distribution and representer theorem

- Using compact notation

$$\bar{\mathbf{f}}_* = \mathbf{k}_*^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$

- The mean prediction is a linear combination of the observations \mathbf{y} .
- It is also a linear combination of n kernel functions, each center at a training point

$$\bar{\mathbf{f}}_* = \sum_{i=1}^n \alpha_i k(\mathbf{x}_i, \mathbf{x}_*)$$

where the $\alpha = (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$.

- This is the representer theorem!
- What's the difference with SVMs?

Predictive distribution and representer theorem

- Using compact notation

$$\bar{\mathbf{f}}_* = \mathbf{k}_*^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$

- The mean prediction is a linear combination of the observations \mathbf{y} .
- It is also a linear combination of n kernel functions, each center at a training point

$$\bar{\mathbf{f}}_* = \sum_{i=1}^n \alpha_i k(\mathbf{x}_i, \mathbf{x}_*)$$

where the $\alpha = (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$.

- This is the representer theorem!
- What's the difference with SVMs?
- Answer: α has closed-form solution.

Predictive distribution and representer theorem

- Using compact notation

$$\bar{\mathbf{f}}_* = \mathbf{k}_*^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$

- The mean prediction is a linear combination of the observations \mathbf{y} .
- It is also a linear combination of n kernel functions, each center at a training point

$$\bar{\mathbf{f}}_* = \sum_{i=1}^n \alpha_i k(\mathbf{x}_i, \mathbf{x}_*)$$

where the $\alpha = (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$.

- This is the representer theorem!
- What's the difference with SVMs?
- Answer: α has closed-form solution.

- The marginal log likelihood is

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

- The negative log marginal likelihood is

$$-\log p(\mathbf{y}|\mathbf{X}) = \frac{1}{2}\mathbf{y}^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K} + \sigma^2\mathbf{I}| + \frac{n}{2}\log 2\pi$$

- The marginal log likelihood is

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

- The negative log marginal likelihood is

$$-\log p(\mathbf{y}|\mathbf{X}) = \frac{1}{2}\mathbf{y}^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K} + \sigma^2\mathbf{I}| + \frac{n}{2}\log 2\pi$$

Learning the GP

- Learning the GP means estimating the hyperparameters.
- We do not need to estimate the weights since we have marginalized them.
- The hyperparameters are typically estimated by maximizing the likelihood, or equivalently by minimizing the negative log likelihood, which ignoring constants is

$$\Theta = \underset{\Theta}{\operatorname{argmin}} \frac{1}{2} \mathbf{y}^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y} + \frac{1}{2} \log |\mathbf{K} + \sigma^2 \mathbf{I}|$$

Learning the GP

- Learning the GP means estimating the hyperparameters.
- We do not need to estimate the weights since we have marginalized them.
- The hyperparameters are typically estimated by maximizing the likelihood, or equivalently by minimizing the negative log likelihood, which ignoring constants is

$$\Theta = \operatorname{argmin}_{\Theta} \frac{1}{2} \mathbf{y}^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y} + \frac{1}{2} \log |\mathbf{K} + \sigma^2 \mathbf{I}|$$

- CV can also be used instead of ML.

Learning the GP

- Learning the GP means estimating the hyperparameters.
- We do not need to estimate the weights since we have marginalized them.
- The hyperparameters are typically estimated by maximizing the likelihood, or equivalently by minimizing the negative log likelihood, which ignoring constants is

$$\Theta = \underset{\Theta}{\operatorname{argmin}} \frac{1}{2} \mathbf{y}^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y} + \frac{1}{2} \log |\mathbf{K} + \sigma^2 \mathbf{I}|$$

- CV can also be used instead of ML.
- Typically only a few parameters Θ to estimate.
- Θ depends on the type of kernel.

Learning the GP

- Learning the GP means estimating the hyperparameters.
- We do not need to estimate the weights since we have marginalized them.
- The hyperparameters are typically estimated by maximizing the likelihood, or equivalently by minimizing the negative log likelihood, which ignoring constants is

$$\Theta = \underset{\Theta}{\operatorname{argmin}} \frac{1}{2} \mathbf{y}^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y} + \frac{1}{2} \log |\mathbf{K} + \sigma^2 \mathbf{I}|$$

- CV can also be used instead of ML.
- Typically only a few parameters Θ to estimate.
- Θ depends on the type of kernel.