Mathematics in Data Science
Department of Mathematics
Technical University of Munich

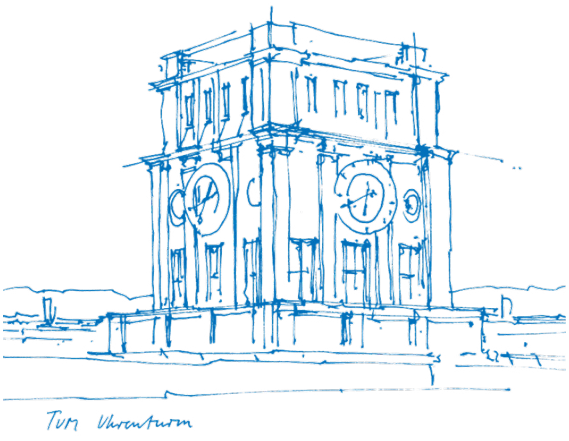
Seminar: Gaussian Processes for Machine Learning

Gaussian processes for regression

Davit Papikyan

Mathematics in Data Science
Department of Mathematics
Technical University of Munich

February 3rd, 2022

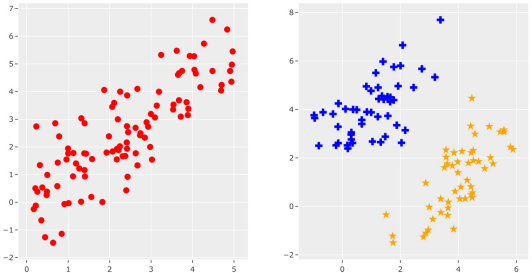


Outline

Introduction

Supervised Learning: Regression vs. Classification

- Regression: **continuous** variables (price, salary, etc.)
- Classification: **discrete** variables (spam/not spam, male/female, etc.)



Outline

Bayesian Linear Regression

Problem statement

Training set: $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \mid i = 1, 2, \ldots, n\}$, where

$$\boldsymbol{x}_i \in \mathbb{R}^d \quad i = 1, 2, \ldots, n$$

$$y_i \in \mathbb{R}$$

Using matrix notations: $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$, where

$\boldsymbol{X} \in \mathbb{R}^{d \times n}$ – design matrix

$\boldsymbol{y} \in \mathbb{R}^n$ – target vector

Bayesian Linear Regression

Problem statement

Training set: $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \mid i = 1, 2, \ldots, n\}$, where

$$\boldsymbol{x}_i \in \mathbb{R}^d \quad i = 1, 2, \ldots, n$$
$$y_i \in \mathbb{R}$$

Using matrix notations: $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$, where

$\boldsymbol{X} \in \mathbb{R}^{d \times n}$ – design matrix

$\boldsymbol{y} \in \mathbb{R}^n$ – target vector

Goal: Make inferences about the relationship between $\boldsymbol{X}$ and $\boldsymbol{y}$, i.e. the conditional distribution of the targets given inputs (without modeling the input distribution itself).

Bayesian Linear Regression

Problem statement

Model: $f(\boldsymbol{x}_i) = \boldsymbol{x}_i^T \boldsymbol{w}$, $\quad y_i = f(\boldsymbol{x}_i) + \epsilon$ where $\epsilon \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$.

$\boldsymbol{w} \in \mathbb{R}^d$ – weight vector

Bayesian Linear Regression

Likelihood

Model: $f(\boldsymbol{x}_i) = \boldsymbol{x}_i^T \boldsymbol{w}, \qquad y_i = f(\boldsymbol{x}_i) + \epsilon$ where $\epsilon \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$

$\boldsymbol{w} \in \mathbb{R}^d$ – weight vector

Likelihood: Probability density of the observations given the parameters.

$$
\begin{aligned}
p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}) &= \prod_{i=1}^{n} p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w}) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(y_i - \boldsymbol{x}_i^T \boldsymbol{w})^2}{2\sigma^2} \right] \\
&= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left[-\frac{1}{2\sigma^2} \|\boldsymbol{y} - \boldsymbol{X}^T \boldsymbol{w}\|^2 \right] = \mathcal{N}(\boldsymbol{X}^T \boldsymbol{w}, \sigma^2 \boldsymbol{I}) \,.
\end{aligned}
$$

Bayesian Linear Regression

Prior

Prior: Probability density of parameters expressing our beliefs about them before having a look at the data

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

Bayesian Linear Regression

Posterior

Posterior: Probability density of parameters given the observations (data).

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}} \qquad p(\boldsymbol{w} \mid \boldsymbol{y}, \boldsymbol{X}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}) \times p(\boldsymbol{w})}{p(\boldsymbol{y} \mid \boldsymbol{X})}$$

where marginal likelihood is independent of $\boldsymbol{w}$, plays a role of normalizing constant and is given by

$$p(\boldsymbol{y} \mid \boldsymbol{X}) = \int p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w})\, p(\boldsymbol{w})\, d\boldsymbol{w}\,.$$

Bayesian Linear Regression

Posterior

$$p(\boldsymbol{w} \mid \boldsymbol{X}, \boldsymbol{y}) \propto \exp\left[-\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}^T\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}^T\boldsymbol{w}) \right] \exp\left[\frac{1}{2}\boldsymbol{w}^T\boldsymbol{\Sigma}_w\boldsymbol{w} \right]$$

$$\propto \exp\left[-\frac{1}{2}(\boldsymbol{w} - \widetilde{\boldsymbol{w}})^T \underbrace{\left(\frac{1}{\sigma^2}\boldsymbol{X}\boldsymbol{X}^T + \boldsymbol{\Sigma}_w^{-1} \right)}_{\boldsymbol{A}} (\boldsymbol{w} - \widetilde{\boldsymbol{w}}) \right]$$

where $\widetilde{\boldsymbol{w}} = \frac{1}{\sigma^2}\boldsymbol{A}\boldsymbol{X}\boldsymbol{y}$ and $\boldsymbol{A} = \frac{1}{\sigma^2}\boldsymbol{X}\boldsymbol{X}^T + \boldsymbol{\Sigma}_w^{-1}$.

$$p(\boldsymbol{w} \mid \boldsymbol{X}, \boldsymbol{y}) \sim \mathcal{N}(\widetilde{\boldsymbol{w}}, \boldsymbol{A}^{-1}) .$$

Bayesian Linear Regression

Posterior predictive distribution

Assume we have a new data point $\boldsymbol{x}_*$ and goal is to predict y_*.

$$p(y_* \,|\, \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) =$$

Bayesian Linear Regression

Posterior predictive distribution

Assume we have a new data point $\boldsymbol{x}_*$ and goal is to predict y_*.

$$p(y_* \mid \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) = \int p(y_*, \boldsymbol{w} \mid \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) \, d\boldsymbol{w}$$

Bayesian Linear Regression

Posterior predictive distribution

Assume we have a new data point $\boldsymbol{x}_*$ and goal is to predict y_*.

$$
\begin{aligned}
p(y_* \mid \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) &= \int p(y_*, \boldsymbol{w} \mid \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y})\, d\boldsymbol{w} \\
&= \int p(y_* \mid \boldsymbol{x}_*, \boldsymbol{w})\, p(\boldsymbol{w} \mid \boldsymbol{X}, \boldsymbol{y})\, d\boldsymbol{w} \\
&= \mathcal{N}\left(\frac{1}{\sigma^2} \boldsymbol{x}_*{}^T \boldsymbol{A}^{-1} \boldsymbol{X} \boldsymbol{y},\ \boldsymbol{x}_*{}^T \boldsymbol{A}^{-1} \boldsymbol{x}_* \right).
\end{aligned}
$$

$p(y_* \mid \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y})$ is known as posterior predictive distribution or just predictive distribution.

Note that we can now estimate the uncertainty of our prediction.

Bayesian Linear Regression

Addressing nonlinear dependency

What if the dependency between y and $\boldsymbol{x}$ is not linear?

Bayesian Linear Regression

Addressing nonlinear dependency

What if the dependency between y and $\boldsymbol{x}$ is not linear?

Idea: Project data into some high dimensional space and then apply the linear model in that space.

Bayesian Linear Regression

Addressing nonlinear dependency

$\boldsymbol{\phi} : \mathbb{R}^d \rightarrow \mathbb{R}^k$ with $\boldsymbol{\Phi} := \Phi(\boldsymbol{X}) \in \mathbb{R}^{k \times n}$ be the aggregation of columns $\boldsymbol{\phi}(\boldsymbol{x})$ for all samples in training set.

Now the model becomes $f(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{w}$ where $\boldsymbol{w} \in \mathbb{R}^k$.

Bayesian Linear Regression

Addressing nonlinear dependency

$\boldsymbol{\phi} : \mathbb{R}^d \to \mathbb{R}^k$ with $\boldsymbol{\Phi} := \Phi(\boldsymbol{X}) \in \mathbb{R}^{k \times n}$ be the aggregation of columns $\boldsymbol{\phi}(\boldsymbol{x})$ for all samples in training set.

Now the model becomes $f(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{w}$ where $\boldsymbol{w} \in \mathbb{R}^k$.

$$p(y_* \mid \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}\left(\frac{1}{\sigma^2}\boldsymbol{\phi}(\boldsymbol{x}_*)^T \boldsymbol{A}^{-1} \boldsymbol{\Phi} \boldsymbol{y},\ \boldsymbol{\phi}(\boldsymbol{x}_*)^T \boldsymbol{A}^{-1} \boldsymbol{\phi}(\boldsymbol{x}_*)\right)$$

where $\boldsymbol{A} = \frac{1}{\sigma^2}\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \boldsymbol{\Sigma}_w^{-1}$.

Bayesian Linear Regression

Addressing nonlinear dependency

$\boldsymbol{\phi} : \mathbb{R}^d \to \mathbb{R}^k$ with $\boldsymbol{\Phi} := \Phi(\boldsymbol{X}) \in \mathbb{R}^{k \times n}$ be the aggregation of columns $\boldsymbol{\phi}(\boldsymbol{x})$ for all samples in training set.

Now the model becomes $f(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{w}$ where $\boldsymbol{w} \in \mathbb{R}^k$.

$$p(y_* \mid \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}\left(\frac{1}{\sigma^2} \boldsymbol{\phi}(\boldsymbol{x}_*)^T \boldsymbol{A}^{-1} \boldsymbol{\Phi} \boldsymbol{y},\ \boldsymbol{\phi}(\boldsymbol{x}_*)^T \boldsymbol{A}^{-1} \boldsymbol{\phi}(\boldsymbol{x}_*) \right)$$

where $\boldsymbol{A} = \frac{1}{\sigma^2} \boldsymbol{\Phi} \boldsymbol{\Phi}^T + \boldsymbol{\Sigma}_w^{-1}$.

Problem: $\boldsymbol{A}^{-1}$ is not tractable for large k!

Bayesian Linear Regression

Addressing nonlinear dependency

Denote $\boldsymbol{K} := \boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi}$ and $\boldsymbol{\phi}_* := \boldsymbol{\phi}(\boldsymbol{x}_*)$. Let's rewrite the mean:

Bayesian Linear Regression

Addressing nonlinear dependency

Denote $\boldsymbol{K} := \boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi}$ and $\boldsymbol{\phi}_* := \phi(\boldsymbol{x}_*)$. Let's rewrite the mean:

$$\boldsymbol{A}\boldsymbol{\Sigma}_w\boldsymbol{\Phi} = \frac{1}{\sigma^2}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{\Sigma}_w\boldsymbol{\Phi} + \boldsymbol{\Phi} = \frac{1}{\sigma^2}\boldsymbol{\Phi}(\boldsymbol{\Phi}^T\boldsymbol{\Sigma}_w\boldsymbol{\Phi} + \sigma^2\boldsymbol{I}) = \frac{1}{\sigma^2}\boldsymbol{\Phi}(\boldsymbol{K} + \sigma^2\boldsymbol{I})$$

Bayesian Linear Regression

Addressing nonlinear dependency

Denote $\boldsymbol{K} := \boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi}$ and $\boldsymbol{\phi}_* := \phi(\boldsymbol{x}_*)$. Let's rewrite the mean:

$$\boldsymbol{A}\boldsymbol{\Sigma}_w \boldsymbol{\Phi} = \frac{1}{\sigma^2}\boldsymbol{\Phi}\boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi} + \boldsymbol{\Phi} = \frac{1}{\sigma^2}\boldsymbol{\Phi}(\boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi} + \sigma^2 \boldsymbol{I}) = \frac{1}{\sigma^2}\boldsymbol{\Phi}(\boldsymbol{K} + \sigma^2 \boldsymbol{I})\,.$$

Multiplying both sides by $\boldsymbol{A}^{-1}$ from left and $(\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1}$ from right we will get

$$\boldsymbol{\Sigma}_w \boldsymbol{\Phi}(\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1} = \frac{1}{\sigma^2}\boldsymbol{A}^{-1}\boldsymbol{\Phi}\,,$$

by using which we can rewrite the mean of posterior predictive distribution as

$$\frac{1}{\sigma^2}\boldsymbol{\phi}_*{}^T \boldsymbol{A}^{-1}\boldsymbol{\Phi}\boldsymbol{y} = \boldsymbol{\phi}_*{}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi}(\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1}\boldsymbol{y}\,.$$

Bayesian Linear Regression

Addressing nonlinear dependency

Now let's rewrite the variance.

Lemma: Matrix inversion lemma or Woodbury matrix identity

$$(\boldsymbol{B}+\boldsymbol{UCV})^{-1}=\boldsymbol{B}^{-1}-\boldsymbol{B}^{-1}\boldsymbol{U}(\boldsymbol{C}^{-1}+\boldsymbol{VB}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VB}^{-1}.$$

$$\boldsymbol{A}^{-1}=\left(\frac{1}{\sigma^2}\boldsymbol{\Phi}\boldsymbol{\Phi}^T+\boldsymbol{\Sigma}_w^{-1}\right)^{-1}=\boldsymbol{\Sigma}_w-\boldsymbol{\Sigma}_w\boldsymbol{\Phi}(\sigma^2\boldsymbol{I}+\underbrace{\boldsymbol{\Phi}^T\boldsymbol{\Sigma}_w\boldsymbol{\Phi}}_{\boldsymbol{K}})^{-1}\boldsymbol{\Phi}^T\boldsymbol{\Sigma}_w$$

Bayesian Linear Regression

Addressing nonlinear dependency

Now let's rewrite the variance.

Lemma: Matrix inversion lemma or Woodbury matrix identity

$$(\boldsymbol{B}+\boldsymbol{UCV})^{-1} = \boldsymbol{B}^{-1} - \boldsymbol{B}^{-1}\boldsymbol{U}(\boldsymbol{C}^{-1}+\boldsymbol{VB}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VB}^{-1}$$

$$\boldsymbol{A}^{-1} = \left(\frac{1}{\sigma^2}\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \boldsymbol{\Sigma}_w^{-1}\right)^{-1} = \boldsymbol{\Sigma}_w - \boldsymbol{\Sigma}_w\boldsymbol{\Phi}(\sigma^2\boldsymbol{I} + \underbrace{\boldsymbol{\Phi}^T\boldsymbol{\Sigma}_w\boldsymbol{\Phi}}_{\boldsymbol{K}})^{-1}\boldsymbol{\Phi}^T\boldsymbol{\Sigma}_w\,,$$

$$\boldsymbol{\phi_*}^T\boldsymbol{A}^{-1}\boldsymbol{\phi_*} = \boldsymbol{\phi_*}^T\boldsymbol{\Sigma}_w\boldsymbol{\phi_*}^T - \boldsymbol{\phi_*}^T\boldsymbol{\Sigma}_w\boldsymbol{\Phi}(\boldsymbol{K}+\sigma^2\boldsymbol{I})^{-1}\boldsymbol{\Phi}^T\boldsymbol{\Sigma}_w\boldsymbol{\phi_*}\,.$$

Bayesian Linear Regression

Addressing nonlinear dependency

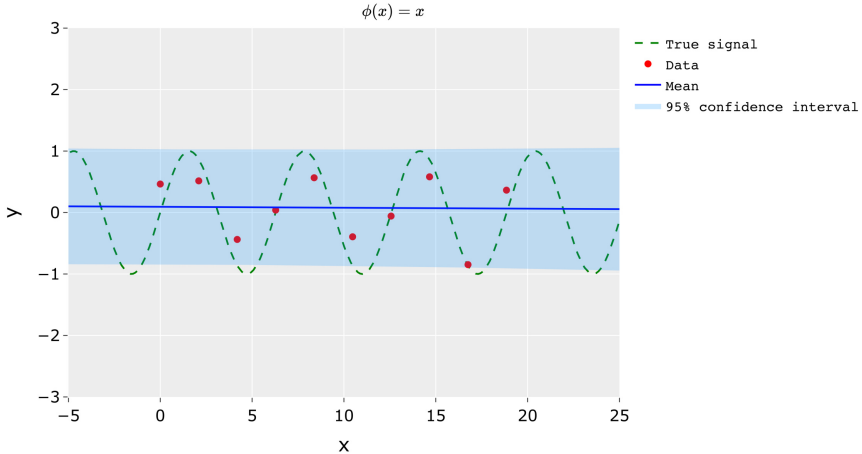
Combining the results we get:

$$p(y_* \mid \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}\Big(\boldsymbol{\phi_*}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi}(\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{y}, \\ \boldsymbol{\phi_*}^T \boldsymbol{\Sigma}_w \boldsymbol{\phi_*}^T - \boldsymbol{\phi_*}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi}(\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \boldsymbol{\phi_*}\Big)$$

where we need to invert matrix of size $n \times n$ which is more convenient when $n < k$.

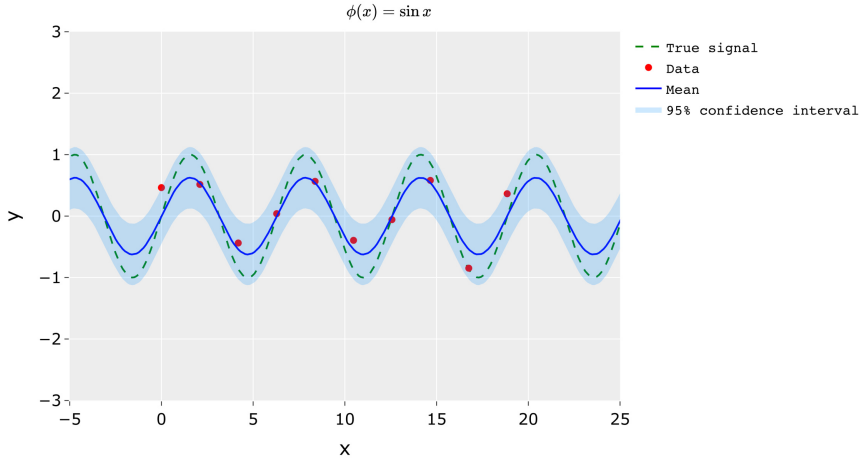
Bayesian Linear Regression

Addressing nonlinear dependency



Bayesian Linear Regression

Addressing nonlinear dependency



Bayesian Linear Regression

Kernel trick

Notice that feature space always enters in the forms of $\boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi}$, ${\boldsymbol{\phi}_*}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi}$, ${\boldsymbol{\phi}_*}^T \boldsymbol{\Sigma}_w \boldsymbol{\phi}_*$.

Entries of these matrices are of the form $\boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\Sigma}_w \boldsymbol{\phi}(\boldsymbol{x'}) =: k(\boldsymbol{x}, \boldsymbol{x'})$.

- $k(\boldsymbol{x}, \boldsymbol{x'})$ is known as *kernel* or *covariance function*

Bayesian Linear Regression

Kernel trick

Notice that feature space always enters in the form of $\boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi}$, ${\boldsymbol{\phi}_*}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi}$, ${\boldsymbol{\phi}_*}^T \boldsymbol{\Sigma}_w \boldsymbol{\phi}_*$.
Entries of these matrices are of the form $\boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\Sigma}_w \boldsymbol{\phi}(\boldsymbol{x'}) =: k(\boldsymbol{x}, \boldsymbol{x'})$.

- $k(\boldsymbol{x}, \boldsymbol{x'})$ is known as *kernel* or *covariance function*

Define $\boldsymbol{\psi}(\boldsymbol{x}) := \boldsymbol{\Sigma}_w^{1/2} \boldsymbol{\phi}(\boldsymbol{x})$ we can rewrite kernel as a simple dot product

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \boldsymbol{\psi}(\boldsymbol{x})^T \boldsymbol{\psi}(\boldsymbol{x'}),$$

where $\boldsymbol{\Sigma}_w^{1/2} = \boldsymbol{U} \boldsymbol{D}^{1/2} \boldsymbol{U}^T$ for $\boldsymbol{U} \boldsymbol{D} \boldsymbol{U}^T$ being the SVD of $\boldsymbol{\Sigma}_w$.

Bayesian Linear Regression

Kernel trick

Notice that feature space always enters in the form of $\boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi}$, ${\boldsymbol{\phi}_*}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi}$, ${\boldsymbol{\phi}_*}^T \boldsymbol{\Sigma}_w \boldsymbol{\phi}_*$.
Entries of these matrices are of the form $\boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\Sigma}_w \boldsymbol{\phi}(\boldsymbol{x'}) =: k(\boldsymbol{x}, \boldsymbol{x'})$.

- $k(\boldsymbol{x}, \boldsymbol{x'})$ is known as *kernel* or *covariance function*

Define $\boldsymbol{\psi}(\boldsymbol{x}) := \boldsymbol{\Sigma}_w^{1/2} \boldsymbol{\phi}(\boldsymbol{x})$ we can rewrite kernel as a simple dot product

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \boldsymbol{\psi}(\boldsymbol{x})^T \boldsymbol{\psi}(\boldsymbol{x'}),$$

where $\boldsymbol{\Sigma}_w^{1/2} = \boldsymbol{U} \boldsymbol{D}^{1/2} \boldsymbol{U}^T$ for $\boldsymbol{U} \boldsymbol{D} \boldsymbol{U}^T$ being the SVD of $\boldsymbol{\Sigma}_w$.

Kernels provide an efficient way to transform data without having to compute coordinates in a new space, i.e. $\boldsymbol{\phi}(\boldsymbol{x})$.

Outline

Gaussian Processes

Problem statement

Goal: Find hypothesis function that fits the data.

Gaussian Process Regression

Problem statement

Goal: Find hypothesis function that fits the data.

Assumption: $y_i = f(\boldsymbol{x}_i) + \epsilon$

- $\epsilon \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$
- f is a function-valued r.v. drawn from a Gaussian Process conditioned on data

Gaussian Processes

Gaussian Process

Definition: Gaussian Process

Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a positive definite kernel and $m : \mathbb{R}^d \to \mathbb{R}$ be a finite real-valued function. Then a random function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be a Gaussian Process (GP) with mean m and covariance kernel k, denoted by $GP(m, k)$, if the following holds:
for any finite set $X = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) \subset \mathbb{R}^d$ of any size $n \in \mathbb{N}$, the random vector

$$f_X = (f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \ldots, f(\boldsymbol{x}_n))^T \in \mathbb{R}^n$$

follows a multivariate normal distribution $\mathcal{N}(m_X, k_{XX})$ with covariance $k_{XX} = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1}^n$ and mean vector $m_X = [m(\boldsymbol{x}_1), m(\boldsymbol{x}_2), \ldots, m(\boldsymbol{x}_n)]^T$. It is written as

$$f \sim GP(m\,, k)\,.$$

Gaussian Processes

Gaussian Process

Since m and k are respectively the mean and covariance functions of a Gaussian process, they can be written as

$$m(\boldsymbol{x}) = \mathbb{E}_{f\sim GP(m,k)}[f(\boldsymbol{x})]\,,$$
$$k(\boldsymbol{x},\boldsymbol{x'}) = \mathbb{E}_{f\sim GP(m,k)}\left[\left(f(\boldsymbol{x})-m(\boldsymbol{x})\right)\left(f(\boldsymbol{x'})-m(\boldsymbol{x'})\right)\right].$$

The function values $f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \ldots, f(\boldsymbol{x}_n)$ are jointly Gaussian for any finite n.

Gaussian Processes

Example: Prior

Define

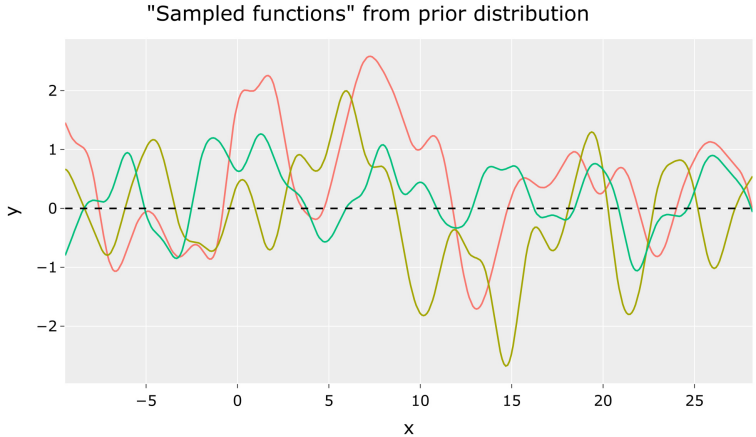
$$k(\boldsymbol{x}, \boldsymbol{x'}) = \mathrm{cov}(f(\boldsymbol{x}), f(\boldsymbol{x'})) = \exp\left(-\frac{1}{2}||\boldsymbol{x} - \boldsymbol{x'}||_2^2 \right).$$

With this covariacne function we can defince covariance matrix $K(X_*, X_*)$ between two sets of points by applying equation above elementwise for each pair of points.

Let's draw some function values $f \sim \mathcal{N}(\mathbf{0}, K(\boldsymbol{X}_*, \boldsymbol{X}_*))$.

Gaussian Processes

Example: Prior



Gaussian Processes

Computing the predictions

Assumption: $y = f(\boldsymbol{x}) + \epsilon$ where noise is i.i.d. $\mathcal{N}(0, \sigma^2)$.

Then,

$$\begin{aligned}\operatorname{cov}(y_i, y_j) = \operatorname{cov}\big(f(\boldsymbol{x}_i) + \epsilon_i, f(\boldsymbol{x}_j) + \epsilon_j\big) = \\ \operatorname{cov}\big(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)\big) + \operatorname{cov}\big(f(\boldsymbol{x}_i), \epsilon_j\big) + \\ \operatorname{cov}\big(\epsilon_i, f(\boldsymbol{x}_j)\big) + \operatorname{cov}\big(\epsilon_i, \epsilon_j\big) = \\ k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sigma^2 \delta_{ij}\end{aligned}$$

or

$$\operatorname{cov}(\boldsymbol{y}) = k(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 \boldsymbol{I}\,.$$

Gaussian Processes

Computing the predictions: Joint distribution of observed and test values

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

Gaussian Processes

Computing the predictions: Posterior predictive distribution

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

$$\boldsymbol{f}_* \,|\, \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{X}_* \sim \mathcal{N}(\overline{\boldsymbol{f}}_*, \operatorname{cov}(\boldsymbol{f}_*))$$

where

$$\overline{\boldsymbol{f}}_* := \mathbb{E}[\boldsymbol{f}_* \,|\, \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{X}_*] = K(\boldsymbol{X}_*, \boldsymbol{X})\big(K(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 \boldsymbol{I}\big)^{-1} \boldsymbol{y}\,,$$
$$\operatorname{cov}(\boldsymbol{f}_*) = K(\boldsymbol{X}_*, \boldsymbol{X}_*) - K(\boldsymbol{X}_*, \boldsymbol{X}_*)\big(K(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 \boldsymbol{I}\big)^{-1} K(\boldsymbol{X}, \boldsymbol{X}_*)\,.$$

Gaussian Processes

Computing the predictions: Posterior predictive distribution

Denote $\boldsymbol{K} := K(\boldsymbol{X}, \boldsymbol{X})$, $\boldsymbol{K_*} := K(\boldsymbol{X}, \boldsymbol{X_*})$,
$\boldsymbol{k}_* := \boldsymbol{k}(\boldsymbol{x}_*) = [k(\boldsymbol{x}_i, \boldsymbol{x}_*)]_{i=1}^n$ – the covariance vector of a test point $\boldsymbol{x}_*$ and n training points.

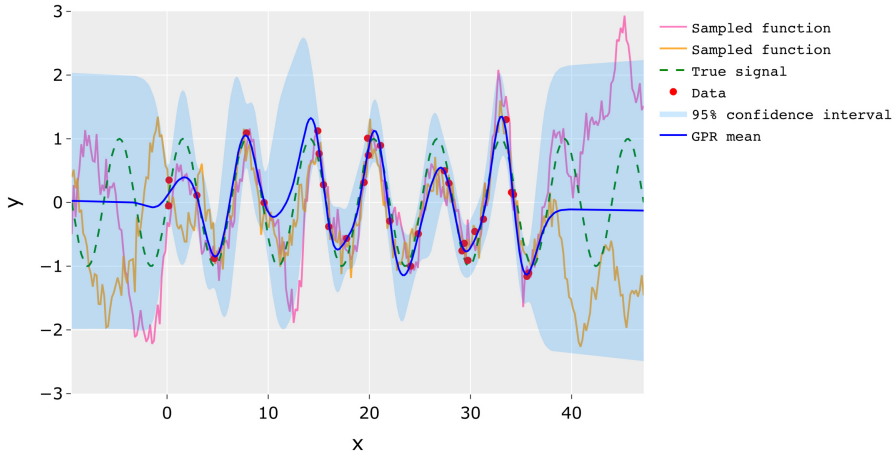
$$f_* \,|\, \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{x}_* \sim \mathcal{N}(\bar{f}_*,\, \mathbb{V}[f_*])$$

where

$$\bar{f}_* = \boldsymbol{k}_*^T (\boldsymbol{K} + \sigma^2 \boldsymbol{I}])^{-1} \boldsymbol{y}\,,$$
$$\mathbb{V}[f_*] = \boldsymbol{k}(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_*^T (\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{k}_*\,.$$

Gaussian Processes

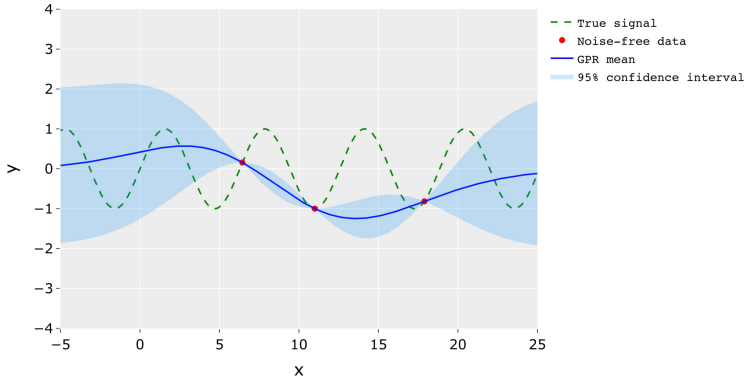
Example



Gaussian Processes

Example: Noise-free case

$$\epsilon = 0 \quad \Leftrightarrow \quad \sigma^2 = 0$$



Gaussian Processes

Equivalence to Bayesian Linear Regression

$$\overline{f}(\boldsymbol{x}_*) = \boldsymbol{k}(\boldsymbol{x}_*)^T(\boldsymbol{K} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{y} = \sum_{i=1}^{n} \alpha_i \boldsymbol{y}_i$$

Corresponds to posterior predictive mean of Bayesian Linear Regression (slide 23)

$$\boldsymbol{\phi}_*{}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi}(\boldsymbol{K} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}$$

if we define $k(\boldsymbol{x}_i, \boldsymbol{x}_j) := \boldsymbol{\phi}(\boldsymbol{x_i})^T \boldsymbol{\Sigma}_w \, \boldsymbol{\phi}(\boldsymbol{x_j})$.

Gaussian Processes

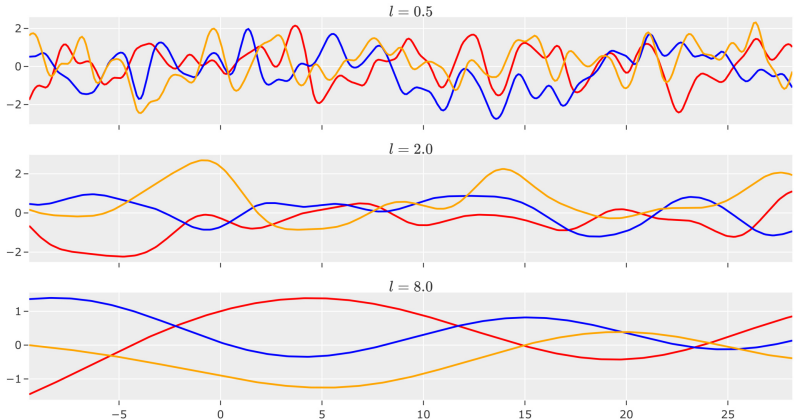
Hyperparameter optimization: RBF kernel

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2}{2\,l^2} \right)$$

- How to choose hyperparameters (e.g. l)?

Gaussian Processes

Hyperparameter optimization: RBF kernel



Gaussian Processes

Hyperparameter optimization

Let's introduce marginal likelihood:

$$p(\boldsymbol{y} \mid \boldsymbol{X}) = \int \overbrace{p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{X})}^{\text{likelihood}} \overbrace{p(\boldsymbol{f} \mid \boldsymbol{X})}^{\text{prior}} \, d\boldsymbol{f}$$

Gaussian Processes

Hyperparameter optimization

Let's introduce marginal likelihood:

$$p(\boldsymbol{y} \mid \boldsymbol{X}) = \int \overbrace{p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{X})}^{\text{likelihood}} \overbrace{p(\boldsymbol{f} \mid \boldsymbol{X})}^{\text{prior}} \, d\boldsymbol{f}$$

Since prior is Gaussian, i.e. $\boldsymbol{f} \mid \boldsymbol{X} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K})$, then

$$\log p(\boldsymbol{f} \mid \boldsymbol{X}) = -\frac{1}{2} \boldsymbol{f}^T \boldsymbol{K}^{-1} \boldsymbol{f} - \frac{1}{2} \log |\boldsymbol{K}| - \frac{n}{2} \log 2\pi \, .$$

Gaussian Processes

Hyperparameter optimization

Using the fact that likelihood is also Gaussian, $\boldsymbol{y} \,|\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{f}, \sigma^2 \boldsymbol{I})$, the final log marginal likelihood is given by

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

Gaussian Processes

Hyperparameter optimization

Using the fact that likelihood is also Gaussian, $\boldsymbol{y} \,|\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{f}, \sigma^2 \boldsymbol{I})$, the final log marginal likelihood is given by

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

Let Θ be the set of all hyperparameters. Note that K is dependent on Θ. Then

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}} \log p(\boldsymbol{y} \,|\, \boldsymbol{X}, \Theta) \,.$$

Gaussian Processes

Advantages vs. Disadvantages

Gaussian Process Regression is a function-space view of regression problem.

Advantages

- Flexibility and interpretability
- Can exactly be optimized (given the hyperparameters)

Disadvantages

- Prone to outliers (use the whole data to perform prediction)
- Not efficient for high-dimensional data

Thank you!

References

[1] M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur. *Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences*. 2018. arXiv: 1807.02582 [stat.ML].

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[3] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006, pp. I–XVIII, 1–248. ISBN: 026218253X.