

Mathematics for Machine Learning

— Linear Regression

Problem Formulation & Parameter Estimation

Joseph Chuang-Chieh Lin

Department of Computer Science & Engineering,
National Taiwan Ocean University

Fall 2025

Credits for the resource

- The slides are based on the textbooks:
 - *Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong: Mathematics for Machine Learning. Cambridge University Press. 2020.*
 - *Arnold J. Insel, Lawrence E. Spence, Stephen H. Friedberg: Linear Algebra, 4th Edition. Prentice Hall. 2013.*
 - *Howard Anton, Chris Rorres, Anton Kaul: Elementary Linear Algebra, 12th Edition. Wiley. 2019.*
- We could partially refer to the monograph:
Francesco Orabona: A Modern Introduction to Online Learning.
<https://arxiv.org/abs/1912.13213>

Outline

- 1 Introduction
- 2 Problem Formulation
- 3 Parameter Estimation
 - Maximum Likelihood Estimation (MLE)
 - Overfitting in Linear Regression
 - Maximum A Posteriori Estimation (MAP)
 - MAP Estimation as Regularization
- 4 Bayesian Linear Regression

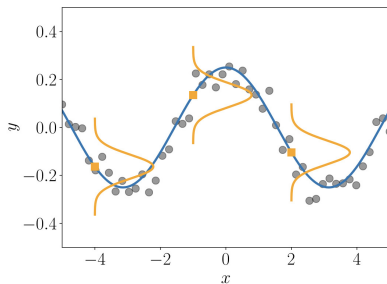
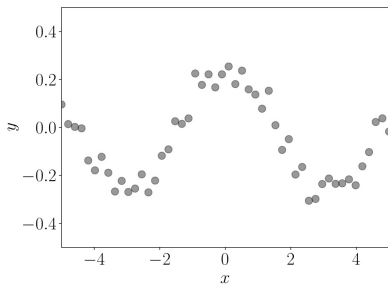
Linear Regression

Aim

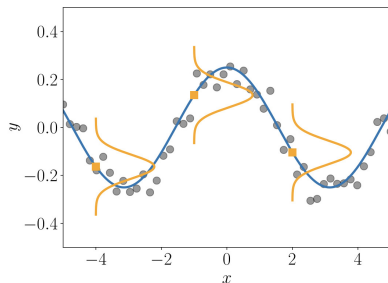
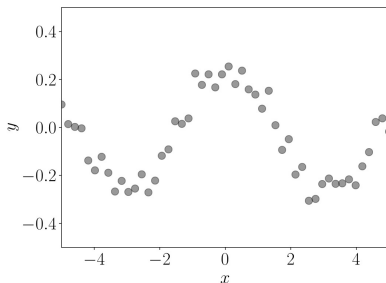
Find (or Infer) a function $f : \mathbb{R}^D \rightarrow \mathbb{R}$ which maps input $\mathbf{x} \in \mathbb{R}^D$ to the corresponding function values $f(\mathbf{x}) \in \mathbb{R}$.

- And we hope f to generalize well to unseen input.
- Training input: $\{\mathbf{x}_i\}_{i=1}^N$
- Assume the noisy observations $\{y_i\}_{i=1}^N$ for $y_i = f(\mathbf{x}_i) + \epsilon$, an i.i.d. random variable ϵ .
 - Consider zero-mean Gaussian noise throughout our discussions.

- Observe (noisy) function values $y_n = f(x_n) + \epsilon$.



- Observe (noisy) function values $y_n = f(x_n) + \epsilon$.



Applications of regression:

- Time series analysis, reinforcement learning, optimization, computer games, classification algorithms, etc.

Problems Involved in Regression

- Choice of the model and the parametrization.
 - Function classes, particular parametrization (e.g., degree of the polynomial)
- Finding good parameters.
 - Loss minimization w.r.t. different loss functions.
- Overfitting and model selection.
- Relationship b/w loss functions and parameter priors.
 - Probabilistic models.
- Uncertainty modeling.
 - We have limited amount of data.
 - The smaller the training set, the more important uncertainty modeling.
 - Equip model predictions with confidence bounds.

Outline

- 1 Introduction
- 2 **Problem Formulation**
- 3 Parameter Estimation
 - Maximum Likelihood Estimation (MLE)
 - Overfitting in Linear Regression
 - Maximum A Posteriori Estimation (MAP)
 - MAP Estimation as Regularization
- 4 Bayesian Linear Regression

Problem Formulation

- Because of observing noise, we adopt a probabilistic approach to explicitly model the noise using a **likelihood function**.
- **Focus:** a regression problem with the likelihood function:

$$p(y \mid \mathbf{x}) = \mathcal{N}(y \mid f(\mathbf{x}), \sigma^2).$$

- $\mathbf{x} \in \mathbb{R}^D$: inputs.
- $y \in \mathbb{R}$: noisy function values (targets).
- The relationship between \mathbf{x} and y :

$$y = f(\mathbf{x}) + \epsilon,$$

for $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

An Example of Linear Regression

- An example of **linear regression**:

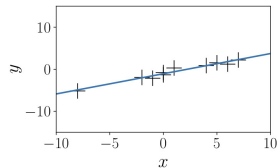
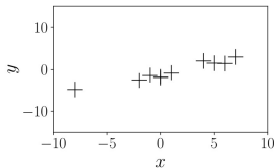
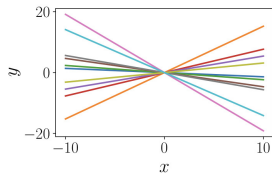
$$p(y \mid \mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y \mid \mathbf{x}^\top \boldsymbol{\theta}, \sigma^2).$$

\Longleftrightarrow

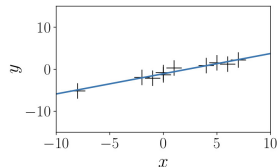
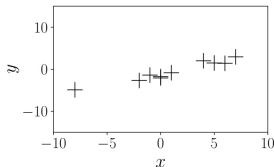
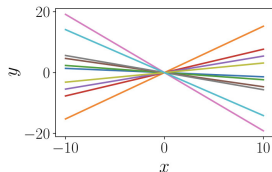
$$y = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon,$$

for $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

- $\boldsymbol{\theta} \in \mathbb{R}^D$: the **parameters** we seek.
- ϵ : the only source of uncertainty.



- “Linear”: linear in the parameters.
 - Parameters: describing a function by a linear combination of input features.



- “Linear”: linear in the parameters.
 - Parameters: describing a function by a linear combination of input features.
- Hence, $y = \phi^\top(\mathbf{x})\theta$ is also regarded as a linear regression (ϕ can be nonlinear).
 - A “feature” here is a representation $\phi(\mathbf{x})$ of the input \mathbf{x} .

Outline

- 1 Introduction
- 2 Problem Formulation
- 3 Parameter Estimation**
 - Maximum Likelihood Estimation (MLE)
 - Overfitting in Linear Regression
 - Maximum A Posteriori Estimation (MAP)
 - MAP Estimation as Regularization
- 4 Bayesian Linear Regression

The Likelihood

- Given a training set $\mathcal{D} := \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$ for $i = 1, \dots, N$.
- By the independence of the input, the likelihood factorizes:

$$p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) = p(y_1, \dots, y_N \mid \mathbf{x}_1, \dots, \mathbf{x}_N, \boldsymbol{\theta})$$

The Likelihood

- Given a training set $\mathcal{D} := \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$ for $i = 1, \dots, N$.
- By the independence of the input, the likelihood factorizes:

$$\begin{aligned} p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) &= p(y_1, \dots, y_N \mid \mathbf{x}_1, \dots, \mathbf{x}_N, \boldsymbol{\theta}) \\ &= \prod_{i=1}^N p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}) = \prod_{i=1}^N \mathcal{N}(y_i \mid \mathbf{x}_i^\top \boldsymbol{\theta}, \sigma^2). \end{aligned}$$

The likelihood and the factors $p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta})$ are Gaussian due to the noise distribution.

The Likelihood

- Given a training set $\mathcal{D} := \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$ for $i = 1, \dots, N$.
- By the independence of the input, the likelihood factorizes:

$$\begin{aligned} p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) &= p(y_1, \dots, y_N \mid \mathbf{x}_1, \dots, \mathbf{x}_N, \boldsymbol{\theta}) \\ &= \prod_{i=1}^N p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}) = \prod_{i=1}^N \mathcal{N}(y_i \mid \mathbf{x}_i^\top \boldsymbol{\theta}, \sigma^2). \end{aligned}$$

The likelihood and the factors $p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta})$ are Gaussian due to the noise distribution.

- Goal:** Find optimal parameters $\boldsymbol{\theta}^* \in \mathbb{R}^D$.

The Likelihood

- Given a training set $\mathcal{D} := \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$ for $i = 1, \dots, N$.
- By the independence of the input, the likelihood factorizes:

$$\begin{aligned} p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) &= p(y_1, \dots, y_N \mid \mathbf{x}_1, \dots, \mathbf{x}_N, \boldsymbol{\theta}) \\ &= \prod_{i=1}^N p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}) = \prod_{i=1}^N \mathcal{N}(y_i \mid \mathbf{x}_i^\top \boldsymbol{\theta}, \sigma^2). \end{aligned}$$

The likelihood and the factors $p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta})$ are Gaussian due to the noise distribution.

- Goal:** Find optimal parameters $\boldsymbol{\theta}^* \in \mathbb{R}^D$.
- Then we can make predictions for an arbitrary test input \mathbf{x}_* and get target y_* with $p(y_* \mid \mathbf{x}_*, \boldsymbol{\theta}^*) = \mathcal{N}(y_* \mid \mathbf{x}_*^\top \boldsymbol{\theta}^*, \sigma^2)$.

Outline

- 1 Introduction
- 2 Problem Formulation
- 3 **Parameter Estimation**
 - **Maximum Likelihood Estimation (MLE)**
 - Overfitting in Linear Regression
 - Maximum A Posteriori Estimation (MAP)
 - MAP Estimation as Regularization
- 4 Bayesian Linear Regression

Maximum Likelihood Estimation (MLE)

Find parameters θ_{ML}

$$\theta_{ML} \in \arg \max_{\theta} p(\mathcal{Y} \mid \mathcal{X}, \theta).$$

Note:

- The likelihood $p(y \mid \mathbf{x}, \theta)$ is **NOT** a probability distribution of θ .

Maximum Likelihood Estimation (MLE)

Find parameters θ_{ML}

$$\theta_{ML} \in \arg \max_{\theta} p(\mathcal{Y} \mid \mathcal{X}, \theta).$$

Note:

- The likelihood $p(y \mid \mathbf{x}, \theta)$ is **NOT** a probability distribution of θ . It's a function of θ (might not be integrable w.r.t θ).
- However, it's a normalized probability distribution in y .

How to find the desired θ_{ML} ?

- 1 Perform gradient ascent (or descent).

How to find the desired θ_{ML} ?

- 1 Perform **gradient ascent (or descent)**.
- 2 For linear regression, we can directly have a **closed-form** solution.

How to find the desired θ_{ML} ?

- ① Perform **gradient ascent (or descent)**.
- ② For linear regression, we can directly have a **closed-form** solution.
- ③ In practice, we do not maximize the likelihood directly. Instead, we apply the **negative log-likelihood**.
 - It does not suffer from **numerical underflow**.
 - The differentiation rules become simpler.

Maximize likelihood \Leftrightarrow Minimize negative log-likelihood

The negative log-likelihood

$$-\log p(\mathcal{Y} \mid \mathcal{X}, \theta) = -\log \prod_{i=1}^N p(y_i \mid \mathbf{x}_i, \theta)$$

Maximize likelihood \Leftrightarrow Minimize negative log-likelihood

The negative log-likelihood

$$-\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) = -\log \prod_{i=1}^N p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}) = -\sum_{i=1}^N \log p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}).$$

★ **Note:** the independence assumption on the training set applies here.

Maximize likelihood \Leftrightarrow Minimize negative log-likelihood

The negative log-likelihood

$$-\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) = -\log \prod_{i=1}^N p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}) = -\sum_{i=1}^N \log p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}).$$

★ **Note:** the independence assumption on the training set applies here.

$$\log p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}) = -\frac{1}{2\sigma^2}(y_i - \mathbf{x}^\top \boldsymbol{\theta})^2 + \text{constant}_{\text{independent of } \boldsymbol{\theta}}.$$

Ignoring the constant terms, we obtain

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &:= \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mathbf{x}_i^\top \boldsymbol{\theta})^2 \\ &= \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) = \frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2,\end{aligned}$$

where $\mathbf{X} := [\mathbf{x}_1, \dots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times D}$ and $\mathbf{y} := [y_1, \dots, y_N]^\top \in \mathbb{R}^N$.

Ignoring the constant terms, we obtain

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &:= \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mathbf{x}_i^\top \boldsymbol{\theta})^2 \\ &= \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) = \frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2,\end{aligned}$$

where $\mathbf{X} := [\mathbf{x}_1, \dots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times D}$ and $\mathbf{y} := [y_1, \dots, y_N]^\top \in \mathbb{R}^N$.

To get $\boldsymbol{\theta}$, we need to solve $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \mathbf{0}^\top$:

Ignoring the constant terms, we obtain

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &:= \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mathbf{x}_i^\top \boldsymbol{\theta})^2 \\ &= \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) = \frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2,\end{aligned}$$

where $\mathbf{X} := [\mathbf{x}_1, \dots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times D}$ and $\mathbf{y} := [y_1, \dots, y_N]^\top \in \mathbb{R}^N$.

To get $\boldsymbol{\theta}$, we need to solve $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \mathbf{0}^\top$:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \mathbf{0}^\top \iff \boldsymbol{\theta}_{ML}^\top \mathbf{X}^\top \mathbf{X} = \mathbf{y}^\top \mathbf{X}$$

Ignoring the constant terms, we obtain

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &:= \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mathbf{x}_i^\top \boldsymbol{\theta})^2 \\ &= \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) = \frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2,\end{aligned}$$

where $\mathbf{X} := [\mathbf{x}_1, \dots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times D}$ and $\mathbf{y} := [y_1, \dots, y_N]^\top \in \mathbb{R}^N$.

To get $\boldsymbol{\theta}$, we need to solve $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \mathbf{0}^\top$:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \mathbf{0}^\top &\iff \boldsymbol{\theta}_{ML}^\top \mathbf{X}^\top \mathbf{X} = \mathbf{y}^\top \mathbf{X} \\ &\iff \boldsymbol{\theta}_{ML}^\top = \mathbf{y}^\top \mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1}\end{aligned}$$

Ignoring the constant terms, we obtain

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &:= \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mathbf{x}_i^\top \boldsymbol{\theta})^2 \\ &= \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) = \frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2,\end{aligned}$$

where $\mathbf{X} := [\mathbf{x}_1, \dots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times D}$ and $\mathbf{y} := [y_1, \dots, y_N]^\top \in \mathbb{R}^N$.

To get $\boldsymbol{\theta}$, we need to solve $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \mathbf{0}^\top$:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \mathbf{0}^\top &\iff \boldsymbol{\theta}_{ML}^\top \mathbf{X}^\top \mathbf{X} = \mathbf{y}^\top \mathbf{X} \\ &\iff \boldsymbol{\theta}_{ML}^\top = \mathbf{y}^\top \mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1} \\ &\iff \boldsymbol{\theta}_{ML} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}.\end{aligned}$$

★ We use the positive definite property of $\mathbf{X}^\top \mathbf{X}$ if $\text{rank}(\mathbf{X}) = D$.

Remark

- We can get a global minimum because the Hessian $\nabla_{\theta}^2 \mathcal{L}(\theta) = \mathbf{X}^T \mathbf{X}$ is positive definite (for full rank \mathbf{X} ?).

MLE with Features

- Note that “linear” regression is linear in the “parameters”.
- We can perform an arbitrary **nonlinear** transformation $\phi(\mathbf{x})$ of the input \mathbf{x} , and then linearly combine these components.

MLE with Features

- Note that “linear” regression is linear in the “parameters”.
- We can perform an arbitrary **nonlinear** transformation $\phi(\mathbf{x})$ of the input \mathbf{x} , and then linearly combine these components.
- The corresponding linear regression turns out to be:

$$p(y \mid \mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y \mid \boldsymbol{\phi}^\top(\mathbf{x})\boldsymbol{\theta}, \sigma^2).$$



$$y = \boldsymbol{\phi}^\top(\mathbf{x})\boldsymbol{\theta} + \epsilon$$

MLE with Features

- Note that “linear” regression is linear in the “parameters”.
- We can perform an arbitrary **nonlinear** transformation $\phi(\mathbf{x})$ of the input \mathbf{x} , and then linearly combine these components.
- The corresponding linear regression turns out to be:

$$p(y \mid \mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y \mid \boldsymbol{\phi}^\top(\mathbf{x})\boldsymbol{\theta}, \sigma^2).$$

 \Longleftrightarrow

$$y = \boldsymbol{\phi}^\top(\mathbf{x})\boldsymbol{\theta} + \epsilon = \sum_{k=0}^{K-1} \theta_k \phi_k(\mathbf{x}) + \epsilon$$

- $\phi : \mathbb{R}^D \rightarrow \mathbb{R}$ is a (nonlinear) transformation of the input \mathbf{x}
- $\phi_k : \mathbb{R}^D \rightarrow \mathbb{R}$: the k th feature vector of ϕ .

Polynomial Regression (Example)

Consider a regression problem $y = \phi^\top(x)\theta + \epsilon$, for $x \in \mathbb{R}$ and $\theta \in \mathbb{R}^K$. A polynomial transformation of x is often used as

$$\phi(x) = \begin{bmatrix} \phi_0(x) \\ \phi_1(x) \\ \vdots \\ \phi_{K-1}(x) \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^{K-1} \end{bmatrix} \in \mathbb{R}^K.$$

- We lift the original one-dimensional input space into a K -dimensional feature space.
- We can model polynomials of degree $\leq K - 1$ as $f(x) = \sum_{k=1}^{K-1} \theta_k x^k = \phi^\top(x)\theta$, for $\theta = [\theta_0, \dots, \theta_{K-1}]^\top \in \mathbb{R}^K$ which contains the linear parameters θ_k .

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

We can also define a feature matrix as

$$\Phi := \begin{bmatrix} \phi^\top(\mathbf{x}_1) \\ \vdots \\ \phi^\top(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \cdots & \phi_{K-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \cdots & \phi_{K-1}(\mathbf{x}_2) \\ \vdots & & \vdots \\ \phi_0(\mathbf{x}_N) & \cdots & \phi_{K-1}(\mathbf{x}_N) \end{bmatrix} \in \mathbb{R}^{N \times K},$$

where $\Phi_{ij} = \phi_j(\mathbf{x}_i)$ and $\phi_j : \mathbb{R}^D \rightarrow \mathbb{R}$.

Example

Feature Matrix for Second-Order Polynomials

$$\Phi := \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 \end{bmatrix}.$$

With the feature matrix Φ :

$$\Phi := \begin{bmatrix} \phi^\top(\mathbf{x}_1) \\ \vdots \\ \phi^\top(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \cdots & \phi_{K-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \cdots & \phi_{K-1}(\mathbf{x}_2) \\ \vdots & & \vdots \\ \phi_0(\mathbf{x}_N) & \cdots & \phi_{K-1}(\mathbf{x}_N) \end{bmatrix} \in \mathbb{R}^{N \times K},$$

The negative log-likelihood can be written as

$$-\log p(\mathcal{Y} \mid \mathcal{X}, \theta) = \frac{1}{2\sigma^2} (\mathbf{y} - \Phi\theta)^\top (\mathbf{y} - \Phi\theta) + \text{constant}.$$

- Replacing \mathbf{X} by Φ .
- Both of them are independent of θ .

¹Requiring $\text{rank}(\Phi) = K$

With the feature matrix Φ :

$$\Phi := \begin{bmatrix} \phi^\top(\mathbf{x}_1) \\ \vdots \\ \phi^\top(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \cdots & \phi_{K-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \cdots & \phi_{K-1}(\mathbf{x}_2) \\ \vdots & & \vdots \\ \phi_0(\mathbf{x}_N) & \cdots & \phi_{K-1}(\mathbf{x}_N) \end{bmatrix} \in \mathbb{R}^{N \times K},$$

The negative log-likelihood can be written as

$$-\log p(\mathcal{Y} \mid \mathcal{X}, \theta) = \frac{1}{2\sigma^2} (\mathbf{y} - \Phi\theta)^\top (\mathbf{y} - \Phi\theta) + \text{constant}.$$

- Replacing \mathbf{X} by Φ .
- Both of them are independent of θ .
- Similarly, we have¹

$$\theta_{ML} = (\Phi^\top \Phi)^{-1} \Phi^\top \mathbf{y}.$$

¹Requiring $\text{rank}(\Phi) = K$

Estimating the Noise Variance (1/2)

- We can also use the principle of MLE to obtain that for σ_{ML}^2 for the noise variance.

Estimating the Noise Variance (1/2)

- We can also use the principle of MLE to obtain that for σ_{ML}^2 for the noise variance.
- Write down the log-likelihood:

$$\begin{aligned}\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}, \sigma^2) &= \sum_{i=1}^N \log \mathcal{N}(y_i \mid \boldsymbol{\phi}^\top(\mathbf{x}_i)\boldsymbol{\theta}, \sigma^2) \\&= \sum_{i=1}^N \left(-\frac{1}{2} \log(2\pi) - \frac{1}{2} \log \sigma^2 - \frac{1}{2\sigma^2} (y_i - \boldsymbol{\phi}^\top(\mathbf{x}_i)\boldsymbol{\theta})^2 \right) \\&= -\frac{N}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \boldsymbol{\phi}^\top(\mathbf{x}_i)\boldsymbol{\theta})^2 + \text{constant}\end{aligned}$$

Estimating the Noise Variance (1/2)

- We can also use the principle of MLE to obtain that for σ_{ML}^2 for the noise variance.
- Write down the log-likelihood:

$$\begin{aligned}\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}, \sigma^2) &= \sum_{i=1}^N \log \mathcal{N}(y_i \mid \boldsymbol{\phi}^\top(\mathbf{x}_i)\boldsymbol{\theta}, \sigma^2) \\ &= \sum_{i=1}^N \left(-\frac{1}{2} \log(2\pi) - \frac{1}{2} \log \sigma^2 - \frac{1}{2\sigma^2} (y_i - \boldsymbol{\phi}^\top(\mathbf{x}_i)\boldsymbol{\theta})^2 \right) \\ &= -\frac{N}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \boldsymbol{\phi}^\top(\mathbf{x}_i)\boldsymbol{\theta})^2 + \text{constant}\end{aligned}$$

$$\text{Let } s := \sum_{i=1}^N (y_i - \boldsymbol{\phi}^\top(\mathbf{x}_i)\boldsymbol{\theta})^2.$$

Estimating the Noise Variance (2/2)

- The partial derivative w.r.t. σ^2 :

$$\frac{\partial \log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}, \sigma^2)}{\partial \sigma^2} = -\frac{N}{2\sigma^2} + \frac{1}{\sigma^4} s = 0$$
$$\iff \frac{N}{2\sigma^2} = \frac{s}{2\sigma^4}.$$

Thus,

$$\sigma_{ML}^2 = \frac{s}{N} = \frac{1}{N} \sum_{i=1}^N (y_i - \boldsymbol{\phi}^\top(\mathbf{x}_i)\boldsymbol{\theta})^2.$$

Outline

- 1 Introduction
- 2 Problem Formulation
- 3 Parameter Estimation**
 - Maximum Likelihood Estimation (MLE)
 - **Overfitting in Linear Regression**
 - Maximum A Posteriori Estimation (MAP)
 - MAP Estimation as Regularization
- 4 Bayesian Linear Regression

Evaluating the Quality of the Model

- We can evaluate the quality of the model by computing the error/loss.
- Given that σ^2 is not a free model parameter, we can ignore that term by scaling by $1/\sigma^2$ and derive a squared-error function $\|\mathbf{y} - \Phi\boldsymbol{\theta}\|^2$.

Evaluating the Quality of the Model

- We can evaluate the quality of the model by computing the error/loss.
- Given that σ^2 is not a free model parameter, we can ignore that term by scaling by $1/\sigma^2$ and derive a squared-error function $\|\mathbf{y} - \Phi\boldsymbol{\theta}\|^2$.
- To compare the errors of datasets with **different sizes** and **the same scale**, we often use the root-mean squared error (RMSE):

Evaluating the Quality of the Model

- We can evaluate the quality of the model by computing the error/loss.
- Given that σ^2 is not a free model parameter, we can ignore that term by scaling by $1/\sigma^2$ and derive a squared-error function $\|\mathbf{y} - \Phi\boldsymbol{\theta}\|^2$.
- To compare the errors of datasets with **different sizes** and **the same scale**, we often use the root-mean squared error (RMSE):

$$\sqrt{\frac{1}{N}\|\mathbf{y} - \Phi\boldsymbol{\theta}\|^2} = \sqrt{\frac{1}{N}\sum_{i=1}^N (y_i - \phi^\top(\mathbf{x}_i)\boldsymbol{\theta})^2}$$

Evaluating the Quality of the Model

- We can evaluate the quality of the model by computing the error/loss.
- Given that σ^2 is not a free model parameter, we can ignore that term by scaling by $1/\sigma^2$ and derive a squared-error function $\|\mathbf{y} - \Phi\boldsymbol{\theta}\|^2$.
- To compare the errors of datasets with **different sizes** and **the same scale**, we often use the root-mean squared error (RMSE):

$$\sqrt{\frac{1}{N}\|\mathbf{y} - \Phi\boldsymbol{\theta}\|^2} = \sqrt{\frac{1}{N}\sum_{i=1}^N (y_i - \phi^\top(\mathbf{x}_i)\boldsymbol{\theta})^2}$$

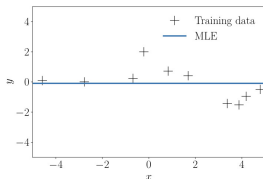
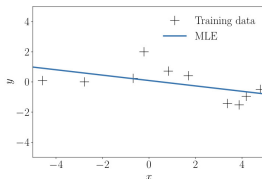
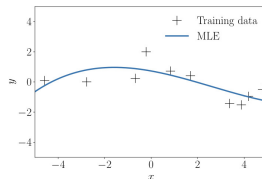
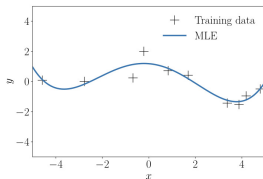
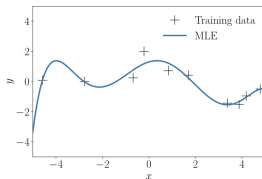
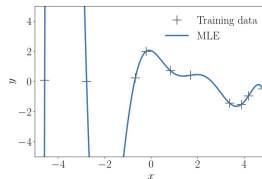
- Model selection:

Evaluating the Quality of the Model

- We can evaluate the quality of the model by computing the error/loss.
- Given that σ^2 is not a free model parameter, we can ignore that term by scaling by $1/\sigma^2$ and derive a squared-error function $\|\mathbf{y} - \Phi\boldsymbol{\theta}\|^2$.
- To compare the errors of datasets with **different sizes** and **the same scale**, we often use the root-mean squared error (RMSE):

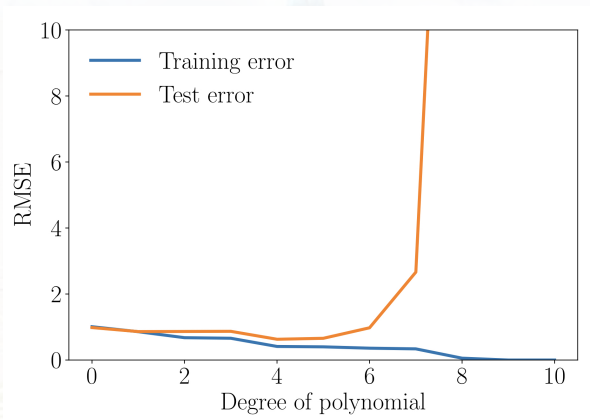
$$\sqrt{\frac{1}{N}\|\mathbf{y} - \Phi\boldsymbol{\theta}\|^2} = \sqrt{\frac{1}{N}\sum_{i=1}^N (y_i - \phi^\top(\mathbf{x}_i)\boldsymbol{\theta})^2}$$

- Model selection: determine the best degree of the polynomial.
 - Brute-force searching and enumerate all reasonable polynomial degrees M .

(a) $M = 0$ (b) $M = 1$ (c) $M = 3$ (d) $M = 4$ (e) $M = 6$ (f) $M = 9$

Goal: a good generalization by making *accurate* predictions for new unseen data.

- A separate test set comprising 200 data points generated using exactly the same procedure used to generate the training set.



Outline

- 1 Introduction
- 2 Problem Formulation
- 3 Parameter Estimation**
 - Maximum Likelihood Estimation (MLE)
 - Overfitting in Linear Regression
 - Maximum A Posteriori Estimation (MAP)**
 - MAP Estimation as Regularization
- 4 Bayesian Linear Regression

Motivation

- MLE is prone to overfitting.
- **Experience:** The parameter values becomes relatively large when the model is overfitting.

Motivation

- MLE is prone to overfitting.
- **Experience:** The parameter values becomes relatively large when the model is overfitting.
- To mitigate the effect of huge parameter values, we place a **prior distribution $p(\theta)$** on the parameters.

Motivation

- MLE is prone to overfitting.
- **Experience:** The parameter values becomes relatively large when the model is overfitting.
- To mitigate the effect of huge parameter values, we place a **prior distribution** $p(\theta)$ on the parameters.
- **Rough idea:** Encode the parameter values that are plausible before seeing any data.
 - For example, a Gaussian prior $p(\theta) = \mathcal{N}(\mathbf{0}, I)$.

Maximum a Posteriori Estimation (1/5)

- Once a dataset $(\mathcal{X}, \mathcal{Y})$ is available, we seek parameters that maximize the posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y})$ instead of maximizing the likelihood.

$$p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) = \frac{p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{Y} \mid \mathcal{X})}.$$

Maximum a Posteriori Estimation (1/5)

- Once a dataset $(\mathcal{X}, \mathcal{Y})$ is available, we seek parameters that maximize the posterior distribution $p(\boldsymbol{\theta} | \mathcal{X}, \mathcal{Y})$ instead of maximizing the likelihood.

$$p(\boldsymbol{\theta} | \mathcal{X}, \mathcal{Y}) = \frac{p(\mathcal{Y} | \mathcal{X}, \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{Y} | \mathcal{X})}.$$

- The prior will have an effect on the parameter vector.

Maximum a Posteriori Estimation (1/5)

- Once a dataset $(\mathcal{X}, \mathcal{Y})$ is available, we seek parameters that maximize the posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y})$ instead of maximizing the likelihood.

$$p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) = \frac{p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{Y} \mid \mathcal{X})}.$$

- The prior will have an effect on the parameter vector.
- $\boldsymbol{\theta}_{MAP}$: the maximizer of the above posterior (i.e., the MAP estimate).

Maximum a Posteriori Estimation (2/5)

The log-transformation of the posterior:

$$\log p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) = \log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) + \text{constant}$$

The constant is independent of $\boldsymbol{\theta}$.

We can see that the MAP estimate is a compromise between the prior and the data-dependent likelihood.

Maximum a Posteriori Estimation (2/5)

The log-transformation of the posterior:

$$\log p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) = \log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) + \text{constant}$$

The constant is independent of $\boldsymbol{\theta}$.

We can see that the MAP estimate is a compromise between the prior and the data-dependent likelihood.

We minimize the negative log-posterior w.r.t. $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}_{MAP} \in \arg \min_{\boldsymbol{\theta}} \{-\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) - \log p(\boldsymbol{\theta})\}.$$

Maximum a Posteriori Estimation (3/5)

$$\boldsymbol{\theta}_{MAP} \in \arg \min_{\boldsymbol{\theta}} \{-\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) - \log p(\boldsymbol{\theta})\}.$$

The gradient:

$$-\frac{d \log p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y})}{d\boldsymbol{\theta}} = -\frac{d \log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta})}{d\boldsymbol{\theta}} - \frac{d \log p(\boldsymbol{\theta})}{d\boldsymbol{\theta}}.$$

Assume the Gaussian prior $p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, b^2 \mathbf{I})$. We have

$$-\log p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) = \frac{1}{2\sigma^2} (\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\theta})^\top (\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\theta}) + \frac{1}{2b^2} \boldsymbol{\theta}^\top \boldsymbol{\theta} + \text{constant}$$

Maximum a Posteriori Estimation (4/5)

$$-\log p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) = \frac{1}{2\sigma^2}(\mathbf{y} - \Phi\boldsymbol{\theta})^\top(\mathbf{y} - \Phi\boldsymbol{\theta}) + \frac{1}{2b^2}\boldsymbol{\theta}^\top\boldsymbol{\theta} + \text{constant}$$

Hence, the gradient of the log-posterior w.r.t. $\boldsymbol{\theta}$ is

$$-\frac{d \log p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y})}{d\boldsymbol{\theta}} = \frac{1}{\sigma^2}(\boldsymbol{\theta}^\top \Phi^\top \Phi - \mathbf{y}^\top \Phi) + \frac{1}{b^2}\boldsymbol{\theta}^\top.$$

Setting the gradient to $\mathbf{0}^\top$ to get $\boldsymbol{\theta}_{MAP}$:

Maximum a Posteriori Estimation (5/5)

$$\begin{aligned} & \frac{1}{\sigma^2}(\boldsymbol{\theta}^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} - \mathbf{y}^\top \boldsymbol{\Phi}) + \frac{1}{b^2} \boldsymbol{\theta}^\top = \mathbf{0}^\top \\ \Leftrightarrow & \boldsymbol{\theta}^\top \left(\frac{1}{\sigma^2} \boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \frac{1}{b^2} \mathbf{I} \right) - \frac{1}{\sigma^2} \mathbf{y}^\top \boldsymbol{\Phi} = \mathbf{0}^\top \\ \Leftrightarrow & \boldsymbol{\theta}^\top \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \frac{\sigma^2}{b^2} \mathbf{I} \right) = \mathbf{y}^\top \boldsymbol{\Phi} \\ \Leftrightarrow & \boldsymbol{\theta}^\top = \mathbf{y}^\top \boldsymbol{\Phi} \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \frac{\sigma^2}{b^2} \mathbf{I} \right)^{-1}. \end{aligned}$$

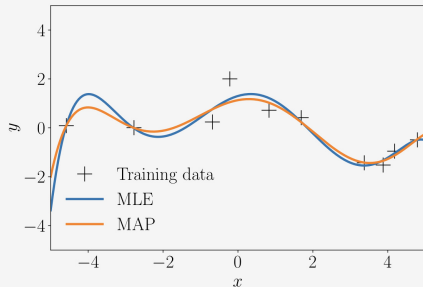
Finally, we have

Maximum a Posteriori Estimation (5/5)

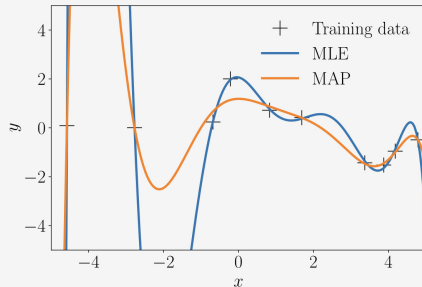
$$\begin{aligned} & \frac{1}{\sigma^2}(\boldsymbol{\theta}^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} - \mathbf{y}^\top \boldsymbol{\Phi}) + \frac{1}{b^2} \boldsymbol{\theta}^\top = \mathbf{0}^\top \\ \Leftrightarrow & \boldsymbol{\theta}^\top \left(\frac{1}{\sigma^2} \boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \frac{1}{b^2} \mathbf{I} \right) - \frac{1}{\sigma^2} \mathbf{y}^\top \boldsymbol{\Phi} = \mathbf{0}^\top \\ \Leftrightarrow & \boldsymbol{\theta}^\top \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \frac{\sigma^2}{b^2} \mathbf{I} \right) = \mathbf{y}^\top \boldsymbol{\Phi} \\ \Leftrightarrow & \boldsymbol{\theta}^\top = \mathbf{y}^\top \boldsymbol{\Phi} \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \frac{\sigma^2}{b^2} \mathbf{I} \right)^{-1}. \end{aligned}$$

Finally, we have

$$\boldsymbol{\theta}_{MAP} = \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \frac{\sigma^2}{b^2} \mathbf{I} \right)^{-1} \boldsymbol{\Phi}^\top \mathbf{y}.$$



(a) Polynomials of degree 6.



(b) Polynomials of degree 8.

Outline

- 1 Introduction
- 2 Problem Formulation
- 3 Parameter Estimation**
 - Maximum Likelihood Estimation (MLE)
 - Overfitting in Linear Regression
 - Maximum A Posteriori Estimation (MAP)
 - **MAP Estimation as Regularization**
- 4 Bayesian Linear Regression

Motivation (I)

- Mitigate the effect of overfitting by **penalizing the amplitude of the parameters by means of regularization**.
- Consider the regularized least squares:

$$\underbrace{\|\mathbf{y} - \Phi\boldsymbol{\theta}\|^2}_{\text{for fitting data}} + \underbrace{\lambda\|\boldsymbol{\theta}\|_2^2}_{\text{regularizer}}$$

for the regularization parameter $\lambda \geq 0$.

- The 2-norm $\|\cdot\|_2$ can be replaced by other types of norm.

Motivation (II)

- The regularizer $\lambda \|\boldsymbol{\theta}\|_2^2$ can be seen as a negative log-Gaussian prior.
- The Gaussian prior $p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, b^2 \mathbf{I})$, so the negative log-Gaussian prior is

$$-\log p(\boldsymbol{\theta}) = \frac{1}{2b^2} \|\boldsymbol{\theta}\|_2^2 + \text{constant}$$

hence we have $\lambda = \frac{1}{2b^2}$.

Minimizing the regularized least-squares loss function yields

$$\theta_{RLS} = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T \mathbf{y}.$$

Minimizing the regularized least-squares loss function yields

$$\theta_{RLS} = (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top \mathbf{y}.$$

This is identical to the MAP estimate for $\lambda = \frac{\sigma^2}{b^2}$.

- σ^2 : the noise variance
- b^2 : the variance of the Gaussian prior $p(\theta) = \mathcal{N}(\mathbf{0}, b^2 I)$.

Outline

- 1 Introduction
- 2 Problem Formulation
- 3 Parameter Estimation
 - Maximum Likelihood Estimation (MLE)
 - Overfitting in Linear Regression
 - Maximum A Posteriori Estimation (MAP)
 - MAP Estimation as Regularization
- 4 Bayesian Linear Regression

From MAP to Bayesian Linear Regression

- So far we have used **point estimates** of the parameters θ :
 - maximum likelihood estimate (MLE) θ_{ML}
 - maximum a posteriori estimate (MAP) θ_{MAP}

From MAP to Bayesian Linear Regression

- So far we have used **point estimates** of the parameters θ :
 - maximum likelihood estimate (MLE) θ_{ML}
 - maximum a posteriori estimate (MAP) θ_{MAP}
- However, both MLE and MAP **ignore remaining uncertainty** about θ .

From MAP to Bayesian Linear Regression

- So far we have used **point estimates** of the parameters θ :
 - maximum likelihood estimate (MLE) θ_{ML}
 - maximum a posteriori estimate (MAP) θ_{MAP}
- However, both MLE and MAP **ignore remaining uncertainty** about θ .
- **Bayesian linear regression:**
 - Treat θ as a random variable.
 - Use Bayes' rule to obtain the **posterior distribution** $p(\theta | \mathcal{X}, \mathcal{Y})$.
 - Make predictions by **integrating over** all plausible parameter values.

From MAP to Bayesian Linear Regression

- So far we have used **point estimates** of the parameters θ :
 - maximum likelihood estimate (MLE) θ_{ML}
 - maximum a posteriori estimate (MAP) θ_{MAP}
- However, both MLE and MAP **ignore remaining uncertainty** about θ .
- **Bayesian linear regression:**
 - Treat θ as a random variable.
 - Use Bayes' rule to obtain the **posterior distribution** $p(\theta \mid \mathcal{X}, \mathcal{Y})$.
 - Make predictions by **integrating over** all plausible parameter values.
- This will give us not only a prediction, but also a **measure of uncertainty**.

Bayesian Linear Regression Model (1/2)

- Recall the feature-based linear regression model

$$y = \phi^\top(\mathbf{x})\boldsymbol{\theta} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2).$$

Bayesian Linear Regression Model (1/2)

- Recall the feature-based linear regression model

$$y = \phi^\top(\mathbf{x})\boldsymbol{\theta} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2).$$

- For a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ we write in matrix form

$$\mathbf{y} = \Phi\boldsymbol{\theta} + \boldsymbol{\epsilon},$$

where

- $\Phi \in \mathbb{R}^{N \times K}$: feature matrix, i -th row $\phi^\top(\mathbf{x}_i)$,
- $\mathbf{y} = [y_1, \dots, y_N]^\top \in \mathbb{R}^N$.

Bayesian Linear Regression Model (1/2)

- Recall the feature-based linear regression model

$$y = \phi^\top(\mathbf{x})\boldsymbol{\theta} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2).$$

- For a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ we write in matrix form

$$\mathbf{y} = \Phi\boldsymbol{\theta} + \boldsymbol{\epsilon},$$

where

- $\Phi \in \mathbb{R}^{N \times K}$: feature matrix, i -th row $\phi^\top(\mathbf{x}_i)$,
 - $\mathbf{y} = [y_1, \dots, y_N]^\top \in \mathbb{R}^N$.
- The **likelihood** is Gaussian:

$$p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y} \mid \Phi\boldsymbol{\theta}, \sigma^2 \mathbf{I}_N).$$

Bayesian Linear Regression Model (2/2)

- We place a **Gaussian prior** on the parameter vector θ :

$$p(\theta) = \mathcal{N}(\theta \mid \mathbf{m}_0, \mathbf{S}_0),$$

where

- $\mathbf{m}_0 \in \mathbb{R}^K$: prior mean,
- $\mathbf{S}_0 \in \mathbb{R}^{K \times K}$: prior covariance matrix.

Bayesian Linear Regression Model (2/2)

- We place a **Gaussian prior** on the parameter vector θ :

$$p(\theta) = \mathcal{N}(\theta \mid \mathbf{m}_0, \mathbf{S}_0),$$

where

- $\mathbf{m}_0 \in \mathbb{R}^K$: prior mean,
 - $\mathbf{S}_0 \in \mathbb{R}^{K \times K}$: prior covariance matrix.
- This prior encodes our **a priori beliefs** about likely parameter values.
 - Large variances in $\mathbf{S}_0 \Rightarrow$ weak prior, parameters can vary a lot.
 - Small variances in $\mathbf{S}_0 \Rightarrow$ strong prior, parameters are strongly regularized (i.e., very concentrated around \mathbf{m}_0).

Bayesian Linear Regression Model (2/2)

- We place a **Gaussian prior** on the parameter vector θ :

$$p(\theta) = \mathcal{N}(\theta \mid \mathbf{m}_0, \mathbf{S}_0),$$

where

- $\mathbf{m}_0 \in \mathbb{R}^K$: prior mean,
 - $\mathbf{S}_0 \in \mathbb{R}^{K \times K}$: prior covariance matrix.
- This prior encodes our **a priori beliefs** about likely parameter values.
 - Large variances in $\mathbf{S}_0 \Rightarrow$ weak prior, parameters can vary a lot.
 - Small variances in $\mathbf{S}_0 \Rightarrow$ strong prior, parameters are strongly regularized (i.e., very concentrated around \mathbf{m}_0).
 - Because both prior and likelihood are Gaussian, the posterior will also be Gaussian (Exercise).

Remark: Product of Gaussian Densities

- Consider two D -dimensional Gaussians in **the same variable \mathbf{x}** :

$$\mathcal{N}(\mathbf{x} \mid \mathbf{a}, \mathbf{A}) \text{ and } \mathcal{N}(\mathbf{x} \mid \mathbf{b}, \mathbf{B}).$$

- Their product is proportional to another Gaussian:

$$\mathcal{N}(\mathbf{x} \mid \mathbf{a}, \mathbf{A}) \mathcal{N}(\mathbf{x} \mid \mathbf{b}, \mathbf{B}) = \rho \mathcal{N}(\mathbf{x} \mid \mathbf{c}, \mathbf{C}),$$

where

$$\mathbf{C} = (\mathbf{A}^{-1} + \mathbf{B}^{-1})^{-1}, \quad \mathbf{c} = \mathbf{C}(\mathbf{A}^{-1}\mathbf{a} + \mathbf{B}^{-1}\mathbf{b}),$$

and the scaling constant ρ is

$$\rho = (2\pi)^{-D/2} \det(\mathbf{A} + \mathbf{B})^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{a} - \mathbf{b})^\top (\mathbf{A} + \mathbf{B})^{-1}(\mathbf{a} - \mathbf{b})\right).$$

- The constant ρ can itself be written as a Gaussian density with “inflated” covariance $\mathbf{A} + \mathbf{B}$:

$$\rho = \mathcal{N}(\mathbf{a} \mid \mathbf{b}, \mathbf{A} + \mathbf{B}) = \mathcal{N}(\mathbf{b} \mid \mathbf{a}, \mathbf{A} + \mathbf{B}).$$

Posterior over Parameters (1/2)

- By Bayes' rule,

$$p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) = \frac{p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{Y} \mid \mathcal{X})}.$$

Posterior over Parameters (1/2)

- By Bayes' rule,

$$p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) = \frac{p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{Y} \mid \mathcal{X})}.$$

- Ignoring the normalizing constant, the posterior is proportional to

$$p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) \propto \underbrace{\exp\left(-\frac{1}{2\sigma^2} \|\mathbf{y} - \Phi\boldsymbol{\theta}\|^2\right)}_{\text{likelihood}} \underbrace{\exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \mathbf{m}_0)^\top \mathbf{S}_0^{-1}(\boldsymbol{\theta} - \mathbf{m}_0)\right)}_{\text{prior}}.$$

Posterior over Parameters (1/2)

- By Bayes' rule,

$$p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) = \frac{p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{Y} \mid \mathcal{X})}.$$

- Ignoring the normalizing constant, the posterior is proportional to

$$p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) \propto \underbrace{\exp\left(-\frac{1}{2\sigma^2} \|\mathbf{y} - \Phi\boldsymbol{\theta}\|^2\right)}_{\text{likelihood}} \underbrace{\exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \mathbf{m}_0)^\top \mathbf{S}_0^{-1}(\boldsymbol{\theta} - \mathbf{m}_0)\right)}_{\text{prior}}.$$

- The exponent is a **quadratic function** of $\boldsymbol{\theta} \Rightarrow$ the posterior is Gaussian.

Posterior over Parameters (2/2)

Posterior of θ (proof skipped)

The posterior is

$$p(\theta \mid \mathcal{X}, \mathcal{Y}) = \mathcal{N}(\theta \mid \mathbf{m}_N, \mathbf{S}_N),$$

where the posterior covariance and mean are given by

$$\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \frac{1}{\sigma^2} \Phi^\top \Phi,$$

$$\mathbf{m}_N = \mathbf{S}_N \left(\mathbf{S}_0^{-1} \mathbf{m}_0 + \frac{1}{\sigma^2} \Phi^\top \mathbf{y} \right).$$

Posterior over Parameters (2/2)

Posterior of θ (proof skipped)

The posterior is

$$p(\theta \mid \mathcal{X}, \mathcal{Y}) = \mathcal{N}(\theta \mid \mathbf{m}_N, \mathbf{S}_N),$$

where the posterior covariance and mean are given by

$$\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \frac{1}{\sigma^2} \Phi^\top \Phi,$$

$$\mathbf{m}_N = \mathbf{S}_N \left(\mathbf{S}_0^{-1} \mathbf{m}_0 + \frac{1}{\sigma^2} \Phi^\top \mathbf{y} \right).$$

- \mathbf{S}_N : the prior uncertainty \mathbf{S}_0 + information from the data.
- \mathbf{m}_N : a **precision-weighted average** of prior mean and data evidence.

Interpretation of the Posterior

- The matrix \mathbf{S}_0^{-1} is the **prior precision** and $\frac{1}{\sigma^2} \Phi^\top \Phi$ is the **data precision**.

Interpretation of the Posterior

- The matrix \mathbf{S}_0^{-1} is the **prior precision** and $\frac{1}{\sigma^2} \Phi^\top \Phi$ is the **data precision**.
- More data (larger N) $\Rightarrow \Phi^\top \Phi$ dominates \Rightarrow posterior is driven mainly by the data.

Interpretation of the Posterior

- The matrix \mathbf{S}_0^{-1} is the **prior precision** and $\frac{1}{\sigma^2} \Phi^\top \Phi$ is the **data precision**.
- More data (larger N) $\Rightarrow \Phi^\top \Phi$ dominates \Rightarrow posterior is driven mainly by the data.
- Fewer data or very noisy data (large σ^2) \Rightarrow posterior is closer to the prior.

Interpretation of the Posterior

- The matrix \mathbf{S}_0^{-1} is the **prior precision** and $\frac{1}{\sigma^2} \Phi^\top \Phi$ is the **data precision**.
- More data (larger N) $\Rightarrow \Phi^\top \Phi$ dominates \Rightarrow posterior is driven mainly by the data.
- Fewer data or very noisy data (large σ^2) \Rightarrow posterior is closer to the prior.
- For a Gaussian posterior, the **MAP estimate** and the **posterior mean** coincide, but
 - MAP uses only the mode θ_{MAP} ,
 - Bayesian prediction uses the *full posterior* $p(\theta \mid \mathcal{X}, \mathcal{Y})$.

Prior Predictive Distribution

- Consider a new input \mathbf{x}_* with feature vector $\phi(\mathbf{x}_*)$.

Prior Predictive Distribution

- Consider a new input \mathbf{x}_* with feature vector $\phi(\mathbf{x}_*)$.
- Before observing any data, predictions are based on the prior:

$$p(y_* | \mathbf{x}_*) = \int p(y_* | \mathbf{x}_*, \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}.$$

Prior Predictive Distribution

- Consider a new input \mathbf{x}_* with feature vector $\phi(\mathbf{x}_*)$.
- Before observing any data, predictions are based on the prior:

$$p(y_* | \mathbf{x}_*) = \int p(y_* | \mathbf{x}_*, \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}.$$

- Since both terms are Gaussian, the **prior predictive** is Gaussian:

$$p(y_* | \mathbf{x}_*) = \mathcal{N}\left(y_* | \phi^\top(\mathbf{x}_*) \mathbf{m}_0, \phi^\top(\mathbf{x}_*) \mathbf{S}_0 \phi(\mathbf{x}_*) + \sigma^2\right).$$

Prior Predictive Distribution

- Consider a new input \mathbf{x}_* with feature vector $\phi(\mathbf{x}_*)$.
- Before observing any data, predictions are based on the prior:

$$p(y_* | \mathbf{x}_*) = \int p(y_* | \mathbf{x}_*, \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}.$$

- Since both terms are Gaussian, the **prior predictive** is Gaussian:

$$p(y_* | \mathbf{x}_*) = \mathcal{N}\left(y_* | \phi^\top(\mathbf{x}_*) \mathbf{m}_0, \phi^\top(\mathbf{x}_*) \mathbf{S}_0 \phi(\mathbf{x}_*) + \sigma^2\right).$$

- It reflects what we expect *before* seeing any training data.

Posterior Predictive Distribution (1/2)

- After observing the dataset \mathcal{D} , we use the posterior to make predictions:

$$p(y_* | \mathbf{x}_*, \mathcal{X}, \mathcal{Y}) = \int p(y_* | \mathbf{x}_*, \boldsymbol{\theta}) p(\boldsymbol{\theta} | \mathcal{X}, \mathcal{Y}) d\boldsymbol{\theta}.$$

Posterior Predictive Distribution (1/2)

- After observing the dataset \mathcal{D} , we use the posterior to make predictions:

$$p(y_* | \mathbf{x}_*, \mathcal{X}, \mathcal{Y}) = \int p(y_* | \mathbf{x}_*, \boldsymbol{\theta}) p(\boldsymbol{\theta} | \mathcal{X}, \mathcal{Y}) d\boldsymbol{\theta}.$$

- Again, both distributions inside the integral are Gaussian \Rightarrow the **posterior predictive** is Gaussian.

Posterior Predictive Distribution (1/2)

- After observing the dataset \mathcal{D} , we use the posterior to make predictions:

$$p(y_* | \mathbf{x}_*, \mathcal{X}, \mathcal{Y}) = \int p(y_* | \mathbf{x}_*, \boldsymbol{\theta}) p(\boldsymbol{\theta} | \mathcal{X}, \mathcal{Y}) d\boldsymbol{\theta}.$$

- Again, both distributions inside the integral are Gaussian \Rightarrow the **posterior predictive** is Gaussian.
- Intuitively:
 - we average predictions over all plausible parameter values,
 - weighted by how probable they are under the posterior.

Posterior Predictive Distribution (2/2)

Posterior predictive of y_*

The predictive distribution at a new input \mathbf{x}_* is

$$p(y_* \mid \mathbf{x}_*, \mathcal{X}, \mathcal{Y}) = \mathcal{N}(y_* \mid \mu_*(\mathbf{x}_*), \sigma_*^2(\mathbf{x}_*)) ,$$

with mean

$$\mu_*(\mathbf{x}_*) = \phi^\top(\mathbf{x}_*) \mathbf{m}_N,$$

and variance

$$\sigma_*^2(\mathbf{x}_*) = \phi^\top(\mathbf{x}_*) \mathbf{S}_N \phi(\mathbf{x}_*) + \sigma^2.$$

Posterior Predictive Distribution (2/2)

Posterior predictive of y_*

The predictive distribution at a new input \mathbf{x}_* is

$$p(y_* | \mathbf{x}_*, \mathcal{X}, \mathcal{Y}) = \mathcal{N}(y_* | \mu_*(\mathbf{x}_*), \sigma_*^2(\mathbf{x}_*)) ,$$

with mean

$$\mu_*(\mathbf{x}_*) = \phi^\top(\mathbf{x}_*) \mathbf{m}_N,$$

and variance

$$\sigma_*^2(\mathbf{x}_*) = \phi^\top(\mathbf{x}_*) \mathbf{S}_N \phi(\mathbf{x}_*) + \sigma^2.$$

- The first term in σ_*^2 quantifies **parameter uncertainty**.
- The second term σ^2 is the **measurement noise**.

Predictive Uncertainty

- Near many training inputs, the features $\phi(\mathbf{x}_*)$ are well-supported by data \Rightarrow parameter uncertainty is small.

Predictive Uncertainty

- Near many training inputs, the features $\phi(\mathbf{x}_*)$ are well-supported by data \Rightarrow parameter uncertainty is small.
- Far away from the training inputs, predictions are more uncertain:

$\phi^\top(\mathbf{x}_*) \mathbf{S}_N \phi(\mathbf{x}_*)$ becomes large.

Predictive Uncertainty

- Near many training inputs, the features $\phi(\mathbf{x}_*)$ are well-supported by data \Rightarrow parameter uncertainty is small.
- Far away from the training inputs, predictions are more uncertain:

$\phi^\top(\mathbf{x}_*) \mathbf{S}_N \phi(\mathbf{x}_*)$ becomes large.

- Bayesian linear regression therefore yields
 - a **mean prediction** $\mu_*(\mathbf{x}_*)$ and
 - a **credible interval** (e.g., mean $\pm 2\sigma_*(\mathbf{x}_*)$).

Predictive Uncertainty

- Near many training inputs, the features $\phi(\mathbf{x}_*)$ are well-supported by data \Rightarrow parameter uncertainty is small.
- Far away from the training inputs, predictions are more uncertain:

$\phi^\top(\mathbf{x}_*)\mathbf{S}_N\phi(\mathbf{x}_*)$ becomes large.

- Bayesian linear regression therefore yields
 - a **mean prediction** $\mu_*(\mathbf{x}_*)$ and
 - a **credible interval** (e.g., mean $\pm 2\sigma_*(\mathbf{x}_*)$).
- This is very useful for model assessment and decision making.

Example: Polynomial Regression Revisited

- Recall the polynomial feature maps

$$\phi(x) = [1, x, x^2, \dots, x^{K-1}]^\top.$$

Example: Polynomial Regression Revisited

- Recall the polynomial feature maps

$$\phi(x) = [1, x, x^2, \dots, x^{K-1}]^\top.$$

- For high-degree polynomials, MLE severely overfits, while MAP regularization improves the fit but still returns a single curve.

Example: Polynomial Regression Revisited

- Recall the polynomial feature maps

$$\phi(x) = [1, x, x^2, \dots, x^{K-1}]^\top.$$

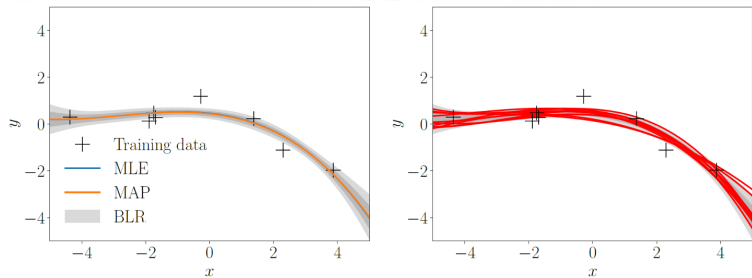
- For high-degree polynomials, MLE severely overfits, while MAP regularization improves the fit but still returns a single curve.
- In contrast, Bayesian linear regression
 - produces a **distribution** over curves,
 - with narrow uncertainty bands where data are dense,
 - and wide bands where there are few or no observations.

Example: Polynomial Regression Revisited

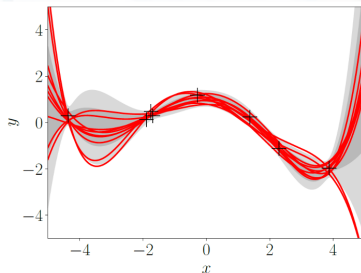
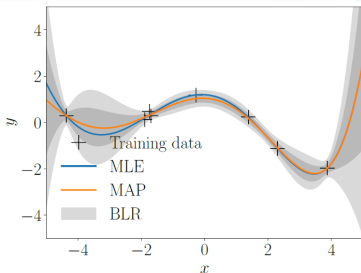
- Recall the polynomial feature maps

$$\phi(x) = [1, x, x^2, \dots, x^{K-1}]^\top.$$

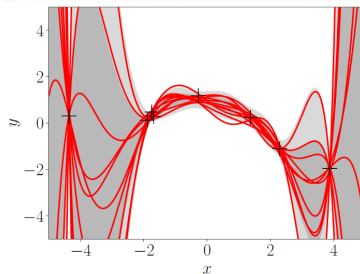
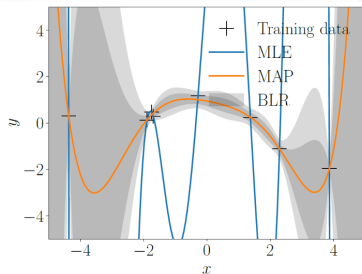
- For high-degree polynomials, MLE severely overfits, while MAP regularization improves the fit but still returns a single curve.
- In contrast, Bayesian linear regression
 - produces a **distribution** over curves,
 - with narrow uncertainty bands where data are dense,
 - and wide bands where there are few or no observations.
- This behaviour matches the qualitative picture in the textbook for Bayesian linear regression.



(a) Posterior distribution for polynomials of degree $M = 3$ (left) and samples from the posterior over functions (right).



(b) Posterior distribution for polynomials of degree $M = 5$ (left) and samples from the posterior over functions (right).



(c) Posterior distribution for polynomials of degree $M = 7$ (left) and samples from the posterior over functions (right).

Marginal Likelihood (1/2)

- The **marginal likelihood** (or **model evidence**) is

$$p(\mathcal{Y} | \mathcal{X}) = \int p(\mathcal{Y} | \mathcal{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}.$$

Marginal Likelihood (1/2)

- The **marginal likelihood** (or **model evidence**) is

$$p(\mathcal{Y} \mid \mathcal{X}) = \int p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}.$$

- It measures how well the model (including the prior) explains the data.

Marginal Likelihood (1/2)

- The **marginal likelihood** (or **model evidence**) is

$$p(\mathcal{Y} | \mathcal{X}) = \int p(\mathcal{Y} | \mathcal{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}.$$

- It measures how well the model (including the prior) explains the data.
- In Bayesian linear regression, the integral can be computed in closed form.

Marginal Likelihood (2/2)

Closed-form marginal likelihood

Using Gaussian identities, we obtain

$$p(\mathcal{Y} \mid \mathcal{X}) = \mathcal{N}(\mathbf{y} \mid \Phi \mathbf{m}_0, \Phi \mathbf{S}_0 \Phi^\top + \sigma^2 \mathbf{I}_N).$$

Marginal Likelihood (2/2)

Closed-form marginal likelihood

Using Gaussian identities, we obtain

$$p(\mathcal{Y} \mid \mathcal{X}) = \mathcal{N}\left(\mathbf{y} \mid \Phi \mathbf{m}_0, \Phi \mathbf{S}_0 \Phi^\top + \sigma^2 \mathbf{I}_N\right).$$

- Different model choices (e.g., different polynomial degrees) lead to different marginal likelihoods.

Marginal Likelihood (2/2)

Closed-form marginal likelihood

Using Gaussian identities, we obtain

$$p(\mathcal{Y} \mid \mathcal{X}) = \mathcal{N}\left(\mathbf{y} \mid \Phi \mathbf{m}_0, \Phi \mathbf{S}_0 \Phi^\top + \sigma^2 \mathbf{I}_N\right).$$

- Different model choices (e.g., different polynomial degrees) lead to different marginal likelihoods.
- The marginal likelihood automatically trades off **data fit** (i.e., likelihood) and **model complexity** (i.e., prior).

Summary: Bayesian Linear Regression

- We treat the parameters θ as random and specify a Gaussian prior $p(\theta)$.

Summary: Bayesian Linear Regression

- We treat the parameters θ as random and specify a Gaussian prior $p(\theta)$.
- Together with the Gaussian likelihood, this yields a Gaussian posterior $p(\theta \mid \mathcal{X}, \mathcal{Y})$.

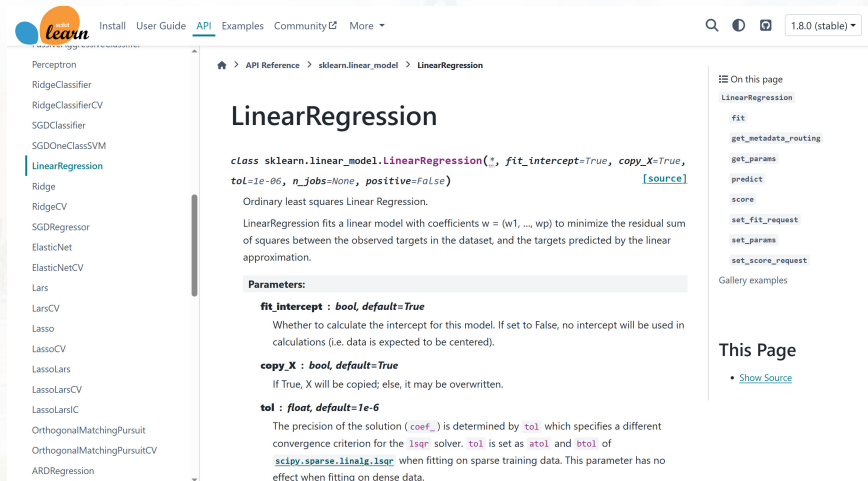
Summary: Bayesian Linear Regression

- We treat the parameters θ as random and specify a Gaussian prior $p(\theta)$.
- Together with the Gaussian likelihood, this yields a Gaussian posterior $p(\theta \mid \mathcal{X}, \mathcal{Y})$.
- Predictions are made by **integrating over the posterior**, resulting in a Gaussian predictive distribution with
 - mean $\mu_*(\mathbf{x}_*) = \phi^\top(\mathbf{x}_*)\mathbf{m}_N$,
 - variance $\sigma_*^2(\mathbf{x}_*) = \phi^\top(\mathbf{x}_*)\mathbf{S}_N\phi(\mathbf{x}_*) + \sigma^2$.

Summary: Bayesian Linear Regression

- We treat the parameters θ as random and specify a Gaussian prior $p(\theta)$.
- Together with the Gaussian likelihood, this yields a Gaussian posterior $p(\theta \mid \mathcal{X}, \mathcal{Y})$.
- Predictions are made by **integrating over the posterior**, resulting in a Gaussian predictive distribution with
 - mean $\mu_*(\mathbf{x}_*) = \phi^\top(\mathbf{x}_*)\mathbf{m}_N$,
 - variance $\sigma_*^2(\mathbf{x}_*) = \phi^\top(\mathbf{x}_*)\mathbf{S}_N\phi(\mathbf{x}_*) + \sigma^2$.
- Bayesian linear regression improves over MLE and MAP by
 - quantifying parameter and predictive uncertainty,
 - enabling principled model comparison via the marginal likelihood.

LinearRegression — scikit-learn [\[link\]](#)



The screenshot shows the scikit-learn documentation page for the `LinearRegression` class. The page layout includes a top navigation bar with links for 'Install', 'User Guide', 'API' (which is highlighted), 'Examples', and 'Community'. A search bar and version selector (1.8.0 (stable)) are on the right. A left sidebar lists various models, with 'LinearRegression' selected. The main content area features the class name 'LinearRegression' in large text, followed by its signature: `class sklearn.linear_model.LinearRegression(*, fit_intercept=True, copy_X=True, tol=1e-06, n_jobs=None, positive=False)` with a '[source]' link. Below this, a paragraph describes it as 'Ordinary least squares Linear Regression.' and explains that it fits a linear model by minimizing the residual sum of squares. A 'Parameters:' section follows, detailing the `fit_intercept`, `copy_X`, and `tol` parameters. On the right, a 'On this page' sidebar lists methods like `fit`, `get_metadata_routing`, `get_params`, `predict`, `score`, `set_fit_request`, `set_params`, and `set_score_request`, along with a 'Gallery examples' link. At the bottom right, a 'This Page' section contains a 'Show Source' link.

scikit-learn

Install User Guide **API** Examples Community More

1.8.0 (stable)

API Reference > sklearn.linear_model > LinearRegression

LinearRegression

```
class sklearn.linear_model.LinearRegression(*, fit_intercept=True, copy_X=True,
tol=1e-06, n_jobs=None, positive=False) \[source\]
```

Ordinary least squares Linear Regression.

LinearRegression fits a linear model with coefficients $w = (w_1, \dots, w_p)$ to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

Parameters:

fit_intercept : *bool, default=True*

Whether to calculate the intercept for this model. If set to False, no intercept will be used in calculations (i.e. data is expected to be centered).

copy_X : *bool, default=True*

If True, X will be copied; else, it may be overwritten.

tol : *float, default=1e-6*

The precision of the solution (`coef_`) is determined by `tol` which specifies a different convergence criterion for the `lsqr` solver. `tol` is set as `atol` and `btol` of `scipy.sparse.linalg.lsqr` when fitting on sparse training data. This parameter has no effect when fitting on dense data.

On this page

- LinearRegression
- `fit`
- `get_metadata_routing`
- `get_params`
- `predict`
- `score`
- `set_fit_request`
- `set_params`
- `set_score_request`

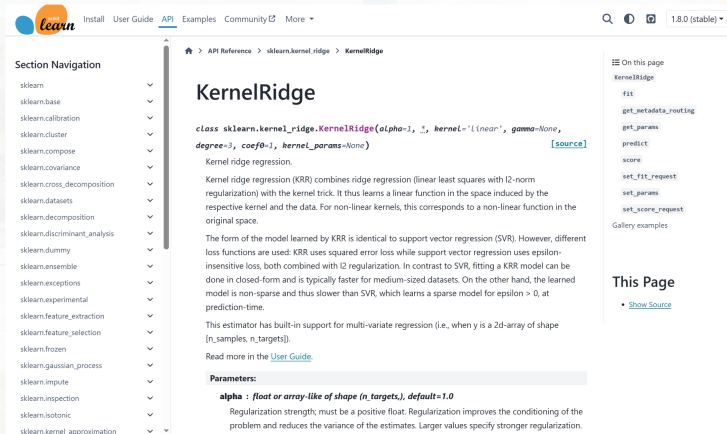
[Gallery examples](#)

This Page

- [Show Source](#)

Kernel Ridge Regression (KRR) — scikit-learn [\[link\]](#)

- In scikit-learn, “linear regression using a kernel” is implemented through Kernel Ridge Regression (KRR).



The screenshot shows the scikit-learn documentation page for `KernelRidge`. The page is titled "KernelRidge" and is part of the "API Reference" section. It includes a sidebar with "Section Navigation" listing various scikit-learn modules. The main content area describes the `KernelRidge` class, its parameters, and its usage. The parameters section lists `alpha` as a float or array-like of shape `(n_targets,)` with a default of `1.0`. The text explains that regularization strength must be a positive float and that larger values specify stronger regularization.

Section Navigation

- sklearn
- sklearn.base
- sklearn.calibration
- sklearn.cluster
- sklearn.compose
- sklearn.covariance
- sklearn.cross_decomposition
- sklearn.datasets
- sklearn.decomposition
- sklearn.discriminant_analysis
- sklearn.dummy
- sklearn.ensemble
- sklearn.exceptions
- sklearn.experimental
- sklearn.feature_extraction
- sklearn.feature_selection
- sklearn.frozen
- sklearn.gaussian_process
- sklearn.impute
- sklearn.inspection
- sklearn.isotonic
- sklearn.kernel_approximation

API Reference > sklearn.kernel.ridge > KernelRidge

KernelRidge

`class sklearn.kernel_ridge.KernelRidge(alpha=1.0, *, kernel='linear', gamma=None, degree=3, coef0=1, kernel_params=None)` [\[source\]](#)

Kernel ridge regression.

Kernel ridge regression (KRR) combines ridge regression (linear least squares with l2-norm regularization) with the kernel trick. It thus learns a linear function in the space induced by the respective kernel and the data. For non-linear kernels, this corresponds to a non-linear function in the original space.

The form of the model learned by KRR is identical to support vector regression (SVR). However, different loss functions are used: KRR uses squared error loss while support vector regression uses epsilon-insensitive loss, both combined with l2 regularization. In contrast to SVR, fitting a KRR model can be done in closed-form and is typically faster for medium-sized datasets. On the other hand, the learned model is non-sparse and thus slower than SVR, which learns a sparse model for epsilon > 0, at prediction-time.

This estimator has built-in support for multi-variate regression (i.e., when y is a 2d-array of shape `[n_samples, n_targets]`).

Read more in the [User Guide](#).

Parameters:

alpha : float or array-like of shape `(n_targets,)`, default=1.0

Regularization strength; must be a positive float. Regularization improves the conditioning of the problem and reduces the variance of the estimates. Larger values specify stronger regularization.

On this page

- [KernelRidge](#)
- [fit](#)
- [get_metadata_routing](#)
- [get_params](#)
- [predict](#)
- [score](#)
- [set_fit_request](#)
- [set_params](#)
- [set_score_request](#)
- [Gallery examples](#)

This Page

- [Show Source](#)

Discussions

Theorem 9.1 (Parameter Posterior)

Consider the linear regression model with Gaussian noise

$$\mathbf{y} \mid \boldsymbol{\theta}, \mathcal{X} \sim \mathcal{N}(\boldsymbol{\Phi}\boldsymbol{\theta}, \sigma^2 \mathbf{I}),$$

and Gaussian prior on the parameters

$$\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{m}_0, \mathbf{S}_0),$$

where $\boldsymbol{\Phi}$ is the design matrix, \mathbf{m}_0 the prior mean and \mathbf{S}_0 the prior covariance.

Theorem 9.1 (Parameter Posterior)

Consider the linear regression model with Gaussian noise

$$\mathbf{y} \mid \boldsymbol{\theta}, \mathcal{X} \sim \mathcal{N}(\boldsymbol{\Phi}\boldsymbol{\theta}, \sigma^2 \mathbf{I}),$$

and Gaussian prior on the parameters

$$\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{m}_0, \mathbf{S}_0),$$

where $\boldsymbol{\Phi}$ is the design matrix, \mathbf{m}_0 the prior mean and \mathbf{S}_0 the prior covariance.

Theorem 9.1 (Parameter Posterior)

The posterior over parameters is Gaussian:

$$p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) = \mathcal{N}(\boldsymbol{\theta} \mid \mathbf{m}_N, \mathbf{S}_N),$$

$$\text{with } \mathbf{S}_N = (\mathbf{S}_0^{-1} + \sigma^{-2} \boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1}, \quad \mathbf{m}_N = \mathbf{S}_N (\mathbf{S}_0^{-1} \mathbf{m}_0 + \sigma^{-2} \boldsymbol{\Phi}^\top \mathbf{y}),$$

where the subscript N indicates dependence on the full training set.

Proof of Theorem 9.1 (1/3)

- By Bayes' rule, up to a normalizing constant,

$$p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) \propto p(\mathbf{y} \mid \mathcal{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta}).$$

Proof of Theorem 9.1 (1/3)

- By Bayes' rule, up to a normalizing constant,

$$p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) \propto p(\mathbf{y} \mid \mathcal{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta}).$$

- Likelihood:

$$p(\mathbf{y} \mid \mathcal{X}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y} \mid \boldsymbol{\Phi}\boldsymbol{\theta}, \sigma^2 \mathbf{I}).$$

- Prior:

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \mid \mathbf{m}_0, \mathbf{S}_0).$$

Proof of Theorem 9.1 (1/3)

- By Bayes' rule, up to a normalizing constant,

$$p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) \propto p(\mathbf{y} \mid \mathcal{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta}).$$

- Likelihood:

$$p(\mathbf{y} \mid \mathcal{X}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y} \mid \boldsymbol{\Phi}\boldsymbol{\theta}, \sigma^2 \mathbf{I}).$$

- Prior:

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \mid \mathbf{m}_0, \mathbf{S}_0).$$

- Work in log-space (and drop constants independent of $\boldsymbol{\theta}$):

$$\log p(\mathbf{y} \mid \mathcal{X}, \boldsymbol{\theta}) = -\frac{1}{2\sigma^2} (\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\theta})^\top (\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\theta}) + \text{const},$$

$$\log p(\boldsymbol{\theta}) = -\frac{1}{2} (\boldsymbol{\theta} - \mathbf{m}_0)^\top \mathbf{S}_0^{-1} (\boldsymbol{\theta} - \mathbf{m}_0) + \text{const}.$$

Proof of Theorem 9.1 (2/3) – completing the squares

- Summing the two log terms (still ignoring constants):

$$\begin{aligned}\log p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) &= -\frac{1}{2\sigma^2}(\mathbf{y} - \Phi\boldsymbol{\theta})^\top(\mathbf{y} - \Phi\boldsymbol{\theta}) \\ &\quad - \frac{1}{2}(\boldsymbol{\theta} - \mathbf{m}_0)^\top \mathbf{S}_0^{-1}(\boldsymbol{\theta} - \mathbf{m}_0) + \text{const.}\end{aligned}$$

Proof of Theorem 9.1 (2/3) – completing the squares

- Summing the two log terms (still ignoring constants):

$$\begin{aligned}\log p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) &= -\frac{1}{2\sigma^2}(\mathbf{y} - \Phi\boldsymbol{\theta})^\top(\mathbf{y} - \Phi\boldsymbol{\theta}) \\ &\quad - \frac{1}{2}(\boldsymbol{\theta} - \mathbf{m}_0)^\top \mathbf{S}_0^{-1}(\boldsymbol{\theta} - \mathbf{m}_0) + \text{const.}\end{aligned}$$

- Expand the quadratic terms in $\boldsymbol{\theta}$:

$$\begin{aligned}(\mathbf{y} - \Phi\boldsymbol{\theta})^\top(\mathbf{y} - \Phi\boldsymbol{\theta}) &= \mathbf{y}^\top \mathbf{y} - 2\mathbf{y}^\top \Phi\boldsymbol{\theta} + \boldsymbol{\theta}^\top \Phi^\top \Phi\boldsymbol{\theta}, \\ (\boldsymbol{\theta} - \mathbf{m}_0)^\top \mathbf{S}_0^{-1}(\boldsymbol{\theta} - \mathbf{m}_0) &= \boldsymbol{\theta}^\top \mathbf{S}_0^{-1}\boldsymbol{\theta} - 2\mathbf{m}_0^\top \mathbf{S}_0^{-1}\boldsymbol{\theta} + \mathbf{m}_0^\top \mathbf{S}_0^{-1}\mathbf{m}_0.\end{aligned}$$

Proof of Theorem 9.1 (2/3) – completing the squares

- Summing the two log terms (still ignoring constants):

$$\begin{aligned}\log p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) &= -\frac{1}{2\sigma^2}(\mathbf{y} - \Phi\boldsymbol{\theta})^\top(\mathbf{y} - \Phi\boldsymbol{\theta}) \\ &\quad - \frac{1}{2}(\boldsymbol{\theta} - \mathbf{m}_0)^\top \mathbf{S}_0^{-1}(\boldsymbol{\theta} - \mathbf{m}_0) + \text{const.}\end{aligned}$$

- Expand the quadratic terms in $\boldsymbol{\theta}$:

$$\begin{aligned}(\mathbf{y} - \Phi\boldsymbol{\theta})^\top(\mathbf{y} - \Phi\boldsymbol{\theta}) &= \mathbf{y}^\top \mathbf{y} - 2\mathbf{y}^\top \Phi\boldsymbol{\theta} + \boldsymbol{\theta}^\top \Phi^\top \Phi\boldsymbol{\theta}, \\ (\boldsymbol{\theta} - \mathbf{m}_0)^\top \mathbf{S}_0^{-1}(\boldsymbol{\theta} - \mathbf{m}_0) &= \boldsymbol{\theta}^\top \mathbf{S}_0^{-1}\boldsymbol{\theta} - 2\mathbf{m}_0^\top \mathbf{S}_0^{-1}\boldsymbol{\theta} + \mathbf{m}_0^\top \mathbf{S}_0^{-1}\mathbf{m}_0.\end{aligned}$$

- Collect all terms that depend on $\boldsymbol{\theta}$:

$$\begin{aligned}\log p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) &= -\frac{1}{2} \left[\boldsymbol{\theta}^\top (\sigma^{-2}\Phi^\top \Phi + \mathbf{S}_0^{-1})\boldsymbol{\theta} \right. \\ &\quad \left. - 2(\sigma^{-2}\Phi^\top \mathbf{y} + \mathbf{S}_0^{-1}\mathbf{m}_0)^\top \boldsymbol{\theta} \right] + \text{const.}\end{aligned}$$

Proof of Theorem 9.1 (3/3)

- A multivariate Gaussian $\mathcal{N}(\boldsymbol{\theta} \mid \mathbf{m}_N, \mathbf{S}_N)$ has log-density (up to an additive constant)

$$\begin{aligned}\log \mathcal{N}(\boldsymbol{\theta} \mid \mathbf{m}_N, \mathbf{S}_N) &= -\frac{1}{2}(\boldsymbol{\theta} - \mathbf{m}_N)^\top \mathbf{S}_N^{-1}(\boldsymbol{\theta} - \mathbf{m}_N) + \text{const} \\ &= -\frac{1}{2} \left[\boldsymbol{\theta}^\top \mathbf{S}_N^{-1} \boldsymbol{\theta} - 2\mathbf{m}_N^\top \mathbf{S}_N^{-1} \boldsymbol{\theta} \right] + \text{const.}\end{aligned}$$

Proof of Theorem 9.1 (3/3)

- A multivariate Gaussian $\mathcal{N}(\boldsymbol{\theta} \mid \mathbf{m}_N, \mathbf{S}_N)$ has log-density (up to an additive constant)

$$\begin{aligned}\log \mathcal{N}(\boldsymbol{\theta} \mid \mathbf{m}_N, \mathbf{S}_N) &= -\frac{1}{2}(\boldsymbol{\theta} - \mathbf{m}_N)^\top \mathbf{S}_N^{-1}(\boldsymbol{\theta} - \mathbf{m}_N) + \text{const} \\ &= -\frac{1}{2} \left[\boldsymbol{\theta}^\top \mathbf{S}_N^{-1} \boldsymbol{\theta} - 2\mathbf{m}_N^\top \mathbf{S}_N^{-1} \boldsymbol{\theta} \right] + \text{const.}\end{aligned}$$

- Compare this with the quadratic form obtained on the previous slide:

$$\boldsymbol{\theta}^\top (\sigma^{-2} \boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \mathbf{S}_0^{-1}) \boldsymbol{\theta} \quad \text{and} \quad (\sigma^{-2} \boldsymbol{\Phi}^\top \mathbf{y} + \mathbf{S}_0^{-1} \mathbf{m}_0)^\top \boldsymbol{\theta}.$$

Proof of Theorem 9.1 (3/3)

- A multivariate Gaussian $\mathcal{N}(\boldsymbol{\theta} \mid \mathbf{m}_N, \mathbf{S}_N)$ has log-density (up to an additive constant)

$$\begin{aligned}\log \mathcal{N}(\boldsymbol{\theta} \mid \mathbf{m}_N, \mathbf{S}_N) &= -\frac{1}{2}(\boldsymbol{\theta} - \mathbf{m}_N)^\top \mathbf{S}_N^{-1}(\boldsymbol{\theta} - \mathbf{m}_N) + \text{const} \\ &= -\frac{1}{2} \left[\boldsymbol{\theta}^\top \mathbf{S}_N^{-1} \boldsymbol{\theta} - 2\mathbf{m}_N^\top \mathbf{S}_N^{-1} \boldsymbol{\theta} \right] + \text{const}.\end{aligned}$$

- Compare this with the quadratic form obtained on the previous slide:

$$\boldsymbol{\theta}^\top (\sigma^{-2} \boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \mathbf{S}_0^{-1}) \boldsymbol{\theta} \quad \text{and} \quad (\sigma^{-2} \boldsymbol{\Phi}^\top \mathbf{y} + \mathbf{S}_0^{-1} \mathbf{m}_0)^\top \boldsymbol{\theta}.$$

- Matching the coefficients of the quadratic and linear terms in $\boldsymbol{\theta}$ gives

$$\mathbf{S}_N^{-1} = \sigma^{-2} \boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \mathbf{S}_0^{-1} \implies \mathbf{S}_N = (\mathbf{S}_0^{-1} + \sigma^{-2} \boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1},$$

$$\mathbf{m}_N^\top \mathbf{S}_N^{-1} = (\sigma^{-2} \boldsymbol{\Phi}^\top \mathbf{y} + \mathbf{S}_0^{-1} \mathbf{m}_0)^\top \implies \mathbf{m}_N = \mathbf{S}_N (\mathbf{S}_0^{-1} \mathbf{m}_0 + \sigma^{-2} \boldsymbol{\Phi}^\top \mathbf{y}).$$