Lecture 3: Linear Models for Regression

Tao LIN

SoE, Westlake University

September 15, 2025

This lecture:

- Normal Equation and Least Squares
- Probabilistic Interpretation of Linear Regression
- Maximum Likelihood Estimation (MLE)
- Over-fitting and Under-fitting
- Polynomial Regression, Ridge Regression, and Lasso

Next lecture:

- Generalization Gap and Model Selection
- Bias-Variance Decomposition

Reading materials

- Chapter 1, Stanford CS 229 Lecture Notes, `https://cs229.stanford.edu/notes2022fall/main_notes.pdf`
- Chapter 3.1 & 3.2, Bishop, Pattern Recognition and Machine Learning
- Read about over-fitting in the paper by Pedro Domingos (Sections 3 and 5 of “A few useful things to know about machine learning”)

Reference

- EPFL, CS-433 Machine Learning, `https://github.com/epfml/ML_course`

Table of Contents

Table of Contents

Gradient Descent for linear regression with MSE

Considering a dataset $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$ and learnable weights $\mathbf{w} \in \mathbb{R}^D$ for $f_{\mathbf{w}}(\mathbf{X}) = \mathbf{X}\mathbf{w}$.

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^N, \qquad \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1D} \\ x_{21} & x_{22} & \dots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{ND} \end{bmatrix} \in \mathbb{R}^{N \times D} \tag{1}$$

Gradient Descent for linear regression with MSE

Considering a dataset $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$ and learnable weights $\mathbf{w} \in \mathbb{R}^D$ for $f_{\mathbf{w}}(\mathbf{X}) = \mathbf{X}\mathbf{w}$.

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^N, \qquad \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1D} \\ x_{21} & x_{22} & \dots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{ND} \end{bmatrix} \in \mathbb{R}^{N \times D} \tag{1}$$

We define the error vector $\mathbf{e}$:

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{w} = \begin{pmatrix} e_1 \\ \vdots \\ e_N \end{pmatrix} \in \mathbb{R}^N, \tag{2}$$

where $e_n := y_n - \mathbf{x}_n^\top \mathbf{w}$.

Gradient Descent for linear regression with MSE

Considering a dataset $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$ and learnable weights $\mathbf{w} \in \mathbb{R}^D$ for $f_{\mathbf{w}}(\mathbf{X}) = \mathbf{X}\mathbf{w}$.

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^N, \qquad \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1D} \\ x_{21} & x_{22} & \dots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{ND} \end{bmatrix} \in \mathbb{R}^{N \times D} \tag{1}$$

We define the error vector $\mathbf{e}$:

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{w} = \begin{pmatrix} e_1 \\ \vdots \\ e_N \end{pmatrix} \in \mathbb{R}^N, \tag{2}$$

where $e_n := y_n - \mathbf{x}_n^\top \mathbf{w}$. The MSE is defined as:

$$\mathcal{L}(\mathbf{w}) := \frac{1}{2N} \sum_{n=1}^{N} \left(y_n - \mathbf{x}_n^\top \mathbf{w}\right)^2 \tag{3}$$

Gradient Descent for linear regression with MSE

Considering a dataset $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$ and learnable weights $\mathbf{w} \in \mathbb{R}^D$ for $f_{\mathbf{w}}(\mathbf{X}) = \mathbf{X}\mathbf{w}$.

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^N, \qquad \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1D} \\ x_{21} & x_{22} & \dots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{ND} \end{bmatrix} \in \mathbb{R}^{N \times D} \tag{1}$$

We define the error vector $\mathbf{e}$:

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{w} = \begin{pmatrix} e_1 \\ \vdots \\ e_N \end{pmatrix} \in \mathbb{R}^N, \tag{2}$$

where $e_n := y_n - \mathbf{x}_n^\top \mathbf{w}$. The MSE is defined as:

$$\mathcal{L}(\mathbf{w}) := \frac{1}{2N} \sum_{n=1}^{N} \left(y_n - \mathbf{x}_n^\top \mathbf{w}\right)^2 = \tfrac{1}{2N} \mathbf{e}^\top \mathbf{e}, \tag{3}$$

and then the gradient is given by $\nabla \mathcal{L}(\mathbf{w}) = -\frac{1}{N} \mathbf{X}^\top \mathbf{e}$.

Stochastic Gradient Descent (SGD)

- **Sum Objectives:** In machine learning, most cost functions are formulated as a sum over the training examples, that is

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n(\mathbf{w}) ,$$

Stochastic Gradient Descent (SGD)

- **Sum Objectives:** In machine learning, most cost functions are formulated as a sum over the training examples, that is

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n(\mathbf{w}) \ ,$$

where $\mathcal{L}_n$ is the cost contributed by the n-th training example.

Stochastic Gradient Descent (SGD)

- **Sum Objectives:** In machine learning, most cost functions are formulated as a sum over the training examples, that is

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n(\mathbf{w}) \ ,$$

  where $\mathcal{L}_n$ is the cost contributed by the n-th training example.

- **The SGD Algorithm:** The Stochastic Gradient Descent (SGD) algorithm is given by the following update rule, at step t:

$$\mathbf{w}^{(t+1)} := \mathbf{w}^{(t)} - \gamma \, \nabla \mathcal{L}_n(\mathbf{w}^{(t)}) \ .$$

Stochastic Gradient Descent (SGD)

- **Sum Objectives:** In machine learning, most cost functions are formulated as a sum over the training examples, that is

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}_n(\mathbf{w}) \, ,$$

where $\mathcal{L}_n$ is the cost contributed by the n-th training example.

- **The SGD Algorithm:** The Stochastic Gradient Descent (SGD) algorithm is given by the following update rule, at step t:

$$\mathbf{w}^{(t+1)} := \mathbf{w}^{(t)} - \gamma \, \nabla\mathcal{L}_n(\mathbf{w}^{(t)}) \, .$$

- **Theoretical Motivation:** In expectation over the random choice of n, we have

$$\mathbb{E}[\nabla\mathcal{L}_n(\mathbf{w})] = \nabla\mathcal{L}(\mathbf{w})$$

which is the true gradient direction.

- **Mini-batch SGD:** There is an intermediate version, using the update direction being

$$\mathbf{g} := \frac{1}{|B|} \sum_{n \in B} \nabla \mathcal{L}_n(\mathbf{w}^{(t)})$$

- **Mini-batch SGD:** There is an intermediate version, using the update direction being

$$\mathbf{g} := \frac{1}{|B|} \sum_{n \in B} \nabla \mathcal{L}_n(\mathbf{w}^{(t)})$$

again with

$$\mathbf{w}^{(t+1)} := \mathbf{w}^{(t)} - \gamma \, \mathbf{g} \, .$$

- **Mini-batch SGD:** There is an intermediate version, using the update direction being

$$\mathbf{g} := \frac{1}{|B|} \sum_{n \in B} \nabla \mathcal{L}_n(\mathbf{w}^{(t)})$$

  again with

$$\mathbf{w}^{(t+1)} := \mathbf{w}^{(t)} - \gamma \, \mathbf{g} \, .$$

- We have randomly chosen a subset $B \subseteq [N]$ of the training examples.

- **Mini-batch SGD:** There is an intermediate version, using the update direction being

$$\mathbf{g} := \frac{1}{|B|} \sum_{n \in B} \nabla \mathcal{L}_n(\mathbf{w}^{(t)})$$

  again with

$$\mathbf{w}^{(t+1)} := \mathbf{w}^{(t)} - \gamma \, \mathbf{g} \, .$$

- We have randomly chosen a subset $B \subseteq [N]$ of the training examples. For each of these selected examples n, we compute the respective gradient $\nabla \mathcal{L}_n$, at the same current point $\mathbf{w}^{(t)}$.

- **Mini-batch SGD:** There is an intermediate version, using the update direction being

$$\mathbf{g} := \frac{1}{|B|} \sum_{n \in B} \nabla \mathcal{L}_n(\mathbf{w}^{(t)})$$

again with

$$\mathbf{w}^{(t+1)} := \mathbf{w}^{(t)} - \gamma\, \mathbf{g}\,.$$

- We have randomly chosen a subset $B \subseteq [N]$ of the training examples. For each of these selected examples n, we compute the respective gradient $\nabla \mathcal{L}_n$, at the same current point $\mathbf{w}^{(t)}$.
- The computation of $\mathbf{g}$ can be parallelized easily. This is how current deep-learning applications utilize GPUs (by running over $|B|$ threads in parallel).

- **Mini-batch SGD:** There is an intermediate version, using the update direction being

$$\mathbf{g} := \frac{1}{|B|} \sum_{n \in B} \nabla \mathcal{L}_n(\mathbf{w}^{(t)})$$

again with

$$\mathbf{w}^{(t+1)} := \mathbf{w}^{(t)} - \gamma \, \mathbf{g} \, .$$

- We have randomly chosen a subset $B \subseteq [N]$ of the training examples. For each of these selected examples n, we compute the respective gradient $\nabla \mathcal{L}_n$, at the same current point $\mathbf{w}^{(t)}$.
- The computation of $\mathbf{g}$ can be parallelized easily. This is how current deep-learning applications utilize GPUs (by running over $|B|$ threads in parallel).
- Note that in the extreme case $B := [N]$, we obtain (batch) gradient descent, i.e. $\mathbf{g} = \nabla \mathcal{L}$.

An alternative view of SGD

Finite-sum empirical risk minimization problem:

$$f(\mathbf{x}^\star) = \min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} \left(f_i(\mathbf{x}) := \boxed{F(\mathbf{x}, \boldsymbol{\xi}_i)} \right) \right\}$$

An alternative view of SGD

Finite-sum empirical risk minimization problem:

$$f(\mathbf{x}^\star) = \min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} \left(f_i(\mathbf{x}) := \boxed{F(\mathbf{x}, \xi_i)} \right) \right\}$$

- The loss function of i-th data $\xi_i := (\mathbf{d}_i, y_i)$

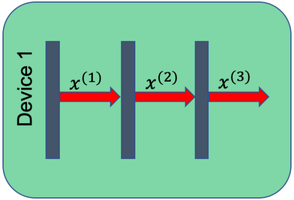
An alternative view of SGD

Finite-sum empirical risk minimization problem:

$$f(\mathbf{x}^\star) = \min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} \left(f_i(\mathbf{x}) := \boxed{F(\mathbf{x}, \boldsymbol{\xi}_i)} \right) \right\}$$

- The loss function of i-th data $\boldsymbol{\xi}_i := (\mathbf{d}_i, y_i)$
- Baseline method: Stochastic Gradient Descent (SGD)

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \eta \nabla F(\mathbf{x}^{(t)}, \boldsymbol{\xi}_i)$$



An alternative view of SGD

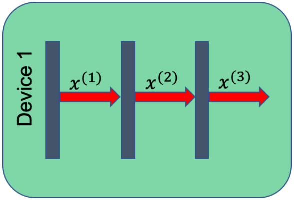
Finite-sum empirical risk minimization problem:

$$f(\mathbf{x}^\star) = \min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} \left(f_i(\mathbf{x}) := \boxed{F(\mathbf{x}, \boldsymbol{\xi}_i)} \right) \right\}$$

- The loss function of i-th data $\boldsymbol{\xi}_i := (\mathbf{d}_i, y_i)$
- Baseline method: Stochastic Gradient Descent (SGD)

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \boxed{\eta}\ \boxed{\nabla F(\mathbf{x}^{(t)}, \boldsymbol{\xi}_i)}$$

  - η is step-size/learning rate



An alternative view of SGD

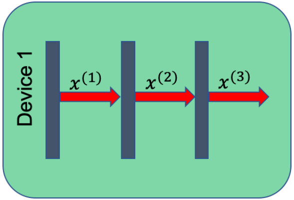
Finite-sum empirical risk minimization problem:

$$f(\mathbf{x}^\star) = \min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} \left(f_i(\mathbf{x}) := F(\mathbf{x}, \boldsymbol{\xi}_i) \right) \right\}$$

- The loss function of i-th data $\boldsymbol{\xi}_i := (\mathbf{d}_i, y_i)$
- Baseline method: Stochastic Gradient Descent (SGD)

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \eta \nabla F(\mathbf{x}^{(t)}, \boldsymbol{\xi}_i)$$

  - η is step-size/learning rate
  - sampled i.i.d. $i \in \{1, \ldots, N\}$



An alternative view of SGD

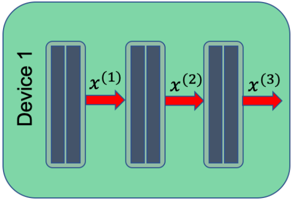
Finite-sum empirical risk minimization problem:

$$f(\mathbf{x}^\star) = \min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} \left(f_i(\mathbf{x}) := F(\mathbf{x}, \boldsymbol{\xi}_i) \right) \right\}$$

- The loss function of i-th data $\boldsymbol{\xi}_i := (\mathbf{d}_i, y_i)$
- Baseline method: Stochastic Gradient Descent (SGD)

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \boxed{\frac{1}{B}\sum_{i\in\mathcal{B}}} \eta \nabla f_i(\mathbf{x})$$

(Using *mini-batching*)



An alternative view of SGD

Finite-sum empirical risk minimization problem:

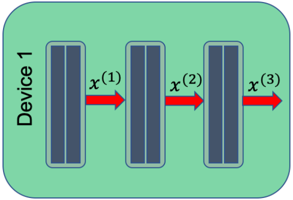
$$f(\mathbf{x}^\star) = \min_{\mathbf{x}\in\mathbb{R}^d} \left\{ f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^N \left(f_i(\mathbf{x}) := F(\mathbf{x}, \boldsymbol{\xi}_i) \right) \right\}$$

- The loss function of i-th data $\boldsymbol{\xi}_i := (\mathbf{d}_i, y_i)$
- Baseline method: Stochastic Gradient Descent (SGD)

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \frac{1}{B}\sum_{i\in\mathcal{B}} \eta \nabla f_i(\mathbf{x})$$

(Using *mini-batching*)

  - $\mathcal{B} \in \{1, \ldots, N\}$ with $|\mathcal{B}| = B$.



Alternative view: Stochastic reformulation of finite-sum problems (SGD with arbitrary sampling)

Random sampling vector $\mathbf{v} = (v_1, \ldots, v_N) \sim \mathcal{D}$ with $\mathbb{E}\left[v_i\right] = 1 \qquad$ for $i = 1, \ldots, N$.

Alternative view: Stochastic reformulation of finite-sum problems (SGD with arbitrary sampling)

Random sampling vector $\mathbf{v} = (v_1, \ldots, v_N) \sim \mathcal{D}$ with $\mathbb{E}[v_i] = 1 \qquad$ for $i = 1, \ldots, N$.

$$f(\mathbf{x}) := \frac{1}{N} \sum_{i=1}^{N} f_i(\mathbf{x}) \qquad \text{(Stochastic Reformulation)}$$

Alternative view: Stochastic reformulation of finite-sum problems (SGD with arbitrary sampling)

Random sampling vector $\mathbf{v} = (v_1, \ldots, v_N) \sim \mathcal{D}$ with $\mathbb{E}\left[v_i\right] = 1 \qquad$ for $i = 1, \ldots, N$.

$$f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}\left[v_i\right] f_i(\mathbf{x}) \qquad \text{(Stochastic Reformulation)}$$

Alternative view: Stochastic reformulation of finite-sum problems (SGD with arbitrary sampling)

Random sampling vector $\mathbf{v} = (v_1, \ldots, v_N) \sim \mathcal{D}$ with $\mathbb{E}[v_i] = 1 \qquad$ for $i = 1, \ldots, N$.

$$f(\mathbf{x}) := \tfrac{1}{N}\sum_{i=1}^N f_i(\mathbf{x}) = \tfrac{1}{N}\sum_{i=1}^N \mathbb{E}[v_i] f_i(\mathbf{x}) = \mathbb{E}\left[\underbrace{\frac{1}{N}\sum_{i=1}^N v_i f_i(\mathbf{x})}_{=: f_v(\mathbf{x})}\right] \qquad \text{(Stochastic Reformulation)}$$

Alternative view: Stochastic reformulation of finite-sum problems (SGD with arbitrary sampling)

Random sampling vector $\mathbf{v} = (v_1, \ldots, v_N) \sim \mathcal{D}$ with $\mathbb{E}[v_i] = 1$ for $i = 1, \ldots, N$.

$$f(\mathbf{x}) := \tfrac{1}{N}\sum_{i=1}^N f_i(\mathbf{x}) = \tfrac{1}{N}\sum_{i=1}^N \mathbb{E}[v_i] f_i(\mathbf{x}) = \mathbb{E}\left[\underbrace{\frac{1}{N}\sum_{i=1}^N v_i f_i(\mathbf{x})}_{=: f_v(\mathbf{x})}\right] \qquad \text{(Stochastic Reformulation)}$$

Original Finite-sum problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{N}\sum_{i=1}^N f_i(\mathbf{x})$$

$\longleftrightarrow$

Stochastic Reformulation

$$\min_{\mathbf{x}\in\mathbb{R}^d} \mathbb{E}[f_v(\mathbf{x})]$$

Minimizing the expectation of **random linear combinations** of original function

Alternative view: Stochastic reformulation of finite-sum problems (SGD with arbitrary sampling)

Random sampling vector $\mathbf{v} = (v_1, \ldots, v_N) \sim \mathcal{D}$ with $\mathbb{E}[v_i] = 1$ for $i = 1, \ldots, N$.

$$f(\mathbf{x}) := \tfrac{1}{N}\textstyle\sum_{i=1}^N f_i(\mathbf{x}) = \tfrac{1}{N}\sum_{i=1}^N \mathbb{E}[v_i] f_i(\mathbf{x}) = \mathbb{E}\left[\underbrace{\frac{1}{N}\sum_{i=1}^N v_i f_i(\mathbf{x})}_{=:f_v(\mathbf{x})}\right] \quad \text{(Stochastic Reformulation)}$$

Original Finite-sum problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{N}\sum_{i=1}^N f_i(\mathbf{x})$$

$\longleftrightarrow$

Stochastic Reformulation

$$\min_{\mathbf{x}\in\mathbb{R}^d} \mathbb{E}[f_v(\mathbf{x})]$$

Minimizing the expectation of **random linear combinations** of original function

$\downarrow$

Sample $v_i^{(t)} \sim \mathcal{D}$

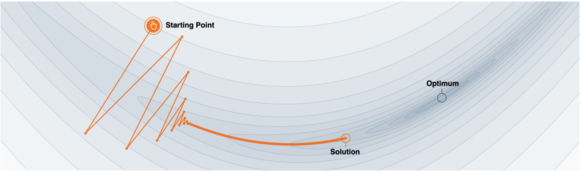
$$\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} - \eta \nabla f_{v_i^{(t)}}(\mathbf{x}^{(t)})$$

The distribution $\mathcal{D}$ encodes any form of mini-batching / non-uniform sampling.

Table of Contents

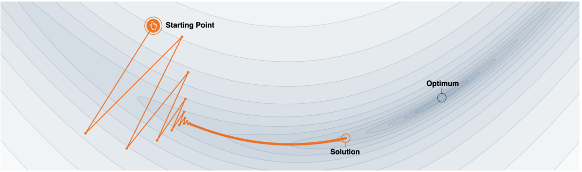
Motivation

Recall that *GD for Linear Regression with MSE loss* is an iterative optimization algorithm, computationally flexibility for handling large datasets.



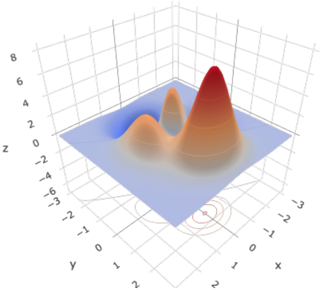
Motivation

Recall that *GD for Linear Regression with MSE loss* is an iterative optimization algorithm, computationally flexibility for handling large datasets.

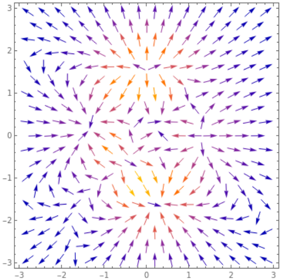


Can we compute the optimum of the cost function analytically?

- Linear regression using an MSE cost function is one such case.

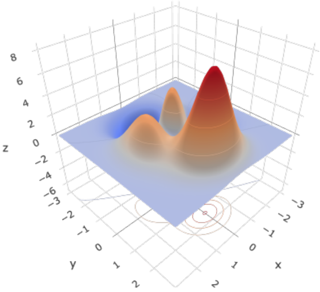


Cost function $f(x, y)$.

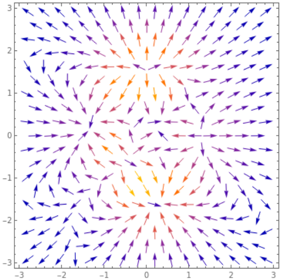


Gradient of f plotted as a vector field.

- Linear regression using an MSE cost function is one such case.

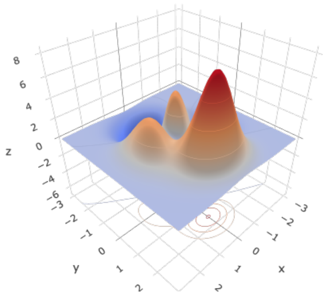


Cost function $f(x, y)$.

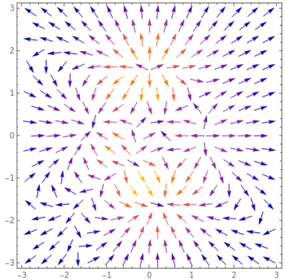


Gradient of f plotted as a vector field.

- Linear regression using an MSE cost function is one such case.
- Its solution can be obtained explicitly, **by solving a linear system of equations**.



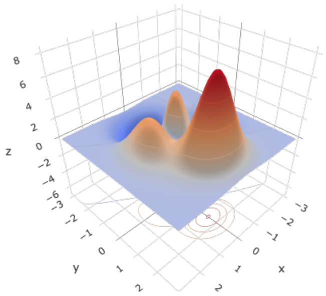
Cost function $f(x, y)$.



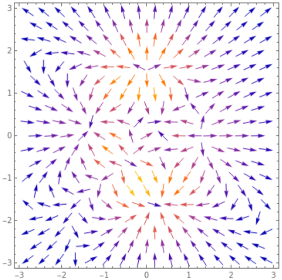
Gradient of f plotted as a vector field.

- Linear regression using an MSE cost function is one such case.
- Its solution can be obtained explicitly, **by solving a linear system of equations**.

$\Rightarrow$ These equations are sometimes called the normal equations.



Cost function $f(x, y)$.



Gradient of f plotted as a vector field.

- Linear regression using an MSE cost function is one such case.
- Its solution can be obtained explicitly, **by solving a linear system of equations**.

$\Rightarrow$ These equations are sometimes called the normal equations.

$\Rightarrow$ Solving the normal equations is called the least squares.

Recall that the cost function for linear regression with MSE is given by

Recall that the cost function for linear regression with MSE is given by

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2N}\sum_{n=1}^{N}\left(y_n - \mathbf{x}_n^\top \mathbf{w}\right)^2 = \frac{1}{2N}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w}), \tag{4}$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^N, \qquad \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1D} \\ x_{21} & x_{22} & \dots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{ND} \end{bmatrix} \in \mathbb{R}^{N \times D}. \tag{5}$$

Recall that the cost function for linear regression with MSE is given by

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2N}\sum_{n=1}^{N}\left(y_n - \mathbf{x}_n^\top \mathbf{w}\right)^2 = \frac{1}{2N}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w}), \tag{4}$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^N, \qquad \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1D} \\ x_{21} & x_{22} & \dots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{ND} \end{bmatrix} \in \mathbb{R}^{N\times D}. \tag{5}$$

What is the meaning of finding the analytical solution?

Steps to form normal equations

Steps to form normal equations

Let's start with some tools:

Steps to form normal equations

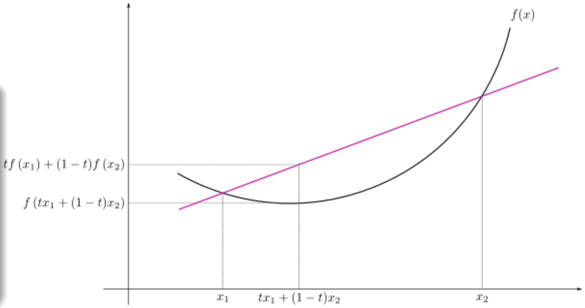
Let's start with some tools:

Definition 1 (Convexity)

Steps to form normal equations

Let's start with some tools:

Definition 1 (Convexity)



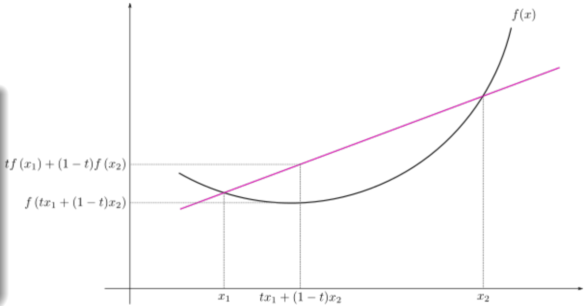
Steps to form normal equations

Let's start with some tools:

Definition 1 (Convexity)

A function $h(\mathbf{u})$ with $\mathbf{u} \in \mathbb{R}^D$ is convex, if for any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^D$ and for any $0 \leq \lambda \leq 1$, we have:

$$h(\lambda \mathbf{u} + (1-\lambda)\mathbf{v}) \leq \lambda h(\mathbf{u}) + (1-\lambda)h(\mathbf{v}) \tag{6}$$



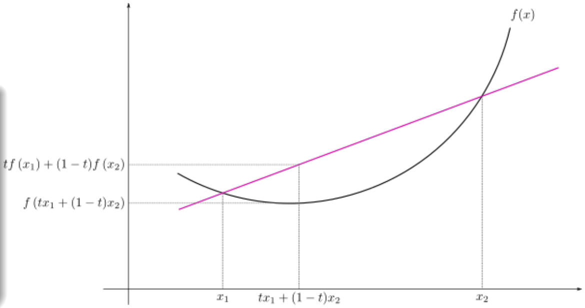
Steps to form normal equations

Let's start with some tools:

Definition 1 (Convexity)

A function $h(\mathbf{u})$ with $\mathbf{u} \in \mathbb{R}^D$ is convex, if for any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^D$ and for any $0 \leq \lambda \leq 1$, we have:

$$h(\lambda \mathbf{u} + (1-\lambda)\mathbf{v}) \leq \lambda h(\mathbf{u}) + (1-\lambda) h(\mathbf{v}) \quad (6)$$



To derive the normal equations,

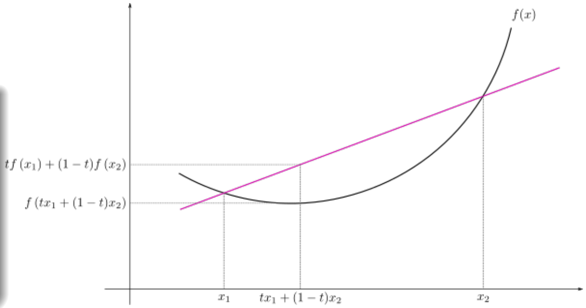
Steps to form normal equations

Let's start with some tools:

Definition 1 (Convexity)

A function $h(\mathbf{u})$ with $\mathbf{u} \in \mathbb{R}^D$ is convex, if for any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^D$ and for any $0 \leq \lambda \leq 1$, we have:

$$h\left(\lambda \mathbf{u} + (1-\lambda)\mathbf{v}\right) \leq \lambda h(\mathbf{u}) + (1-\lambda) h(\mathbf{v}) \tag{6}$$



To derive the normal equations,

1. we first show that the problem is convex.

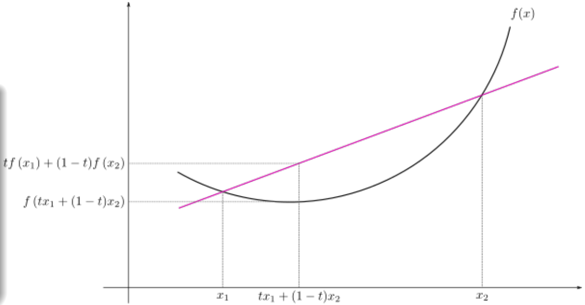
Steps to form normal equations

Let's start with some tools:

Definition 1 (Convexity)

A function $h(\mathbf{u})$ with $\mathbf{u} \in \mathbb{R}^D$ is convex, if for any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^D$ and for any $0 \leq \lambda \leq 1$, we have:

$$h\left(\lambda \mathbf{u} + (1-\lambda)\mathbf{v}\right) \leq \lambda h(\mathbf{u}) + (1-\lambda)h(\mathbf{v}) \tag{6}$$



To derive the normal equations,

1. we first show that the problem is convex.
2. we then use the optimality conditions for convex functions, i.e.,

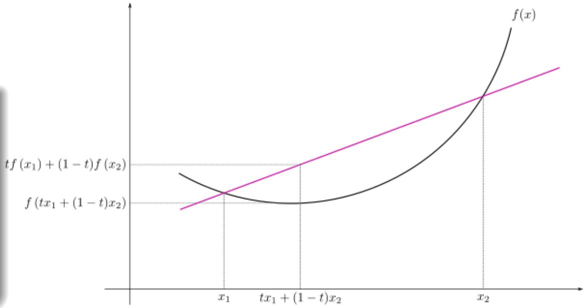
Steps to form normal equations

Let's start with some tools:

Definition 1 (Convexity)

A function $h(\mathbf{u})$ with $\mathbf{u} \in \mathbb{R}^D$ is convex, if for any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^D$ and for any $0 \leq \lambda \leq 1$, we have:

$$h(\lambda \mathbf{u} + (1-\lambda)\mathbf{v}) \leq \lambda h(\mathbf{u}) + (1-\lambda) h(\mathbf{v}) \quad (6)$$



To derive the normal equations,

1. we first show that the problem is convex.
2. we then use the optimality conditions for convex functions, i.e.,

$$\nabla \mathcal{L}(\mathbf{w}^\star) = \mathbf{0}\,, \quad (7)$$

where $\mathbf{w}^\star$ corresponds to the parameter at the optimum point.

Derivation (step 1): the MSE is *convex* in the $\mathbf{w}$

There are several ways of proving this:

Derivation (step 1): the MSE is *convex* in the $\mathbf{w}$

There are several ways of proving this:

1. **Way 1.** Recall the definition of $\mathcal{L}$

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left(\mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w}\right)^2 , \tag{8}$$

   where each $\mathcal{L}_n$ is the composition of a linear function with a convex function.

Derivation (step 1): the MSE is *convex* in the $\mathbf{w}$

There are several ways of proving this:

1. **Way 1.** Recall the definition of $\mathcal{L}$

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left(\mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w} \right)^2 , \tag{8}$$

   where each $\mathcal{L}_n$ is the composition of a linear function with a convex function.
   $\Rightarrow$ We conclude the proof by

Derivation (step 1): the MSE is *convex* in the $\mathbf{w}$

There are several ways of proving this:

1. **Way 1.** Recall the definition of $\mathcal{L}$

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left(\mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w} \right)^2 , \tag{8}$$

where each $\mathcal{L}_n$ is the composition of a linear function with a convex function.
$\Rightarrow$ We conclude the proof by "the sum of convex functions is still a convex function".

Derivation (step 1): the MSE is *convex* in the $\mathbf{w}$

There are several ways of proving this:

1. **Way 1.** Recall the definition of $\mathcal{L}$

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left(\mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w} \right)^2 , \tag{8}$$

where each $\mathcal{L}_n$ is the composition of a linear function with a convex function.
$\Rightarrow$ We conclude the proof by "the sum of convex functions is still a convex function".

2. **Way 2.** By verifying the definition of convexity, that for any $\lambda \in [0, 1]$ and $\mathbf{w}, \mathbf{w}'$,

$$\mathcal{L}(\lambda \mathbf{w} + (1-\lambda)\mathbf{w}') - (\lambda \mathcal{L}(\mathbf{w}) + (1-\lambda)\mathcal{L}(\mathbf{w}')) \le 0 \,. \tag{9}$$

Derivation (step 1): the MSE is *convex* in the $\mathbf{w}$

There are several ways of proving this:

1. **Way 1.** Recall the definition of $\mathcal{L}$

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left(\mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w}\right)^2 , \tag{8}$$

   where each $\mathcal{L}_n$ is the composition of a linear function with a convex function.
   $\Rightarrow$ We conclude the proof by "the sum of convex functions is still a convex function".

2. **Way 2.** By verifying the definition of convexity, that for any $\lambda \in [0,1]$ and $\mathbf{w}, \mathbf{w}'$,

$$\mathcal{L}(\lambda \mathbf{w} + (1-\lambda)\mathbf{w}') - (\lambda \mathcal{L}(\mathbf{w}) + (1-\lambda)\mathcal{L}(\mathbf{w}')) \leq 0 . \tag{9}$$

   Exercise.

Derivation (step 1): the MSE is *convex* in the $\mathbf{w}$

There are several ways of proving this:

1. **Way 1.** Recall the definition of $\mathcal{L}$

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left(\mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w} \right)^2 , \tag{8}$$

   where each $\mathcal{L}_n$ is the composition of a linear function with a convex function.
   $\Rightarrow$ We conclude the proof by "the sum of convex functions is still a convex function".

2. **Way 2.** By verifying the definition of convexity, that for any $\lambda \in [0, 1]$ and $\mathbf{w}, \mathbf{w}'$,

$$\mathcal{L}(\lambda \mathbf{w} + (1-\lambda)\mathbf{w}') - (\lambda \mathcal{L}(\mathbf{w}) + (1-\lambda)\mathcal{L}(\mathbf{w}')) \leq 0 \, . \tag{9}$$

   Exercise. The LHS of our case $-\frac{1}{2N}\lambda(1-\lambda) \left\| \mathbf{X}(\mathbf{w} - \mathbf{w}') \right\|_2^2$ indeed is non-positive.

Derivation (step 1): the MSE is *convex* in the $\mathbf{w}$

There are several ways of proving this:

1. **Way 1.** Recall the definition of $\mathcal{L}$

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left(\mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w} \right)^2 , \tag{8}$$

   where each $\mathcal{L}_n$ is the composition of a linear function with a convex function.
   $\Rightarrow$ We conclude the proof by "the sum of convex functions is still a convex function".

2. **Way 2.** By verifying the definition of convexity, that for any $\lambda \in [0, 1]$ and $\mathbf{w}, \mathbf{w}'$,

$$\mathcal{L}(\lambda \mathbf{w} + (1-\lambda)\mathbf{w}') - (\lambda \mathcal{L}(\mathbf{w}) + (1-\lambda)\mathcal{L}(\mathbf{w}')) \leq 0 \,. \tag{9}$$

   Exercise. The LHS of our case $-\frac{1}{2N}\lambda(1-\lambda) \left\| \mathbf{X}(\mathbf{w} - \mathbf{w}') \right\|_2^2$ indeed is non-positive.

3. **Way 3.** Check the second derivative (the Hessian) and show that it is positive semi-definite (all its eigenvalues are non-negative).

Derivation (step 1): the MSE is *convex* in the $\mathbf{w}$

There are several ways of proving this:

1. **Way 1.** Recall the definition of $\mathcal{L}$

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left(\mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w} \right)^2 , \tag{8}$$

where each $\mathcal{L}_n$ is the composition of a linear function with a convex function.
$\Rightarrow$ We conclude the proof by "the sum of convex functions is still a convex function".

2. **Way 2.** By verifying the definition of convexity, that for any $\lambda \in [0,1]$ and $\mathbf{w}, \mathbf{w}'$,

$$\mathcal{L}(\lambda \mathbf{w} + (1-\lambda)\mathbf{w}') - (\lambda \mathcal{L}(\mathbf{w}) + (1-\lambda)\mathcal{L}(\mathbf{w}')) \leq 0 \,. \tag{9}$$

Exercise. The LHS of our case $-\frac{1}{2N}\lambda(1-\lambda) \left\| \mathbf{X}(\mathbf{w}-\mathbf{w}') \right\|_2^2$ indeed is non-positive.

3. **Way 3.** Check the second derivative (the Hessian) and show that it is positive semi-definite (all its eigenvalues are non-negative). Exercise.

Derivation (step 1): the MSE is *convex* in the $\mathbf{w}$

There are several ways of proving this:

1. **Way 1.** Recall the definition of $\mathcal{L}$

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left(\mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w}\right)^2 , \tag{8}$$

   where each $\mathcal{L}_n$ is the composition of a linear function with a convex function.
   $\Rightarrow$ We conclude the proof by "the sum of convex functions is still a convex function".

2. **Way 2.** By verifying the definition of convexity, that for any $\lambda \in [0, 1]$ and $\mathbf{w}, \mathbf{w}'$,

$$\mathcal{L}(\lambda \mathbf{w} + (1-\lambda)\mathbf{w}') - (\lambda \mathcal{L}(\mathbf{w}) + (1-\lambda)\mathcal{L}(\mathbf{w}')) \leq 0 . \tag{9}$$

   Exercise. The LHS of our case $-\frac{1}{2N}\lambda(1-\lambda)\left\|\mathbf{X}(\mathbf{w}-\mathbf{w}')\right\|_2^2$ indeed is non-positive.

3. **Way 3.** Check the second derivative (the Hessian) and show that it is positive semi-definite (all its eigenvalues are non-negative). Exercise.
   $\Rightarrow$ the Hessian has the form $\frac{1}{N}\mathbf{X}^\top\mathbf{X}$,

Derivation (step 1): the MSE is *convex* in the $\mathbf{w}$

There are several ways of proving this:

1. **Way 1.** Recall the definition of $\mathcal{L}$

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left(\mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w} \right)^2 , \tag{8}$$

   where each $\mathcal{L}_n$ is the composition of a linear function with a convex function.
   $\Rightarrow$ We conclude the proof by "the sum of convex functions is still a convex function".

2. **Way 2.** By verifying the definition of convexity, that for any $\lambda \in [0, 1]$ and $\mathbf{w}, \mathbf{w}'$,

$$\mathcal{L}(\lambda \mathbf{w} + (1-\lambda)\mathbf{w}') - (\lambda \mathcal{L}(\mathbf{w}) + (1-\lambda)\mathcal{L}(\mathbf{w}')) \leq 0 \,. \tag{9}$$

   Exercise. The LHS of our case $-\frac{1}{2N}\lambda(1-\lambda)\left\|\mathbf{X}(\mathbf{w}-\mathbf{w}')\right\|_2^2$ indeed is non-positive.

3. **Way 3.** Check the second derivative (the Hessian) and show that it is positive semi-definite (all its eigenvalues are non-negative). Exercise.
   $\Rightarrow$ the Hessian has the form $\frac{1}{N}\mathbf{X}^\top\mathbf{X}$, which is indeed positive semi-definite.

Derivation (step 1): the MSE is *convex* in the $\mathbf{w}$

There are several ways of proving this:

1. **Way 1.** Recall the definition of $\mathcal{L}$

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left(\mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w} \right)^2 , \tag{8}$$

   where each $\mathcal{L}_n$ is the composition of a linear function with a convex function.
   $\Rightarrow$ We conclude the proof by "the sum of convex functions is still a convex function".

2. **Way 2.** By verifying the definition of convexity, that for any $\lambda \in [0, 1]$ and $\mathbf{w}, \mathbf{w}'$,

$$\mathcal{L}(\lambda \mathbf{w} + (1-\lambda)\mathbf{w}') - (\lambda \mathcal{L}(\mathbf{w}) + (1-\lambda)\mathcal{L}(\mathbf{w}')) \leq 0 \,. \tag{9}$$

   Exercise. The LHS of our case $-\frac{1}{2N}\lambda(1-\lambda) \left\| \mathbf{X}(\mathbf{w} - \mathbf{w}') \right\|_2^2$ indeed is non-positive.

3. **Way 3.** Check the second derivative (the Hessian) and show that it is positive semi-definite (all its eigenvalues are non-negative). Exercise.
   $\Rightarrow$ the Hessian has the form $\frac{1}{N}\mathbf{X}^\top\mathbf{X}$, which is indeed positive semi-definite.
   $\Rightarrow$ its non-zero eigenvalues are the squares of the non-zero singular values of the matrix $\mathbf{X}$.

Derivation (step 2): finding the minimum of a convex function

Derivation (step 2): finding the minimum of a convex function

By taking the gradient of $\mathcal{L}(\mathbf{w}) = \frac{1}{2N}\sum_{n=1}^{N}\left(y_n - \mathbf{x}_n^\top\mathbf{w}\right)^2 = \frac{1}{2N}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w})$, we have

$$\nabla\mathcal{L}(\mathbf{w}) = -\frac{1}{N}\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w})\,. \tag{10}$$

Derivation (step 2): finding the minimum of a convex function

By taking the gradient of $\mathcal{L}(\mathbf{w}) = \frac{1}{2N}\sum_{n=1}^{N}\left(y_n - \mathbf{x}_n^\top \mathbf{w}\right)^2 = \frac{1}{2N}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w})$, we have

$$\nabla\mathcal{L}(\mathbf{w}) = -\frac{1}{N}\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w})\,. \tag{10}$$

Given the property of convexity $\nabla\mathcal{L}(\mathbf{w}^\star) = \mathbf{0}$,

Derivation (step 2): finding the minimum of a convex function

By taking the gradient of $\mathcal{L}(\mathbf{w}) = \frac{1}{2N}\sum_{n=1}^{N}\left(y_n - \mathbf{x}_n^\top \mathbf{w}\right)^2 = \frac{1}{2N}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w})$, we have

$$\nabla\mathcal{L}(\mathbf{w}) = -\frac{1}{N}\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w})\,. \tag{10}$$

Given the property of convexity $\nabla\mathcal{L}(\mathbf{w}^\star) = \mathbf{0}$, we can get the normal equations for linear regression:

$$\mathbf{X}^\top \underbrace{(\mathbf{y} - \mathbf{X}\mathbf{w})}_{\text{error}} = \mathbf{0}\,, \tag{11}$$

where the error $\mathbf{e} := \mathbf{y} - \mathbf{X}\mathbf{w}$ is orthogonal to all columns of $\mathbf{X}$.

Geometric interpretation of Normal Equations

Definition 2 (Span of a set of vectors)

The **span** of a set of vectors, $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$, is the set of all possible linear combinations of these vectors; i.e. $\mathrm{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_k\} = \{\alpha_1 \mathbf{x}_1 + \ldots + \alpha_k \mathbf{x}_k \,|\, \alpha_1, \ldots, \alpha_k \in \mathbb{R}\}$.

Geometric interpretation of Normal Equations

Definition 2 (Span of a set of vectors)

The **span** of a set of vectors, $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$, is the set of all possible linear combinations of these vectors; i.e. span$\{\mathbf{x}_1, \ldots, \mathbf{x}_k\} = \{\alpha_1 \mathbf{x}_1 + \ldots + \alpha_k \mathbf{x}_k \,|\, \alpha_1, \ldots, \alpha_k \in \mathbb{R}\}$.

- The span of $\mathbf{X} \in \mathbb{R}^{N \times D}$ is the space spanned by the columns of $\mathbf{X}$

$$\mathcal{S} := span(\mathbf{X}) = \{\mathbf{u} := \mathbf{X}\mathbf{w} | \mathbf{w} \in \mathbb{R}^D\}$$

Geometric interpretation of Normal Equations

Definition 2 (Span of a set of vectors)

The **span** of a set of vectors, $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$, is the set of all possible linear combinations of these vectors; i.e. $\text{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_k\} = \{\alpha_1\mathbf{x}_1 + \ldots + \alpha_k\mathbf{x}_k \,|\, \alpha_1, \ldots, \alpha_k \in \mathbb{R}\}$.

- The span of $\mathbf{X} \in \mathbb{R}^{N\times D}$ is the space spanned by the columns of $\mathbf{X}$

$$\mathcal{S} := span(\mathbf{X}) = \{\mathbf{u} := \mathbf{X}\mathbf{w} | \mathbf{w} \in \mathbb{R}^D\}$$

Which element $\mathbf{u}$ of $span(\mathbf{X})$ shall we take such that the normal equation $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}$?

Geometric interpretation of Normal Equations

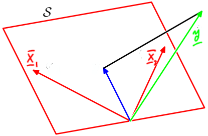
Definition 2 (Span of a set of vectors)

The **span** of a set of vectors, $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$, is the set of all possible linear combinations of these vectors; i.e. $\text{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_k\} = \{\alpha_1 \mathbf{x}_1 + \ldots + \alpha_k \mathbf{x}_k \mid \alpha_1, \ldots, \alpha_k \in \mathbb{R}\}$.

- The span of $\mathbf{X} \in \mathbb{R}^{N \times D}$ is the space spanned by the columns of $\mathbf{X}$

$$\mathcal{S} := span(\mathbf{X}) = \{\mathbf{u} := \mathbf{X}\mathbf{w} | \mathbf{w} \in \mathbb{R}^D\}$$

Which element $\mathbf{u}$ of $span(\mathbf{X})$ shall we take such that the normal equation $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}$?



(taken from Bishop's book)

Geometric interpretation of Normal Equations

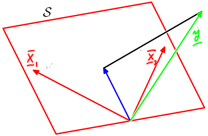
Definition 2 (Span of a set of vectors)

The **span** of a set of vectors, $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$, is the set of all possible linear combinations of these vectors; i.e. $\text{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_k\} = \{\alpha_1 \mathbf{x}_1 + \ldots + \alpha_k \mathbf{x}_k \,|\, \alpha_1, \ldots, \alpha_k \in \mathbb{R}\}$.

- The span of $\mathbf{X} \in \mathbb{R}^{N \times D}$ is the space spanned by the columns of $\mathbf{X}$

$$\mathcal{S} := span(\mathbf{X}) = \{\mathbf{u} := \mathbf{X}\mathbf{w} | \mathbf{w} \in \mathbb{R}^D\}$$

Which element $\mathbf{u}$ of $span(\mathbf{X})$ shall we take such that the normal equation $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}$?



(taken from Bishop's book)

From $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}$, we have:

Geometric interpretation of Normal Equations

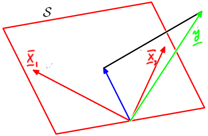
Definition 2 (Span of a set of vectors)

The **span** of a set of vectors, $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$, is the set of all possible linear combinations of these vectors; i.e. $\text{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_k\} = \{\alpha_1 \mathbf{x}_1 + \ldots + \alpha_k \mathbf{x}_k \,|\, \alpha_1, \ldots, \alpha_k \in \mathbb{R}\}$.

- The span of $\mathbf{X} \in \mathbb{R}^{N \times D}$ is the space spanned by the columns of $\mathbf{X}$

$$\mathcal{S} := span(\mathbf{X}) = \{\mathbf{u} := \mathbf{X}\mathbf{w} | \mathbf{w} \in \mathbb{R}^D\}$$

Which element $\mathbf{u}$ of $span(\mathbf{X})$ shall we take such that the normal equation $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}$?



(taken from Bishop's book)

From $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}$, we have:

- the optimum choice for $\mathbf{u}$, i.e. $\mathbf{u}^\star$, requires $\mathbf{y} - \mathbf{u}^\star$ to be orthogonal to $span(\mathbf{X})$.

Geometric interpretation of Normal Equations

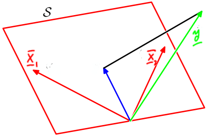
Definition 2 (Span of a set of vectors)

The **span** of a set of vectors, $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$, is the set of all possible linear combinations of these vectors; i.e. $\text{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_k\} = \{\alpha_1 \mathbf{x}_1 + \ldots + \alpha_k \mathbf{x}_k \mid \alpha_1, \ldots, \alpha_k \in \mathbb{R}\}$.

- The span of $\mathbf{X} \in \mathbb{R}^{N \times D}$ is the space spanned by the columns of $\mathbf{X}$

$$\mathcal{S} := span(\mathbf{X}) = \{\mathbf{u} := \mathbf{X}\mathbf{w} | \mathbf{w} \in \mathbb{R}^D\}$$

Which element $\mathbf{u}$ of $span(\mathbf{X})$ shall we take such that the normal equation $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}$?



(taken from Bishop's book)

From $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}$, we have:

- the optimum choice for $\mathbf{u}$, i.e. $\mathbf{u}^\star$, requires $\mathbf{y} - \mathbf{u}^\star$ to be orthogonal to $span(\mathbf{X})$.
- $\mathbf{u}^\star$ should be equal to *the projection of* $\mathbf{y}$ *onto* $span(\mathbf{X})$.

Least Squares

We need to solve the linear system of the normal equation $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}$, where

$$\mathbf{X}^\top y = \underbrace{\mathbf{X}^\top\mathbf{X}}_{\text{Gram matrix}} \mathbf{w} \tag{12}$$

Least Squares

We need to solve the linear system of the normal equation $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}$, where

$$\mathbf{X}^\top y = \underbrace{\mathbf{X}^\top \mathbf{X}}_{\text{Gram matrix}} \mathbf{w} \tag{12}$$

If the Gram matrix is invertible,

Least Squares

We need to solve the linear system of the normal equation $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}$, where

$$\mathbf{X}^\top y = \underbrace{\mathbf{X}^\top\mathbf{X}}_{\text{Gram matrix}} \mathbf{w} \tag{12}$$

If the Gram matrix is invertible, we can multiply the normal equation by the inverse of the Gram matrix from the left:

$$\mathbf{w}^\star = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}\,, \tag{13}$$

Least Squares

We need to solve the linear system of the normal equation $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}$, where

$$\mathbf{X}^\top y = \underbrace{\mathbf{X}^\top\mathbf{X}}_{\text{Gram matrix}} \mathbf{w} \tag{12}$$

If the Gram matrix is invertible, we can multiply the normal equation by the inverse of the Gram matrix from the left:

$$\mathbf{w}^\star = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}\,, \tag{13}$$

where we can get a closed-form expression for the minimum.

Least Squares

We need to solve the linear system of the normal equation $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}$, where

$$\mathbf{X}^\top y = \underbrace{\mathbf{X}^\top \mathbf{X}}_{\text{Gram matrix}} \mathbf{w} \tag{12}$$

If the Gram matrix is invertible, we can multiply the normal equation by the inverse of the Gram matrix from the left:

$$\mathbf{w}^\star = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}\,, \tag{13}$$

where we can get a closed-form expression for the minimum.

We can use this model to predict a new value for an unseen datapoint (test point) $\mathbf{x}_m$:

$$\hat{y}_m := \mathbf{x}_m^\top \mathbf{w}^\star = \mathbf{x}_m^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}\,. \tag{14}$$

Least Squares

We need to solve the linear system of the normal equation $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}$, where

$$\mathbf{X}^\top y = \underbrace{\mathbf{X}^\top\mathbf{X}}_{\text{Gram matrix}} \quad \mathbf{w} \tag{12}$$

If the Gram matrix is invertible, we can multiply the normal equation by the inverse of the Gram matrix from the left:

$$\mathbf{w}^\star = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}\,, \tag{13}$$

where we can get a closed-form expression for the minimum.

We can use this model to predict a new value for an unseen datapoint (test point) $\mathbf{x}_m$:

$$\hat{y}_m := \mathbf{x}_m^\top\mathbf{w}^\star = \mathbf{x}_m^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}\,. \tag{14}$$

Remark 3

The Gram matrix $\mathbf{X}^\top\mathbf{X} \in \mathbb{R}^{D\times D}$ is invertible if and only if $\mathbf{X}$ has full column rank, or in other words $rank(\mathbf{X}) = D$.

Rank deficiency and ill-conditioning

Unfortunately, in practice, $\mathbf{X} \in \mathbb{R}^{N \times D}$ is often rank deficient.

Rank deficiency and ill-conditioning

Unfortunately, in practice, $\mathbf{X} \in \mathbb{R}^{N \times D}$ is often rank deficient.

For example,

Rank deficiency and ill-conditioning

Unfortunately, in practice, $\mathbf{X} \in \mathbb{R}^{N \times D}$ is often rank deficient.

For example,

- If $D > N$ (namely over-parameterized), we always have $rank(\mathbf{X}) < D$

Rank deficiency and ill-conditioning

Unfortunately, in practice, $\mathbf{X} \in \mathbb{R}^{N \times D}$ is often rank deficient.

For example,

- If $D > N$ (namely over-parameterized), we always have $rank(\mathbf{X}) < D$ (since row rank = col. rank)

Rank deficiency and ill-conditioning

Unfortunately, in practice, $\mathbf{X} \in \mathbb{R}^{N \times D}$ is often rank deficient.

For example,

- If $D > N$ (namely over-parameterized), we always have $rank(\mathbf{X}) < D$ (since row rank = col. rank)
- If $D \leq N$, but some of the columns $\mathbf{x}_{:d}$ are (nearly) collinear

Rank deficiency and ill-conditioning

Unfortunately, in practice, $\mathbf{X} \in \mathbb{R}^{N \times D}$ is often rank deficient.

For example,

- If $D > N$ (namely over-parameterized), we always have $rank(\mathbf{X}) < D$ (since row rank = col. rank)
- If $D \leq N$, but some of the columns $\mathbf{x}_{:d}$ are (nearly) collinear $\Rightarrow$ $\mathbf{X}$ is ill-conditioned, leading to numerical issues when solving the linear system.

Rank deficiency and ill-conditioning

Unfortunately, in practice, $\mathbf{X} \in \mathbb{R}^{N \times D}$ is often rank deficient.

For example,

- If $D > N$ (namely over-parameterized), we always have $rank(\mathbf{X}) < D$ (since row rank = col. rank)
- If $D \leq N$, but some of the columns $\mathbf{x}_{:d}$ are (nearly) collinear $\Rightarrow \mathbf{X}$ is ill-conditioned, leading to numerical issues when solving the linear system.

Can we solve least squares if $\mathbf{X}$ is rank deficient?

Rank deficiency and ill-conditioning

Unfortunately, in practice, $\mathbf{X} \in \mathbb{R}^{N \times D}$ is often rank deficient.

For example,

- If $D > N$ (namely over-parameterized), we always have $rank(\mathbf{X}) < D$ (since row rank = col. rank)
- If $D \leq N$, but some of the columns $\mathbf{x}_{:d}$ are (nearly) collinear $\Rightarrow$ $\mathbf{X}$ is ill-conditioned, leading to numerical issues when solving the linear system.

Can we solve least squares if $\mathbf{X}$ is rank deficient?
Yes, using a linear system solver, e.g., $\mathrm{np.linalg.solve}(\mathbf{X}, \mathbf{y})$.

Table of Contents

Recall: Gaussian distribution and independence

Definition 4 (A Gaussian random variable)

The definition of a Gaussian random variable in $\mathbb{R}$ with mean μ and variance σ^2. It has a density of

Recall: Gaussian distribution and independence

Definition 4 (A Gaussian random variable)

The definition of a Gaussian random variable in $\mathbb{R}$ with mean μ and variance σ^2. It has a density of

$$p(y \mid \mu, \sigma^2) = \mathcal{N}(y \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(y-\mu)^2}{2\sigma^2}\right\}. \tag{15}$$

Recall: Gaussian distribution and independence

Definition 4 (A Gaussian random variable)

The definition of a Gaussian random variable in $\mathbb{R}$ with mean μ and variance σ^2. It has a density of

$$p(y \mid \mu, \sigma^2) = \mathcal{N}(y \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(y-\mu)^2}{2\sigma^2} \right\}. \tag{15}$$

Definition 5 (The density of a Gaussian random vector)

The density of a Gaussian random vector with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ (which must be a positive semi-definite matrix) is

Recall: Gaussian distribution and independence

Definition 4 (A Gaussian random variable)

The definition of a Gaussian random variable in $\mathbb{R}$ with mean μ and variance σ^2. It has a density of

$$p(y \,|\, \mu, \sigma^2) = \mathcal{N}(y \,|\, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(y-\mu)^2}{2\sigma^2} \right\}. \tag{15}$$

Definition 5 (The density of a Gaussian random vector)

The density of a Gaussian random vector with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ (which must be a positive semi-definite matrix) is

$$\mathcal{N}(\mathbf{y} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D \, det(\boldsymbol{\Sigma})}} \exp\left\{ -\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{y}-\boldsymbol{\mu}) \right\}. \tag{16}$$

Recall: Gaussian distribution and independence

Definition 4 (A Gaussian random variable)

The definition of a Gaussian random variable in $\mathbb{R}$ with mean μ and variance σ^2. It has a density of

$$p(y \mid \mu, \sigma^2) = \mathcal{N}(y \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(y-\mu)^2}{2\sigma^2} \right\}. \tag{15}$$

Definition 5 (The density of a Gaussian random vector)

The density of a Gaussian random vector with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ (which must be a positive semi-definite matrix) is

$$\mathcal{N}(\mathbf{y} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D \, det(\boldsymbol{\Sigma})}} \exp\left\{ -\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu}) \right\}. \tag{16}$$

Two random variables X and Y are called *independent* when $p(x, y) = p(x)p(y)$.

A probabilistic model for linear regression

Definition 6 (Data generation process)

We assume that the data is generated by the model,

$$y_n = \mathbf{x}_n^\top \mathbf{w} + \epsilon_n, \tag{17}$$

where

A probabilistic model for linear regression

Definition 6 (Data generation process)

We assume that the data is generated by the model,

$$y_n = \mathbf{x}_n^\top \mathbf{w} + \epsilon_n, \tag{17}$$

where

- the ϵ_n (the noise) is a zero-mean Gaussian random variable with variance σ^2

A probabilistic model for linear regression

Definition 6 (Data generation process)

We assume that the data is generated by the model,

$$y_n = \mathbf{x}_n^\top \mathbf{w} + \epsilon_n, \tag{17}$$

where

- the ϵ_n (the noise) is a zero-mean Gaussian random variable with variance σ^2
- the noise is independent of each other and independent of the input.

A probabilistic model for linear regression

Definition 6 (Data generation process)

We assume that the data is generated by the model,

$$y_n = \mathbf{x}_n^\top \mathbf{w} + \epsilon_n, \tag{17}$$

where

- the ϵ_n (the noise) is a zero-mean Gaussian random variable with variance σ^2
- the noise is independent of each other and independent of the input.
- the model $\mathbf{w}$ is unknown.

A probabilistic model for linear regression

Definition 6 (Data generation process)

We assume that the data is generated by the model,

$$y_n = \mathbf{x}_n^\top \mathbf{w} + \epsilon_n, \tag{17}$$

where

- the ϵ_n (the noise) is a zero-mean Gaussian random variable with variance σ^2
- the noise is independent of each other and independent of the input.
- the model $\mathbf{w}$ is unknown.

The likelihood of the data vector $\mathbf{y} = (y_1, \cdots, y_N)$ given the input $\mathbf{X}$ and the model $\mathbf{w}$ is

A probabilistic model for linear regression

Definition 6 (Data generation process)

We assume that the data is generated by the model,

$$y_n = \mathbf{x}_n^\top \mathbf{w} + \epsilon_n, \tag{17}$$

where

- the ϵ_n (the noise) is a zero-mean Gaussian random variable with variance σ^2
- the noise is independent of each other and independent of the input.
- the model $\mathbf{w}$ is unknown.

The likelihood of the data vector $\mathbf{y} = (y_1, \cdots, y_N)$ given the input $\mathbf{X}$ and the model $\mathbf{w}$ is

$$p(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} p(y_n \mid \mathbf{x}_n, \mathbf{w}) = \prod_{n=1}^{N} \mathcal{N}(y_n \mid \mathbf{x}_n^\top \mathbf{w}, \sigma^2). \tag{18}$$

A probabilistic model for linear regression

Definition 6 (Data generation process)

We assume that the data is generated by the model,

$$y_n = \mathbf{x}_n^\top \mathbf{w} + \epsilon_n, \tag{17}$$

where

- the ϵ_n (the noise) is a zero-mean Gaussian random variable with variance σ^2
- the noise is independent of each other and independent of the input.
- the model $\mathbf{w}$ is unknown.

The likelihood of the data vector $\mathbf{y} = (y_1, \cdots, y_N)$ given the input $\mathbf{X}$ and the model $\mathbf{w}$ is

$$p(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} p(y_n \mid \mathbf{x}_n, \mathbf{w}) = \prod_{n=1}^{N} \mathcal{N}(y_n \mid \mathbf{x}_n^\top \mathbf{w}, \sigma^2). \tag{18}$$

The probabilistic view point: maximize this likelihood over the choice of model $\mathbf{w}$.

Maximum-likelihood estimator (MLE)

Instead of maximizing the likelihood, we can maximize the logarithm of the likelihood:

$$\mathcal{L}_{\mathsf{LL}}(\mathbf{w}) := \log p(\mathbf{y} \,|\, \mathbf{X}, \mathbf{w}) = -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\top \mathbf{w})^2 + \text{cnst}. \qquad \text{(log-likelihood (LL))}$$

Maximum-likelihood estimator (MLE)

Instead of maximizing the likelihood, we can maximize the logarithm of the likelihood:

$$\mathcal{L}_{\mathsf{LL}}(\mathbf{w}) := \log p(\mathbf{y} \,|\, \mathbf{X}, \mathbf{w}) = -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\top \mathbf{w})^2 + \text{cnst}. \qquad \text{(log-likelihood (LL))}$$

Compare the LL to the MSE (Mean Squared Error)

$$\mathcal{L}_{\mathsf{LL}}(\mathbf{w}) = -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\top \mathbf{w})^2 + \text{cnst} \tag{19}$$

$$\mathcal{L}_{\mathsf{MSE}}(\mathbf{w}) = \frac{1}{2N} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\top \mathbf{w})^2 \tag{20}$$

Maximum-likelihood estimator (MLE)

Instead of maximizing the likelihood, we can maximize the logarithm of the likelihood:

$$\mathcal{L}_{\text{LL}}(\mathbf{w}) := \log\, p(\mathbf{y}\,|\,\mathbf{X}, \mathbf{w}) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - \mathbf{x}_n^\top \mathbf{w})^2 + \text{cnst.}$$ (log-likelihood (LL))

Compare the LL to the MSE (Mean Squared Error)

$$\mathcal{L}_{\text{LL}}(\mathbf{w}) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - \mathbf{x}_n^\top \mathbf{w})^2 + \text{cnst} \tag{19}$$

$$\mathcal{L}_{\text{MSE}}(\mathbf{w}) = \frac{1}{2N}\sum_{n=1}^{N}(y_n - \mathbf{x}_n^\top \mathbf{w})^2 \tag{20}$$

Maximizing the LL is equivalent to minimizing the MSE:

$$\arg\min_{\mathbf{w}}\ \mathcal{L}_{\text{MSE}}(\mathbf{w}) = \arg\max_{\mathbf{w}}\ \mathcal{L}_{\text{LL}}(\mathbf{w}). \tag{21}$$

Properties of MLE

MLE is a *sample* approximation to the *expected log-likelihood*:

$$\mathcal{L}_{\mathsf{LL}}(\mathbf{w}) \approx \mathbb{E}_{p(y,\mathbf{x})}\left[\log p(y \mid \mathbf{x}, \mathbf{w})\right] \tag{22}$$

Properties of MLE

MLE is a *sample* approximation to the *expected log-likelihood*:

$$\mathcal{L}_{\mathsf{LL}}(\mathbf{w}) \approx \mathbb{E}_{p(y,\mathbf{x})}\left[\log p(y \mid \mathbf{x}, \mathbf{w})\right] \tag{22}$$

1. This gives us another way to design cost functions.

Properties of MLE

MLE is a *sample* approximation to the *expected log-likelihood*:

$$\mathcal{L}_{\mathsf{LL}}(\mathbf{w}) \approx \mathbb{E}_{p(y,\mathbf{x})}\left[\log p(y \mid \mathbf{x}, \mathbf{w})\right] \tag{22}$$

1. This gives us another way to design cost functions.
   MLE can also be interpreted as *finding the model under which the observed data is most likely to have been generated from (probabilistically)*.

Properties of MLE

MLE is a *sample* approximation to the *expected log-likelihood*:

$$\mathcal{L}_{\text{LL}}(\mathbf{w}) \approx \mathbb{E}_{p(y,\mathbf{x})}\left[\log p(y \mid \mathbf{x}, \mathbf{w})\right] \tag{22}$$

1. This gives us another way to design cost functions.
   MLE can also be interpreted as *finding the model under which the observed data is most likely to have been generated from (probabilistically)*.

2. MLE is consistent, i.e., it will give us the correct model assuming that we have a sufficient amount of data. (can be proven under some weak conditions)

$$\mathbf{w}_{\text{MLE}} \quad \longrightarrow^{p} \quad \mathbf{w}_{\text{true}} \qquad \text{in probability} \tag{23}$$

Properties of MLE

MLE is a *sample* approximation to the *expected log-likelihood*:

$$\mathcal{L}_{\mathsf{LL}}(\mathbf{w}) \approx \mathbb{E}_{p(y,\mathbf{x})}\left[\log p(y \mid \mathbf{x}, \mathbf{w})\right] \tag{22}$$

1. This gives us another way to design cost functions.
   MLE can also be interpreted as *finding the model under which the observed data is most likely to have been generated from (probabilistically)*.

2. MLE is consistent, i.e., it will give us the correct model assuming that we have a sufficient amount of data. (can be proven under some weak conditions)

$$\mathbf{w}_{\mathsf{MLE}} \quad \longrightarrow^{p} \quad \mathbf{w}_{\mathsf{true}} \qquad \text{in probability} \tag{23}$$

3. The MLE is asymptotically normal, i.e.,

$$(\mathbf{w}_{\mathsf{MLE}} - \mathbf{w}_{\mathsf{true}}) \quad \longrightarrow^{d} \quad \frac{1}{\sqrt{N}}\mathcal{N}\left(\mathbf{w}_{\mathsf{MLE}} \mid \mathbf{0}, \mathbf{F}^{-1}(\mathbf{w}_{\mathsf{true}})\right), \tag{24}$$

where $\mathbf{F}(\mathbf{w}) = -\mathbb{E}_{p(\mathbf{y})}\left[\frac{\partial^2 \mathcal{L}}{\partial \mathbf{w} \partial \mathbf{w}^\top}\right]$ is the Fisher information.

Properties of MLE

MLE is a *sample* approximation to the *expected log-likelihood*:

$$\mathcal{L}_{\text{LL}}(\mathbf{w}) \approx \mathbb{E}_{p(y,\mathbf{x})}\left[\log p(y \mid \mathbf{x}, \mathbf{w})\right] \tag{22}$$

1. This gives us another way to design cost functions.
   MLE can also be interpreted as *finding the model under which the observed data is most likely to have been generated from (probabilistically)*.

2. MLE is consistent, i.e., it will give us the correct model assuming that we have a sufficient amount of data. (can be proven under some weak conditions)

$$\mathbf{w}_{\text{MLE}} \quad \longrightarrow^{p} \quad \mathbf{w}_{\text{true}} \qquad \text{in probability} \tag{23}$$

3. The MLE is asymptotically normal, i.e.,

$$(\mathbf{w}_{\text{MLE}} - \mathbf{w}_{\text{true}}) \quad \longrightarrow^{d} \quad \frac{1}{\sqrt{N}}\mathcal{N}(\mathbf{w}_{\text{MLE}} \mid \mathbf{0}, \mathbf{F}^{-1}(\mathbf{w}_{\text{true}}))\,, \tag{24}$$

   where $\mathbf{F}(\mathbf{w}) = -\mathbb{E}_{p(\mathbf{y})}\left[\frac{\partial^2 \mathcal{L}}{\partial \mathbf{w} \partial \mathbf{w}^\top}\right]$ is the Fisher information.

4. MLE is efficient, i.e. it achieves the Cramer-Rao lower bound.

Another example

What if we replace the Gaussian distribution with a Laplace distribution?

$$p(y_n \mid \mathbf{x}_n, \mathbf{w}) = \frac{1}{2b} e^{-\frac{1}{b}\left|y_n - \mathbf{x}_n^\top \mathbf{w}\right|} \tag{26}$$

Another example

What if we replace the Gaussian distribution with a Laplace distribution?

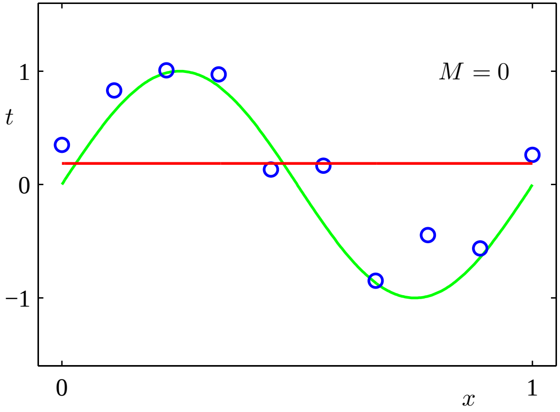
$$p(y_n \mid \mathbf{x}_n, \mathbf{w}) = \frac{1}{2b} e^{-\frac{1}{b}\left|y_n - \mathbf{x}_n^\top \mathbf{w}\right|} \tag{26}$$

We can recover the MAE cost function!

Table of Contents

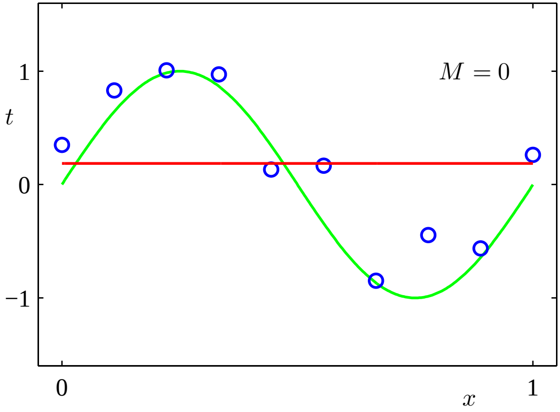
Table of Contents

Under-fitting with Linear Models $f_{\mathbf{w}}(\mathbf{X}) := \mathbf{X}\mathbf{w}$



Settings:

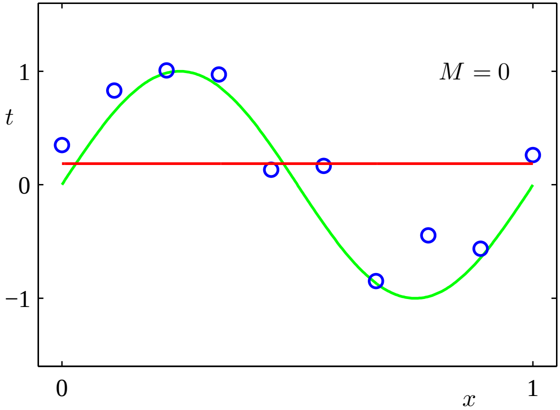
Under-fitting with Linear Models $f_{\mathbf{w}}(\mathbf{X}) := \mathbf{X}\mathbf{w}$



Settings:

- A scalar function $g(x)$

Under-fitting with Linear Models $f_{\mathbf{w}}(\mathbf{X}) := \mathbf{X}\mathbf{w}$

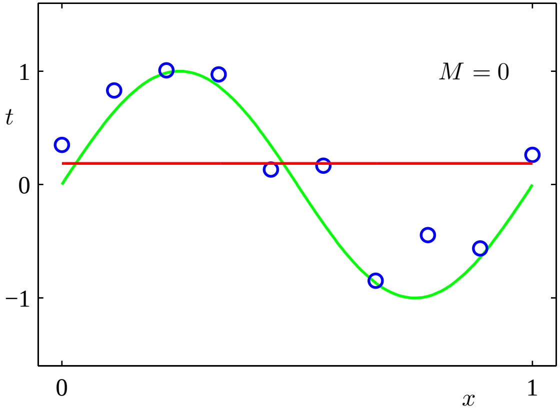


Settings:

- A scalar function $g(x)$
- We do not observe $g(x_n)$ directly:

$$y_n = g(x_n) + Z_n \,. \tag{27}$$

Under-fitting with Linear Models $f_{\mathbf{w}}(\mathbf{X}) := \mathbf{X}\mathbf{w}$



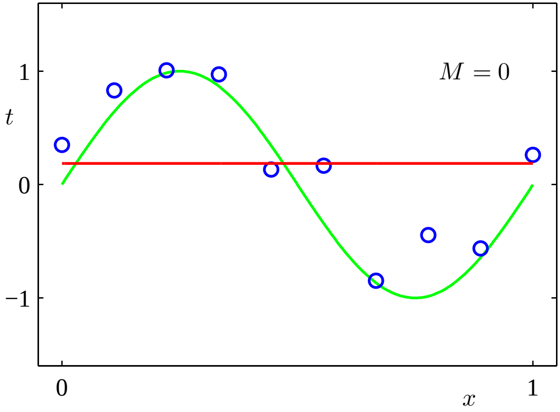
Settings:

- A scalar function $g(x)$
- We do not observe $g(x_n)$ directly:

$$y_n = g(x_n) + Z_n \,. \tag{27}$$

- The noise Z_n might be due to some measurement inaccuracies.

Under-fitting with Linear Models $f_{\mathbf{w}}(\mathbf{X}) := \mathbf{X}\mathbf{w}$



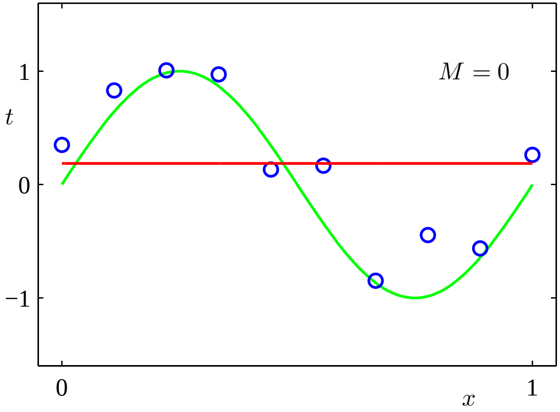
Settings:

- A scalar function $g(x)$
- We do not observe $g(x_n)$ directly:

$$y_n = g(x_n) + Z_n \,. \tag{27}$$

- The noise Z_n might be due to some measurement inaccuracies.
- The y_n are shown as blue circles.

Under-fitting with Linear Models $f_{\mathbf{w}}(\mathbf{X}) := \mathbf{X}\mathbf{w}$



Settings:

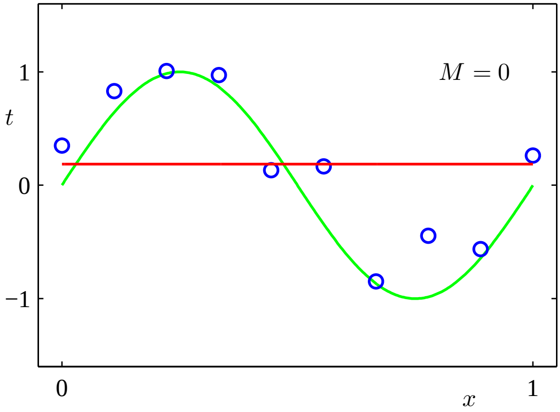
- A scalar function $g(x)$
- We do not observe $g(x_n)$ directly:

$$y_n = g(x_n) + Z_n \,. \tag{27}$$

- The noise Z_n might be due to some measurement inaccuracies.
- The y_n are shown as blue circles.
- Our model family consists of only linear functions of the scalar input x:

$$\mathcal{F} = \{f_w(x) = xw\} \,. \tag{28}$$

Under-fitting with Linear Models $f_{\mathbf{w}}(\mathbf{X}) := \mathbf{X}\mathbf{w}$



Observations:

Settings:

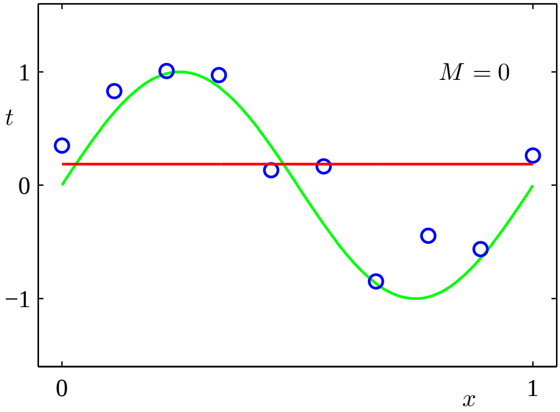
- A scalar function $g(x)$
- We do not observe $g(x_n)$ directly:

$$y_n = g(x_n) + Z_n \,. \tag{27}$$

- The noise Z_n might be due to some measurement inaccuracies.
- The y_n are shown as blue circles.
- Our model family consists of only linear functions of the scalar input x:

$$\mathcal{F} = \{f_w(x) = xw\} \,. \tag{28}$$

Under-fitting with Linear Models $f_{\mathbf{w}}(\mathbf{X}) := \mathbf{X}\mathbf{w}$



Settings:

- A scalar function $g(x)$
- We do not observe $g(x_n)$ directly:

$$y_n = g(x_n) + Z_n \,. \tag{27}$$

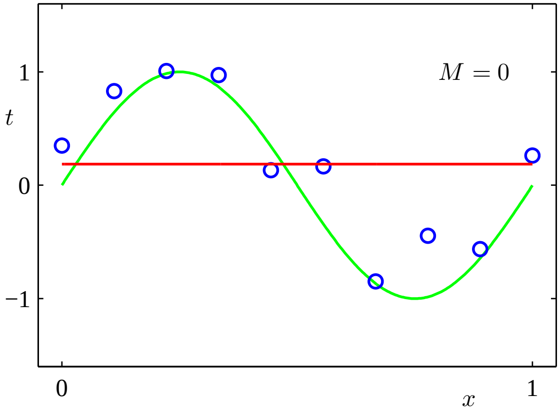
- The noise Z_n might be due to some measurement inaccuracies.
- The y_n are shown as blue circles.
- Our model family consists of only linear functions of the scalar input x:

$$\mathcal{F} = \{f_w(x) = xw\} \,. \tag{28}$$

Observations:

- The solid curve is the underlying function (the slope of the function).

Under-fitting with Linear Models $f_{\mathbf{w}}(\mathbf{X}) := \mathbf{X}\mathbf{w}$



Settings:

- A scalar function $g(x)$
- We do not observe $g(x_n)$ directly:

$$y_n = g(x_n) + Z_n \,. \tag{27}$$

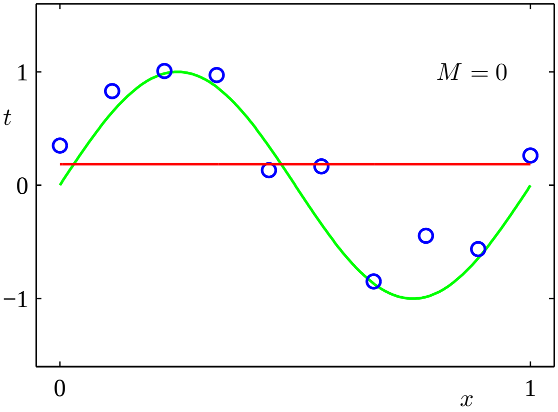
- The noise Z_n might be due to some measurement inaccuracies.
- The y_n are shown as blue circles.
- Our model family consists of only linear functions of the scalar input x:

$$\mathcal{F} = \{f_w(x) = xw\} \,. \tag{28}$$

Observations:

- The solid curve is the underlying function (the slope of the function).
- We cannot match the given function accurately,

Under-fitting with Linear Models $f_{\mathbf{w}}(\mathbf{X}) := \mathbf{X}\mathbf{w}$



Settings:

- A scalar function $g(x)$
- We do not observe $g(x_n)$ directly:

$$y_n = g(x_n) + Z_n \,. \tag{27}$$

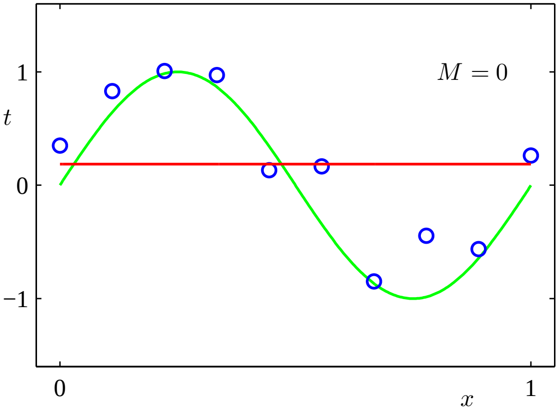
- The noise Z_n might be due to some measurement inaccuracies.
- The y_n are shown as blue circles.
- Our model family consists of only linear functions of the scalar input x:

$$\mathcal{F} = \{f_w(x) = xw\} \,. \tag{28}$$

Observations:

- The solid curve is the underlying function (the slope of the function).
- We cannot match the given function accurately,
  regardless of how many samples we get and how small the noise is.

Under-fitting with Linear Models $f_{\mathbf{w}}(\mathbf{X}) := \mathbf{X}\mathbf{w}$



Settings:

- A scalar function $g(x)$
- We do not observe $g(x_n)$ directly:

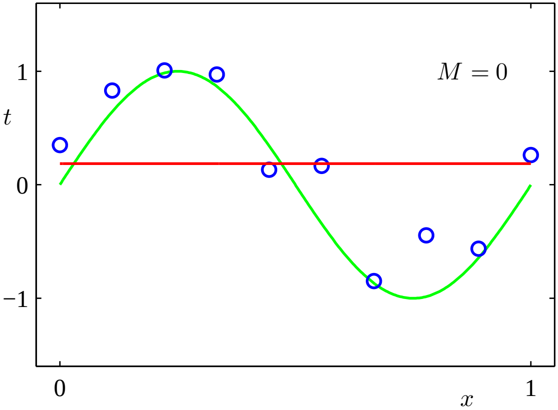
$$y_n = g(x_n) + Z_n \,. \tag{27}$$

- The noise Z_n might be due to some measurement inaccuracies.
- The y_n are shown as blue circles.
- Our model family consists of only linear functions of the scalar input x:

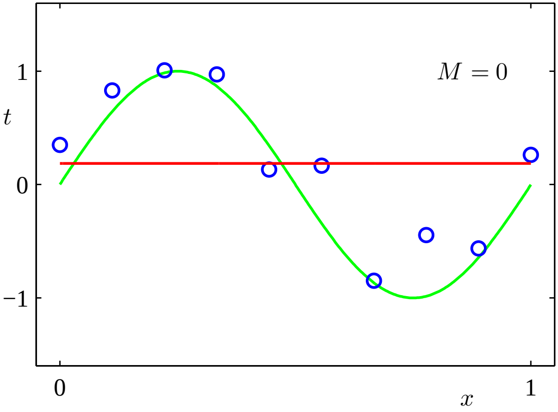
$$\mathcal{F} = \{f_w(x) = xw\} \,. \tag{28}$$

Linear Model might under-fit!

Table of Contents



Linear Model might under-fit!



~~Linear Model might under-fit!~~

Extended/Augmented feature vectors

We can increase the representational power of linear models by “augmenting” the input!

Extended/Augmented feature vectors

We can increase the representational power of linear models by "augmenting" the input!
Using a one-dimensional input (feature) x_n as an example:

Extended/Augmented feature vectors

We can increase the representational power of linear models by "augmenting" the input!
Using a one-dimensional input (feature) x_n as an example:

- Instead of only using the input feature x_n,

Extended/Augmented feature vectors

We can increase the representational power of linear models by "augmenting" the input!
Using a one-dimensional input (feature) x_n as an example:

- Instead of only using the input feature x_n,
- we might add a polynomial basis to get an extended feature vector $\phi(x_n)$, i.e.,

$$\phi(x_n) := \left[1,\, x_n,\, x_n^2,\, x_n^3,\, \ldots,\, x_n^M\right] \,, \tag{29}$$

Extended/Augmented feature vectors

We can increase the representational power of linear models by "augmenting" the input!
Using a one-dimensional input (feature) x_n as an example:

- Instead of only using the input feature x_n,
- we might add a polynomial basis to get an extended feature vector $\phi(x_n)$, i.e.,

$$\phi(x_n) := \left[1,\, x_n,\, x_n^2,\, x_n^3,\, \ldots,\, x_n^M\right] , \tag{29}$$

  where M is of arbitrary degree.

Extended/Augmented feature vectors

We can increase the representational power of linear models by “augmenting” the input!
Using a one-dimensional input (feature) x_n as an example:

- Instead of only using the input feature x_n,
- we might add a polynomial basis to get an extended feature vector $\phi(x_n)$, i.e.,

$$\phi(x_n) := \left[1,\, x_n,\, x_n^2,\, x_n^3,\, \ldots,\, x_n^M\right] \,, \tag{29}$$

  where M is of arbitrary degree.
- We then fit a linear model to this extended feature vector $\phi(x_n)$:

Extended/Augmented feature vectors

We can increase the representational power of linear models by “augmenting” the input!
Using a one-dimensional input (feature) x_n as an example:

- Instead of only using the input feature x_n,
- we might add a polynomial basis to get an extended feature vector $\phi(x_n)$, i.e.,

$$\phi(x_n) := \left[1,\, x_n,\, x_n^2,\, x_n^3,\, \ldots,\, x_n^M\right] , \tag{29}$$

  where M is of arbitrary degree.
- We then fit a linear model to this extended feature vector $\phi(x_n)$:

$$\begin{aligned} y_n &\approx w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_M x_n^M \qquad (30) \\ &=: \phi(x_n)^\top \mathbf{w} \,. \qquad (31) \end{aligned}$$

Extended/Augmented feature vectors

We can increase the representational power of linear models by "augmenting" the input!
Using a one-dimensional input (feature) x_n as an example:

- Instead of only using the input feature x_n,
- we might add a polynomial basis to get an extended feature vector $\phi(x_n)$, i.e.,

$$\phi(x_n) := \left[1,\, x_n,\, x_n^2,\, x_n^3,\, \dots,\, x_n^M\right] \,, \tag{29}$$

  where M is of arbitrary degree.
- We then fit a linear model to this extended feature vector $\phi(x_n)$:

$$y_n \approx w_0 + w_1 x_n + w_2 x_n^2 + \dots + w_M x_n^M \tag{30}$$

$$=: \phi(x_n)^\top \mathbf{w} \,. \tag{31}$$

Is it all good?

Table of Contents

Interpolation between under-fitting and over-fitting

Let’s consider the polynomial regression problem (for a one-dimensional input feature x_n):

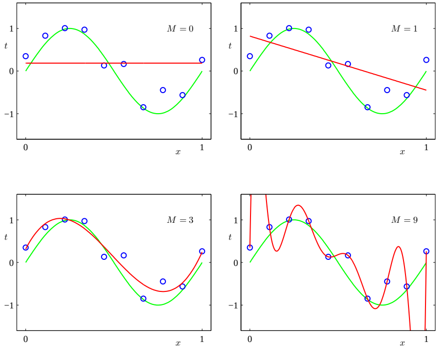
$$y_n \approx w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_M x_n^M =: \phi(x_n)^\top \mathbf{w} \,. \tag{32}$$

Interpolation between under-fitting and over-fitting

Let's consider the polynomial regression problem (for a one-dimensional input feature x_n):

$$y_n \approx w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_M x_n^M =: \phi(x_n)^\top \mathbf{w} \,. \tag{32}$$

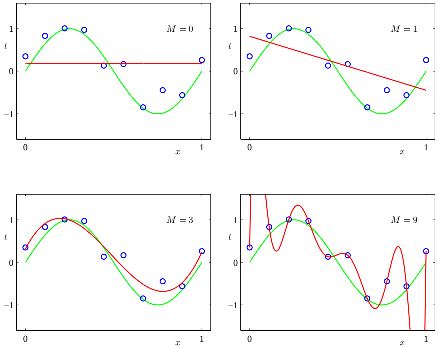
Settings:



Interpolation between under-fitting and over-fitting

Let's consider the polynomial regression problem (for a one-dimensional input feature x_n):

$$y_n \approx w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_M x_n^M =: \phi(x_n)^\top \mathbf{w}\,. \tag{32}$$



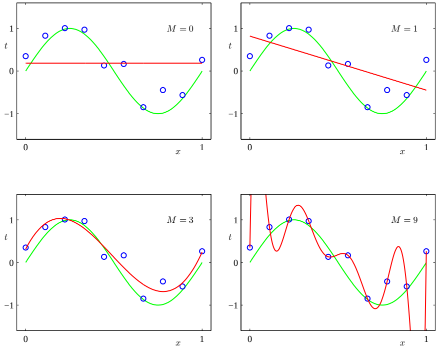
Settings:

- The circles are data points

Interpolation between under-fitting and over-fitting

Let's consider the polynomial regression problem (for a one-dimensional input feature x_n):

$$y_n \approx w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_M x_n^M =: \phi(x_n)^\top \mathbf{w}\,. \tag{32}$$



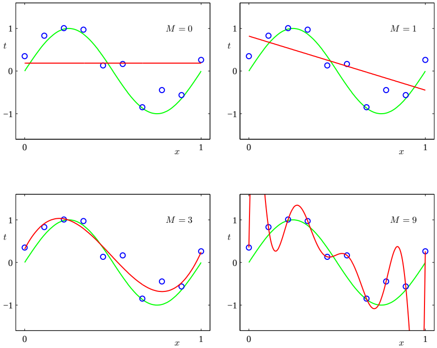
Settings:

- The circles are data points
- The green line represents the "true function"

Interpolation between under-fitting and over-fitting

Let's consider the polynomial regression problem (for a one-dimensional input feature x_n):

$$y_n \approx w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_M x_n^M =: \phi(x_n)^\top \mathbf{w}\,. \tag{32}$$



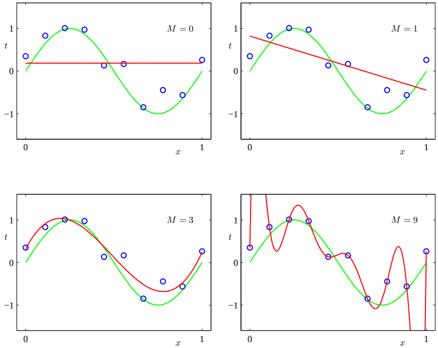
Settings:

- The circles are data points
- The green line represents the "true function"
- The red line is the learned model.

Interpolation between under-fitting and over-fitting

Let's consider the polynomial regression problem (for a one-dimensional input feature x_n):

$$y_n \approx w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_M x_n^M =: \phi(x_n)^\top \mathbf{w} \,. \tag{32}$$



Settings:

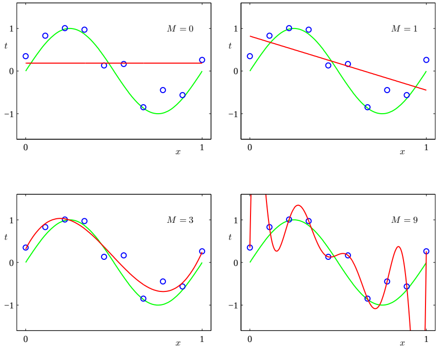
- The circles are data points
- The green line represents the "true function"
- The red line is the learned model.

Observations:

Interpolation between under-fitting and over-fitting

Let's consider the polynomial regression problem (for a one-dimensional input feature x_n):

$$y_n \approx w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_M x_n^M =: \phi(x_n)^\top \mathbf{w} \,. \tag{32}$$



Settings:

- The circles are data points
- The green line represents the "true function"
- The red line is the learned model.

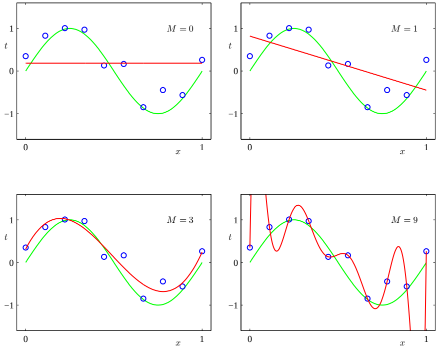
Observations:

- The model is under-fitting for $M = 0$ & $M = 1$

Interpolation between under-fitting and over-fitting

Let's consider the polynomial regression problem (for a one-dimensional input feature x_n):

$$y_n \approx w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_M x_n^M =: \phi(x_n)^\top \mathbf{w} \,. \tag{32}$$



Settings:

- The circles are data points
- The green line represents the "true function"
- The red line is the learned model.

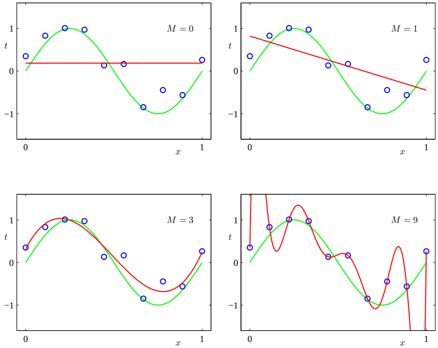
Observations:

- The model is under-fitting for $M = 0$ & $M = 1$
- For $M = 3$, the model fits the data

Interpolation between under-fitting and over-fitting

Let's consider the polynomial regression problem (for a one-dimensional input feature x_n):

$$y_n \approx w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_M x_n^M =: \phi(x_n)^\top \mathbf{w}\,. \tag{32}$$



Settings:

- The circles are data points
- The green line represents the "true function"
- The red line is the learned model.

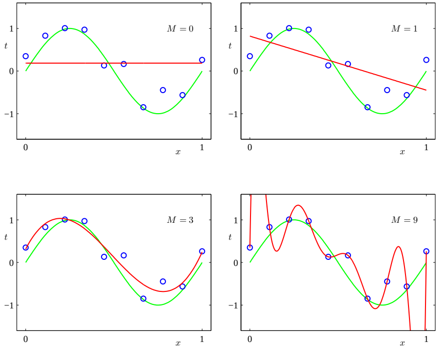
Observations:

- The model is under-fitting for $M = 0$ & $M = 1$
- For $M = 3$, the model fits the data $\rightarrow$ but can be improved

Interpolation between under-fitting and over-fitting

Let's consider the polynomial regression problem (for a one-dimensional input feature x_n):

$$y_n \approx w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_M x_n^M =: \phi(x_n)^\top \mathbf{w} \,. \tag{32}$$



Settings:

- The circles are data points
- The green line represents the “true function”
- The red line is the learned model.

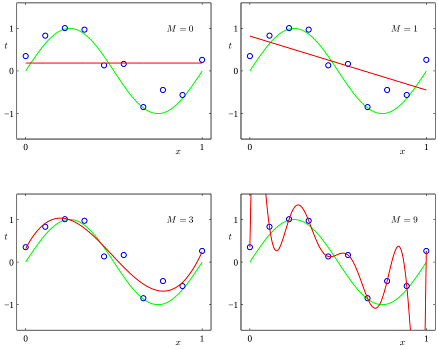
Observations:

- The model is under-fitting for $M = 0$ & $M = 1$
- For $M = 3$, the model fits the data
  $\to$ but can be improved
- For $M = 9$, the model fits every single points

Interpolation between under-fitting and over-fitting

Let's consider the polynomial regression problem (for a one-dimensional input feature x_n):

$$y_n \approx w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_M x_n^M =: \phi(x_n)^\top \mathbf{w}\,. \tag{32}$$



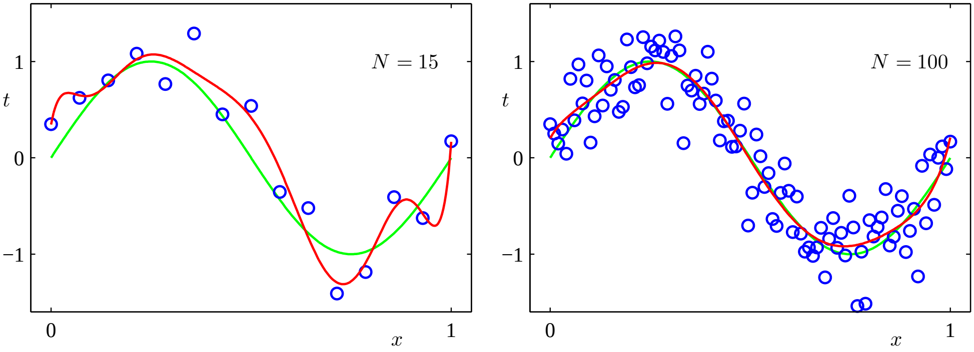
Settings:

- The circles are data points
- The green line represents the "true function"
- The red line is the learned model.

Observations:

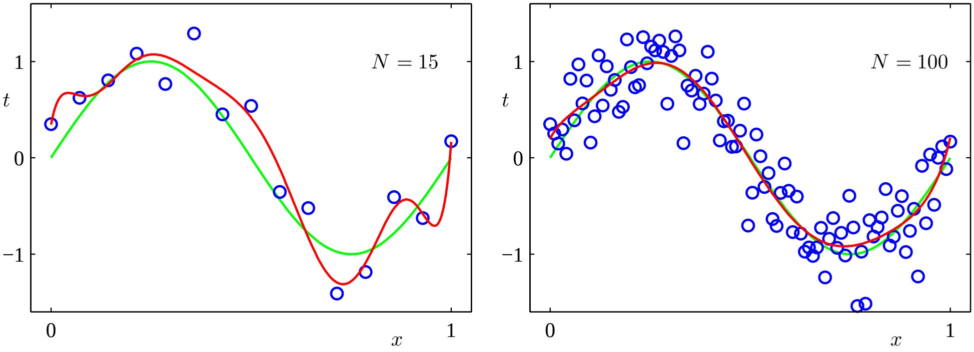
- The model is under-fitting for $M = 0$ & $M = 1$
- For $M = 3$, the model fits the data $\rightarrow$ but can be improved
- For $M = 9$, the model fits every single points $\rightarrow$ severe over-fitting

- **Question:** How to avoid over-fitting?



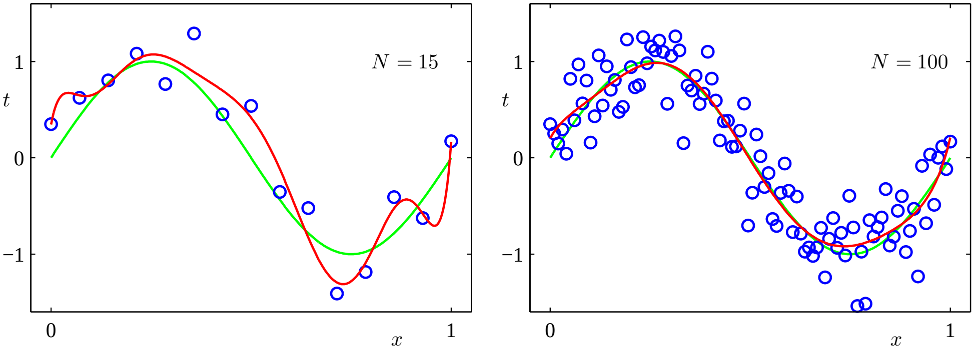
Increasing N but keeping M fixed.

- **Question:** How to avoid over-fitting?
- **Potential solution:** Increasing the amount of data might reduce over-fitting.



Increasing N but keeping M fixed.

- **Question:** How to avoid over-fitting?
- **Potential solution:** Increasing the amount of data might reduce over-fitting.



Increasing N but keeping M fixed.

We will elaborate on this question later!

Definitions of under-fit and over-fit

Definitions of under-fit and over-fit

- under-fit: cannot find the *the underlying function* of the data.

Definitions of under-fit and over-fit

- under-fit: cannot find the *the underlying function* of the data.
- over-fit: can fit both *the underlying function* and *the noise in the data*.

Definitions of under-fit and over-fit

- under-fit: cannot find the *the underlying function* of the data.
- over-fit: can fit both *the underlying function* and *the noise in the data*.

Discussion:

- Both of these phenomena (under-fit and over-fit) are undesirable.

Definitions of under-fit and over-fit

- under-fit: cannot find the *the underlying function* of the data.
- over-fit: can fit both *the underlying function* and *the noise in the data*.

Discussion:

- Both of these phenomena (under-fit and over-fit) are undesirable.
- This discussion is made more difficult:

Definitions of under-fit and over-fit

- under-fit: cannot find the *the underlying function* of the data.
- over-fit: can fit both *the underlying function* and *the noise in the data*.

Discussion:

- Both of these phenomena (under-fit and over-fit) are undesirable.
- This discussion is made more difficult:
  - since all we have is data

Definitions of under-fit and over-fit

- under-fit: cannot find the *the underlying function* of the data.
- over-fit: can fit both *the underlying function* and *the noise in the data*.

Discussion:

- Both of these phenomena (under-fit and over-fit) are undesirable.
- This discussion is made more difficult:
  - since all we have is data
  - we do not know a priori what part is the underlying signal and what part is noise.

Table of Contents

We have seen that by augmenting the feature vector:

$$y_n \approx w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_M x_n^M =: \phi(x_n)^\top \mathbf{w}\,, \tag{33}$$

we can increase the representation power (or complexity) of the linear model.

We have seen that by augmenting the feature vector:

$$y_n \approx w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_M x_n^M =: \phi(x_n)^\top \mathbf{w}\,, \tag{33}$$

we can increase the representation power (or complexity) of the linear model.

The issue of over-fitting!

We have seen that by augmenting the feature vector:

$$y_n \approx w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_M x_n^M =: \phi(x_n)^\top \mathbf{w}\,, \tag{33}$$

we can increase the representation power (or complexity) of the linear model.

The issue of over-fitting!

Regularization is a way to mitigate this undesirable behavior.

We have seen that by augmenting the feature vector:

$$y_n \approx w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_M x_n^M =: \phi(x_n)^\top \mathbf{w}\,, \tag{33}$$

we can increase the representation power (or complexity) of the linear model.

The issue of over-fitting!

Regularization is a way to mitigate this undesirable behavior.

- We will discuss regularization in the context of linear models

We have seen that by augmenting the feature vector:

$$y_n \approx w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_M x_n^M =: \phi(x_n)^\top \mathbf{w}\,, \tag{33}$$

we can increase the representation power (or complexity) of the linear model.

The issue of over-fitting!

Regularization is a way to mitigate this undesirable behavior.

- We will discuss regularization in the context of linear models
- The same principle applies also to more complex models such as neural nets.

Regularization

Through regularization, we can penalize complex models and favor simpler ones:

$$\min_{\mathbf{w}} \quad \mathcal{L}(\mathbf{w}) + \boxed{\Omega(\mathbf{w})} \tag{34}$$

Regularization

Through regularization, we can penalize complex models and favor simpler ones:

$$\min_{\mathbf{w}} \quad \mathcal{L}(\mathbf{w}) + \boxed{\Omega(\mathbf{w})} \tag{34}$$

- Ω is a regularizer.

Regularization

Through regularization, we can penalize complex models and favor simpler ones:

$$\min_{\mathbf{w}} \quad \mathcal{L}(\mathbf{w}) + \Omega(\mathbf{w}) \tag{34}$$

- Ω is a regularizer.
- Ω measures the complexity of the model given by $\mathbf{w}$.

Regularization

Through regularization, we can penalize complex models and favor simpler ones:

$$\min_{\mathbf{w}} \quad \mathcal{L}(\mathbf{w}) + \boxed{\Omega(\mathbf{w})} \tag{34}$$

- Ω is a regularizer.
- Ω measures the complexity of the model given by $\mathbf{w}$.
- Model Complexity $\Longleftrightarrow$ The richness of the model space.

Table of Contents

L_2-regularization: Ridge Regression

The most frequently used regularizer is the standard Euclidean norm (L_2-norm), which is

$$\Omega(\mathbf{w}) = \lambda \left\|\mathbf{w}\right\|_2^2 \qquad \text{where } \left\|\mathbf{w}\right\|_2^2 = \sum_i w_i^2 \ . \tag{35}$$

L_2-regularization: Ridge Regression

The most frequently used regularizer is the standard Euclidean norm (L_2-norm), which is

$$\Omega(\mathbf{w}) = \lambda \|\mathbf{w}\|_2^2 \qquad \text{where } \|\mathbf{w}\|_2^2 = \sum_i w_i^2 \ . \tag{35}$$

- The main effect: large model weights w_i will be penalized (avoided).

L_2-regularization: Ridge Regression

The most frequently used regularizer is the standard Euclidean norm (L_2-norm), which is

$$\Omega(\mathbf{w}) = \lambda \|\mathbf{w}\|_2^2 \qquad \text{where } \|\mathbf{w}\|_2^2 = \sum_i w_i^2 \ . \tag{35}$$

- The main effect: large model weights w_i will be penalized (avoided).
- It encourages weight values to decay toward zero unless supported by the data.

L_2-regularization: Ridge Regression

The most frequently used regularizer is the standard Euclidean norm (L_2-norm), which is

$$\Omega(\mathbf{w}) = \lambda \|\mathbf{w}\|_2^2 \qquad \text{where } \|\mathbf{w}\|_2^2 = \sum_i w_i^2 \; . \tag{35}$$

- The main effect: large model weights w_i will be penalized (avoided).
- It encourages weight values to decay toward zero unless supported by the data.
- When $\mathcal{L}$ is MSE, we can define the ridge regression:

$$\min_{\mathbf{w}} \quad \frac{1}{2N} \sum_{n=1}^{N} \left[y_n - \mathbf{x}_n^\top \mathbf{w} \right]^2 \; + \; \lambda \|\mathbf{w}\|_2^2 \tag{36}$$

L_2-regularization: Ridge Regression

The most frequently used regularizer is the standard Euclidean norm (L_2-norm), which is

$$\Omega(\mathbf{w}) = \lambda \|\mathbf{w}\|_2^2 \qquad \text{where } \|\mathbf{w}\|_2^2 = \sum_i w_i^2 \ . \tag{35}$$

- The main effect: large model weights w_i will be penalized (avoided).
- It encourages weight values to decay toward zero unless supported by the data.
- When $\mathcal{L}$ is MSE, we can define the ridge regression:

$$\min_{\mathbf{w}} \quad \frac{1}{2N} \sum_{n=1}^{N} \left[y_n - \mathbf{x}_n^\top \mathbf{w}\right]^2 \ + \ \lambda \|\mathbf{w}\|_2^2 \tag{36}$$

- *Linear Regression* is a special case of this: by setting $\lambda := 0$.

Explicit solution of Ridge Regression for $\mathbf{w}$

Ridge Regression can be defined as:

$$\min_{\mathbf{w}} \quad \frac{1}{2N} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \; + \; \lambda \|\mathbf{w}\|_2^2 \; . \tag{37}$$

Explicit solution of Ridge Regression for $\mathbf{w}$

Ridge Regression can be defined as:

$$\min_{\mathbf{w}} \quad \frac{1}{2N} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \; + \; \lambda \|\mathbf{w}\|_2^2 \; . \tag{37}$$

Differentiating and setting to zero (following a similar procedure as least squares):

Explicit solution of Ridge Regression for $\mathbf{w}$

Ridge Regression can be defined as:

$$\min_{\mathbf{w}} \quad \frac{1}{2N} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \; + \; \lambda \|\mathbf{w}\|_2^2 \, . \tag{37}$$

Differentiating and setting to zero (following a similar procedure as least squares):

$$\mathbf{w}^{\star}_{\text{ridge}} = (\mathbf{X}^{\top}\mathbf{X} + \lambda'\mathbf{I})^{-1}\mathbf{X}^{\top}\mathbf{y} \tag{38}$$

(here for simpler notation $\lambda'/2N = \lambda$)

Ridge Regression to fight ill-conditioning

Ridge Regression to fight ill-conditioning

Lemma 7 (Lifting the eigenvalues)

The eigenvalues of $(\mathbf{X}^\top \mathbf{X} + \lambda' \mathbf{I})$ *are all at least* λ' *and so the inverse always exists.*

Ridge Regression to fight ill-conditioning

Lemma 7 (Lifting the eigenvalues)

The eigenvalues of $(\mathbf{X}^\top\mathbf{X} + \lambda'\mathbf{I})$ *are all at least* λ' *and so the inverse always exists.*

Proof.

Write the Eigenvalue decomposition of $\mathbf{X}^\top\mathbf{X}$ as $\mathbf{USU}^\top$.

$\square$

Ridge Regression to fight ill-conditioning

Lemma 7 (Lifting the eigenvalues)

The eigenvalues of $(\mathbf{X}^\top \mathbf{X} + \lambda' \mathbf{I})$ *are all at least* λ' *and so the inverse always exists.*

Proof.

Write the Eigenvalue decomposition of $\mathbf{X}^\top \mathbf{X}$ as $\mathbf{U}\mathbf{S}\mathbf{U}^\top$. We then have

$$\mathbf{X}^\top \mathbf{X} + \lambda' \mathbf{I} = \mathbf{U}\mathbf{S}\mathbf{U}^\top + \lambda' \mathbf{U}\mathbf{I}\mathbf{U}^\top \tag{39}$$

$$= \mathbf{U}[\mathbf{S} + \lambda' \mathbf{I}]\mathbf{U}^\top. \tag{40}$$

□

Ridge Regression to fight ill-conditioning

Lemma 7 (Lifting the eigenvalues)

The eigenvalues of $(\mathbf{X}^\top\mathbf{X} + \lambda'\mathbf{I})$ *are all at least* λ' *and so the inverse always exists.*

Proof.

Write the Eigenvalue decomposition of $\mathbf{X}^\top\mathbf{X}$ as $\mathbf{U}\mathbf{S}\mathbf{U}^\top$. We then have

$$\mathbf{X}^\top\mathbf{X} + \lambda'\mathbf{I} = \mathbf{U}\mathbf{S}\mathbf{U}^\top + \lambda'\mathbf{U}\mathbf{I}\mathbf{U}^\top \tag{39}$$

$$= \mathbf{U}[\mathbf{S} + \lambda'\mathbf{I}]\mathbf{U}^\top. \tag{40}$$

We see now that every Eigenvalue is “lifted” by an amount λ'. □

An alternative proof (optional reading).

Recall that for a symmetric matrix $\mathbf{A}$ we can also compute eigenvalues by looking at the so-called Rayleigh ratio,

$$R(\mathbf{A}, \mathbf{v}) = \frac{\mathbf{v}^\top \mathbf{A}\mathbf{v}}{\mathbf{v}^\top \mathbf{v}}. \tag{41}$$

Note that if $\mathbf{v}$ is an eigenvector with eigenvalue λ, then the Rayleigh coefficient indeed gives us λ.

We can find the smallest and largest eigenvalue by minimizing and maximizing this coefficient.

But note that if we apply this to the symmetric matrix $\mathbf{X}^\top\mathbf{X} + \lambda'\mathbf{I}$, then for any vector $\mathbf{v}$ we have

$$\frac{\mathbf{v}^\top(\mathbf{X}^\top\mathbf{X} + \lambda'\mathbf{I})\mathbf{v}}{\mathbf{v}^\top\mathbf{v}} \geq \frac{\lambda'\mathbf{v}^\top\mathbf{v}}{\mathbf{v}^\top\mathbf{v}} = \lambda'\,. \tag{42}$$

$\square$

Table of Contents

L_1-regularization: the Lasso

Previously, we regularize the complexity of the model:

- e.g., L_2-regularization: Ridge Regression

L_1-regularization: the Lasso

Previously, we regularize the complexity of the model:

- e.g., L_2-regularization: Ridge Regression
- What if we consider an alternative complexity measure?

L_1-regularization: the Lasso

Previously, we regularize the complexity of the model:

- e.g., L_2-regularization: Ridge Regression
- What if we consider an alternative complexity measure?
- Just by considering a different norm!

L_1-regularization: the Lasso

Previously, we regularize the complexity of the model:

- e.g., L_2-regularization: Ridge Regression
- What if we consider an alternative complexity measure?
- Just by considering a different norm!
- A very important case is the L_1-norm, leading to L_1-regularization.

L_1-regularization: the Lasso

Previously, we regularize the complexity of the model:

- e.g., L_2-regularization: Ridge Regression
- What if we consider an alternative complexity measure?
- Just by considering a different norm!
- A very important case is the L_1-norm, leading to L_1-regularization.

In combination with the MSE cost function (i.e., L_1-norm), this is known as the Lasso:

$$\min_{\mathbf{w}} \quad \frac{1}{2N}\sum_{n=1}^{N}\left[y_n - \mathbf{x}_n^\top \mathbf{w}\right]^2 + \lambda \left\|\mathbf{w}\right\|_1 \qquad \text{where } \left\|\mathbf{w}\right\|_1 := \sum_i |w_i| \,. \tag{43}$$

Table of Contents

Geometric interpretation for the Ridge Regression

Recall the definition of the Ridge Regression:

$$\min_{\mathbf{w}} \quad \frac{1}{2N} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \;+\; \lambda \|\mathbf{w}\|_2^2 \,. \tag{44}$$

Geometric interpretation for the Ridge Regression

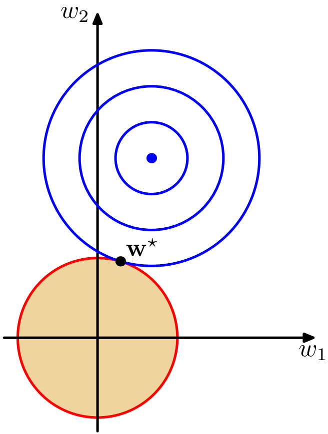
Recall the definition of the Ridge Regression:

$$\min_{\mathbf{w}} \quad \frac{1}{2N} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \; + \; \lambda \|\mathbf{w}\|_2^2 \; . \tag{44}$$

The Ridge Regression can be reformulated as

$$\min_{\mathbf{w}} \quad \frac{1}{2N} \sum_{n=1}^{N} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \quad \text{such that } \|\mathbf{w}\|_2^2 \leq \lambda \tag{45}$$

Geometric interpretation for the Ridge Regression



• Blue lines indicate the level sets of the un-regularized MSE.

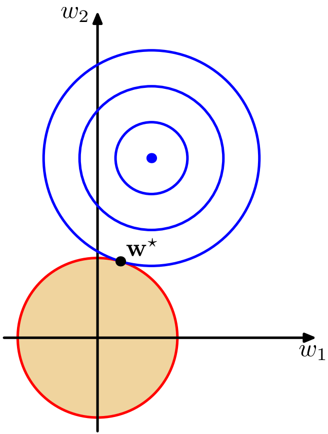
Recall the definition of the Ridge Regression:

$$\min_{\mathbf{w}} \quad \frac{1}{2N} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \; + \; \lambda \|\mathbf{w}\|_2^2 \, . \tag{44}$$

The Ridge Regression can be reformulated as

$$\min_{\mathbf{w}} \quad \frac{1}{2N} \sum_{n=1}^{N} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \quad \text{such that } \|\mathbf{w}\|_2^2 \leq \lambda \tag{45}$$

Geometric interpretation for the Ridge Regression



- Blue lines indicate the level sets of the un-regularized MSE.
- The optimum value for $\mathbf{w}$ is $\mathbf{w}^\star$.

Recall the definition of the Ridge Regression:

$$\min_{\mathbf{w}} \quad \frac{1}{2N} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \; + \; \lambda \|\mathbf{w}\|_2^2 \; . \tag{44}$$

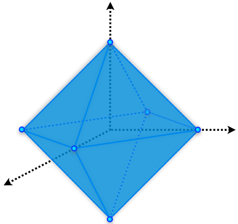
The Ridge Regression can be reformulated as

$$\min_{\mathbf{w}} \quad \frac{1}{2N} \sum_{n=1}^{N} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \quad \text{such that } \|\mathbf{w}\|_2^2 \leq \lambda \tag{45}$$

Geometric interpretation for the Lasso Regression

Similarly, the Lasso Regression can be reformulated as

$$\min_{\mathbf{w}} \quad \frac{1}{2N}\sum_{n=1}^{N} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \quad \text{such that } \|\mathbf{w}\|_1 \leq \lambda$$

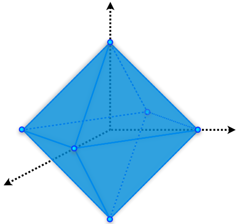


A “ball” of constant L_1 norm

Geometric interpretation for the Lasso Regression

Similarly, the Lasso Regression can be reformulated as

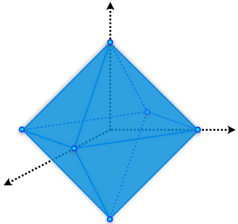
$$\min_{\mathbf{w}} \quad \frac{1}{2N}\sum_{n=1}^{N} \|\mathbf{y}-\mathbf{X}\mathbf{w}\|_2^2 \quad \text{such that } \|\mathbf{w}\|_1 \leq \lambda$$



A “ball” of constant L_1 norm

- The optimal point is somewhere on the surface of this “ball”.

Geometric interpretation for the Lasso Regression

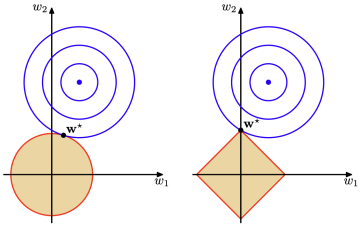


A "ball" of constant L_1 norm

Similarly, the Lasso Regression can be reformulated as

$$\min_{\mathbf{w}} \quad \frac{1}{2N}\sum_{n=1}^{N} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \quad \text{such that } \|\mathbf{w}\|_1 \leq \lambda$$

- The optimal point is somewhere on the surface of this "ball".

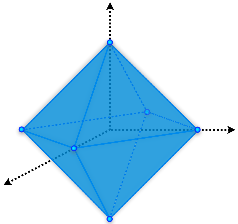


L_2-norm. L_1-norm.

Geometric interpretation for the Lasso Regression

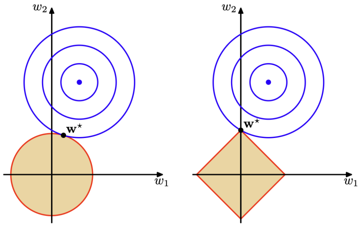
Similarly, the Lasso Regression can be reformulated as

$$\min_{\mathbf{w}} \quad \frac{1}{2N}\sum_{n=1}^{N} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \quad \text{such that } \|\mathbf{w}\|_1 \leq \lambda$$



A “ball” of constant L_1 norm

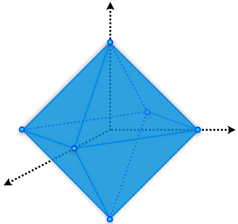
- The optimal point is somewhere on the surface of this “ball”.



L_2-norm. L_1-norm.

- This forces some of the elements of $\mathbf{w}$ to be strictly 0.

Geometric interpretation for the Lasso Regression

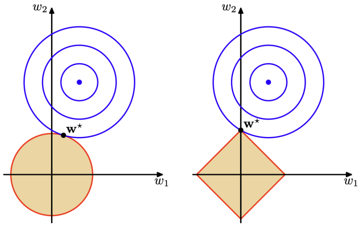


A “ball” of constant L_1 norm

Similarly, the Lasso Regression can be reformulated as

$$\min_{\mathbf{w}} \quad \frac{1}{2N}\sum_{n=1}^{N} \|\mathbf{y}-\mathbf{X}\mathbf{w}\|_2^2 \quad \text{such that } \|\mathbf{w}\|_1 \leq \lambda$$

- The optimal point is somewhere on the surface of this “ball”.



L_2-norm. L_1-norm.

- This forces some of the elements of $\mathbf{w}$ to be strictly 0.
- As λ is increased, an increasing number of parameters are driven to 0.

Table of Contents

The probabilistic interpretation of least-squares linear regression

Least-Squares linear regression can be interpreted as the Maximum Likelihood Estimator:

$$\mathbf{w}_{\text{lse}} \overset{(a)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{y}, \mathbf{X}|\mathbf{w}) \qquad \text{(by the definition of log-likelihood)}$$

$$(47)$$

The probabilistic interpretation of least-squares linear regression

Least-Squares linear regression can be interpreted as the Maximum Likelihood Estimator:

$$
\begin{aligned}
\mathbf{w}_{\text{lse}} &\overset{(a)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{y}, \mathbf{X}|\mathbf{w}) && \text{(by the definition of log-likelihood)} \\
&\overset{(b)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{X}|\mathbf{w})p(\mathbf{y}|\mathbf{X}, \mathbf{w}) && \text{(by factoring the likelihood)}
\end{aligned}
\tag{47}
$$

The probabilistic interpretation of least-squares linear regression

Least-Squares linear regression can be interpreted as the Maximum Likelihood Estimator:

$$
\begin{aligned}
\mathbf{w}_{\text{lse}} &\overset{(a)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{y}, \mathbf{X}|\mathbf{w}) && \text{(by the definition of log-likelihood)} \\
&\overset{(b)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{X}|\mathbf{w}) p(\mathbf{y}|\mathbf{X}, \mathbf{w}) && \text{(by factoring the likelihood)} \\
&\overset{(c)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{X}) p(\mathbf{y}|\mathbf{X}, \mathbf{w}) && \text{(the choice of the input } \mathbf{x}_n \text{ is independent of } \mathbf{w}\text{)}
\end{aligned}
\tag{47}
$$

The probabilistic interpretation of least-squares linear regression

Least-Squares linear regression can be interpreted as the Maximum Likelihood Estimator:

$$
\begin{aligned}
\mathbf{w}_{\text{lse}} &\overset{(a)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{y}, \mathbf{X}|\mathbf{w}) && \text{(by the definition of log-likelihood)} \\
&\overset{(b)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{X}|\mathbf{w}) p(\mathbf{y}|\mathbf{X}, \mathbf{w}) && \text{(by factoring the likelihood)} \\
&\overset{(c)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{X}) p(\mathbf{y}|\mathbf{X}, \mathbf{w}) && \text{(the choice of the input } \mathbf{x}_n \text{ is independent of } \mathbf{w}) \\
&\overset{(d)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{y}|\mathbf{X}, \mathbf{w}) && (p(\mathbf{X}) \text{ is independent of } \mathbf{w})
\end{aligned}
\tag{47}
$$

The probabilistic interpretation of least-squares linear regression

Least-Squares linear regression can be interpreted as the Maximum Likelihood Estimator:

$$
\begin{aligned}
\mathbf{w}_{\text{lse}} &\overset{(a)}{=} \arg\min_{\mathbf{w}} \; -\log p(\mathbf{y}, \mathbf{X}|\mathbf{w}) && \text{(by the definition of log-likelihood)} \\
&\overset{(b)}{=} \arg\min_{\mathbf{w}} \; -\log p(\mathbf{X}|\mathbf{w}) p(\mathbf{y}|\mathbf{X}, \mathbf{w}) && \text{(by factoring the likelihood)} \\
&\overset{(c)}{=} \arg\min_{\mathbf{w}} \; -\log p(\mathbf{X}) p(\mathbf{y}|\mathbf{X}, \mathbf{w}) && \text{(the choice of the input } \mathbf{x}_n \text{ is independent of } \mathbf{w}) \\
&\overset{(d)}{=} \arg\min_{\mathbf{w}} \; -\log p(\mathbf{y}|\mathbf{X}, \mathbf{w}) && (p(\mathbf{X}) \text{ is independent of } \mathbf{w}) \\
&\overset{(e)}{=} \arg\min_{\mathbf{w}} \; -\log \left[\prod_{n=1}^{N} p(y_n|\mathbf{x}_n, \mathbf{w}) \right] && \text{(we assume samples are iid.)}
\end{aligned}
\tag{47}
$$

The probabilistic interpretation of least-squares linear regression

Least-Squares linear regression can be interpreted as the Maximum Likelihood Estimator:

$$
\begin{aligned}
\mathbf{w}_{\text{lse}} &\overset{(a)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{y},\mathbf{X}|\mathbf{w}) && \text{(by the definition of log-likelihood)}\\
&\overset{(b)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{X}|\mathbf{w})p(\mathbf{y}|\mathbf{X},\mathbf{w}) && \text{(by factoring the likelihood)}\\
&\overset{(c)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{X})p(\mathbf{y}|\mathbf{X},\mathbf{w}) && \text{(the choice of the input } \mathbf{x}_n \text{ is independent of } \mathbf{w})\\
&\overset{(d)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{y}|\mathbf{X},\mathbf{w}) && (p(\mathbf{X}) \text{ is independent of } \mathbf{w})\\
&\overset{(e)}{=} \arg\min_{\mathbf{w}} \quad -\log\left[\prod_{n=1}^{N} p(y_n|\mathbf{x}_n,\mathbf{w})\right] && \text{(we assume samples are iid.)}\\
&= \arg\min_{\mathbf{w}} \quad \frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - \mathbf{x}_n^\top\mathbf{w})^2 && \text{(by definition and calculus)}
\end{aligned}
\tag{47}
$$

The probabilistic interpretation of Ridge Regression

We start with the posterior $p(\mathbf{w}|\mathbf{X}, \mathbf{y})$ and chose $\mathbf{w}$ to maximize this posterior.

The Maximum-A-Posteriori (MAP) estimate:

$$\mathbf{w}_{\text{ridge}} = \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{w}|\mathbf{X}, \mathbf{y}) \qquad \text{(by the definition of posterior)} \tag{48}$$

The probabilistic interpretation of Ridge Regression

We start with the posterior $p(\mathbf{w}|\mathbf{X},\mathbf{y})$ and chose $\mathbf{w}$ to maximize this posterior.

The Maximum-A-Posteriori (MAP) estimate:

$$\begin{aligned}
\mathbf{w}_{\text{ridge}} &= \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{w}|\mathbf{X},\mathbf{y}) && \text{(by the definition of posterior)} \\
&\overset{(a)}{=} \arg\min_{\mathbf{w}} \quad -\log \frac{p(\mathbf{y},\mathbf{X}|\mathbf{w})p(\mathbf{w})}{p(\mathbf{y},\mathbf{X})} && \text{(by the Bayes' law)}
\end{aligned} \tag{48}$$

The probabilistic interpretation of Ridge Regression

We start with the posterior $p(\mathbf{w}|\mathbf{X}, \mathbf{y})$ and chose $\mathbf{w}$ to maximize this posterior.

The Maximum-A-Posteriori (MAP) estimate:

$$\begin{aligned}
\mathbf{w}_{\text{ridge}} &= \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{w}|\mathbf{X}, \mathbf{y}) && \text{(by the definition of posterior)} \\
&\overset{(a)}{=} \arg\min_{\mathbf{w}} \quad -\log \frac{p(\mathbf{y}, \mathbf{X}|\mathbf{w})p(\mathbf{w})}{p(\mathbf{y}, \mathbf{X})} && \text{(by the Bayes' law)} \\
&\overset{(b)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{y}, \mathbf{X}|\mathbf{w})p(\mathbf{w}) && \text{(eliminate quantities that do not depend on } \mathbf{w}\text{)}
\end{aligned} \tag{48}$$

The probabilistic interpretation of Ridge Regression

We start with the posterior $p(\mathbf{w}|\mathbf{X}, \mathbf{y})$ and chose $\mathbf{w}$ to maximize this posterior.

The Maximum-A-Posteriori (MAP) estimate:

$$\begin{aligned}
\mathbf{w}_{\text{ridge}} &= \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{w}|\mathbf{X}, \mathbf{y}) && \text{(by the definition of posterior)} \\
&\overset{(a)}{=} \arg\min_{\mathbf{w}} \quad -\log \frac{p(\mathbf{y}, \mathbf{X}|\mathbf{w})p(\mathbf{w})}{p(\mathbf{y}, \mathbf{X})} && \text{(by the Bayes' law)} \\
&\overset{(b)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{y}, \mathbf{X}|\mathbf{w})p(\mathbf{w}) && \text{(eliminate quantities that do not depend on } \mathbf{w}\text{)} \\
&\overset{(c)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w}) && \text{(eliminate quantities that do not depend on } \mathbf{w}\text{)}
\end{aligned} \tag{48}$$

The probabilistic interpretation of Ridge Regression

We start with the posterior $p(\mathbf{w}|\mathbf{X}, \mathbf{y})$ and chose $\mathbf{w}$ to maximize this posterior.

The Maximum-A-Posteriori (MAP) estimate:

$$
\begin{aligned}
\mathbf{w}_{\text{ridge}} &= \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{w}|\mathbf{X}, \mathbf{y}) && \text{(by the definition of posterior)} \\
&\overset{(a)}{=} \arg\min_{\mathbf{w}} \quad -\log \frac{p(\mathbf{y}, \mathbf{X}|\mathbf{w})p(\mathbf{w})}{p(\mathbf{y}, \mathbf{X})} && \text{(by the Bayes' law)} \\
&\overset{(b)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{y}, \mathbf{X}|\mathbf{w})p(\mathbf{w}) && \text{(eliminate quantities that do not depend on } \mathbf{w}\text{)} \\
&\overset{(c)}{=} \arg\min_{\mathbf{w}} \quad -\log p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w}) && \text{(eliminate quantities that do not depend on } \mathbf{w}\text{)} \\
&= \arg\min_{\mathbf{w}} \quad \sum_{n=1}^{N} \frac{1}{2\sigma^2}(y_n - \mathbf{x}_n^\top \mathbf{w})^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2 \,. && (48)
\end{aligned}
$$

This lecture:

- Normal Equation and Least Squares
- Probabilistic Interpretation of Linear Regression
- Maximum Likelihood Estimation (MLE)
- Over-fitting and Under-fitting
- Polynomial Regression, Ridge Regression, and Lasso

Next lecture:

- Generalization Gap and Model Selection
- Bias-Variance Decomposition