# Lecture 2: Supervised Learning

**Tao LIN**

February 23, 2023

WESTLAKE UNIVERSITY | SCHOOL OF ENGINEERING

# Feedback on the Questionnaire

- Course Project evaluation protocol
  - CS student: either (by-team) or (by-team-and-supervisor)
  - non-CS student: (by-team-and-supervisor)

# Feedback on the Questionnaire

- Course Project evaluation protocol
  - CS student: either (by-team) or (by-team-and-supervisor)
  - non-CS student: (by-team-and-supervisor)
- Homework due time
  - Every Wednesday at 9 pm

# Feedback on the Questionnaire

- Course Project evaluation protocol
  - CS student: either (by-team) or (by-team-and-supervisor)
  - non-CS student: (by-team-and-supervisor)
- Homework due time
  - Every Wednesday at 9 pm
- We will provide the reading materials for each lecture

# Feedback on the Questionnaire

- Course Project evaluation protocol
  - CS student: either (by-team) or (by-team-and-supervisor)
  - non-CS student: (by-team-and-supervisor)
- Homework due time
  - Every Wednesday at 9 pm
- We will provide the reading materials for each lecture
- QA session
  - No office hours for the moment
  - Please post your question as a GitHub issue

# Feedback on the Questionnaire

- Course Project evaluation protocol
  - CS student: either (by-team) or (by-team-and-supervisor)
  - non-CS student: (by-team-and-supervisor)
- Homework due time
  - Every Wednesday at 9 pm
- We will provide the reading materials for each lecture
- QA session
  - No office hours for the moment
  - Please post your question as a GitHub issue
- Slides and prerequisites
  - We will upload the slides every Wednesday at noon.

# Feedback on the Questionnaire

- Course Project evaluation protocol
  - CS student: either (by-team) or (by-team-and-supervisor)
  - non-CS student: (by-team-and-supervisor)
- Homework due time
  - Every Wednesday at 9 pm
- We will provide the reading materials for each lecture
- QA session
  - No office hours for the moment
  - Please post your question as a GitHub issue
- Slides and prerequisites
  - We will upload the slides every Wednesday at noon.
- Theoretical foundation for DL
  - We will explain the mathematical intuitions and insights behind DL methods.

**This lecture:**

- Basic concept of regression and classification
- Linear regression
    - Definition
    - Gradient Descent (GD) optimization
    - Least Square
    - The probabilistic interpretation of linear regression

**Next lecture:**

- Over-fitting and under-fitting
- Polynomial regression and Ridge regression
- 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, Bishop, Pattern Recognition and Machine Learning

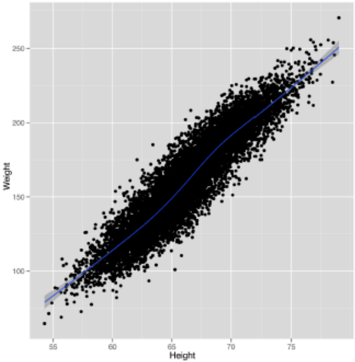
# Reference

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

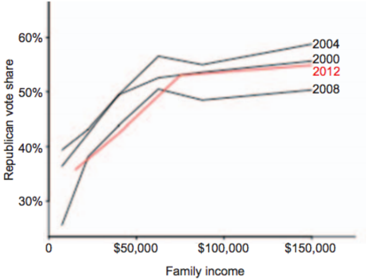
# Table of Contents

# Table of Contents

# What is regression?




(a) Height is correlated with weight. Taken from “Machine Learning for Hackers"



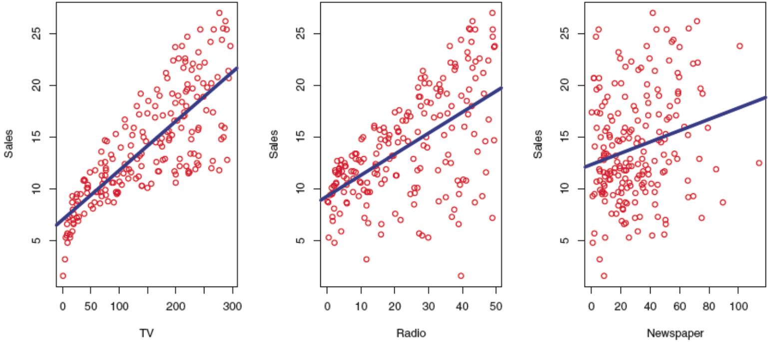

(b) Do rich people vote for republicans? Taken from Avi Feller et. al. 2013, Red state/blue state in 2012 elections.

Regression is to relate input variables to the output variable.

# Dataset for regression

$$\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N} \in \mathcal{X} \times \mathcal{Y} \quad (1)$$

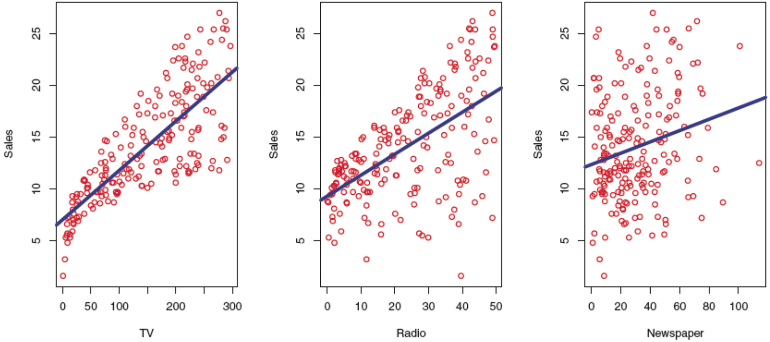
- data consists of pairs $(\mathbf{x}_n, y_n)$, where $y_n$ is the $n$'th output and $\mathbf{x}_n$ is a vector of $D$ inputs.



# Dataset for regression

$$\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N} \in \mathcal{X} \times \mathcal{Y} \quad (1)$$

- data consists of pairs $(\mathbf{x}_n, y_n)$, where $y_n$ is the $n$'th output and $\mathbf{x}_n$ is a vector of $D$ inputs.
- The number of pairs $N$ is the data-size and $D$ is the dimensionality.



# Two goals of regression

**The regression function** approximates the output $y_n$ “well enough” given inputs $\mathbf{x}_n$.

$$y_n \approx f(\mathbf{x}_n), \text{ for all } n \tag{2}$$

# Two goals of regression

**The regression function** approximates the output $y_n$ "well enough" given inputs $\mathbf{x}_n$.

$$y_n \approx f(\mathbf{x}_n), \text{ for all } n \tag{2}$$

1. prediction: predict outputs for new inputs.

# Two goals of regression

**The regression function** approximates the output $y_n$ “well enough” given inputs $\mathbf{x}_n$.

$$y_n \approx f(\mathbf{x}_n), \text{ for all } n \tag{2}$$

1. prediction: predict outputs for new inputs.

   e.g., what is the weight of a person who is 170 cm tall?

# Two goals of regression

**The regression function** approximates the output $y_n$ "well enough" given inputs $\mathbf{x}_n$.

$$y_n \approx f(\mathbf{x}_n), \text{ for all } n \tag{2}$$

1. prediction: predict outputs for new inputs.

   e.g., what is the weight of a person who is 170 cm tall?

2. interpretation: understand the effect of the input on the output.

# Two goals of regression

**The regression function** approximates the output $y_n$ “well enough” given inputs $\mathbf{x}_n$.

$$y_n \approx f(\mathbf{x}_n), \text{ for all } n \tag{2}$$

1. prediction: predict outputs for new inputs.

   e.g., what is the weight of a person who is 170 cm tall?

2. interpretation: understand the effect of the input on the output.

   e.g., are taller people heavier too?

# Two goals of regression

**The regression function** approximates the output $y_n$ “well enough” given inputs $\mathbf{x}_n$.

$$y_n \approx f(\mathbf{x}_n), \text{ for all } n \tag{2}$$

1. prediction: predict outputs for new inputs.

   e.g., what is the weight of a person who is 170 cm tall?

2. interpretation: understand the effect of the input on the output.

   e.g., are taller people heavier too?

**Remark 1 (Correlation $\neq$ Causation)**

*Regression finds a correlation not a causal relationship, so interpret your results with caution.*

# Two goals of regression

**The regression function** approximates the output $y_n$ "well enough" given inputs $\mathbf{x}_n$.

$$y_n \approx f(\mathbf{x}_n), \text{ for all } n \tag{2}$$

1. prediction: predict outputs for new inputs.

   e.g., what is the weight of a person who is 170 cm tall?

2. interpretation: understand the effect of the input on the output.

   e.g., are taller people heavier too?

**Remark 1 (Correlation $\neq$ Causation)**

*Regression finds a correlation not a causal relationship, so interpret your results with caution.*

**Remark 2 (Shortcut learning in Deep Learning)**

*Models may only learn spurious correlation (and thus sensitive to distribution shifts).*

# Table of Contents

# Classification

We observe some data

$$\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N} \in \mathcal{X} \times \underbrace{\mathcal{Y}}_{\text{Discrete set}} \tag{3}$$

# Classification

We observe some data

$$\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N} \in \mathcal{X} \times \underbrace{\mathcal{Y}}_{\text{Discrete set}} \tag{3}$$

- **Binary classification:** $y \in \{\mathcal{C}_1, \mathcal{C}_2\} \quad \Rightarrow \quad$ The $\mathcal{C}_i$ are called class labels or classes.

# Classification

We observe some data

$$\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N} \in \mathcal{X} \times \underbrace{\mathcal{Y}}_{\text{Discrete set}} \tag{3}$$

- **Binary classification:** $y \in \{\mathcal{C}_1, \mathcal{C}_2\} \quad \Rightarrow \quad$ The $\mathcal{C}_i$ are called class labels or classes.
- **Multi-class classification:** $y \in \{\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_{K-1}\}$ for a $K$-class problem.

# Classification

We observe some data

$$\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^N \in \mathcal{X} \times \underbrace{\mathcal{Y}}_{\text{Discrete set}} \tag{3}$$

- **Binary classification:** $y \in \{\mathcal{C}_1, \mathcal{C}_2\}$ $\quad\Rightarrow\quad$ The $\mathcal{C}_i$ are called class labels or classes.
- **Multi-class classification:** $y \in \{\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_{K-1}\}$ for a $K$-class problem.

**Remark 3**

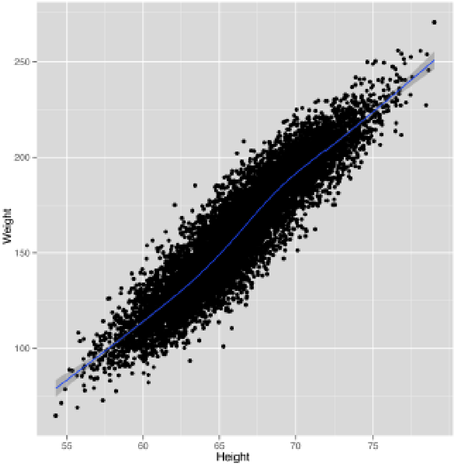
*no ordering between classes.*

# Table of Contents

# Table of Contents

# Definition

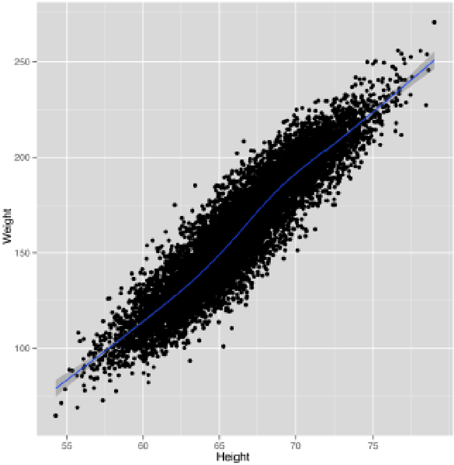
Linear regression is a model:



# Definition

Linear regression is a model:

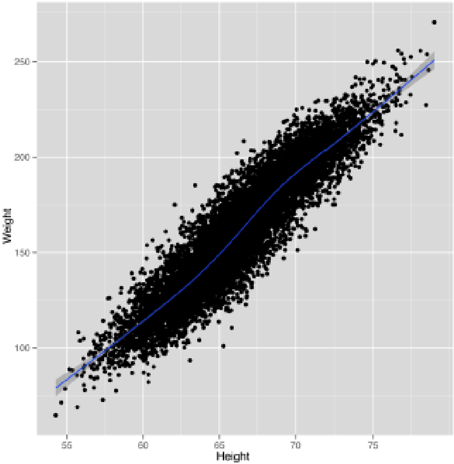
- $y_n \approx f(\mathbf{x}_n)$ for all $n$ and $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^N \in \mathcal{X} \times \mathcal{Y}$



# Definition

Linear regression is a model:

- $y_n \approx f(\mathbf{x}_n)$ for all $n$ and $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^N \in \mathcal{X} \times \mathcal{Y}$
- a linear relationship is assumed for $f$



# Detailed definition

**Simple linear regression** (w/ only one input dimension):

$$y_n \approx f(\mathbf{x}_n) := w_0 + w_1 x_{n1}$$

Here, $\mathbf{w} = (w_0, w_1)$ are the two parameters of the model. They describe $f$.

# Detailed definition

**Simple linear regression** (w/ only one input dimension):

$$y_n \approx f(\mathbf{x}_n) := w_0 + w_1 x_{n1}$$

Here, $\mathbf{w} = (w_0, w_1)$ are the two parameters of the model. They describe $f$.

**Multiple linear regression** (multiple input dimension):

$$\begin{aligned} y_n \approx f(\mathbf{x}_n) &:= w_0 + w_1 x_{n1} + \ldots + w_D x_{nD} && (4) \\ &= w_0 + \mathbf{x}_n^\top \begin{pmatrix} w_1 \\ \vdots \\ w_D \end{pmatrix} && (5) \\ &=: \tilde{\mathbf{x}}_n^\top \tilde{\mathbf{w}} && (6) \end{aligned}$$

# Detailed definition

**Simple linear regression** (w/ only one input dimension):

$$y_n \approx f(\mathbf{x}_n) := w_0 + w_1 x_{n1}$$

Here, $\mathbf{w} = (w_0, w_1)$ are the two parameters of the model. They describe $f$.

**Multiple linear regression** (multiple input dimension):

$$\begin{aligned} y_n \approx f(\mathbf{x}_n) &:= w_0 + w_1 x_{n1} + \ldots + w_D x_{nD} && (4) \\ &= w_0 + \mathbf{x}_n^\top \begin{pmatrix} w_1 \\ \vdots \\ w_D \end{pmatrix} && (5) \\ &=: \tilde{\mathbf{x}}_n^\top \tilde{\mathbf{w}} && (6) \end{aligned}$$

We add a tilde over the input vector & weights, to indicate containing the additional offset term (a.k.a. bias term).

# Detailed definition

**Simple linear regression** (w/ only one input dimension):

$$y_n \approx f(\mathbf{x}_n) := w_0 + w_1 x_{n1}$$

Here, $\mathbf{w} = (w_0, w_1)$ are the two parameters of the model. They describe $f$.

**Multiple linear regression** (multiple input dimension):

$$\begin{aligned} y_n \approx f(\mathbf{x}_n) &:= w_0 + w_1 x_{n1} + \ldots + w_D x_{nD} && (4) \\ &= w_0 + \mathbf{x}_n^\top \begin{pmatrix} w_1 \\ \vdots \\ w_D \end{pmatrix} && (5) \\ &=: \tilde{\mathbf{x}}_n^\top \tilde{\mathbf{w}} && (6) \end{aligned}$$

We add a tilde over the input vector & weights, to indicate containing the additional offset term (a.k.a. bias term).

**Goal: Learning / Estimation / Fitting**

Given data $\mathcal{D}$, we would like to find $\tilde{\mathbf{w}} = [w_0, w_1, \ldots, w_D]$.

# Detailed definition

**Simple linear regression** (w/ only one input dimension):

$$y_n \approx f(\mathbf{x}_n) := w_0 + w_1 x_{n1}$$

Here, $\mathbf{w} = (w_0, w_1)$ are the two parameters of the model. They describe $f$.

**Multiple linear regression** (multiple input dimension):

$$\begin{aligned} y_n \approx f(\mathbf{x}_n) &:= w_0 + w_1 x_{n1} + \ldots + w_D x_{nD} && (4) \\ &= w_0 + \mathbf{x}_n^\top \begin{pmatrix} w_1 \\ \vdots \\ w_D \end{pmatrix} && (5) \\ &=: \tilde{\mathbf{x}}_n^\top \tilde{\mathbf{w}} && (6) \end{aligned}$$

We add a tilde over the input vector & weights, to indicate containing the additional offset term (a.k.a. bias term).

**Goal: Learning / Estimation / Fitting**

Given data $\mathcal{D}$, we would like to find $\tilde{\mathbf{w}} = [w_0, w_1, \ldots, w_D]$.

We need an optimization algorithm!

# Why learn about *linear* regression?

- simple

# Why learn about *linear* regression?

- simple
- easy to understand

# Why learn about *linear* regression?

- simple
- easy to understand
- widely used

# Why learn about *linear* regression?

- simple
- easy to understand
- widely used
- easily generalized to non-linear models

# Why learn about *linear* regression?

- simple
- easy to understand
- widely used
- easily generalized to non-linear models
- we can learn almost all fundamental concepts of ML with regression alone

# Table of Contents

# Motivation

Consider the following models.

$$\text{1-parameter model: } y_n \approx w_0$$
$$\text{2-parameter model: } y_n \approx w_0 + w_1 x_{n1}$$

Q: How can we estimate values of $\mathbf{w}$ given the data $\mathcal{D}$?

# Motivation

Consider the following models.

$$\text{1-parameter model: } y_n \approx w_0$$
$$\text{2-parameter model: } y_n \approx w_0 + w_1 x_{n1}$$

Q: How can we estimate values of $\mathbf{w}$ given the data $\mathcal{D}$?

A: Optimizing the **cost function** (or energy, loss, training objective)

# Motivation

Consider the following models.

$$\text{1-parameter model: } y_n \approx w_0$$
$$\text{2-parameter model: } y_n \approx w_0 + w_1 x_{n1}$$

Q: How can we estimate values of $\mathbf{w}$ given the data $\mathcal{D}$?

A: Optimizing the **cost function** (or energy, loss, training objective) to quantify how well the learned parameter does

# Motivation

Consider the following models.

$$\text{1-parameter model: } y_n \approx w_0$$
$$\text{2-parameter model: } y_n \approx w_0 + w_1 x_{n1}$$

Q: How can we estimate values of $\mathbf{w}$ given the data $\mathcal{D}$?

A: Optimizing the **cost function** (or energy, loss, training objective) to quantify how well the learned parameter does

**Two desirable properties of cost functions**

# Motivation

Consider the following models.

$$\text{1-parameter model: } y_n \approx w_0$$
$$\text{2-parameter model: } y_n \approx w_0 + w_1 x_{n1}$$

Q: How can we estimate values of $\mathbf{w}$ given the data $\mathcal{D}$?

A: Optimizing the **cost function** (or energy, loss, training objective) to quantify how well the learned parameter does

**Two desirable properties of cost functions**

- the cost is symmetric around $0$ (penalize positive and negative errors equally)

# Motivation

Consider the following models.

$$\text{1-parameter model: } y_n \approx w_0$$
$$\text{2-parameter model: } y_n \approx w_0 + w_1 x_{n1}$$

Q: How can we estimate values of $\mathbf{w}$ given the data $\mathcal{D}$?

A: Optimizing the **cost function** (or energy, loss, training objective)
to quantify how well the learned parameter does

**Two desirable properties of cost functions**

- the cost is symmetric around $0$ (penalize positive and negative errors equally)
- the cost penalizes "large" mistakes and "very-large" mistakes similarly

# Mean Squared Error (MSE) and Outliers

$$\text{MSE}(\mathbf{w}) := \frac{1}{N}\sum_{n=1}^{N}\left[y_n - f_{\mathbf{w}}(\mathbf{x}_n)\right]^2 \tag{7}$$

# Mean Squared Error (MSE) and Outliers

$$\text{MSE}(\mathbf{w}) := \frac{1}{N}\sum_{n=1}^{N}\left[y_n - f_{\mathbf{w}}(\mathbf{x}_n)\right]^2 \tag{7}$$

Does this cost function have both mentioned properties?

# Mean Squared Error (MSE) and Outliers

$$\text{MSE}(\mathbf{w}) := \frac{1}{N} \sum_{n=1}^{N} \left[ y_n - f_{\mathbf{w}}(\mathbf{x}_n) \right]^2 \tag{7}$$

Does this cost function have both mentioned properties?

No!

# Mean Squared Error (MSE) and Outliers

$$\text{MSE}(\mathbf{w}) := \frac{1}{N}\sum_{n=1}^{N}\left[y_n - f_{\mathbf{w}}(\mathbf{x}_n)\right]^2 \tag{7}$$

Does this cost function have both mentioned properties?

No!

**Definition 4 (Outliers)**

Outliers are data examples that are far away from most of the other examples.

# Mean Squared Error (MSE) and Outliers

$$\text{MSE}(\mathbf{w}) := \frac{1}{N} \sum_{n=1}^{N} \left[ y_n - f_{\mathbf{w}}(\mathbf{x}_n) \right]^2 \tag{7}$$

Does this cost function have both mentioned properties?

No!

## Definition 4 (Outliers)

Outliers are data examples that are far away from most of the other examples.

MSE is not a good cost function when outliers are present.

# Mean Squared Error (MSE) and Outliers

$$\text{MSE}(\mathbf{w}) := \frac{1}{N}\sum_{n=1}^{N}\left[y_n - f_{\mathbf{w}}(\mathbf{x}_n)\right]^2 \tag{7}$$

Does this cost function have both mentioned properties?

No!

## Definition 4 (Outliers)

Outliers are data examples that are far away from most of the other examples.

MSE is not a good cost function when outliers are present.

- **Pros:** It ensures that *trained model has no outlier predictions with huge errors*.

# Mean Squared Error (MSE) and Outliers

$$\text{MSE}(\mathbf{w}) := \frac{1}{N}\sum_{n=1}^{N}\left[y_n - f_{\mathbf{w}}(\mathbf{x}_n)\right]^2 \tag{7}$$

Does this cost function have both mentioned properties?

No!

## Definition 4 (Outliers)

Outliers are data examples that are far away from most of the other examples.

MSE is not a good cost function when outliers are present.

- **Pros:** It ensures that *trained model has no outlier predictions with huge errors*.
- **Cons:** It is very sensitive to outliers.

# Mean Absolute Error (MAE)

Handling outliers well is a desired *statistical* property.

# Mean Absolute Error (MAE)

Handling outliers well is a desired *statistical* property.

$$\text{MAE}(\mathbf{w}) := \frac{1}{N}\sum_{n=1}^{N} |y_n - f_{\mathbf{w}}(\mathbf{x}_n)| \tag{8}$$

# Mean Absolute Error (MAE)

Handling outliers well is a desired *statistical* property.

$$\text{MAE}(\mathbf{w}) := \frac{1}{N}\sum_{n=1}^{N} |y_n - f_{\mathbf{w}}(\mathbf{x}_n)| \tag{8}$$

+ MAE is more robust to outliers.

# Mean Absolute Error (MAE)

Handling outliers well is a desired *statistical* property.

$$\text{MAE}(\mathbf{w}) := \frac{1}{N} \sum_{n=1}^{N} |y_n - f_{\mathbf{w}}(\mathbf{x}_n)| \tag{8}$$

+ MAE is more robust to outliers.
– MAE is not differentiable at zero.

# Learning / Estimation / Fitting

**Definition 5** (*Learning* problem can be formulated as optimization problem)

Given a cost function $\mathcal{L}(\mathbf{w})$, we wish to find $\mathbf{w}^\star$ which minimizes the cost:

# Learning / Estimation / Fitting

**Definition 5** (*Learning* problem can be formulated as optimization problem)

Given a cost function $\mathcal{L}(\mathbf{w})$, we wish to find $\mathbf{w}^\star$ which minimizes the cost:

$$\min_{\mathbf{w}} \ \mathcal{L}(\mathbf{w}) \quad \text{subject to } \mathbf{w} \in \mathbb{R}^D \tag{9}$$

# Learning / Estimation / Fitting

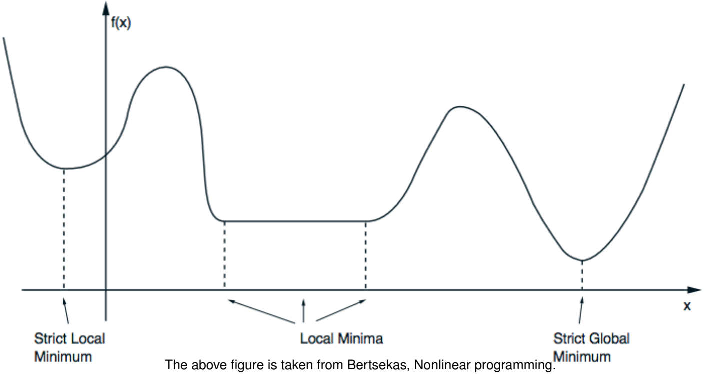
**Definition 5** (*Learning* problem can be formulated as optimization problem)

Given a cost function $\mathcal{L}(\mathbf{w})$, we wish to find $\mathbf{w}^\star$ which minimizes the cost:

$$\min_{\mathbf{w}} \; \mathcal{L}(\mathbf{w}) \quad \text{subject to } \mathbf{w} \in \mathbb{R}^D \tag{9}$$

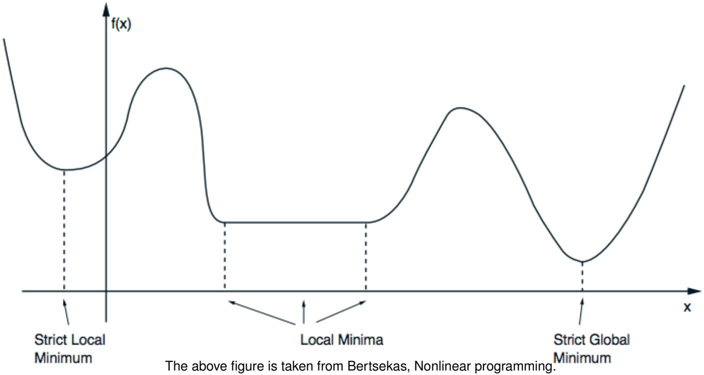
We will use an optimization algorithm to solve the problem (to find a good $\mathbf{w}$).

# Optimization Landscapes




The above figure is taken from Bertsekas, Nonlinear programming.

# Optimization Landscapes

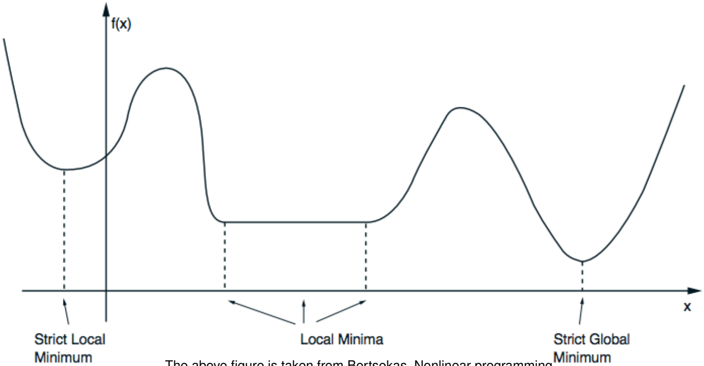



The above figure is taken from Bertsekas, Nonlinear programming.

- A vector $\mathbf{w}^\star$ is a local minimum of $\mathcal{L}$ if it is no worse than its neighbors; i.e. there exists an $\epsilon > 0$ such that,

$$\mathcal{L}(\mathbf{w}^\star) \leq \mathcal{L}(\mathbf{w}), \quad \forall \mathbf{w} \text{ with } \|\mathbf{w} - \mathbf{w}^\star\| < \epsilon$$

# Optimization Landscapes




The above figure is taken from Bertsekas, Nonlinear programming.

- A vector $\mathbf{w}^\star$ is a local minimum of $\mathcal{L}$ if it is no worse than its neighbors; i.e. there exists an $\epsilon > 0$ such that,

$$\mathcal{L}(\mathbf{w}^\star) \leq \mathcal{L}(\mathbf{w}), \quad \forall \mathbf{w} \text{ with } \|\mathbf{w} - \mathbf{w}^\star\| < \epsilon$$

- A vector $\mathbf{w}^\star$ is a global minimum of $\mathcal{L}$ if it is no worse than all others,

$$\mathcal{L}(\mathbf{w}^\star) \leq \mathcal{L}(\mathbf{w}), \quad \forall \mathbf{w} \in \mathbb{R}^D$$

# Smooth Optimization: Follow the Gradient

# Smooth Optimization: Follow the Gradient

## Definition 6 (Gradient)

A gradient $\nabla\mathcal{L}(\mathbf{w})$ (at a point) is the slope of the ***tangent*** to the function (at that point):

# Smooth Optimization: Follow the Gradient

## Definition 6 (Gradient)

A gradient $\nabla \mathcal{L}(\mathbf{w})$ (at a point) is the slope of the ***tangent*** to the function (at that point):

$$\nabla \mathcal{L}(\mathbf{w}) := \left[ \frac{\partial \mathcal{L}(\mathbf{w})}{\partial w_1}, \dots, \frac{\partial \mathcal{L}(\mathbf{w})}{\partial w_D} \right]^\top \in \mathbb{R}^D, \tag{10}$$

# Smooth Optimization: Follow the Gradient

## Definition 6 (Gradient)

A gradient $\nabla\mathcal{L}(\mathbf{w})$ (at a point) is the slope of the ***tangent*** to the function (at that point):

$$\nabla\mathcal{L}(\mathbf{w}) := \left[\frac{\partial\mathcal{L}(\mathbf{w})}{\partial w_1}, \ldots, \frac{\partial\mathcal{L}(\mathbf{w})}{\partial w_D}\right]^\top \in \mathbb{R}^D, \tag{10}$$

where it points to the direction of the largest increase of the function.

# Smooth Optimization: Follow the Gradient

## Definition 6 (Gradient)

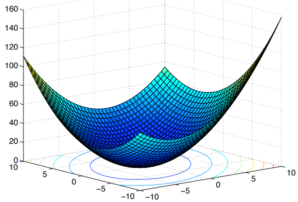
A gradient $\nabla\mathcal{L}(\mathbf{w})$ (at a point) is the slope of the ***tangent*** to the function (at that point):

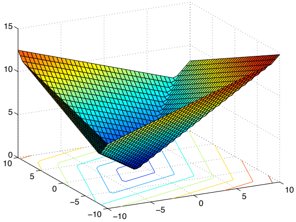
$$\nabla\mathcal{L}(\mathbf{w}) := \left[\frac{\partial\mathcal{L}(\mathbf{w})}{\partial w_1}, \ldots, \frac{\partial\mathcal{L}(\mathbf{w})}{\partial w_D}\right]^\top \in \mathbb{R}^D, \tag{10}$$

where it points to the direction of the largest increase of the function.

For a 2-parameter model, $\text{MSE}(\mathbf{w})$ and $\text{MAE}(\mathbf{w})$ are shown below.

(We used $\mathbf{y}_n \approx w_0 + w_1 x_{n1}$ with $\mathbf{y}^\top = [2, -1, 1.5]$ and $\mathbf{x}^\top = [-1, 1, -1]$).



# Gradient Descent

**Definition 7 (Gradient Descent)**

To minimize the cost function, we iteratively take a step in the (opposite) direction of the gradient

# Gradient Descent

**Definition 7 (Gradient Descent)**

To minimize the cost function, we iteratively take a step in the (opposite) direction of the gradient

$$\mathbf{w}^{(t+1)} := \mathbf{w}^{(t)} - \gamma \nabla \mathcal{L}(\mathbf{w}^{(t)}) \tag{11}$$

# Gradient Descent

### Definition 7 (Gradient Descent)

To minimize the cost function, we iteratively take a step in the (opposite) direction of the gradient

$$\mathbf{w}^{(t+1)} := \mathbf{w}^{(t)} - \gamma \nabla \mathcal{L}(\mathbf{w}^{(t)}) \tag{11}$$

where $\gamma > 0$ is the step-size (or learning rate). Then repeat with the next $t$.

# 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 $f(\mathbf{w}, \mathbf{X}) = \mathbf{X}\mathbf{w}$.

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

# 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 $f(\mathbf{w}, \mathbf{X}) = \mathbf{X}\mathbf{w}$.

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

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

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{w} = \begin{pmatrix} e_1 \\ \vdots \\ e_N \end{pmatrix} \in \mathbb{R}^N, \tag{13}$$

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

# 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 $f(\mathbf{w}, \mathbf{X}) = \mathbf{X}\mathbf{w}$.

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

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

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{w} = \begin{pmatrix} e_1 \\ \vdots \\ e_N \end{pmatrix} \in \mathbb{R}^N, \tag{13}$$

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

$$\mathcal{L}(\mathbf{w}) := \frac{1}{2N} \sum_{n=1}^{N} \left( y_n - \mathbf{x}_n^\top \mathbf{w} \right)^2 = \tfrac{1}{2N} \mathbf{e}^\top \mathbf{e}, \tag{14}$$

# 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 $f(\mathbf{w}, \mathbf{X}) = \mathbf{X}\mathbf{w}$.

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

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

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{w} = \begin{pmatrix} e_1 \\ \vdots \\ e_N \end{pmatrix} \in \mathbb{R}^N, \tag{13}$$

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

$$\mathcal{L}(\mathbf{w}) := \frac{1}{2N} \sum_{n=1}^{N} \left(y_n - \mathbf{x}_n^\top \mathbf{w}\right)^2 = \tfrac{1}{2N} \mathbf{e}^\top \mathbf{e}, \tag{14}$$

and then the gradient is given by

$$\nabla \mathcal{L}(\mathbf{w}) = -\tfrac{1}{N} \mathbf{X}^\top \mathbf{e} \tag{15}$$

# Table of Contents

# Motivation

- In rare cases, one can compute the optimum of the cost function analytically.

# Motivation

- In rare cases, one can compute the optimum of the cost function analytically.
- Linear regression using an MSE cost function is one such case.

# Motivation

- In rare cases, one can compute the optimum of the cost function analytically.
- Linear regression using an MSE cost function is one such case.
- Here its solution can be obtained explicitly, by solving a linear system of equations.

# Motivation

- In rare cases, one can compute the optimum of the cost function analytically.
- Linear regression using an MSE cost function is one such case.
- Here its solution can be obtained explicitly, by solving a linear system of equations.

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

# Motivation

- In rare cases, one can compute the optimum of the cost function analytically.
- Linear regression using an MSE cost function is one such case.
- Here 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

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2N}\sum_{n=1}^{N}\left(y_n - \mathbf{x}_n^\top \mathbf{w}\right)^2 = \frac{1}{2N}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w}), \tag{16}$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^N, \qquad \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1D} \\ x_{21} & x_{22} & \dots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{ND} \end{bmatrix} \in \mathbb{R}^{N\times D}. \tag{17}$$

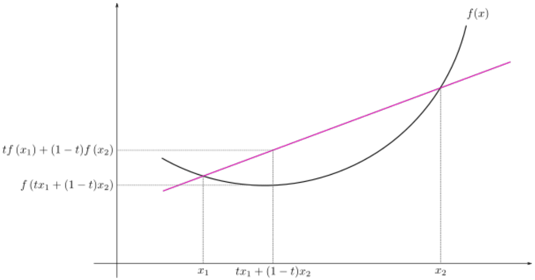
# Steps to form normal equations

# Steps to form normal equations

## Definition 8 (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:

$$h\left(\lambda \mathbf{u} + (1-\lambda)\mathbf{v}\right) \leq \lambda h(\mathbf{u}) + (1-\lambda) h(\mathbf{v}) \tag{18}$$

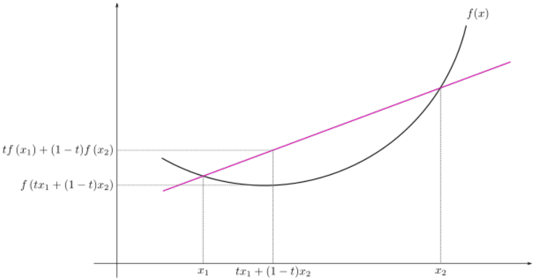


# Steps to form normal equations

## Definition 8 (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:

$$h\left(\lambda \mathbf{u} + (1-\lambda)\mathbf{v}\right) \leq \lambda h(\mathbf{u}) + (1-\lambda)h(\mathbf{v}) \tag{18}$$



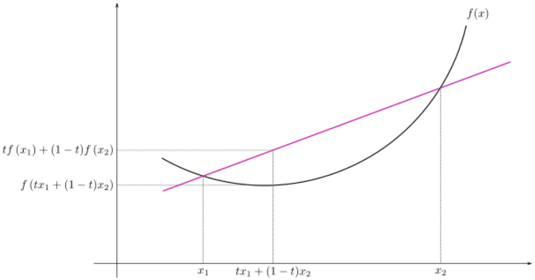

To derive the normal equations,

# Steps to form normal equations

## Definition 8 (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:

$$h\left(\lambda \mathbf{u} + (1-\lambda)\mathbf{v}\right) \leq \lambda h(\mathbf{u}) + (1-\lambda) h(\mathbf{v}) \quad (18)$$




To derive the normal equations,

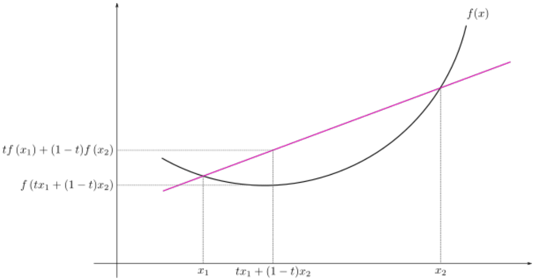
1. we first show that the problem is convex.

# Steps to form normal equations

**Definition 8 (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 $0 \le \lambda \le 1$, we have:

$$h\left(\lambda \mathbf{u} + (1-\lambda)\mathbf{v}\right) \le \lambda h(\mathbf{u}) + (1-\lambda)h(\mathbf{v}) \tag{18}$$




To derive the normal equations,

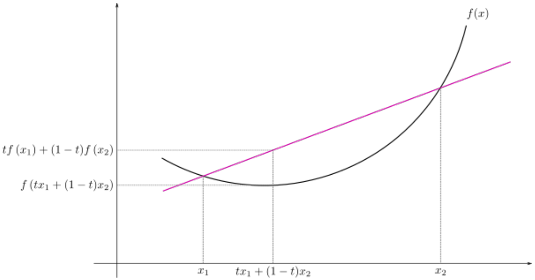
1. we first show that the problem is convex.
2. we then use the optimality conditions for convex functions, i.e.,

# Steps to form normal equations

**Definition 8 (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:

$$h(\lambda \mathbf{u} + (1-\lambda)\mathbf{v}) \leq \lambda h(\mathbf{u}) + (1-\lambda) h(\mathbf{v}) \tag{18}$$




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.,

$$\nabla \mathcal{L}(\mathbf{w}^\star) = \mathbf{0}\,, \tag{19}$$

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:

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

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left( \mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w} \right)^2 \ , \tag{20}$$

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:

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

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left( \mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w} \right)^2 , \tag{20}$$

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:

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

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left( \mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w} \right)^2 , \tag{20}$$

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".

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

$$\mathcal{L}(\lambda \mathbf{w} + (1-\lambda)\mathbf{w}') - (\lambda \mathcal{L}(\mathbf{w}) + (1-\lambda)\mathcal{L}(\mathbf{w}')) \leq 0 \,. \tag{21}$$

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

There are several ways of proving this:

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

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left(\mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w}\right)^2 \,, \tag{20}$$

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".

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

$$\mathcal{L}(\lambda \mathbf{w} + (1-\lambda)\mathbf{w}') - (\lambda \mathcal{L}(\mathbf{w}) + (1-\lambda)\mathcal{L}(\mathbf{w}')) \leq 0 \,. \tag{21}$$

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:

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

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left( \mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w} \right)^2 , \tag{20}$$

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".

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

$$\mathcal{L}(\lambda \mathbf{w} + (1-\lambda)\mathbf{w}') - (\lambda \mathcal{L}(\mathbf{w}) + (1-\lambda)\mathcal{L}(\mathbf{w}')) \leq 0 \,. \tag{21}$$

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.

**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:

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

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left( \mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w} \right)^2 , \tag{20}$$

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".

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

$$\mathcal{L}(\lambda \mathbf{w} + (1-\lambda)\mathbf{w}') - (\lambda \mathcal{L}(\mathbf{w}) + (1-\lambda)\mathcal{L}(\mathbf{w}')) \leq 0 \,. \tag{21}$$

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.

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

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

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left( \mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w} \right)^2 , \tag{20}$$

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”.

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

$$\mathcal{L}(\lambda \mathbf{w} + (1-\lambda)\mathbf{w}') - (\lambda \mathcal{L}(\mathbf{w}) + (1-\lambda)\mathcal{L}(\mathbf{w}')) \leq 0 \,. \tag{21}$$

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.

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

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

$$\mathcal{L} := \frac{1}{2N} \sum_{n=1}^{N} \left(\mathcal{L}_n := y_n - \mathbf{x}_n^\top \mathbf{w}\right)^2 , \tag{20}$$

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".

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

$$\mathcal{L}(\lambda \mathbf{w} + (1-\lambda)\mathbf{w}') - (\lambda \mathcal{L}(\mathbf{w}) + (1-\lambda)\mathcal{L}(\mathbf{w}')) \leq 0 \,. \tag{21}$$

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.

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

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

$$\nabla\mathcal{L}(\mathbf{w}) = -\frac{1}{N}\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}). \tag{22}$$

# 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

$$\nabla \mathcal{L}(\mathbf{w}) = -\frac{1}{N}\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}). \tag{22}$$

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

$$\nabla \mathcal{L}(\mathbf{w}) = -\frac{1}{N}\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}). \tag{22}$$

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

$$\mathbf{X}^\top \underbrace{(\mathbf{y} - \mathbf{X}\mathbf{w})}_{\text{error}} = \mathbf{0}\,, \tag{23}$$

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

# Geometric Interpretation

## Definition 9 (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$\{\mathbf{x}_1, \ldots, \mathbf{x}_k\} = \{\alpha_1 \mathbf{x}_1 + \ldots + \alpha_k \mathbf{x}_k \,|\, \alpha_1, \ldots, \alpha_k \in \mathbb{R}\}$.

# Geometric Interpretation

**Definition 9 (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$\{\mathbf{x}_1, \ldots, \mathbf{x}_k\} = \{\alpha_1\mathbf{x}_1 + \ldots + \alpha_k\mathbf{x}_k \mid \alpha_1, \ldots, \alpha_k \in \mathbb{R}\}$.

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

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

# Geometric Interpretation

## Definition 9 (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 $\mathbf{X}$

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

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

# Geometric Interpretation

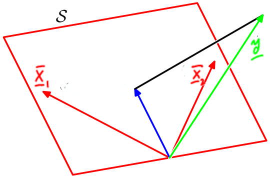
**Definition 9 (Span of a set of vectors)**

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 $\mathbf{X}$

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

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




(taken from Bishop's book)

# Geometric Interpretation

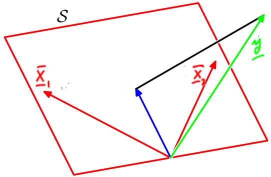
## Definition 9 (Span of a set of vectors)

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$\{\mathbf{x}_1, \ldots, \mathbf{x}_k\} = \{\alpha_1 \mathbf{x}_1 + \ldots + \alpha_k \mathbf{x}_k \mid \alpha_1, \ldots, \alpha_k \in \mathbb{R}\}$.

- 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? (for 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

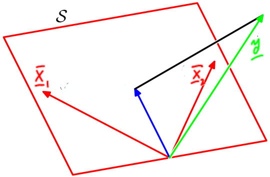
## Definition 9 (Span of a set of vectors)

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 $\mathbf{X}$

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

Which element $\mathbf{u}$ of $span(\mathbf{X})$ shall we take? (for 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

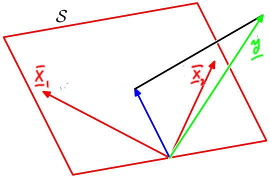
**Definition 9 (Span of a set of vectors)**

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$\{\mathbf{x}_1, \ldots, \mathbf{x}_k\} = \{\alpha_1 \mathbf{x}_1 + \ldots + \alpha_k \mathbf{x}_k \mid \alpha_1, \ldots, \alpha_k \in \mathbb{R}\}$.

- 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? (for 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

$$\mathbf{X}^\top y = \underbrace{\mathbf{X}^\top\mathbf{X}}_{\text{Gram matrix}} \mathbf{w} \tag{24}$$

# 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

$$\mathbf{X}^\top y = \underbrace{\mathbf{X}^\top\mathbf{X}}_{\text{Gram matrix}} \mathbf{w} \tag{24}$$

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

$$\mathbf{X}^\top y = \underbrace{\mathbf{X}^\top \mathbf{X}}_{\text{Gram matrix}} \mathbf{w} \tag{24}$$

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

$$\mathbf{w}^\star = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{y}\,, \tag{25}$$

# 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

$$\mathbf{X}^\top y = \underbrace{\mathbf{X}^\top\mathbf{X}}_{\text{Gram matrix}} \mathbf{w} \tag{24}$$

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

$$\mathbf{w}^\star = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}\,, \tag{25}$$

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

$$\mathbf{X}^\top y = \underbrace{\mathbf{X}^\top\mathbf{X}}_{\text{Gram matrix}} \mathbf{w} \tag{24}$$

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

$$\mathbf{w}^\star = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}\,, \tag{25}$$

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) $\mathbf{x}_m$:

$$\hat{y}_m := \mathbf{x}_m^\top\mathbf{w}^\star = \mathbf{x}_m^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}. \tag{26}$$

# 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

$$\mathbf{X}^\top y = \underbrace{\mathbf{X}^\top \mathbf{X}}_{\text{Gram matrix}} \mathbf{w} \tag{24}$$

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

$$\mathbf{w}^\star = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}\,, \tag{25}$$

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) $\mathbf{x}_m$:

$$\hat{y}_m := \mathbf{x}_m^\top \mathbf{w}^\star = \mathbf{x}_m^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}. \tag{26}$$

## Remark 10

*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., $\text{np.linalg.solve}(\mathbf{X}, \mathbf{y})$.

# Table of Contents

# Recall: Gaussian distribution and independence

## Definition 11 (A Gaussian random variable)

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

$$p(y \mid \mu, \sigma^2) = \mathcal{N}(y \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(y-\mu)^2}{2\sigma^2}\right\}. \tag{27}$$

# Recall: Gaussian distribution and independence

## Definition 11 (A Gaussian random variable)

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

$$p(y \mid \mu, \sigma^2) = \mathcal{N}(y \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(y-\mu)^2}{2\sigma^2}\right\}. \tag{27}$$

## Definition 12 (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

$$\mathcal{N}(\mathbf{y} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D \, det(\boldsymbol{\Sigma})}} \exp\left\{-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right\}. \tag{28}$$

# Recall: Gaussian distribution and independence

## Definition 11 (A Gaussian random variable)

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

$$p(y \mid \mu, \sigma^2) = \mathcal{N}(y \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(y-\mu)^2}{2\sigma^2}\right\}. \tag{27}$$

## Definition 12 (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

$$\mathcal{N}(\mathbf{y} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D \, det(\boldsymbol{\Sigma})}} \exp\left\{-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right\}. \tag{28}$$

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

# A probabilistic model for least-squares

## Definition 13 (Data generation process)

We assume that the data is generated by the model,

$$y_n = \mathbf{x}_n^\top \mathbf{w} + \epsilon_n, \tag{29}$$

where

# A probabilistic model for least-squares

## Definition 13 (Data generation process)

We assume that the data is generated by the model,

$$y_n = \mathbf{x}_n^\top \mathbf{w} + \epsilon_n, \tag{29}$$

where

- the $\epsilon_n$ (the noise) is a zero-mean Gaussian random variable with variance $\sigma^2$

# A probabilistic model for least-squares

## Definition 13 (Data generation process)

We assume that the data is generated by the model,

$$y_n = \mathbf{x}_n^\top \mathbf{w} + \epsilon_n, \tag{29}$$

where

- the $\epsilon_n$ (the noise) is a zero-mean Gaussian random variable with variance $\sigma^2$
- the noise is independent of each other and independent of the input.

# A probabilistic model for least-squares

## Definition 13 (Data generation process)

We assume that the data is generated by the model,

$$y_n = \mathbf{x}_n^\top \mathbf{w} + \epsilon_n, \tag{29}$$

where

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

# A probabilistic model for least-squares

## Definition 13 (Data generation process)

We assume that the data is generated by the model,

$$y_n = \mathbf{x}_n^\top \mathbf{w} + \epsilon_n, \tag{29}$$

where

- the $\epsilon_n$ (the noise) is a zero-mean Gaussian random variable with variance $\sigma^2$
- 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

$$p(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} p(y_n \mid \mathbf{x}_n, \mathbf{w}) = \prod_{n=1}^{N} \mathcal{N}(y_n \mid \mathbf{x}_n^\top \mathbf{w}, \sigma^2). \tag{30}$$

# A probabilistic model for least-squares

## Definition 13 (Data generation process)

We assume that the data is generated by the model,

$$y_n = \mathbf{x}_n^\top \mathbf{w} + \epsilon_n, \tag{29}$$

where

- the $\epsilon_n$ (the noise) is a zero-mean Gaussian random variable with variance $\sigma^2$
- 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

$$p(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} p(y_n \mid \mathbf{x}_n, \mathbf{w}) = \prod_{n=1}^{N} \mathcal{N}(y_n \mid \mathbf{x}_n^\top \mathbf{w}, \sigma^2). \tag{30}$$

**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 take the logarithm of the likelihood, i.e., log-likelihood (LL):

$$\mathcal{L}_{\mathsf{LL}}(\mathbf{w}) := \log\, p(\mathbf{y} \,|\, \mathbf{X}, \mathbf{w}) = -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\top \mathbf{w})^2 + \text{cnst.} \tag{31}$$

# Maximum-likelihood estimator (MLE)

Instead of maximizing the likelihood, we can maximize take the logarithm of the likelihood, i.e., log-likelihood (LL):

$$\mathcal{L}_{\mathsf{LL}}(\mathbf{w}) := \log p(\mathbf{y} \,|\, \mathbf{X}, \mathbf{w}) = -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\top \mathbf{w})^2 + \text{cnst}. \tag{31}$$

Compare the LL to the MSE (Mean Squared Error)

$$\mathcal{L}_{\mathsf{LL}}(\mathbf{w}) = -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\top \mathbf{w})^2 + \text{cnst} \tag{32}$$

$$\mathcal{L}_{\mathsf{MSE}}(\mathbf{w}) = \frac{1}{2N} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\top \mathbf{w})^2 \tag{33}$$

# Maximum-likelihood estimator (MLE)

Instead of maximizing the likelihood, we can maximize take the logarithm of the likelihood, i.e., log-likelihood (LL):

$$\mathcal{L}_{\text{LL}}(\mathbf{w}) := \log p(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) = -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\top \mathbf{w})^2 + \text{cnst}. \tag{31}$$

Compare the LL to the MSE (Mean Squared Error)

$$\mathcal{L}_{\text{LL}}(\mathbf{w}) = -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\top \mathbf{w})^2 + \text{cnst} \tag{32}$$

$$\mathcal{L}_{\text{MSE}}(\mathbf{w}) = \frac{1}{2N} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\top \mathbf{w})^2 \tag{33}$$

Maximizing the LL is equivalent to minimizing the MSE:

$$\arg\min_{\mathbf{w}} \mathcal{L}_{\text{MSE}}(\mathbf{w}) = \arg\max_{\mathbf{w}} \mathcal{L}_{\text{LL}}(\mathbf{w}). \tag{34}$$

# Properties of MLE

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

$$\mathcal{L}_{\mathsf{LL}}(\mathbf{w}) \approx \mathbb{E}_{p(y,\mathbf{x})}\left[\log p(y \mid \mathbf{x}, \mathbf{w})\right] \tag{35}$$

# Properties of MLE

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

$$\mathcal{L}_{\text{LL}}(\mathbf{w}) \approx \mathbb{E}_{p(y,\mathbf{x})}\left[\log p(y \mid \mathbf{x}, \mathbf{w})\right] \tag{35}$$

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

# Properties of MLE

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

$$\mathcal{L}_{\mathsf{LL}}(\mathbf{w}) \approx \mathbb{E}_{p(y,\mathbf{x})}\left[\log p(y \mid \mathbf{x}, \mathbf{w})\right] \tag{35}$$

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*:

$$\mathcal{L}_{\mathsf{LL}}(\mathbf{w}) \approx \mathbb{E}_{p(y,\mathbf{x})}\left[\log p(y \mid \mathbf{x}, \mathbf{w})\right] \tag{35}$$

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)

$$\mathbf{w}_{\mathsf{MLE}} \longrightarrow^{p} \mathbf{w}_{\mathsf{true}} \quad \text{in probability} \tag{36}$$

# Properties of MLE

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

$$\mathcal{L}_{\text{LL}}(\mathbf{w}) \approx \mathbb{E}_{p(y,\mathbf{x})}\left[\log p(y \mid \mathbf{x}, \mathbf{w})\right] \tag{35}$$

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)

$$\mathbf{w}_{\text{MLE}} \quad \longrightarrow^{p} \quad \mathbf{w}_{\text{true}} \qquad \text{in probability} \tag{36}$$

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

$$(\mathbf{w}_{\text{MLE}} - \mathbf{w}_{\text{true}}) \quad \longrightarrow^{d} \quad \frac{1}{\sqrt{N}}\mathcal{N}(\mathbf{w}_{\text{MLE}} \mid \mathbf{0}, \mathbf{F}^{-1}(\mathbf{w}_{\text{true}}))\,, \tag{37}$$

   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*:

$$\mathcal{L}_{\text{LL}}(\mathbf{w}) \approx \mathbb{E}_{p(y,\mathbf{x})}\left[\log p(y \mid \mathbf{x}, \mathbf{w})\right] \tag{35}$$

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)

$$\mathbf{w}_{\text{MLE}} \;\xrightarrow{}^{p}\; \mathbf{w}_{\text{true}} \quad \text{in probability} \tag{36}$$

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

$$(\mathbf{w}_{\text{MLE}} - \mathbf{w}_{\text{true}}) \;\xrightarrow{}^{d}\; \frac{1}{\sqrt{N}}\mathcal{N}(\mathbf{w}_{\text{MLE}} \mid \mathbf{0}, \mathbf{F}^{-1}(\mathbf{w}_{\text{true}}))\,, \tag{37}$$

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.

$$\text{Covariance}(\mathbf{w}_{\text{MLE}}) = \mathbf{F}^{-1}(\mathbf{w}_{\text{true}}) \tag{38}$$

# Another example

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

$$p(y_n \mid \mathbf{x}_n, \mathbf{w}) = \frac{1}{2b} e^{-\frac{1}{b}\left|y_n - \mathbf{x}_n^\top \mathbf{w}\right|} \tag{39}$$

# Another example

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

$$p(y_n \mid \mathbf{x}_n, \mathbf{w}) = \frac{1}{2b} e^{-\frac{1}{b}\left|y_n - \mathbf{x}_n^\top \mathbf{w}\right|} \tag{39}$$

we can recover the MAE cost function!

**This lecture:**

- Basic concept of regression and classification
- Linear regression
  - Definition
  - Gradient Descent (GD) optimization
  - Least Square
  - The probabilistic interpretation of Linear Regression

**Next lecture:**

- Over-fitting and under-fitting
- Polynomial regression and Ridge regression
- Model selection
- Bias-Variance Decomposition