

# Maximum Likelihood Estimation (MLE)

## Regularizations

Faculty of Computer Science  
University of Information Technology (UIT)  
Vietnam National University - Ho Chi Minh City (VNU-HCM)

Math for Computer Science, Fall 2023

The contents of this document are taken mainly from the follow sources:

- Kevin P. Murphy. Probabilistic Machine Learning: An Introduction. <sup>1</sup>

---

<sup>1</sup><https://probml.github.io/pml-book/book1.html>

# Table of Contents

- 1 Introduction
- 2 Maximum Likelihood Estimation
- 3 Examples
- 4 MLE for Linear Regression
- 5 Regularization

# Table of Contents

- 1 Introduction
- 2 Maximum Likelihood Estimation
- 3 Examples
- 4 MLE for Linear Regression
- 5 Regularization

# Introduction

- The process of estimating  $\theta$  from  $\mathcal{D}$  is called **model fitting**, or **training**, is at the heart of machine learning.
- There are many methods for estimating  $\theta$ , and they involve an optimization problem of the form

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \mathcal{L}(\theta)$$

where  $\mathcal{L}(\theta)$  is some kind of loss function or objective function.

- The process of quantifying uncertainty about an unknown quantity estimated from a finite sample of data is called **inference**.
- In deep learning, the term “inference” refers to “prediction”, namely computing

$$p(y|x, \hat{\theta})$$

# Table of Contents

- 1 Introduction
- 2 Maximum Likelihood Estimation
- 3 Examples
- 4 MLE for Linear Regression
- 5 Regularization

# Maximum Likelihood Estimation

- The most common approach to parameter estimation is to pick the parameters that assign the highest probability to the training data. This is called **maximum likelihood estimation** or **MLE**.

$$\hat{\theta}_{\text{mle}} = \underset{\theta}{\operatorname{argmax}} p(\mathcal{D}|\theta)$$

- We usually assume the training examples are “independent and identically distributed”, and are sampled from the same distribution (i.e., the **iid** assumption). The conditional likelihood becomes

$$p(\mathcal{D}|\theta) = p(y_1, y_2, \dots, y_N | x_1, x_2, \dots, x_N, \theta) = \prod_{n=1}^N p(y_n | x_n, \theta)$$

- We usually work with the **log likelihood**, which decomposes into a sum of terms, one per example.

$$\text{LL}(\theta) = \log p(\mathcal{D}|\theta) = \log \prod_{n=1}^N p(y_n | x_n, \theta) = \sum_{n=1}^N \log p(y_n | x_n, \theta)$$

# Maximum Likelihood Estimation

- The MLE is given by

$$\hat{\theta}_{\text{MLE}} = \underset{\theta}{\operatorname{argmax}} \sum_{n=1}^N \log p(y_n | \mathbf{x}_n, \theta)$$

- Because most optimization algorithms are designed to *minimize* cost functions, we redefine the objective function to be the conditional **negative log likelihood** or **NLL**:

$$\text{NLL}(\theta) = -\log p(\mathcal{D} | \theta) = -\sum_{n=1}^N \log p(y_n | \mathbf{x}_n, \theta)$$

- Minimizing this will give the MLE.

$$\hat{\theta}_{\text{MLE}} = \underset{\theta}{\operatorname{argmin}} -\sum_{n=1}^N \log p(y_n | \mathbf{x}_n, \theta)$$



# Table of Contents

- 1 Introduction
- 2 Maximum Likelihood Estimation
- 3 Examples**
- 4 MLE for Linear Regression
- 5 Regularization

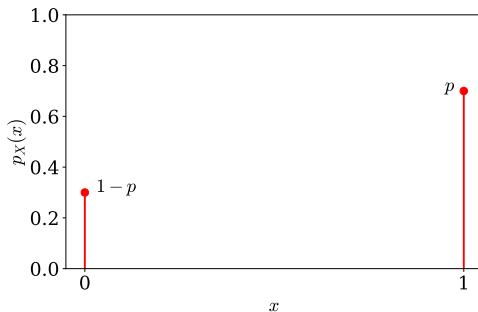
- A Bernoulli r.v.  $X$  takes two possible values, usually 0 and 1, modeling random experiments that have two possible outcomes (e.g., “success” and “failure”).
  - e.g., tossing a coin. The outcome is either Head or Tail.
  - e.g., taking an exam. The result is either Pass or Fail.
  - e.g., classifying images. An image is either Cat or Non-cat.

# Bernoulli Random Variables

## Definition

A random variable  $X$  is a Bernoulli random variable with parameter  $p \in [0, 1]$ , written as  $X \sim \text{Bernoulli}(p)$  if its PMF is given by

$$P_X(x) = \begin{cases} p, & \text{for } x = 1 \\ 1 - p, & \text{for } x = 0. \end{cases}$$



## Example

- A bag contains 3 balls, each ball is either red or blue.
- The number of blue balls  $\theta$  can be 0, 1, 2, 3.
- Choose 4 balls randomly **with replacement**.
- Random variables  $X_1, X_2, X_3, X_4$  are defined as

$$X_i = \begin{cases} 1, & \text{if the } i\text{-th chosen ball is blue} \\ 0, & \text{if the } i\text{-th chosen ball is red} \end{cases}$$

- After doing the experiment, the following values for  $X_i$ 's are observed:  $x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1$ .
- Note that  $X_i$ 's are i.i.d. (independent and identically distributed) and  $X_i \sim \text{Bernoulli}(\frac{\theta}{3})$ . For which value of  $\theta$  is the probability of the observed sample is the largest?

## Example

$$P_{X_i}(x) = \begin{cases} \frac{\theta}{3}, & \text{for } x = 1 \\ 1 - \frac{\theta}{3}, & \text{for } x = 0 \end{cases}$$

$X_i$ 's are independent, the joint PMF of  $X_1, X_2, X_3, X_4$  can be written

$$P_{X_1 X_2 X_3 X_4}(x_1, x_2, x_3, x_4) = P_{X_1}(x_1)P_{X_2}(x_2)P_{X_3}(x_3)P_{X_4}(x_4)$$

$$P_{X_1 X_2 X_3 X_4}(1, 0, 1, 1) = \frac{\theta}{3} \cdot \left(1 - \frac{\theta}{3}\right) \cdot \frac{\theta}{3} \cdot \frac{\theta}{3} = \left(\frac{\theta}{3}\right)^3 \left(1 - \frac{\theta}{3}\right)$$

$\theta$	$P_{X_1 X_2 X_3 X_4}(1, 0, 1, 1; \theta)$
0	0
1	0.0247
2	0.0988
3	0

The observed data is most likely to occur for  $\theta = 2$ .

We may choose  $\hat{\theta} = 2$  as our estimate of  $\theta$ .

# MLE for the Bernoulli distribution

- Suppose  $Y$  is a random variable representing a coin toss.
- The event  $Y = 1$  corresponds to heads,  $Y = 0$  corresponds to tails.
- The probability distribution for this rv is the Bernoulli. The NLL for the Bernoulli distribution is

$$\begin{aligned}\text{NLL}(\theta) &= -\log \prod_{n=1}^N p(y_n|\theta) = -\log \prod_{n=1}^N \theta^{\mathbb{I}(y_n=1)}(1-\theta)^{\mathbb{I}(y_n=0)} \\ &= -\sum_{n=1}^N \mathbb{I}(y_n = 1) \log \theta + \mathbb{I}(y_n = 0) \log(1 - \theta) \\ &= -[N_1 \log \theta + N_0 \log(1 - \theta)]\end{aligned}$$

where  $N_1 = \sum_{n=1}^N \mathbb{I}(y_n = 1)$  is the number of heads, and  $N_0 = \sum_{n=1}^N \mathbb{I}(y_n = 0)$  is the number of tails.

- $N = N_0 + N_1$  is the **sample size**.

# MLE for the Bernoulli distribution

$$\text{NLL}(\theta) = -[N_1 \log \theta + N_0 \log(1 - \theta)]$$

- The derivative of the NLL is

$$\frac{d}{d\theta} \text{NLL}(\theta) = \frac{-N_1}{\theta} + \frac{N_0}{1 - \theta}$$

- The MLE can be found by solving  $\frac{d}{d\theta} \text{NLL}(\theta) = 0$ .
- The MLE is given by

$$\hat{\theta}_{\text{mle}} = \frac{N_1}{N_0 + N_1}$$

which is the **empirical** fraction of heads.

# MLE for the categorical distribution

- Suppose we roll a  $K$ -sided dice  $N$  times.
- Let  $Y_n \in \{1, \dots, K\}$  be the  $n$ -th outcome, where  $Y_n \sim \text{Cat}(\boldsymbol{\theta})$ .
- We want to estimate  $\boldsymbol{\theta}$  from the dataset  $\mathcal{D}\{y_n : n = 1 : N\}$ .
- The NLL is given by

$$\text{NLL}(\boldsymbol{\theta}) = - \sum_k N_k \log \theta_k$$

where  $N_k$  is the number of times the event  $Y = k$  is observed.

- To compute the MLE, we have to minimize the NLL subject to the **constraint** that

$$\sum_{k=1}^K \theta_k = 1$$



# MLE for the categorical distribution

- We use the method of Lagrange multipliers. The Lagrangian is as

$$\mathcal{L}(\boldsymbol{\theta}, \lambda) = - \sum_k N_k \log \theta_k - \lambda \left( 1 - \sum_k \theta_k \right)$$

- Taking derivatives with respect to  $\lambda$  yields the original constraint

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 1 - \sum_k \theta_k = 0$$

- Taking derivatives with respect to  $\theta_k$  yields

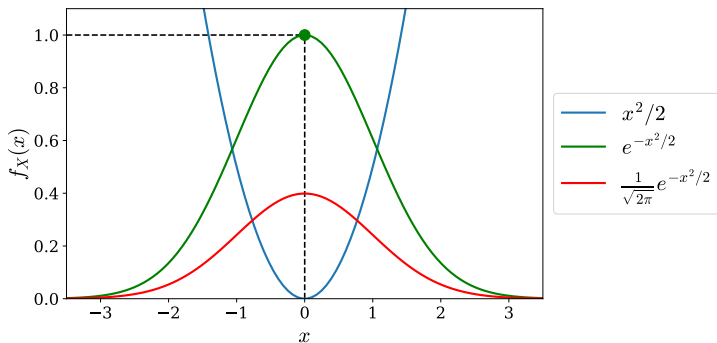
$$\frac{\partial \mathcal{L}}{\partial \theta_k} = -\frac{N_k}{\theta_k} + \lambda = 0 \longrightarrow N_k = \lambda \theta_k$$

- We can solve for  $\lambda$  using the sum-to-one constraint

$$\sum_k N_k = N = \lambda \sum_k \theta_k = \lambda$$

- Thus the MLE is given by  $\hat{\theta}_k = \frac{N_k}{\lambda} = \frac{N_k}{N}$ , the **empirical** fraction of times event  $k$  occurs.

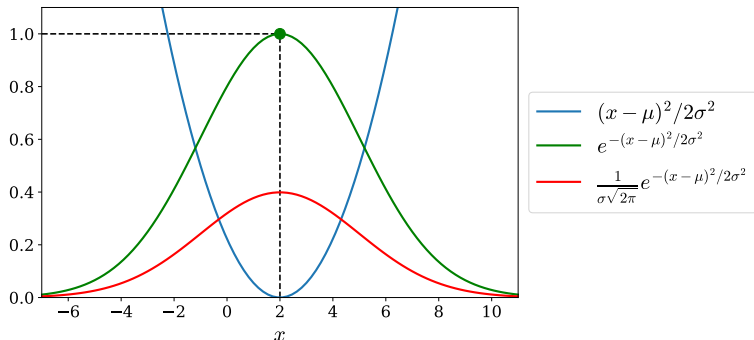
# Standard Normal (Gaussian) Random Variable $N(0, 1)$



$$\int_{-\infty}^{\infty} e^{-x^2/2} dx = \sqrt{2\pi}$$

$$f_X(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

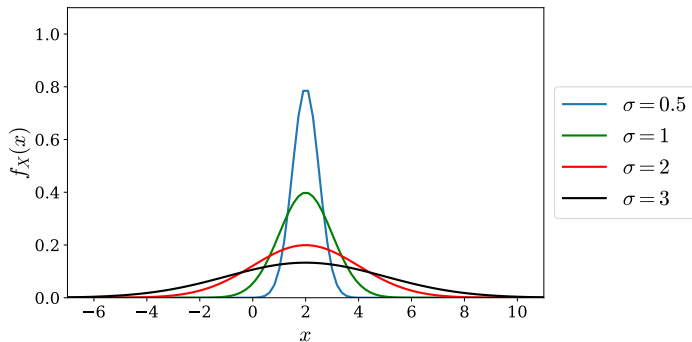
# General Normal (Gaussian) Random Variable $N(\mu, \sigma^2)$



$$\begin{aligned} f_X(x) &= \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \\ &= \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{1}{2\sigma^2}(y-\mu)^2\right) \end{aligned}$$

$$\mathbb{E}[X] = \mu \quad \text{Var}(X) = \sigma^2$$

# General Normal (Gaussian) Random Variable $N(\mu, \sigma^2)$



- Smaller  $\sigma$ , narrower PDF.
- Let  $Y = aX + b$        $N \sim N(\mu, \sigma^2)$
- Then,  $E[Y] = aE[X] + b$        $\text{Var}(Y) = a^2\sigma^2$  (always true)
- But also,  $Y \sim N(a\mu + b, a^2\sigma^2)$

# Example

- We have  $N = 3$  data points  $y_1 = 1$ ,  $y_2 = 0.5$ ,  $y_3 = 1.5$  which are independent and Gaussian with **unknown** mean  $\mu$  and variance 1:

$$y_i \sim \mathcal{N}(\mu, 1)$$

- Likelihood  $P(y_1 y_2 y_3 | \mu) = P(y_1 | \mu) P(y_2 | \mu) P(y_3 | \mu)$ .
- Consider two guesses  $\mu = 1.0$  and  $\mu = 2.5$ . Which has higher likelihood?
- Finding the  $\mu$  that maximizes the likelihood is equivalent to moving the Gaussian until the product  $P(y_1 | \mu) P(y_2 | \mu) P(y_3 | \mu)$  is maximized.

# MLE for the univariate Gaussian

- $Y \sim \mathcal{N}(\mu, \sigma^2)$  and  $\mathcal{D} = \{y_n : n = 1 : N\}$  be an iid sample of size  $N$ .

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

- We can estimate the parameters  $\theta = (\mu, \sigma^2)$  using MLE.
- We derive the NLL, which is given by

$$\begin{aligned} \text{NLL}(\mu, \sigma^2) &= -\sum_{n=1}^N \log \left[ \left( \frac{1}{2\pi\sigma^2} \right)^{\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2}(y_n - \mu)^2\right) \right] \\ &= \frac{1}{2\sigma^2} \sum_{n=1}^N (y_n - \mu)^2 + \frac{N}{2} \log(2\pi\sigma^2) \end{aligned}$$

- The minimum of this function must satisfy the following conditions

$$\frac{\partial}{\partial \mu} \text{NLL}(\mu, \sigma^2) = 0, \quad \frac{\partial}{\partial \sigma^2} \text{NLL}(\mu, \sigma^2) = 0$$

# MLE for the univariate Gaussian

- The solution is given by

$$\hat{\mu}_{\text{MLE}} = \frac{1}{N} \sum_{n=1}^N y_n = \bar{y}$$

$$\hat{\sigma}_{\text{MLE}}^2 = \frac{1}{N} \sum_{n=1}^N (y_n - \hat{\mu}_{\text{MLE}})^2 = \frac{1}{N} \left[ \sum_{n=1}^N y_n^2 + \hat{\mu}_{\text{MLE}}^2 - 2y_n \hat{\mu}_{\text{MLE}} \right] = s^2 - \bar{y}^2$$

$$s^2 \triangleq \frac{1}{N} \sum_{n=1}^N y_n^2$$

- The quantities  $\bar{y}$  and  $s^2$  are called the **sufficient statistics** of the data because they are sufficient to compute the MLE.
- Sometimes, we might see the estimate for the variance as

$$\hat{\sigma}^2 = \frac{1}{N-1} \sum_{n=1}^N (y_n - \hat{\mu}_{\text{MLE}})^2$$

which is not the MLE, but is a different kind of estimate.

# Table of Contents

- 1 Introduction
- 2 Maximum Likelihood Estimation
- 3 Examples
- 4 MLE for Linear Regression**
- 5 Regularization



# MLE for linear regression

- We can make the parameters of the Gaussian to be functions of some input variables

$$p(y|\mathbf{x}; \boldsymbol{\theta}) = \mathcal{N}(y|f_{\mu}(\mathbf{x}; \boldsymbol{\theta}), f_{\sigma}(\mathbf{x}; \boldsymbol{\theta})^2)$$

$f_{\mu}(\mathbf{x}; \boldsymbol{\theta}) \in \mathbb{R}$  predicts mean, and  $f_{\sigma}(\mathbf{x}; \boldsymbol{\theta}) \in \mathbb{R}_+$  predicts variance.

- It is common to assume that the variance is *fixed*, and is *independent* of the input. This is called **homoscedastic regression**.
- Furthermore, it is common to assume the mean is a linear function of the input. The resulting model is called **linear regression**.

$$p(y|\mathbf{x}; \boldsymbol{\theta}) = \mathcal{N}(y|\mathbf{w}^T \mathbf{x} + b, \sigma^2)$$

where  $\boldsymbol{\theta} = (\mathbf{w}, b, \sigma)$ .

# MLE for linear regression

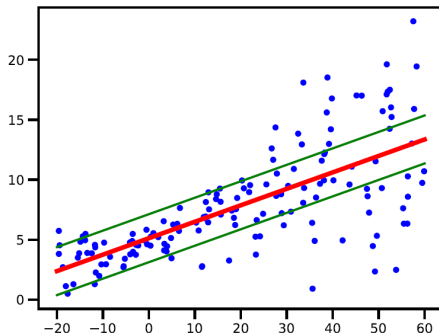


Figure: Linear regression using Gaussian output with **mean**  $\mu(x) = b + wx$  and fixed **variance**  $\sigma^2$ .

- The figure plots the 95% predictive interval  $[\mu(x) - 2\sigma, \mu(x) + 2\sigma]$ .
- This is the uncertainty in the predicted *observation*  $y$  given  $x$ , and capture the variability in the **blue dots**.

# MLE for linear regression

- Linear regression model

$$p(y|\mathbf{x}; \boldsymbol{\theta}) = \mathcal{N}(y|\mathbf{w}^T \mathbf{x}, \sigma^2)$$

where  $\boldsymbol{\theta} = (\mathbf{w}, \sigma^2)$ , and  $\mathbf{w} = (b, w_1, w_2, \dots, w_D)$ .

- Assume that  $\sigma^2$  is fixed, we estimate the weights  $\mathbf{w}$ . The NLL is

$$\text{NLL}(\mathbf{w}) = - \sum_{n=1}^N \log \left[ \left( \frac{1}{2\pi\sigma^2} \right)^{\frac{1}{2}} \exp \left( - \frac{1}{2\sigma^2} (y_n - \mathbf{w}^T \mathbf{x}_n)^2 \right) \right]$$

- Dropping the *irrelevant* **additive constants** gives the simplified objective, known as the **residual sum of squares** or **RSS**:

$$\text{RSS}(\mathbf{w}) = \sum_{n=1}^N (y_n - \mathbf{w}^T \mathbf{x}_n)^2 = \sum_{n=1}^N r_n^2$$

where  $r_n$  is the  $n$ -th **residual error**.

# MLE for linear regression

- **Residual sum of squares** or **RSS**:

$$\text{RSS}(\mathbf{w}) = \sum_{n=1}^N (y_n - \mathbf{w}^\top \mathbf{x}_n)^2$$

- **Mean squared error** or **MSE**:

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

- **Root mean squared error** or **RMSE**:

$$\text{RMSE}(\mathbf{w}) = \sqrt{\text{MSE}(\mathbf{w})} = \sqrt{\frac{1}{N} \sum_{n=1}^N (y_n - \mathbf{w}^\top \mathbf{x}_n)^2}$$

- We can compute the MLE by minimizing the NLL, RSS, MSE, or RMSE. All give the same results.

# MLE for linear regression

- The RSS can be written in matrix notation as follows

$$\text{RSS}(\mathbf{w}) = \sum_{n=1}^N (y_n - \mathbf{w}^\top \mathbf{x}_n)^2 = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 = (\mathbf{X}\mathbf{w} - \mathbf{y})^\top (\mathbf{X}\mathbf{w} - \mathbf{y})$$

- The gradient is given by

$$\nabla_{\mathbf{w}} \text{RSS}(\mathbf{w}) = \mathbf{X}^\top \mathbf{X} \mathbf{w} - \mathbf{X}^\top \mathbf{y}$$

- Setting the gradient to zero  $\nabla_{\mathbf{w}} \text{RSS}(\mathbf{w}) = \mathbf{0}$  and solving gives

$$\mathbf{X}^\top \mathbf{X} \mathbf{w} = \mathbf{X}^\top \mathbf{y}$$

- These are known as the **normal equations**.
- The MLE solution  $\hat{\mathbf{w}}_{\text{MLE}}$  is called the **ordinary least squares (OLS)** solution:

$$\hat{\mathbf{w}}_{\text{MLE}} = \underset{\mathbf{w}}{\text{argmin}} \text{RSS}(\mathbf{w}) = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

# MLE for linear regression

$$\hat{\mathbf{w}}_{\text{MLE}} = \underset{\mathbf{w}}{\operatorname{argmin}} \operatorname{RSS}(\mathbf{w}) = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

- The quantity  $\mathbf{X}^\dagger = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top$  is the (left) pseudo-inverse of the (non-square) matrix  $\mathbf{X}$ .
- Is the solution  $\hat{\mathbf{w}}_{\text{MLE}}$  unique?
- The gradient is  $\nabla_{\mathbf{w}} \operatorname{RSS}(\mathbf{w}) = \mathbf{X}^\top \mathbf{X} \mathbf{w} - \mathbf{X}^\top \mathbf{y}$ . Then, the **Hessian** is

$$\mathbf{H}(\mathbf{w}) = \frac{\partial^2}{\partial \mathbf{w}^2} \operatorname{RSS}(\mathbf{w}) = \mathbf{X}^\top \mathbf{X}$$

- If  $\mathbf{X}$  is **full rank** (i.e., the columns of  $\mathbf{X}$  are linearly independent), then  $\mathbf{H}$  is **positive definite**, since for any  $\mathbf{v}$ , we have

$$\mathbf{v}^\top (\mathbf{X}^\top \mathbf{X}) \mathbf{v} = (\mathbf{X} \mathbf{v})^\top (\mathbf{X} \mathbf{v}) = \|\mathbf{X} \mathbf{v}\|^2 > 0$$

- In the full rank case, the  $\operatorname{RSS}(\mathbf{w})$  has a **unique global minimum**.

# Table of Contents

- 1 Introduction
- 2 Maximum Likelihood Estimation
- 3 Examples
- 4 MLE for Linear Regression
- 5 Regularization**

# Overfitting

- MLE will try to pick parameters that minimize loss on the training set, but this may not result in a model that has low loss on future data. This is called **overfitting**.
- Ex: We want to predict the probability of heads when tossing a coin.
- We toss it  $N = 3$  times and observe 3 heads. The MLE is

$$\hat{\theta}_{\text{MLE}} = \frac{N_1}{N_0 + N_1} = \frac{3}{3 + 0} = 1$$

- If we use this  $\text{Ber}(y|\hat{\theta}_{\text{MLE}})$  to make predictions, we will predict that all future coin tosses will also be heads!!!
- The model has enough parameters to perfectly fit the observed training data, so it can perfectly match the **empirical** distribution.
- In most cases, the **empirical** distribution is not the same as the **true** distribution. Putting all the probability mass on the observed set of  $N$  examples will not leave over any probability for novel data in the future. The model may not **generalize**.



# Example: MLE for Linear Regression

## Example 1:

- Training data:  $\mathbf{x}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ ,  $y_1 = 1$     $\mathbf{x}_2 = \begin{bmatrix} 1 \\ \epsilon \end{bmatrix}$ ,  $y_2 = 1$ .
- $\mathbf{X} = \begin{bmatrix} 1 & 0 \\ 1 & \epsilon \end{bmatrix}$ ,    $\mathbf{y} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
- $\hat{\mathbf{w}}_{\text{mle}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} = ?$

## Example 2:

- Training data:  $\mathbf{x}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ ,  $y_1 = 1 + \epsilon$     $\mathbf{x}_2 = \begin{bmatrix} 1 \\ \epsilon \end{bmatrix}$ ,  $y_2 = 1$ .
- $\mathbf{X} = \begin{bmatrix} 1 & 0 \\ 1 & \epsilon \end{bmatrix}$ ,    $\mathbf{y} = \begin{bmatrix} 1 + \epsilon \\ 1 \end{bmatrix}$
- $\hat{\mathbf{w}}_{\text{mle}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} = ?$

# Example: MLE for Linear Regression

Example 1:

- Training data:  $\mathbf{x}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ ,  $y_1 = 1$     $\mathbf{x}_2 = \begin{bmatrix} 1 \\ \epsilon \end{bmatrix}$ ,  $y_2 = 1$ .
- $\mathbf{X} = \begin{bmatrix} 1 & 0 \\ 1 & \epsilon \end{bmatrix}$ ,  $\mathbf{y} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
- $\hat{\mathbf{w}}_{\text{mle}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$
- $\mathbf{X}^\top \mathbf{X} = \begin{bmatrix} 1 & 1 \\ 0 & \epsilon \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & \epsilon \end{bmatrix} = \begin{bmatrix} 2 & \epsilon \\ \epsilon & \epsilon^2 \end{bmatrix}$
- $(\mathbf{X}^\top \mathbf{X})^{-1} = \begin{bmatrix} 1 & -1/\epsilon \\ -1/\epsilon & 2/\epsilon^2 \end{bmatrix}$
- $\hat{\mathbf{w}}_{\text{mle}} = \begin{bmatrix} 1 & -1/\epsilon \\ -1/\epsilon & 2/\epsilon^2 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & \epsilon \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ .

# Example: MLE for Linear Regression

Example 2:

- Training data:  $\mathbf{x}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ ,  $y_1 = 1 + \epsilon$     $\mathbf{x}_2 = \begin{bmatrix} 1 \\ \epsilon \end{bmatrix}$ ,  $y_2 = 1$ .
- $\mathbf{X} = \begin{bmatrix} 1 & 0 \\ 1 & \epsilon \end{bmatrix}$ ,  $\mathbf{y} = \begin{bmatrix} 1 + \epsilon \\ 1 \end{bmatrix}$
- $\hat{\mathbf{w}}_{\text{mle}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$
- $\mathbf{X}^\top \mathbf{X} = \begin{bmatrix} 1 & 1 \\ 0 & \epsilon \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & \epsilon \end{bmatrix} = \begin{bmatrix} 2 & \epsilon \\ \epsilon & \epsilon^2 \end{bmatrix}$
- $(\mathbf{X}^\top \mathbf{X})^{-1} = \begin{bmatrix} 1 & -1/\epsilon \\ -1/\epsilon & 2/\epsilon^2 \end{bmatrix}$
- $\hat{\mathbf{w}}_{\text{mle}} = \begin{bmatrix} 1 & -1/\epsilon \\ -1/\epsilon & 2/\epsilon^2 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & \epsilon \end{bmatrix} \begin{bmatrix} 1 + \epsilon \\ 1 \end{bmatrix} = \begin{bmatrix} 1 + \epsilon \\ -1 \end{bmatrix}$ .

# Regularization

- The main solution to overfitting is to use **regularization**.
- We add a penalty term to the NLL (or empirical risk):

$$\mathcal{L}(\boldsymbol{\theta}; \lambda) = \left[ \frac{1}{N} \sum_{n=1}^N \ell(y_n, f(\mathbf{x}_n; \boldsymbol{\theta})) \right] + \lambda C(\boldsymbol{\theta})$$

where  $\lambda \geq 0$  is the **regularization parameter**, and  $C(\boldsymbol{\theta})$  is some form of **complexity penalty**.

- A common complexity penalty is to use  $C(\boldsymbol{\theta}) = -\log p(\boldsymbol{\theta})$ , where  $p(\boldsymbol{\theta})$  is the **prior** for  $\boldsymbol{\theta}$ .
- If  $\ell$  is the log loss, the regularized objective becomes

$$\mathcal{L}(\boldsymbol{\theta}; \lambda) = -\frac{1}{N} \sum_{n=1}^N \log p(y_n | \mathbf{x}_n, \boldsymbol{\theta}) - \lambda \log p(\boldsymbol{\theta})$$

# Maximum a posteriori estimation (MAP)

$$\mathcal{L}(\boldsymbol{\theta}; \lambda) = -\frac{1}{N} \sum_{n=1}^N \log p(y_n | x_n, \boldsymbol{\theta}) - \lambda \log p(\boldsymbol{\theta})$$

- By setting  $\lambda = 1$  and rescaling  $p(\boldsymbol{\theta})$  appropriately, we can equivalently minimize the following

$$\mathcal{L}(\boldsymbol{\theta}; \lambda) = -\left[ \sum_{n=1}^N \log p(y_n | x_n, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) \right] = -[\log p(\mathcal{D} | \boldsymbol{\theta}) + \log p(\boldsymbol{\theta})]$$

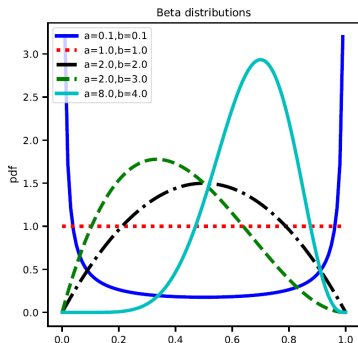
- Minimizing this is equivalent to maximizing the log posterior:

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{\text{map}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log p(\boldsymbol{\theta} | \mathcal{D}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log \frac{p(\mathcal{D} | \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D})} \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} [\log p(\mathcal{D} | \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) - \text{const}] \end{aligned}$$

- This is **MAP estimation**, or **maximum a posteriori estimation**.

# MAP estimation for Bernoulli distribution

- Coin tossing. If we observe just one head, the MLE is  $\hat{\theta}_{\text{MLE}} = 1$ .
- To avoid this, we can add a penalty to  $\theta$  to discourage “extreme” values, such as  $\theta = 0$  or  $\theta = 1$ .
- We can use a beta distribution as our prior  $p(\theta) = \text{Beta}(\theta|a, b)$ , where  $a, b > 1$  encourages values of  $\theta$  near to  $a/(a + b)$ .



- If  $a = b = 1$ , we get uniform distribution
- If  $a$  and  $b$  are both less than 1, we get bimodal distribution.
- If  $a$  and  $b$  are both greater than 1, the distribution is unimodal.

$$\text{mean} = \frac{a}{a + b}$$

$$\text{var} = \frac{ab}{(a + b)^2(a + b + 1)}$$

# MAP estimate for Bernoulli distribution

- Using the beta distribution as our prior  $p(\theta) = \text{Beta}(\theta|a, b)$ , the log likelihood plus log prior becomes

$$\begin{aligned}\text{LL}(\theta) &= \log p(\mathcal{D}|\theta) + \log p(\theta) \\ &= [N_1 \log \theta + N_0 \log(1 - \theta)] + [(a - 1) \log \theta + (b - 1) \log(1 - \theta)]\end{aligned}$$

- The MAP estimate is

$$\hat{\theta}_{\text{map}} = \frac{N_1 + a - 1}{N_1 + N_0 + a + b - 2}$$

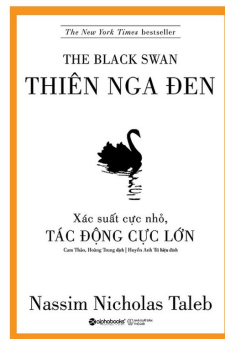
- If we set  $a = b = 2$ , that weakly favor a value of  $\theta$  near 0.5, the estimate becomes

$$\hat{\theta}_{\text{map}} = \frac{N_1 + 1}{N_1 + N_0 + 2}$$

- This is called **add-one smoothing** to avoid the **zero count problem**.

# Black swan paradox

- The zero-count problem, and overfitting, is analogous to the **black swan paradox**.
- It is used to illustrate the problem of **induction**: how to draw general conclusions about the future from specific observations from the past.
- The solution to the paradox is to admit that induction is *in general* impossible.
- The best we can do is to make plausible guesses by combining the **empirical data** with **prior knowledge**.





# Weight decay

- Polynomial regression with too much degree of freedom can result in overfitting. One solution is to reduce the degree of the polynomial.
- A more general solution is to **penalize** the **magnitude** of the weights (regression coefficients).
- We use a zero-mean Gaussian prior  $p(\mathbf{w})$ . The MAP estimate is

$$\hat{\mathbf{w}}_{\text{map}} = \underset{\mathbf{w}}{\operatorname{argmin}} \operatorname{NLL}(\mathbf{w}) + \lambda \|\mathbf{w}\|_2^2$$

where  $\|\mathbf{w}\|_2^2 = \sum_{d=1}^D w_d^2$ . We penalize the magnitude of weight vectors  $\mathbf{w}$ , rather than the bias term  $b$ .

- The equation is called  $\ell_2$  **regularization** or **weight decay**.
- The larger the value of  $\lambda$ , the more the parameters are penalized for being large (i.e., deviating from the zero-mean prior), and thus the less flexible the model.

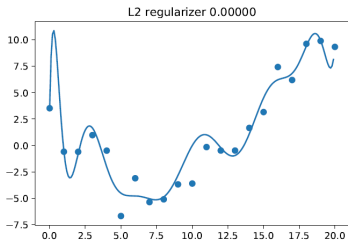
# Ridge regression

- In the case of linear regression, the weight decay penalization scheme is called **ridge regression**.
- Consider polynomial regression, where the predictor has the form

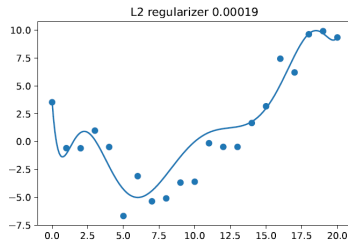
$$f(x; \mathbf{w}) = \sum_{d=0}^D w_d x^d = \mathbf{w}^\top [1, x, x^2, \dots, x^D]$$

- Suppose we use a high degree polynomial, say  $D = 14$ , even though we have a small dataset with just  $N = 21$  examples.
- MLE for the parameters will enable the model to fit the data very well, but the resulting function is very “wiggly”, thus resulting in overfitting.
- Increasing  $\lambda$  can reduce overfitting.

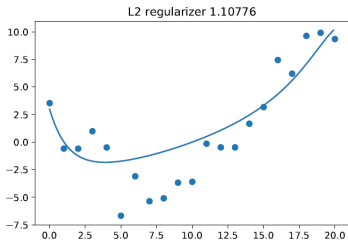
# Ridge regression



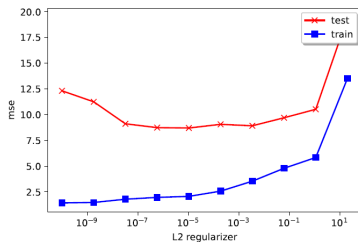
(a)



(b)



(c)



(d)

# Ridge regression

- MAP estimation with a zero-mean Gaussian prior

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|0, \lambda^{-1}\mathbf{I}).$$

$$\begin{aligned}\hat{\mathbf{w}}_{\text{map}} &= \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) + \frac{1}{2\tau^2}\mathbf{w}^\top\mathbf{w} \\ &= \underset{\mathbf{w}}{\operatorname{argmin}} \text{RSS}(\mathbf{w}) + \lambda\|\mathbf{w}\|_2^2\end{aligned}$$

where  $\lambda = \frac{\sigma^2}{\tau^2}$  is proportional to the strength of the prior, and

$$\|\mathbf{w}\|_2 = \sqrt{\sum_{d=1}^D |w_d|^2} = \sqrt{\mathbf{w}^\top\mathbf{w}}$$

is the  $\ell_2$  norm of the vector  $\mathbf{w}$ .

- We do not penalize the offset  $w_0$ , since that only affects the global mean of the output, and does not contribute to the overfitting.

- The MAP estimate corresponds to minimizing the penalized objective:

$$J(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \|\mathbf{w}\|_2^2$$

where  $\lambda = \frac{\sigma^2}{\tau^2}$  is the strength of the regularizer.

- The derivative is given by

$$\nabla_{\mathbf{w}} J(\mathbf{w}) = 2(\mathbf{X}^\top \mathbf{X} \mathbf{w} - \mathbf{X}^\top \mathbf{y} + \lambda \mathbf{w})$$

- Therefore,

$$\begin{aligned} \hat{\mathbf{w}}_{\text{map}} &= (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_D)^{-1} \mathbf{X}^\top \mathbf{y} \\ &= \left( \sum_n \mathbf{x}_n \mathbf{x}_n^\top + \lambda \mathbf{I}_D \right)^{-1} \left( \sum_n y_n \mathbf{x}_n \right) \end{aligned}$$

## Example: MAP for Linear Regression

- Maximum likelihood estimation. Let  $\epsilon = 0.1$
- Ex. 1:  $\hat{\mathbf{w}}_{\text{mle}} = \begin{bmatrix} 1 & -1/\epsilon \\ -1/\epsilon & 2/\epsilon^2 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & \epsilon \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ .
- Ex. 2:  $\hat{\mathbf{w}}_{\text{mle}} = \begin{bmatrix} 1 & -1/\epsilon \\ -1/\epsilon & 2/\epsilon^2 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & \epsilon \end{bmatrix} \begin{bmatrix} 1 + \epsilon \\ 1 \end{bmatrix} = \begin{bmatrix} 1 + \epsilon \\ -1 \end{bmatrix} = \begin{bmatrix} 1.1 \\ -1 \end{bmatrix}$ .
- Maximum a posteriori estimation. Let  $\lambda = 0.05$
- $(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_D) = \begin{bmatrix} 2 + \lambda & \epsilon \\ \epsilon & \epsilon^2 + \lambda \end{bmatrix} = \begin{bmatrix} 2.05 & 0.1 \\ 0.1 & 0.06 \end{bmatrix}$
- $(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_D)^{-1} = \begin{bmatrix} 0.531 & -0.885 \\ -0.885 & 18.1416 \end{bmatrix}$
- Ex. 1:  $\hat{\mathbf{w}}_{\text{map}} = \begin{bmatrix} 0.531 & -0.885 \\ -0.885 & 18.1416 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & \epsilon \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.9735 \\ 0.0442 \end{bmatrix}$
- Ex. 2:  $\hat{\mathbf{w}}_{\text{map}} = \begin{bmatrix} 0.531 & -0.885 \\ -0.885 & 18.1416 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & \epsilon \end{bmatrix} \begin{bmatrix} 1 + \epsilon \\ 1 \end{bmatrix} = \begin{bmatrix} 1.0265 \\ -0.0442 \end{bmatrix}$