Probability & Statistics for DS & AI

Estimation

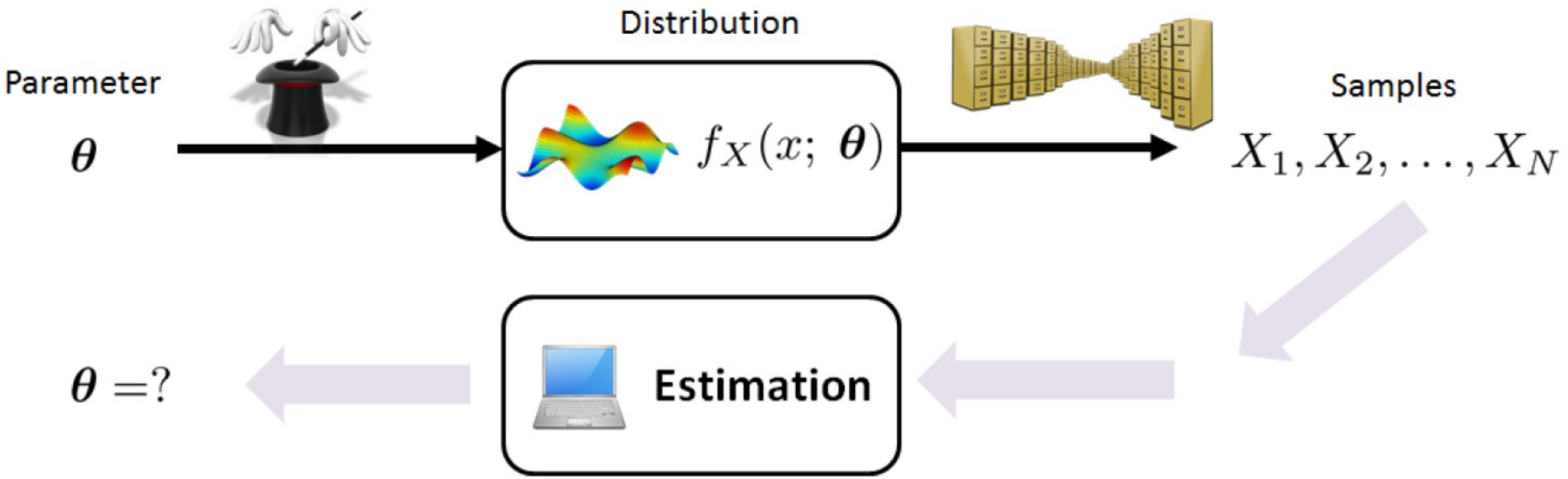
Michele Guindani

Summer

Estimation

- Estimation is an inverse problem with the goal of recovering the underlying parameter $\boldsymbol{\theta}$ of a distribution $f_X(x; \boldsymbol{\theta})$ based on the observed samples $X_1, \ldots, X_N$

8



Estimation is an inverse problem of recovering the unknown parameters that were used by the distribution. In this figure, the PDF of X using a parameter $\boldsymbol{\theta}$ is denoted as $f_X(x; \boldsymbol{\theta})$. The forward data-generation process takes the parameter $\boldsymbol{\theta}$ and creates the random samples $X_1, \ldots, X_N$. Estimation takes these observed random samples and recovers the underlying model parameter $\boldsymbol{\theta}$.

What are parameters?

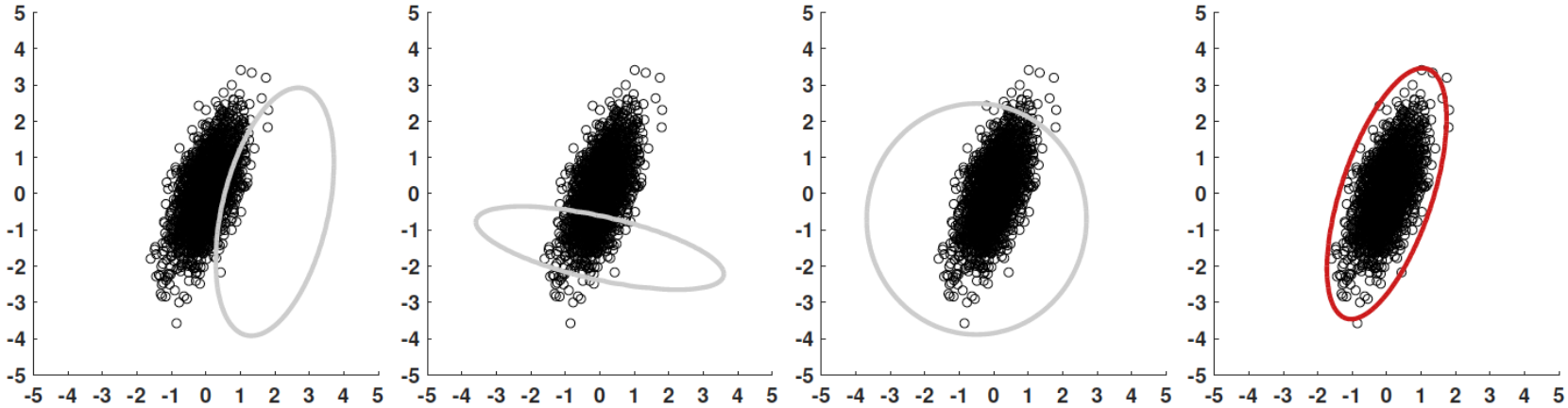
- All probability density functions (PDFs) have parameters.
- A Bernoulli random variable is characterized by a parameter p that defines the probability of obtaining a "head"
- A Gaussian random variable is characterized by two parameters: the mean μ and variance σ^2:

$$f_{X_n}(x_n; \underbrace{\boldsymbol{\theta}}_{=(\mu,\sigma)}) =| \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x_n-\mu)^2}{2\sigma^2}\right\}$$

If we know that $\sigma = 1$, then the PDF is

$$f_{X_n}(x_n; \underbrace{\theta}_{=\mu}) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(x_n-\mu)^2}{2}\right\}$$

where θ is the mean



Bad estimate

$$\boldsymbol{\mu} = \begin{bmatrix} 2 \\ -0.5 \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} 0.25 & 0.2 \\ 0.2 & 1 \end{bmatrix}$$

Bad estimate

$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ -1.5 \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} 1 & -0.2 \\ -0.2 & 0.1 \end{bmatrix}$$

Bad estimate

$$\boldsymbol{\mu} = \begin{bmatrix} -0.5 \\ -0.7 \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Good estimate

$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} 0.25 & 0.3 \\ 0.3 & 1 \end{bmatrix}$$

An estimation problem. Given a set of 1000 data points drawn from a Gaussian distribution with unknown mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, we propose several candidate Gaussians and see which one would be the best fit to the data. Visually, we observe that the right-most Gaussian has the best fit. The goal of this chapter is to develop a systematic way of solving estimation problems of this type.

Estimation methods

- We will be looking at two estimation methods:

① Maximum Likelihood methods

② Maximum a posteriori method (Bayesian but used a lot in ML)

Probability & Statistics for DS & AI

Maximum Likelihood

Michele Guindani

Summer

Likelihood function

Likelihood function

Consider a set of N data points $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$.

Since we have N data points, based on the problem at hand, we can postulate a data generating model:

$$X_1, \ldots, X_N \sim f(\boldsymbol{x}; \boldsymbol{\theta})$$

which means $\boldsymbol{x} = (x_1, \ldots, x_N)$, $f_{\boldsymbol{X}}(\boldsymbol{x}; \boldsymbol{\theta}) =$ PDF of the random vector $\boldsymbol{X}$ with parameter $\boldsymbol{\theta}$.

When you express the joint PDF as a function of $\boldsymbol{x}$ and $\boldsymbol{\theta}$, you have two variables to play with:

- observation $\boldsymbol{x}$, given by the measured data (known)
- parameter $\boldsymbol{\theta}$ $\Rightarrow$ our interest in an estimation problem.

Likelihood function

Consider a set of N data points $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$.

Since we have N data points, based on the problem at hand, we can postulate a data generating model:

$$X_1, \ldots, X_N \sim f(\boldsymbol{x}; \boldsymbol{\theta})$$

which means $\boldsymbol{x} = (x_1, \ldots, x_N)$, $f_{\boldsymbol{X}}(\boldsymbol{x}; \boldsymbol{\theta}) =$ PDF of the random vector $\boldsymbol{X}$ with parameter $\boldsymbol{\theta}$.

When you express the joint PDF as a function of $\boldsymbol{x}$ and $\boldsymbol{\theta}$, you have two variables to play with:

- observation $\boldsymbol{x}$, given by the measured data (known)
- parameter $\boldsymbol{\theta}$ ⇨ our interest in an estimation problem.

- GOAL: find value of $\boldsymbol{\theta}$ that offers the "best explanation" to data $\boldsymbol{x}$

⇨ maximize the likelihood

Likelihood function

Let $\boldsymbol{X} = [X_1, \ldots, X_N]^T$ be a random vector drawn from a joint PDF $f_{\boldsymbol{X}}(\boldsymbol{x}; \boldsymbol{\theta})$, and let $\boldsymbol{x} = [x_1, \ldots, x_N]^T$ be the realizations. The likelihood function is a function of the parameter $\boldsymbol{\theta}$ given the realizations $\boldsymbol{x}$:

$$\mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{x}) \stackrel{\text{def}}{=} f_{\boldsymbol{X}}(\boldsymbol{x}; \boldsymbol{\theta})$$

Likelihood function

Let $\boldsymbol{X} = [X_1, \ldots, X_N]^T$ be a random vector drawn from a joint PDF $f_{\boldsymbol{X}}(\boldsymbol{x}; \boldsymbol{\theta})$, and let $\boldsymbol{x} = [x_1, \ldots, x_N]^T$ be the realizations. The likelihood function is a function of the parameter $\boldsymbol{\theta}$ given the realizations $\boldsymbol{x}$:

$$\mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{x}) \stackrel{\text{def}}{=} f_{\boldsymbol{X}}(\boldsymbol{x}; \boldsymbol{\theta})$$

- ⚠ $\mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{x})$ is not a conditional PDF because $\boldsymbol{\theta}$ is not a random variable.

  The correct way to interpret $\mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{x})$ is to view it as a function of $\boldsymbol{\theta}$.

  This function changes its shape according the observed data $\boldsymbol{x}$. We will return to this point shortly.

Independent observations

- If we measure the interarrival times of a bus for several days, it is quite likely the measurements are not correlated

Independent observations

- If we measure the interarrival times of a bus for several days, it is quite likely the measurements are not correlated
- Assumption: the data points are independent and drawn from an identical distribution $f_X(x)$:

$$f_{\boldsymbol{X}}(\boldsymbol{x};\boldsymbol{\theta}) = f_{X_1,\ldots,X_N}(x_1,\ldots,x_N;\boldsymbol{\theta}) = \prod_{n=1}^{N} f_{X_n}(x_n;\boldsymbol{\theta})$$

Independent observations

- If we measure the interarrival times of a bus for several days, it is quite likely the measurements are not correlated
- Assumption: the data points are independent and drawn from an identical distribution $f_X(x)$:

$$f_{\boldsymbol{X}}(\boldsymbol{x};\boldsymbol{\theta}) = f_{X_1,\ldots,X_N}(x_1,\ldots,x_N;\boldsymbol{\theta}) = \prod_{n=1}^{N} f_{X_n}(x_n;\boldsymbol{\theta})$$

- so the likelihood is

$$\mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{x}) \stackrel{\text{def}}{=} \prod_{n=1}^{N} f_{X_n}(x_n;\boldsymbol{\theta})$$

- and the log-likelihood is

$$\log \mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{x}) = \log f_{\boldsymbol{X}}(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{n=1}^{N} \log f_{X_n}(x_n;\boldsymbol{\theta})$$

Example (Bernoulli)

Find the log-likelihood of a sequence of i.i.d. Bernoulli random variables $X_1, \ldots, X_N$ with parameter θ.

Example (Bernoulli)

Find the log-likelihood of a sequence of i.i.d. Bernoulli random variables $X_1, \ldots, X_N$ with parameter θ.

Solution: If $X_1, \ldots, X_N$ are i.i.d. Bernoulli random variables, we have

$$f_{\boldsymbol{X}}(\boldsymbol{x}; \theta) = \prod_{n=1}^{N} \left\{ \theta^{x_n} (1-\theta)^{1-x_n} \right\}$$

Taking the log on both sides of the equation yields the log-likelihood function:

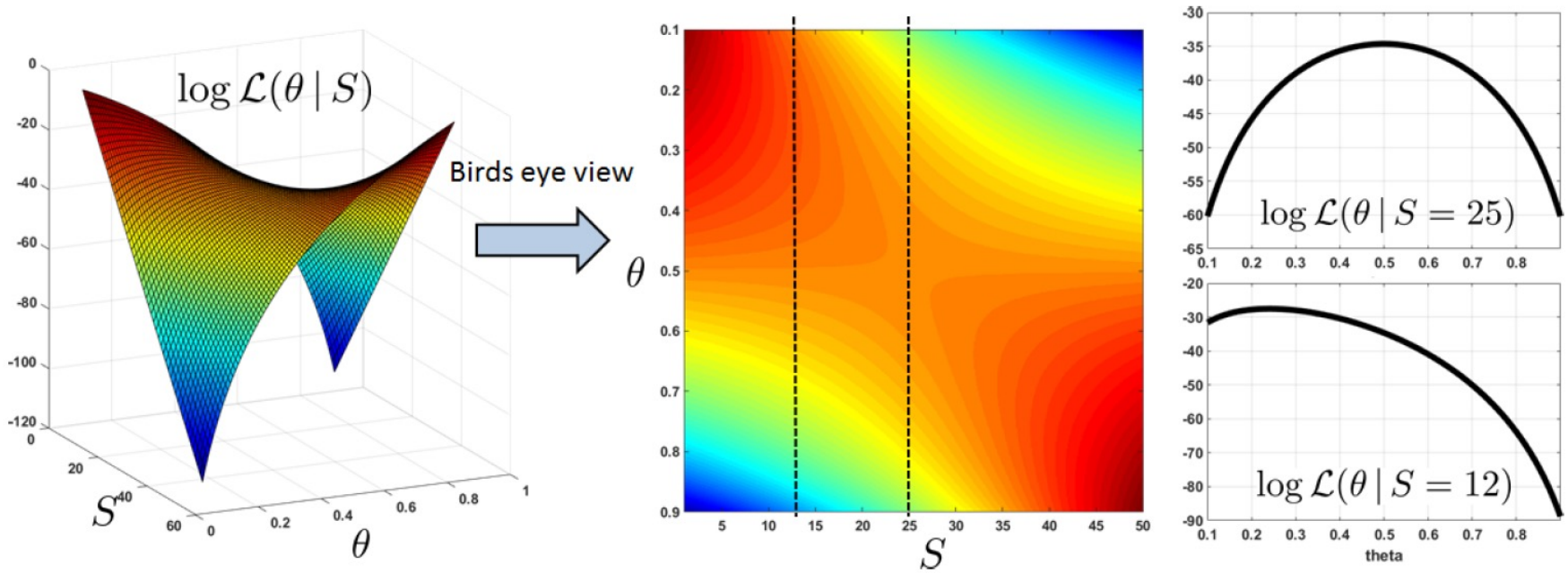
$$\begin{aligned}
\log \mathcal{L}(\theta \mid \boldsymbol{x}) &= \log \left\{ \prod_{n=1}^{N} \left\{ \theta^{x_n} (1-\theta)^{1-x_n} \right\} \right\} \\
&= \sum_{n=1}^{N} \log \left\{ \theta^{x_n} (1-\theta)^{1-x_n} \right\} \\
&= \sum_{n=1}^{N} x_n \log \theta + (1-x_n) \log(1-\theta) \\
&= \left(\sum_{n=1}^{N} x_n \right) \cdot \log \theta + \left(N - \sum_{n=1}^{N} x_n \right) \cdot \log(1-\theta)
\end{aligned}$$

- We can write

$$\log \mathcal{L}(\theta \mid \boldsymbol{x}) = \underbrace{\left(\sum_{n=1}^{N} x_n \right)}_{S} \cdot \log \theta + \underbrace{\left(N - \sum_{n=1}^{N} x_n \right)}_{N-S} \cdot \log(1-\theta)$$

  That is: $\log \mathcal{L}(\theta \mid S) = S \log \theta + (N - S) \log(1 - \theta)$.

- We plot the surface of $L(\theta \mid S)$ as a function of S and θ, assuming that $N = 50$



We plot the log-likelihood function as a function of $S = \sum_{n=1}^{N} x_n$ and θ. [Left] We show the surtace plot of $\mathcal{L}(\theta|S) = S \log \theta + (N - S) \log(1 - \theta)$. Note that the surface has a saddle shape. [Middle] By taking a bird's-eye view of the surface plot, we obtain a 2-dimensional contour plot of the surface, where the color code matches the height of the log-likelihood function. [Right] We take two cross sections along $S = 25$ and $S = 12$. Observe how the shape changes.

Example (Gaussian)

Find the log-likelihood of a sequence of i.i.d. Gaussian random variables $X_1, ., X_N$ with mean μ and variance σ^2

Example (Gaussian)

Find the log-likelihood of a sequence of i.i.d. Gaussian random variables $X_1, ., X_N$ with mean μ and variance σ^2

Solution Since the random variables $X_1, \ldots, X_N$ are i.i.d. Gaussian, the PDF is

$$f_{\boldsymbol{X}}\left(\boldsymbol{x}; \mu, \sigma^2\right) = \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\}$$

Example (Gaussian)

Find the log-likelihood of a sequence of i.i.d. Gaussian random variables $X_1, ., X_N$ with mean μ and variance σ^2

Solution Since the random variables $X_1, \ldots, X_N$ are i.i.d. Gaussian, the PDF is

$$f_{\boldsymbol{X}}\left(\boldsymbol{x}; \mu, \sigma^2\right) = \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\}$$

Taking the log on both sides yields the log-likelihood function:

$$\log \mathcal{L}\left(\mu, \sigma^2 \mid \boldsymbol{x}\right)$$

Example (Gaussian)

Find the log-likelihood of a sequence of i.i.d. Gaussian random variables $X_1, ., X_N$ with mean μ and variance σ^2

Solution Since the random variables $X_1, \ldots, X_N$ are i.i.d. Gaussian, the PDF is

$$f_{\boldsymbol{X}}\left(\boldsymbol{x}; \mu, \sigma^2\right) = \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\}$$

Taking the log on both sides yields the log-likelihood function:

$$\log \mathcal{L}\left(\mu, \sigma^2 \mid \boldsymbol{x}\right) = \log f_{\boldsymbol{X}}\left(\boldsymbol{x}; \mu, \sigma^2\right)$$

Example (Gaussian)

Find the log-likelihood of a sequence of i.i.d. Gaussian random variables $X_1, ., X_N$ with mean μ and variance σ^2

Solution Since the random variables $X_1, \ldots, X_N$ are i.i.d. Gaussian, the PDF is

$$f_{\boldsymbol{X}}\left(\boldsymbol{x}; \mu, \sigma^2\right) = \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\}$$

Taking the log on both sides yields the log-likelihood function:

$$\log \mathcal{L}\left(\mu, \sigma^2 \mid \boldsymbol{x}\right) = \log f_{\boldsymbol{X}}\left(\boldsymbol{x}; \mu, \sigma^2\right)$$

Example (Gaussian)

Find the log-likelihood of a sequence of i.i.d. Gaussian random variables $X_1, ., X_N$ with mean μ and variance σ^2

Solution Since the random variables $X_1, \ldots, X_N$ are i.i.d. Gaussian, the PDF is

$$f_{\boldsymbol{X}}\left(\boldsymbol{x}; \mu, \sigma^2\right) = \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\}$$

Taking the log on both sides yields the log-likelihood function:

$$\begin{aligned} \log \mathcal{L}\left(\mu, \sigma^2 \mid \boldsymbol{x}\right) &= \log f_{\boldsymbol{X}}\left(\boldsymbol{x}; \mu, \sigma^2\right) \\ &= \log \left\{ \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\} \right\} \end{aligned}$$

Example (Gaussian)

Find the log-likelihood of a sequence of i.i.d. Gaussian random variables $X_1, ., X_N$ with mean μ and variance σ^2

Solution Since the random variables $X_1, \ldots, X_N$ are i.i.d. Gaussian, the PDF is

$$f_{\boldsymbol{X}}\left(\boldsymbol{x}; \mu, \sigma^2\right) = \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\}$$

Taking the log on both sides yields the log-likelihood function:

$$\begin{aligned}
\log \mathcal{L}\left(\mu, \sigma^2 \mid \boldsymbol{x}\right) &= \log f_{\boldsymbol{X}}\left(\boldsymbol{x}; \mu, \sigma^2\right) \\
&= \log \left\{ \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\} \right\} \\
&= \sum_{n=1}^{N} \log \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\}
\end{aligned}$$

Example (Gaussian)

Find the log-likelihood of a sequence of i.i.d. Gaussian random variables $X_1, ., X_N$ with mean μ and variance σ^2

Solution Since the random variables $X_1, \ldots, X_N$ are i.i.d. Gaussian, the PDF is

$$f_{\boldsymbol{X}}\left(\boldsymbol{x}; \mu, \sigma^2\right) = \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\}$$

Taking the log on both sides yields the log-likelihood function:

$$\begin{aligned}
\log \mathcal{L}\left(\mu, \sigma^2 \mid \boldsymbol{x}\right) &= \log f_{\boldsymbol{X}}\left(\boldsymbol{x}; \mu, \sigma^2\right) \\
&= \log \left\{ \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\} \right\} \\
&= \sum_{n=1}^{N} \log \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\}
\end{aligned}$$

Example (Gaussian)

Find the log-likelihood of a sequence of i.i.d. Gaussian random variables $X_1, ., X_N$ with mean μ and variance σ^2

Solution Since the random variables $X_1, \ldots, X_N$ are i.i.d. Gaussian, the PDF is

$$f_{\boldsymbol{X}}\left(\boldsymbol{x}; \mu, \sigma^2\right) = \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\}$$

Taking the log on both sides yields the log-likelihood function:

$$\begin{aligned}
\log \mathcal{L}\left(\mu, \sigma^2 \mid \boldsymbol{x}\right) &= \log f_{\boldsymbol{X}}\left(\boldsymbol{x}; \mu, \sigma^2\right) \\
&= \log \left\{ \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\} \right\} \\
&= \sum_{n=1}^{N} \log \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\} \\
&= \sum_{n=1}^{N} \left\{ -\frac{1}{2} \log\left(2\pi\sigma^2\right) - \frac{\left(x_n-\mu\right)^2}{2\sigma^2} \right\}
\end{aligned}$$

Example (Gaussian)

Find the log-likelihood of a sequence of i.i.d. Gaussian random variables $X_1, ., X_N$ with mean μ and variance σ^2

Solution Since the random variables $X_1, \ldots, X_N$ are i.i.d. Gaussian, the PDF is

$$f_{\boldsymbol{X}}\left(\boldsymbol{x}; \mu, \sigma^2\right) = \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\}$$

Taking the log on both sides yields the log-likelihood function:

$$\begin{aligned}
\log \mathcal{L}\left(\mu, \sigma^2 \mid \boldsymbol{x}\right) &= \log f_{\boldsymbol{X}}\left(\boldsymbol{x}; \mu, \sigma^2\right) \\
&= \log \left\{ \prod_{n=1}^{N} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\} \right\} \\
&= \sum_{n=1}^{N} \log \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}} \right\} \\
&= \sum_{n=1}^{N} \left\{ -\frac{1}{2} \log\left(2\pi\sigma^2\right) - \frac{\left(x_n-\mu\right)^2}{2\sigma^2} \right\} \\
&= -\frac{N}{2} \log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2} \sum_{n=1}^{N} \left(x_n-\mu\right)^2
\end{aligned}$$

Maximum Likelihood estimate

Let $\mathcal{L}(\boldsymbol{\theta})$ be the likelihood function of the parameter $\boldsymbol{\theta}$ given the measurements $\boldsymbol{x} = [x_1, \ldots, x_N]^T$. The maximum-likelihood estimate of the parameter $\boldsymbol{\theta}$ is a parameter that maximizes the likelihood:

$$\widehat{\boldsymbol{\theta}}_{ML} \stackrel{\text{def}}{=} \underset{\boldsymbol{\theta}}{\text{argmax}} \mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{x})$$

Example (Bernoulli)

Find the ML estimate for a set of i.i.d. Bernoulli random variables $\{X_1, \ldots, X_N\}$ with $X_n \sim \text{Bernoulli}(\theta)$ for $n = 1, \ldots, N$

Example (Bernoulli)

Find the ML estimate for a set of i.i.d. Bernoulli random variables $\{X_1, \ldots, X_N\}$ with $X_n \sim \text{Bernoulli}(\theta)$ for $n = 1, \ldots, N$

Solution. The log-likelihood function of a set of i.i.d. Bernoulli random variables is

$$\log \mathcal{L}(\theta \mid \boldsymbol{x}) = \left(\sum_{n=1}^{N} x_n \right) \cdot \log \theta + \left(N - \sum_{n=1}^{N} x_n \right) \cdot \log(1 - \theta)$$

Thus, to find the ML estimate, we need to solve the optimization problem

$$\widehat{\theta}_{\text{ML}} = \underset{\theta}{\text{argmax}} \left\{ \left(\sum_{n=1}^{N} x_n \right) \cdot \log \theta + \left(N - \sum_{n=1}^{N} x_n \right) \cdot \log(1 - \theta) \right\}$$

Example (Bernoulli)

Find the ML estimate for a set of i.i.d. Bernoulli random variables $\{X_1, \ldots, X_N\}$ with $X_n \sim \text{Bernoulli}(\theta)$ for $n = 1, \ldots, N$

Solution. The log-likelihood function of a set of i.i.d. Bernoulli random variables is

$$\log \mathcal{L}(\theta \mid \boldsymbol{x}) = \left(\sum_{n=1}^{N} x_n \right) \cdot \log \theta + \left(N - \sum_{n=1}^{N} x_n \right) \cdot \log(1 - \theta)$$

Thus, to find the ML estimate, we need to solve the optimization problem

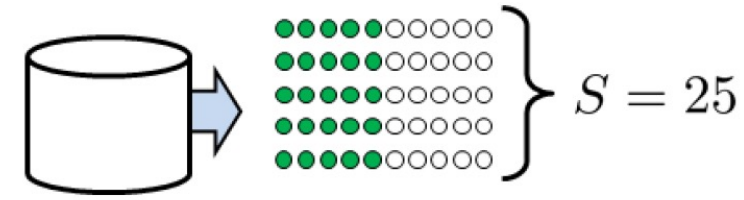
$$\widehat{\theta}_{\text{ML}} = \underset{\theta}{\text{argmax}} \left\{ \left(\sum_{n=1}^{N} x_n \right) \cdot \log \theta + \left(N - \sum_{n=1}^{N} x_n \right) \cdot \log(1 - \theta) \right\}$$

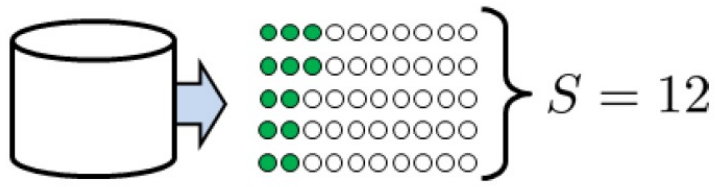
Taking the derivative with respect to θ and setting it to zero,

$$\frac{\left(\sum_{n=1}^{N} x_n \right)}{\theta} - \frac{N - \sum_{n=1}^{N} x_n}{1 - \theta} = 0$$

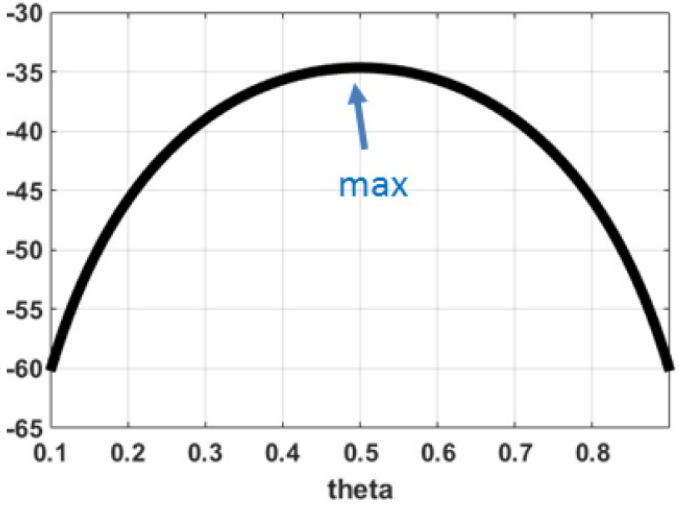
Rearranging the terms yields

$$\widehat{\theta}_{\text{ML}} = \frac{1}{N} \sum_{n=1}^{N} x_n$$





$\log \mathcal{L}(\theta \mid S = 25)$



$\log \mathcal{L}(\theta \mid S = 12)$

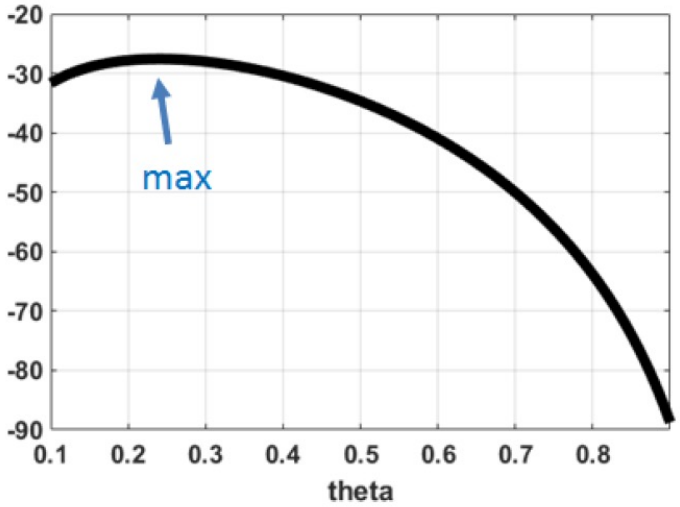
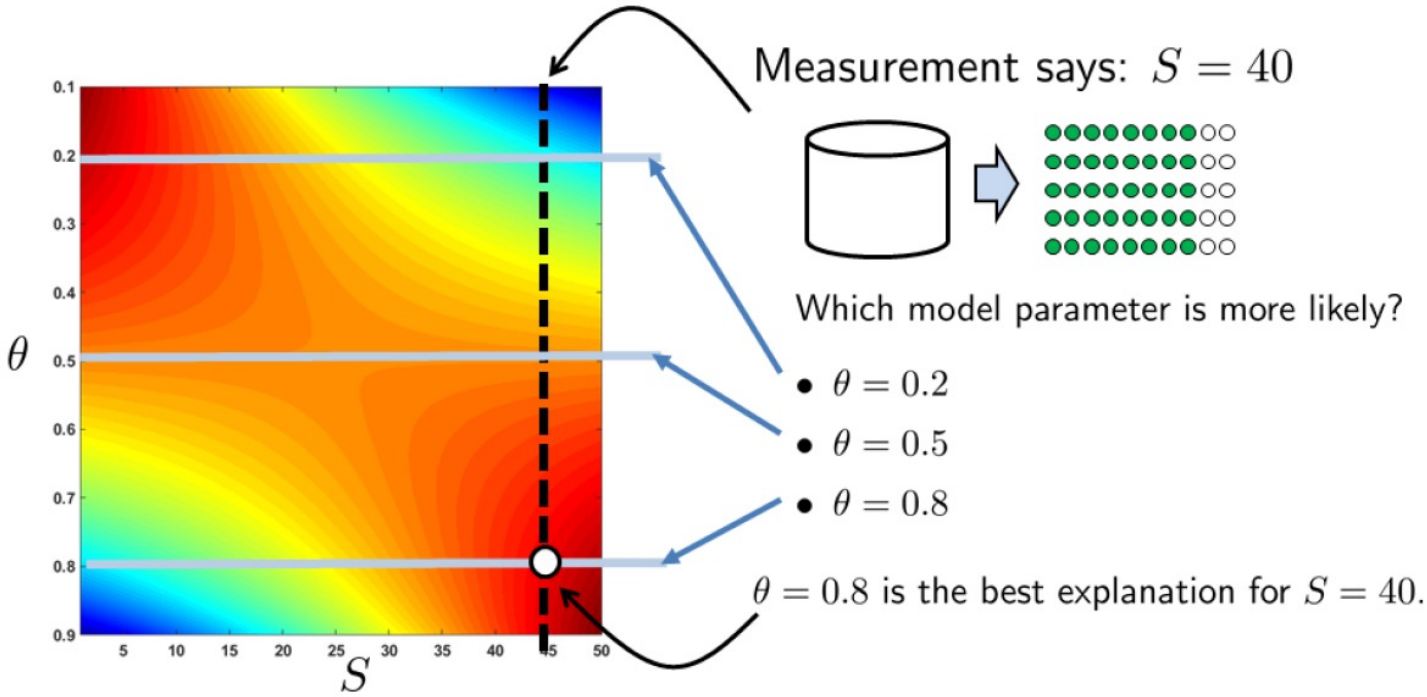


Illustration of how the maximum-likelihood estimate of a set of i.i.d. Bernoulli random variables is determined. The subfigures above show two particular scenarios at $S = 25$ and $S = 12$, assuming that $N = 50$. When $S = 25$, the likelihood function has a quadratic shape centered at $\theta = 0.5$. This point is also the peak of the likelihood function when $S = 25$. Therefore, the ML estimate is $\widehat{\theta}_{\text{ML}} = 0.5$. The second case is when $S = 12$. The quadratic likelihood is shifted toward the left. The ML estimate is $\widehat{\theta}_{\text{MI}} = 0.24$.



Suppose that we have a set of measurements such that $S = 40$. To determine the ML estimate, we look at the vertical cross section at $S = 40$. Among the different candidate parameters, e.g., $\theta = 0.2$, $\theta = 0.5$ and $\theta = 0.8$, we pick the one that has the maximum response to the likelihood function. For $S = 40$, it is more likely that the underlying parameter is $\theta = 0.8$ than $\theta = 0.2$ or $\theta = 0.5$.

Example (Social Network Analysis)

- **Recall:** The Erdos-Renyi graph is one of the simplest models for social networks. The Erdos-Renyi graph is a single-membership network that assumes that all users belong to the same cluster. Thus the connectivity between users is specified by a single parameter:

$$X_{ij} \sim \text{Bernoulli}(p)$$

- In other words, the edge X_{ij} linking user i and user j in the network is either $X_{ij} = 1$ with probability p, or $X_{ij} = 0$ with probability $1 - p$.
- The resulting matrix $\boldsymbol{X} \in \mathbb{R}^{N \times N}$ as the adjacency matrix, with the (i, j) th element being X_{ij}.

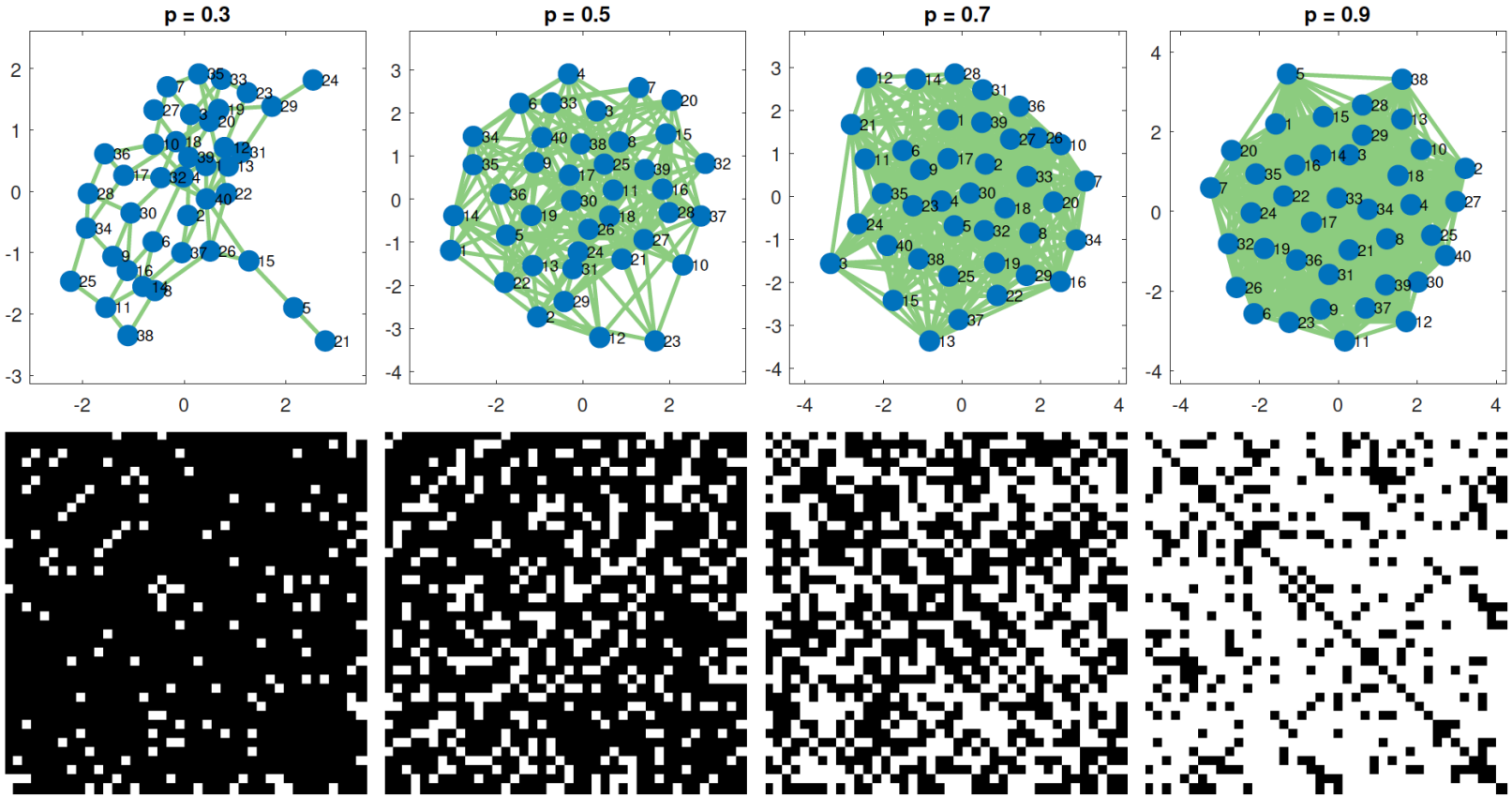


Figure 3.25: The Erdős-Rényi graph. [Top] The graphs. [Bottom] The adjacency matrices.

A single-membership Erdos-Renyi graph is a graph structure in which the edge between node i and node j is defined as a Bernoulli random variable with parameter p. As p increases, the graph has a higher probability of having more edges. The adjacent matrices shown in the bottom row are the mathematical representations of the graphs.

Log-likelihood function of the Erdos-Renyi graph

The probability mass function of X_{ij} is

$$\mathbb{P}[X_{ij} = 1] = p \quad \text{and} \quad \mathbb{P}[X_{ij} = 0] = 1 - p$$

This can be compactly expressed as

$$f_{\boldsymbol{X}}(\boldsymbol{x}; p) = \prod_{i=1}^{N} \prod_{j=1}^{N} p^{x_{ij}} (1-p)^{1-x_{ij}}$$

Hence, the log-likelihood is

$$\log \mathcal{L}(p \mid \boldsymbol{x}) = \sum_{i=1}^{N} \sum_{j=1}^{N} \{x_{ij} \log p + (1 - x_{ij}) \log(1-p)\}$$

To solve the ML estimation problem:

$$\widehat{p}_{\text{ML}} = \underset{p}{\text{argmax}} \log \mathcal{L}(p \mid \boldsymbol{x})$$

To solve the ML estimation problem:

$$\widehat{p}_{\text{ML}} = \underset{p}{\text{argmax}} \log \mathcal{L}(p \mid \boldsymbol{x})$$

Taking the derivative and setting it to zero,

$$\begin{aligned}\frac{d}{dp} \text{lpg}\, \mathcal{L}(p \mid \boldsymbol{x}) =& \frac{d}{dp} \left\{ \sum_{i=1}^{N} \sum_{j=1}^{N} \{x_{ij} \log p + (1 - x_{ij}) \log(1 - p)\} \right\} \\ =& \sum_{i=1}^{N} \sum_{j=1}^{N} \left\{ \frac{x_{ij}}{p} - \frac{1 - x_{ij}}{1 - p} \right\} = 0\end{aligned}$$

To solve the ML estimation problem:

$$\widehat{p}_{\text{ML}} = \underset{p}{\text{argmax}} \log \mathcal{L}(p \mid \boldsymbol{x})$$

Taking the derivative and setting it to zero,

$$\begin{aligned}\frac{d}{dp} \text{lpg}\, \mathcal{L}(p \mid \boldsymbol{x}) &= \frac{d}{dp} \left\{ \sum_{i=1}^{N} \sum_{j=1}^{N} \{x_{ij} \log p + (1 - x_{ij}) \log(1 - p)\} \right\} \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} \left\{ \frac{x_{ij}}{p} - \frac{1 - x_{ij}}{1 - p} \right\} = 0\end{aligned}$$

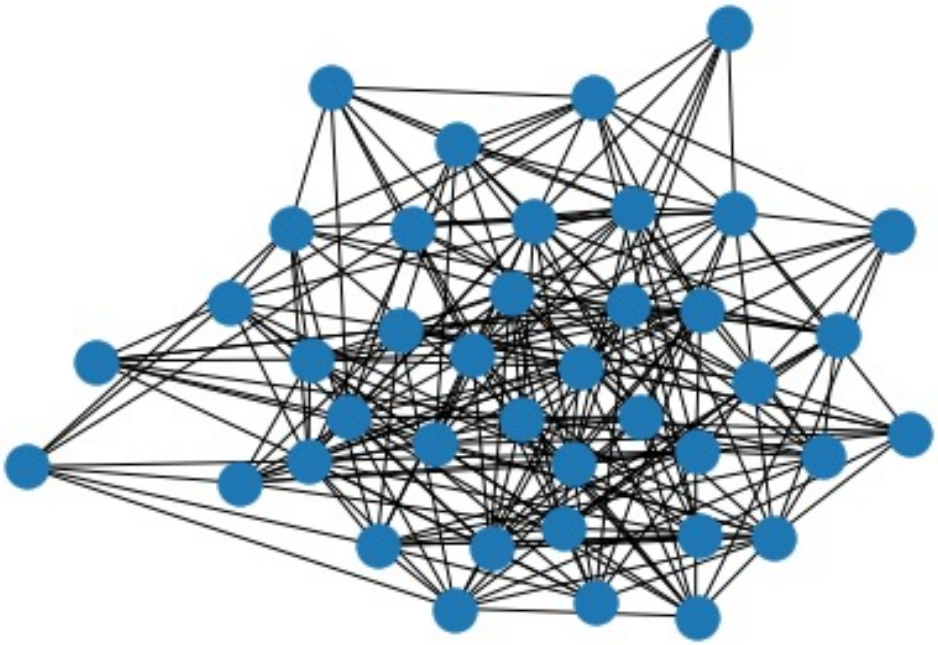
Let $S = \sum_{i=1}^{N} \sum_{j=1}^{N} x_{ij}$. The equation above then becomes

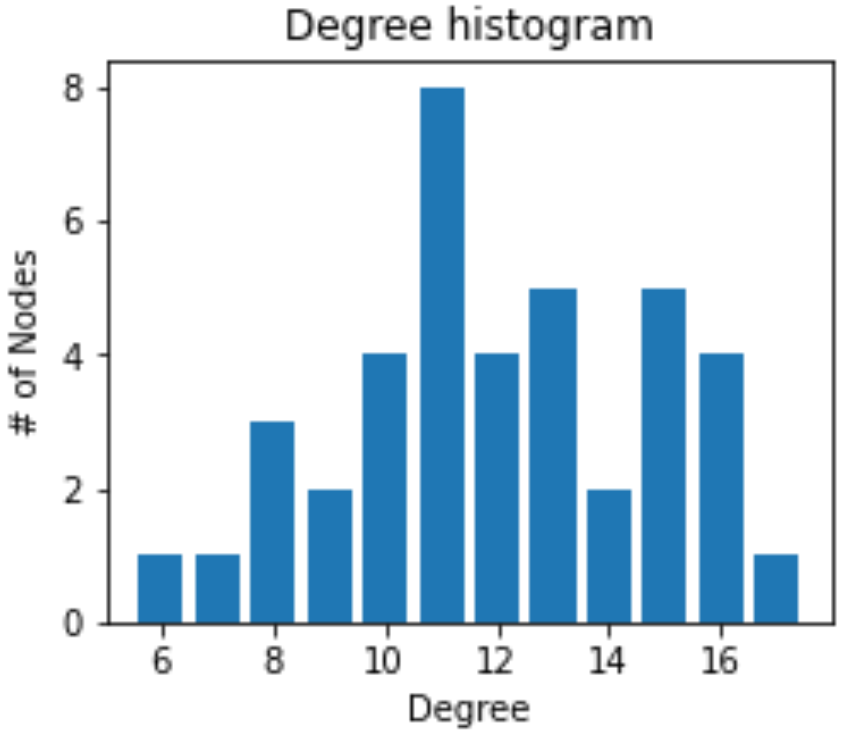
$$\frac{S}{p} - \frac{N^2 - S}{1 - p} = 0$$

which leads to

$$\widehat{p}_{\text{ML}} = \frac{S}{N^2} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} x_{ij}$$

```
# Python code to visualize a graph
import networkx as nx
import numpy as np
import random
np.random.seed(12345)
n = 40 # Number of nodes
p = 0.3 # probability
m = np.round(((n ** 2)/2)*p) # Number of edges
G = nx.gnm_random_graph(n,m) # Graph
A = nx.adjacency_matrix(G) # Adj matrix
nx.draw(G) # Drawing
p_ML = np.mean(A) # ML estimate
print(p_ML) #0.3
```




Degree histogram
# of Nodes
Degree
0
2
4
6
8
6
8
10
12
14
16

```
import matplotlib.pyplot as plt
import networkx as nx
import numpy as np
degree_sequence = sorted((d for n, d in G.degree()), reverse=True)
fig = plt.figure("Degree of a random graph", figsize=(8, 8))
# Create a gridspec for adding subplots of different sizes
axgrid = fig.add_gridspec(5, 4)
ax2 = fig.add_subplot(axgrid[3:, 2:])
ax2.bar(*np.unique(degree_sequence, return_counts=True))
ax2.set_title("Degree histogram")
ax2.set_xlabel("Degree")
ax2.set_ylabel("# of Nodes")
```

Example (Supermarket data sales)

- Data from Kaggle
- Supermarket sales can vary depending on the time of the day, the month of the year, and characteristics of the custormers: the gender, SES, and so on...
- Here, we would like to estimate the rate of sales in a particular branch.
- Of course, we could have different questions in mind: are there differences in the sales' patterns between members and non-members of a loyalty program, between females and males, etc...

Supermarket data sales

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt

df=pd.read_csv('supermarket_sales_reduced.csv')
df.head()
```

Cost of goods sold

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	cogs	gross margin percentage	gross income	Rating
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7.0	26.1415	548.9715	1/5/19	13:08	Ewallet	522.83	4.761905	26.1415	9.1
1	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7.0	16.2155	340.5255	3/3/19	13:23	Credit card	324.31	4.761905	16.2155	7.4
2	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8.0	23.2880	489.0480	1/27/19	20:33	Ewallet	465.76	4.761905	23.2880	8.4
3	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7.0	30.2085	634.3785	2/8/19	10:37	Ewallet	604.17	4.761905	30.2085	5.3
4	355-53-5943	A	Yangon	Member	Female	Electronic accessories	68.84	6.0	20.6520	433.6920	2/25/19	14:36	Ewallet	413.04	4.761905	20.6520	5.8

Supermarket data sales

```
df.describe()
```

	Unit price	Quantity	Tax 5%	Total	cogs	gross margin percentage	gross income	Rating
count	340.000000	340.000000	340.000000	340.000000	340.000000	3.400000e+02	340.000000	340.000000
mean	54.780853	5.467647	14.874001	312.354031	297.480029	4.761905e+00	14.874001	7.027059
std	26.132127	2.859876	11.030477	231.640025	220.609547	1.778975e-14	11.030477	1.731345
min	10.080000	1.000000	0.604500	12.694500	12.090000	4.761905e+00	0.604500	4.000000
25%	32.250000	3.000000	6.547125	137.489625	130.942500	4.761905e+00	6.547125	5.600000
50%	53.235000	5.000000	11.468000	240.828000	229.360000	4.761905e+00	11.468000	7.100000
75%	75.095000	8.000000	21.873375	459.340875	437.467500	4.761905e+00	21.873375	8.500000
max	99.830000	10.000000	49.490000	1039.290000	989.800000	4.761905e+00	49.490000	10.000000

```
df.Gender.value_counts()
```

```
Male      179
Female    161
Name: Gender, dtype: int64
```

```
df.Payment.value_counts()
```

```
Ewallet        126
Cash           110
Credit card    104
Name: Payment, dtype: int64
```

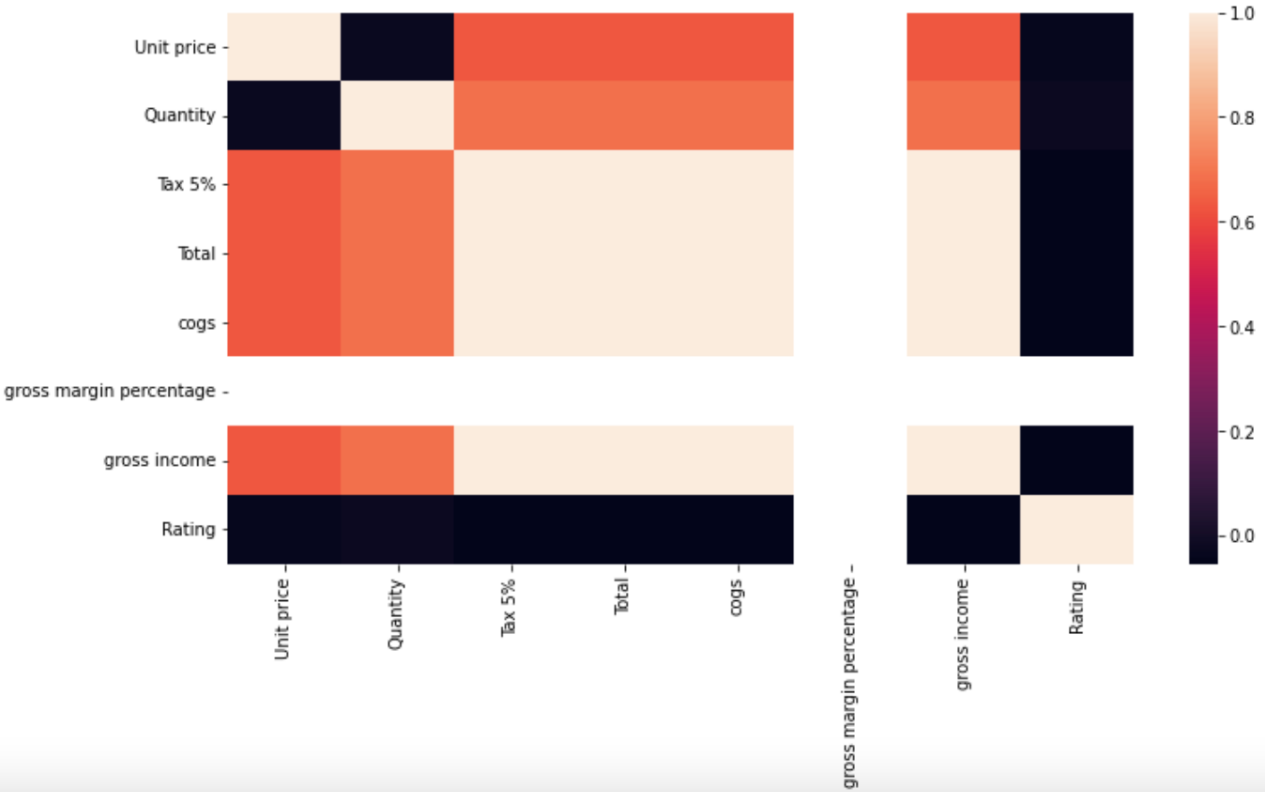
```
df['Product line'].value_counts()
```

```
Home and lifestyle        65
Electronic accessories    60
Sports and travel         59
Food and beverages        58
Fashion accessories       51
Health and beauty         47
Name: Product line, dtype: int64
```

Supermarket data sales

```
In [63]: import seaborn as sns
         #Seaborn is a Python data visualization library based on matplotlib.
         # It provides a high-level interface
         # for drawing attractive and
         #informative statistical graphics.
         plt.figure(figsize=(12,6))
         sns.heatmap(df.corr())
         #gives relation between different columns if any
```

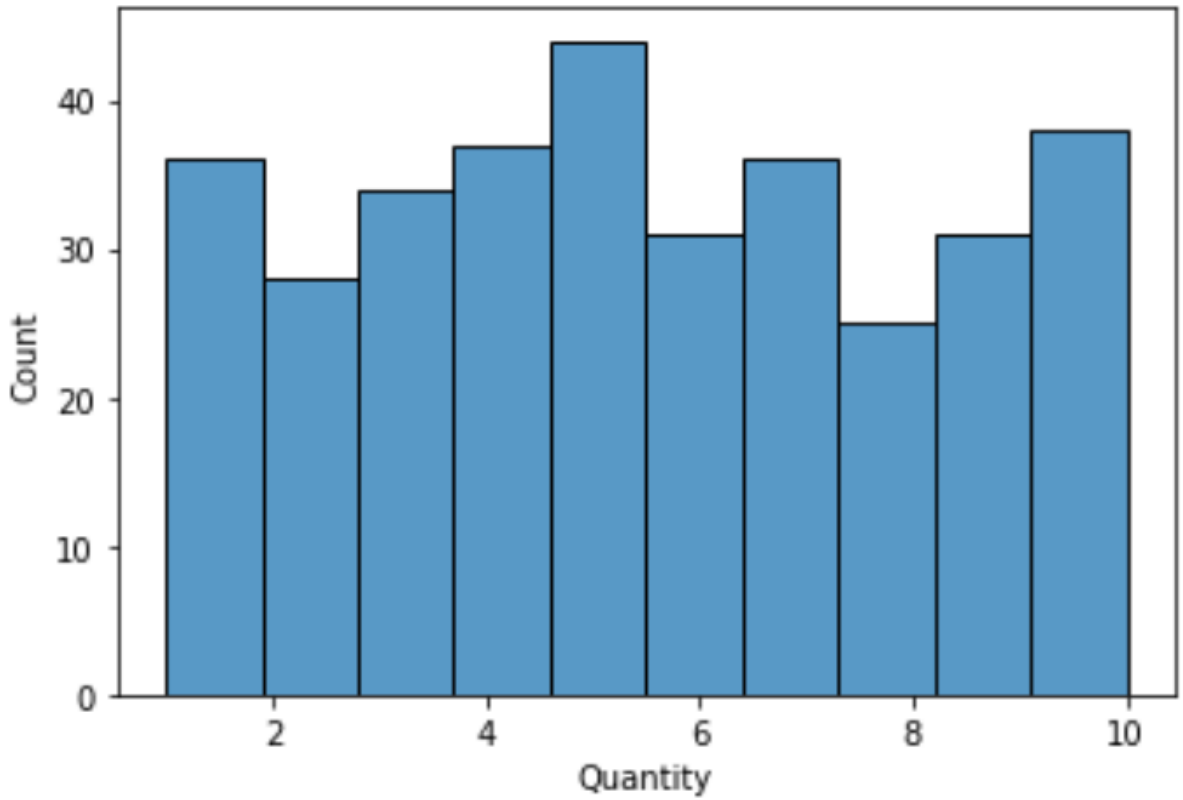
```
Out[63]: <AxesSubplot:>
```



Supermarket data sales

```
import seaborn as sns
sns.histplot(data=df, x="Quantity")
```

```
<AxesSubplot:xlabel='Quantity', ylabel='Count'>
```



Supermarket data sales

We are interested in estimating the rate of sales in a particular branch of a supermarket

The rate of sales may vary across the day, product line, and depend on customer-characteristics

For now, we do not segment our dataset further.

Supermarket data sales

We are interested in estimating the rate of sales in a particular branch of a supermarket

The rate of sales may vary across the day, product line, and depend on customer-characteristics

For now, we do not segment our dataset further.
If we are willing to believe that the customers shop independently from the same distribution, then we can assume

$$X_i \sim \text{Poisson}(\lambda)$$

The MLE λ_{ML} for a Poisson is the sample mean, $\widehat{\lambda}_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n$. (If you don't believe me, see Practice Exercise 8.6) in the textbook.

Supermarket data sales

- Then it is very easy to compute in Python

```
df [ "Quantity"].mean()
#5.467647
```

- The average rate of sale is about 5.47 items per customer.

Supermarket data sales

- Then it is very easy to compute in Python

```
df [ "Quantity"].mean()
#5.467647
```

- The average rate of sale is about 5.47 items per customer.

- Alternatively, one could have used direct minimization of the objective function (likelihood)

Direct minimization of the negative log-likelihood

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import minimize
from scipy.special import factorial
from scipy import stats


def poisson(k, lamb):
    """poisson pdf, parameter lamb is the fit parameter"""
    return (lamb**k/factorial(k)) * np.exp(-lamb)


def negative_log_likelihood(params, data):
    """
    The negative log-Likelihood-Function
    """

    lnl = - np.sum(np.log(poisson(data, params[0])))
    return lnl

def negative_log_likelihood(params, data):
    ''' better alternative using scipy '''
    return -stats.poisson.logpmf(data, params[0]).sum()


filtered_df = df[df['Quantity'].notnull()] #exclude NaN
```

Direct minimization of the negative log-likelihood

```
# minimize the negative log-Likelihood

result = minimize(negative_log_likelihood,  # function to minimize
                  x0=np.ones(1),            # start value
                  args=(filtered_df['Quantity'],),             # additional arguments for function
                  method='Powell',          # minimization method, see docs
                  )
# result is a scipy optimize result object, the fit parameters
# are stored in result.x
print(result)
```

```
   direc: array([[0.00014485]])
     fun: 856.2084501766212
 message: 'Optimization terminated successfully.'
    nfev: 45
     nit: 2
  status: 0
 success: True
       x: array([5.46764711])
```

Direct minimization of the negative log-likelihood

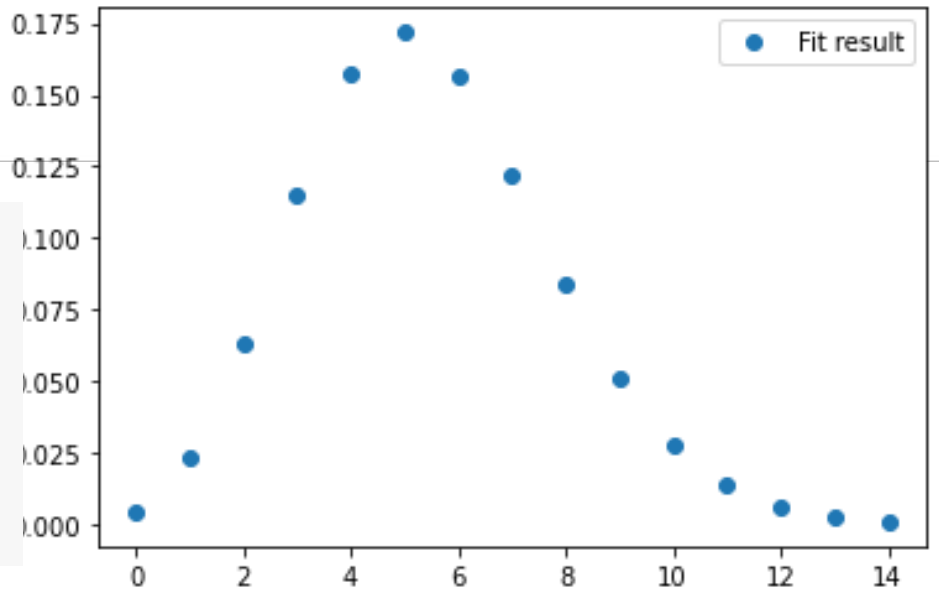
```
# minimize the negative log-Likelihood

result = minimize(negative_log_likelihood,  # function to minimize
                  x0=np.ones(1),            # start value
                  args=(filtered_df['Quantity'],),             # additional arguments for function
                  method='Powell',          # minimization method, see docs
                  )
# result is a scipy optimize result object, the fit parameters
# are stored in result.x
print(result)
```

```
   direc: array([[0.00014485]])
     fun: 856.2084501766212
 message: 'Optimization terminated successfully.'
    nfev: 45
     nit: 2
  status: 0
 success: True
       x: array([5.46764711])
```

```
#plot poisson-distribution with fitted parameter
x_plot = np.arange(0, 15)

plt.plot(
    x_plot,
    stats.poisson.pmf(x_plot, 5.46764711),
    marker='o', linestyle='',
    label='Fit result',
)
plt.legend()
plt.show()
```



Real Estate Evaluation data

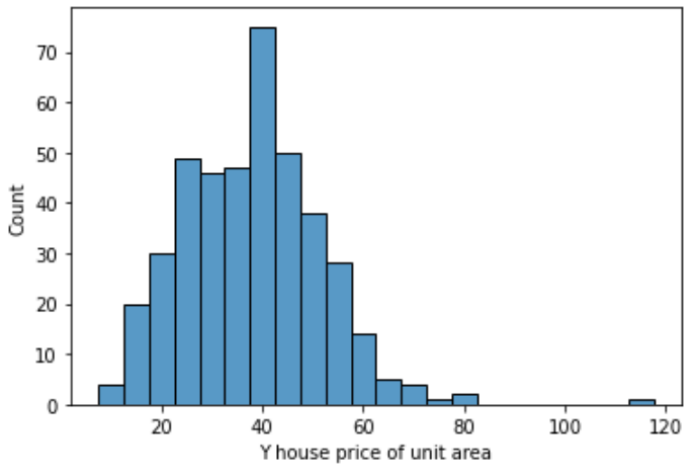
- We now consider market historical data set of real estate valuation collected from the Sindian Dist., New Taipei City, Taiwan (2012-

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt

df=pd.read_csv('Realestate.csv')
df.head()
```

	No	X1 transaction date	X2 house age	X3 distance to the nearest MRT station	X4 number of convenience stores	X5 latitude	X6 longitude	Y house price of unit area
0	1	2012.917	32.0	84.87882	10	24.98298	121.54024	37.9
1	2	2012.917	19.5	306.59470	9	24.98034	121.53951	42.2
2	3	2013.583	13.3	561.98450	5	24.98746	121.54391	47.3
3	4	2013.500	13.3	561.98450	5	24.98746	121.54391	54.8
4	5	2012.833	5.0	390.56840	5	24.97937	121.54245	43.1

```
import seaborn as sns
sns.histplot(data=df, x="Y house price of unit area")
```



Real Estate Evaluation data

- Real Estate evaluation data are continuous measurements.
- We can think of Gaussian data. The Gaussian distribution depends on two parameters, μ and σ^2.

Real Estate Evaluation data

- Real Estate evaluation data are continuous measurements.
- We can think of Gaussian data. The Gaussian distribution depends on two parameters, μ and σ^2.
- The ML estimate of $\boldsymbol{\theta} = \left[\mu, \sigma^2\right]^T$ is the vector

$$\widehat{\mu}_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n \quad \text{and} \quad \widehat{\sigma}^2_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N} (x_n - \widehat{\mu}_{\text{ML}})^2$$

If you don't believe me, check Practice Exercise 8.5 in the textbook

Real Estate Evaluation data

- Real Estate evaluation data are continuous measurements.
- We can think of Gaussian data. The Gaussian distribution depends on two parameters, μ and σ^2.
- The ML estimate of $\boldsymbol{\theta} = \left[\mu, \sigma^2\right]^T$ is the vector

$$\widehat{\mu}_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n \quad \text{and} \quad \widehat{\sigma}^2_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N} (x_n - \widehat{\mu}_{\text{ML}})^2$$

If you don't believe me, check Practice Exercise 8.5 in the textbook

Real Estate Evaluation data

```
import math
import matplotlib.pyplot as plt
import numpy as np
import scipy
import scipy.stats

def normal_mu_MLE(X):
    # Get the number of observations
    T = len(X)
    # Sum the observations
    s = sum(X)
    return 1.0/T * s

def normal_sigma_MLE(X):
    T = len(X)
    # Get the mu MLE
    mu = normal_mu_MLE(X)
    # Sum the square of the differences
    s = sum( np.power((X - mu), 2) )
    # Compute sigma^2
    sigma_squared = 1.0/T * s
    return math.sqrt(sigma_squared)

print(normal_mu_MLE(df["Y house price of unit area"]))
print(normal_sigma_MLE(df["Y house price of unit area"]))
```

```
37.980193236714975
13.59004480629316
```

Real Estate Evaluation data

...or simply....

```
print(np.mean(df["Y house price of unit area"]))
print(np.std(df["Y house price of unit area"]))
```

```
37.980193236714975
13.590044806293161
```

Real Estate Evaluation data

...or simply....

```
print(np.mean(df["Y house price of unit area"]))
print(np.std(df["Y house price of unit area"]))
```

```
37.980193236714975
13.590044806293161
```

...or even more simply....

```
mu, std = scipy.stats.norm.fit(df["Y house price of unit area"])
print("mu estimate:", str(mu))
print("std estimate:", str(std))
```

```
mu estimate: 37.980193236714975
std estimate: 13.590044806293161
```

Real Estate Evaluation data

…or simply….

```
print(np.mean(df["Y house price of unit area"]))
print(np.std(df["Y house price of unit area"]))
```

```
37.980193236714975
13.590044806293161
```

…or even more simply….

```
mu, std = scipy.stats.norm.fit(df["Y house price of unit area"])
print("mu estimate:", str(mu))
print("std estimate:", str(std))
```

```
mu estimate: 37.980193236714975
std estimate: 13.590044806293161
```

Alternatively, program direct minimization…

Transplant survival data

- This data set is contained in the Python module "statsmodels", that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration.
- This data contains the survival time after receiving a heart transplant, the age of the patient and whether or not the survival time was censored.

Transplant survival data

- This data set is contained in the Python module "statsmodels", that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration.
- This data contains the survival time after receiving a heart transplant, the age of the patient and whether or not the survival time was censored.

```
import numpy as np
import scipy.stats as st
import statsmodels.datasets
import matplotlib.pyplot as plt
%matplotlib inline
data = statsmodels.datasets.heart.load_pandas().data

data.tail()
```

	survival	censors	age
64	14.0	1.0	40.3
65	167.0	0.0	26.7
66	110.0	0.0	23.7
67	13.0	0.0	28.9
68	1.0	0.0	35.2

Transplant survival data

- This dataset contains censored and uncensored data: a censor of 0 means that the patient was alive at the end of the study, and thus we don't know the exact survival time.
- We only know that the patient survived at least the indicated number of days.
- For simplicity here, we only keep uncensored data (we thereby introduce a bias toward patients that did not survive very long after their transplant)
- The methods of "survival data analysis" deal with the analysis of censored data.

Transplant survival data

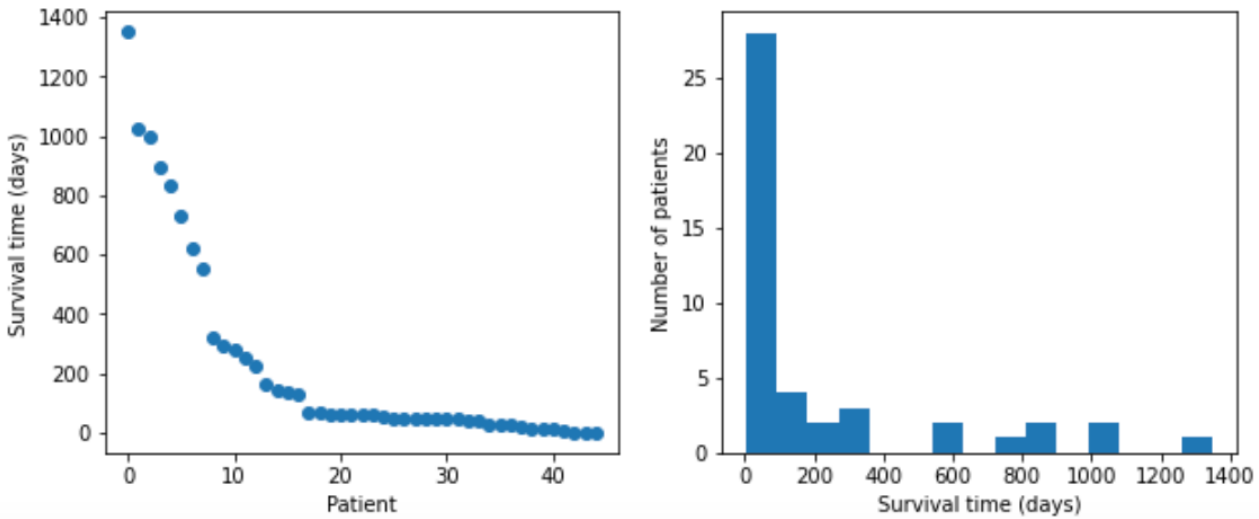
```
data = data[data.censors == 1]
survival = data.survival

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 4))

#Plot raw survival data
ax1.plot(sorted(survival)[::-1], 'o')
ax1.set_xlabel('Patient')
ax1.set_ylabel('Survival time (days)')

#Plot histogram
ax2.hist(survival, bins=15)
ax2.set_xlabel('Survival time (days)')
ax2.set_ylabel('Number of patients')
```

```
Text(0, 0.5, 'Number of patients')
```



Transplant survival data

- We fit an exponential model to the data. The ML estimate is $\frac{1}{\bar{x}}$, the survival rate

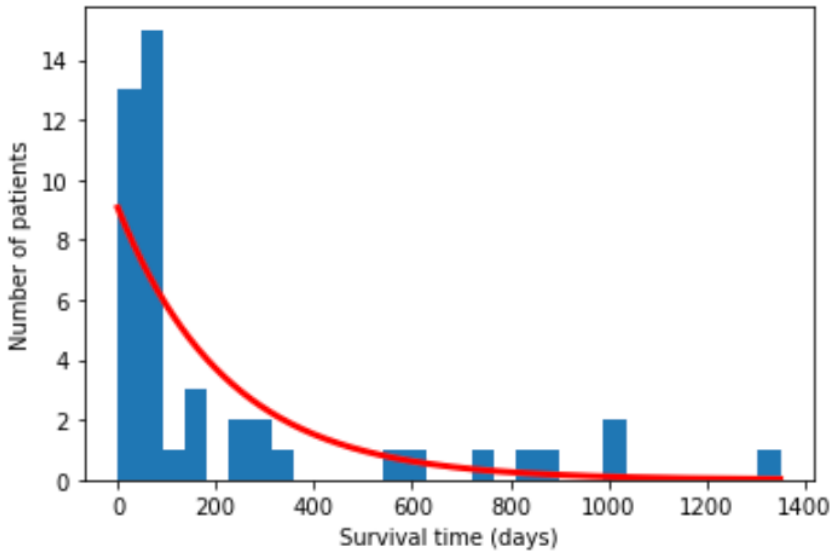
```
smean = survival.mean()
rate = 1. / smean
print(rate) #0.0044785
```

- Alternatively, we can esort to numerical methods. SciPy actually integrates numerical maximum likelihood routines for a large number of distributions.

```
dist = st.expon
args = dist.fit(survival)
args #(1.0, 222.29)
#1/222.29=0.004498627
```

Plot of the fit

```
smax = survival.max()
days = np.linspace(0., smax, 1000)
# bin size: interval between two
# consecutive values in `days`
dt = smax / 999.
dist_exp = st.expon.pdf(days, scale=1. / rate)
nbins = 30
fig, ax = plt.subplots(1, 1, figsize=(6, 4))
ax.hist(survival, nbins)
ax.plot(days, dist_exp * len(survival) * smax / nbins,
        '-r', lw=3)
ax.set_xlabel("Survival time (days)")
ax.set_ylabel("Number of patients")
```



Not a great fit…

Probability & Statistics for DS & AI

Properties of Maximum likelihood estimators

Michele Guindani

Summer

Estimate vs Estimator

- An estimate is a number, e.g., $\widehat{\theta}_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n$. It is the random realization of a random variable.
- An estimator is a random variable, e.g., $\widehat{\Theta}_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N} X_n$. It takes a set of random variables and generates another random variable.

Estimate vs Estimator

- An estimate is a number, e.g., $\widehat{\theta}_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n$. It is the random realization of a random variable.
- An estimator is a random variable, e.g., $\widehat{\Theta}_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N} X_n$. It takes a set of random variables and generates another random variable.
- An estimator is any function that takes the data points $X_1, \ldots, X_N$ and maps them to a number (or a vector of numbers). That is, an estimator is

$$\widehat{\Theta}\left(X_1, \ldots, X_N\right)$$

We call $\widehat{\Theta}$ the estimator of the true parameter θ.

Estimate vs Estimator

- An estimate is a number, e.g., $\widehat{\theta}_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n$. It is the random realization of a random variable.
- An estimator is a random variable, e.g., $\widehat{\Theta}_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N} X_n$. It takes a set of random variables and generates another random variable.
- An estimator is any function that takes the data points $X_1, \ldots, X_N$ and maps them to a number (or a vector of numbers). That is, an estimator is

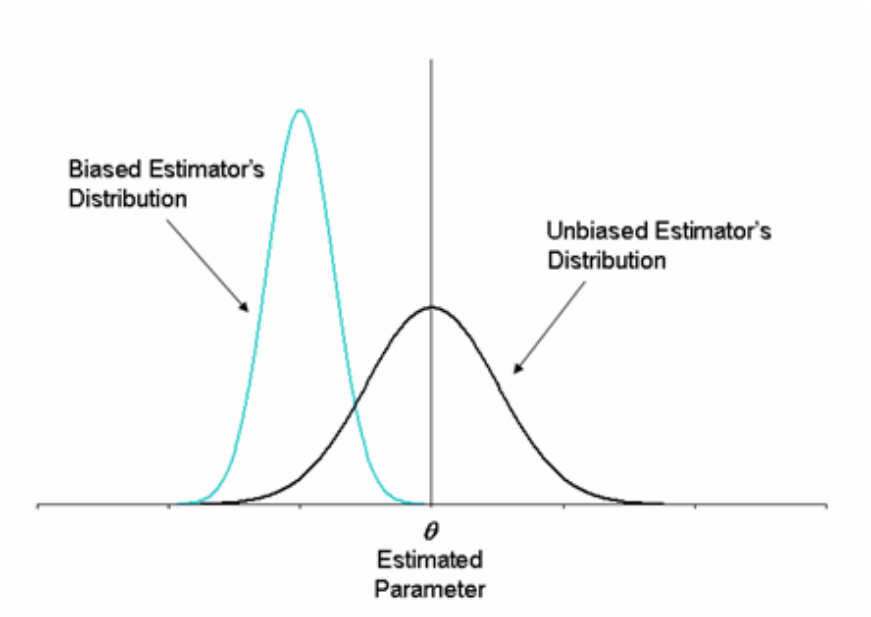
$$\widehat{\Theta}\left(X_1, \ldots, X_N\right)$$

We call $\widehat{\Theta}$ the estimator of the true parameter θ.

- The ML estimators are just one type of estimator, namely those that maximize the likelihood functions. If we do not want to maximize the likelihood we can still define an estimator.

Unbiasedness

- What is an unbiased estimator?
- An estimator $\widehat{\Theta}$ is unbiased if $\mathbb{E}[\widehat{\Theta}] = \theta$
- Unbiased means that the statistical average of $\widehat{\Theta}$ is the true parameter θ.
- If $X_n \sim$ Gaussian $\left(\theta, \sigma^2\right)$, then $\widehat{\Theta} = (1/N) \sum_{n=1}^{N} X_n$ is unbiased, but $\widehat{\Theta} = X_1$ is biased.
- A MLE is not necessarily unbiased.



An ideal illustration

A MLE may be biased

Example

Let $X_1, \ldots, X_N$ be i.i.d. Gaussian random variables with unknown mean μ and unknown variance σ^2. The ML estimators are

$$\widehat{\mu}_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N} X_n \quad \text{and } \widehat{\sigma}^2_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N} (X_n - \widehat{\mu}_{\text{ML}})^2$$

A MLE may be biased

Example

Let $X_1, \ldots, X_N$ be i.i.d. Gaussian random variables with unknown mean μ and unknown variance σ^2. The ML estimators are

$$\widehat{\mu}_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} X_n \quad \text{and } \widehat{\sigma}^2_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} (X_n - \widehat{\mu}_{\mathrm{ML}})^2$$

It is easy to show that $\mathbb{E}\left[\widehat{\mu}_{\mathrm{ML}}\right] = \mu$.

However,

$$\mathbb{E}\left[\widehat{\sigma}^2_{\mathrm{ML}}\right] = \frac{N-1}{N}\sigma^2$$

which is not equal to σ^2. Therefore, $\widehat{\sigma}^2_{\mathrm{ML}}$ is a biased estimator of σ^2

Unbiased estimator of the variance

- In the previous example, it is possible to construct an unbiased estimator for the variance. To do so, we can use

$$\widehat{\sigma}^2_{\text{unbias}} = \frac{1}{N-1}\sum_{n=1}^{N}(X_n - \widehat{\mu}_{\text{ML}})^2$$

so that $\mathbb{E}\left[\widehat{\sigma}^2_{\text{unbias}}\right] = \sigma^2$.

⚠ $\widehat{\sigma}^2_{\text{unbias}}$ does not maximize the likelihood, so while you can get unbiasedness, you cannot maximize the likelihood. If you want to maximize the likelihood, you cannot get unbiasedness.

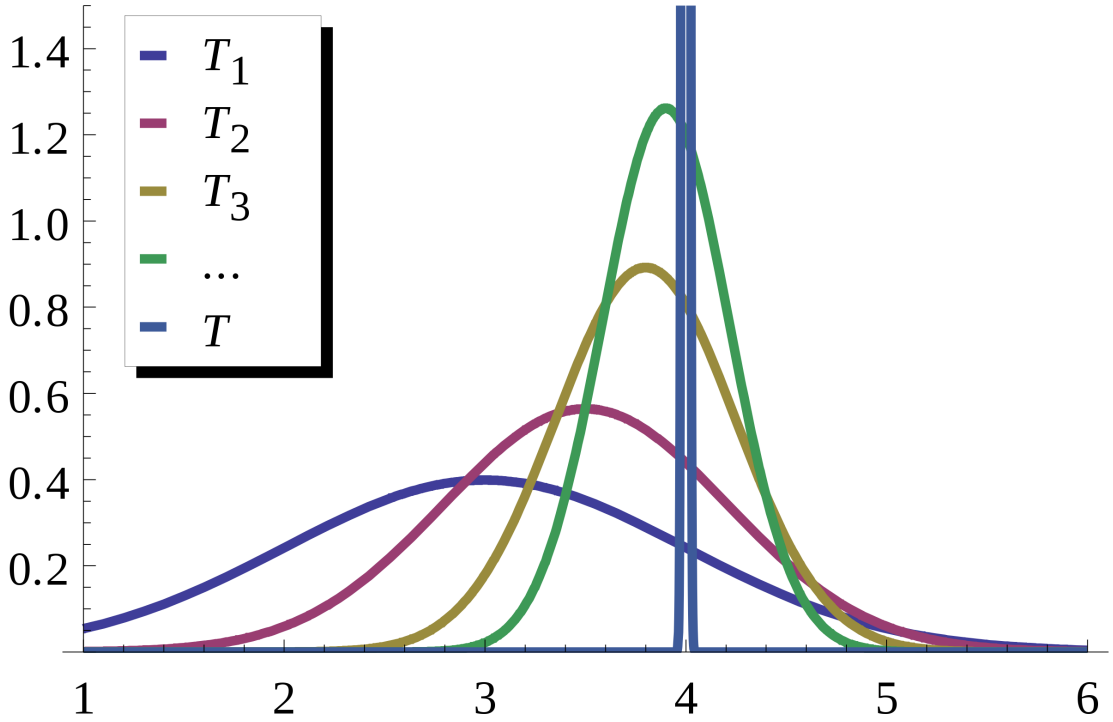
Consistency

Consistency

- Consistent Estimator

Consistency

- Consistent Estimator = If you have enough samples, then the estimator $\widehat{\Theta}$ will converge to the true parameter. See textbook for more details.



$\{T_1, T_2, T_3, ...\}$ is a sequence of estimators for parameter θ_0, the true value of which is 4. This sequence is consistent: the estimators are getting more and more concentrated near the true value θ_0; at the same time, these estimators are biased. The limiting distribution of the sequence is a degenerate random variable which equals ϑ_0 with probability 1.

From: Wikipedia

Consistency vs Unbiasedness

- Unbiasedness does not imply consistency.

Consistency vs Unbiasedness

- Unbiasedness does not imply consistency.

  For example (Gaussian), if

  $$\widehat{\Theta} = X_1$$

  then $\mathbb{E}[X_1] = \mu$. But the absolute difference between the estimator and the true mean does not converge to 0 (in probability) as N grows. So this estimator is inconsistent. See Example 8.16 in the textbook.

Consistency vs Unbiasedness

- Unbiasedness does not imply consistency.

  For example (Gaussian), if

  $$\widehat{\Theta} = X_1$$

  then $\mathbb{E}[X_1] = \mu$. But the absolute difference between the estimator and the true mean does not converge to 0 (in probability) as N grows. So this estimator is inconsistent. See Example 8.16 in the textbook.

- Consistency does not imply unbiasedness.

Consistency vs Unbiasedness

- Unbiasedness does not imply consistency.

  For example (Gaussian), if

  $$\widehat{\Theta} = X_1$$

  then $\mathbb{E}[X_1] = \mu$. But the absolute difference between the estimator and the true mean does not converge to 0 (in probability) as N grows. So this estimator is inconsistent. See Example 8.16 in the textbook.

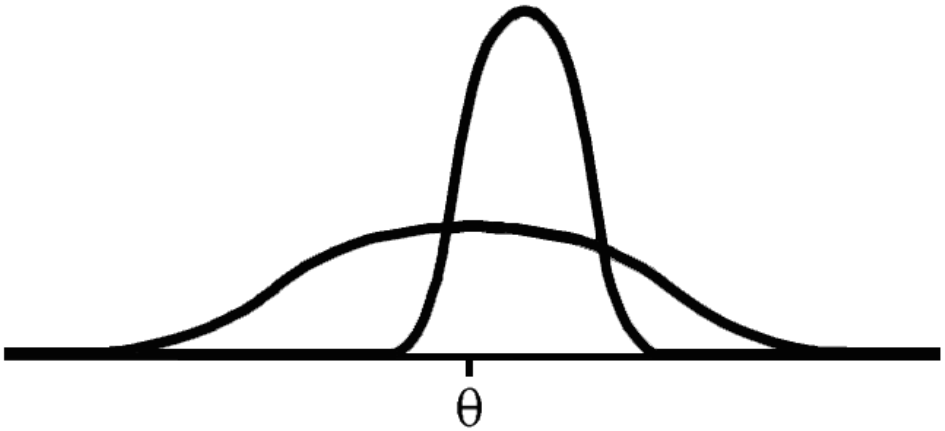
- Consistency does not imply unbiasedness.

  For example (Gaussian),

  $$\widehat{\Theta} = \frac{1}{N}\sum_{n=1}^{N}(X_n - \mu)^2$$

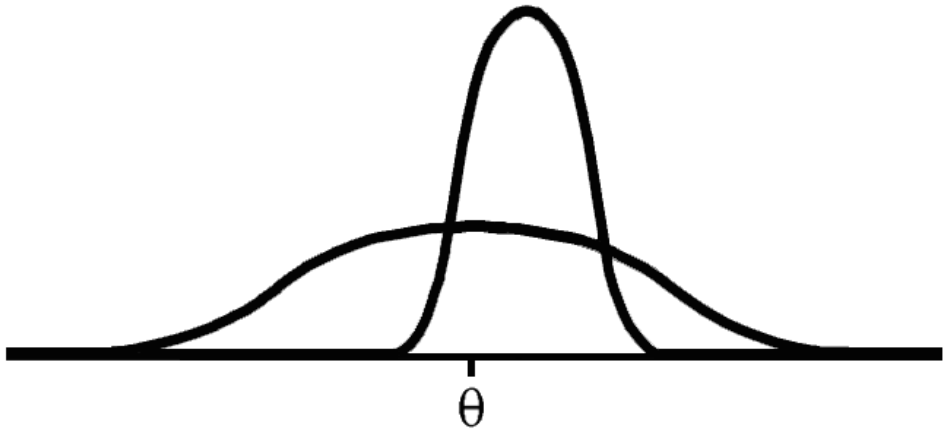
  is a biased estimate for variance, but it is consistent. See Example 8.17 in the textbook.

A first encounter with bias-variance trade-off



PDFs are indicated for two estimators of a parameter θ. One is unbiased. The other is biased but has lower standard error.

A first encounter with bias-variance trade-off



PDFs are indicated for two estimators of a parameter θ. One is unbiased. The other is biased but has lower standard error.

Me

squared error (MSE) combines the notions of bias and standard error. It is defined as

$$\text{MSE} = E\left([X-\theta]^2\right) = \underbrace{E\left([X-E(X)]^2\right)}_{\text{standard error }^2} + \underbrace{(E(X)-\theta)^2}_{\text{bias }^2}$$

Invariance of MLE

- What is the invariance principle?
- There is a monotonic function h.
- There is an ML estimate $\widehat{\theta}_{\text{ML}}$ for θ.
- The monotonic function h maps the true parameter $\theta \longmapsto h(\theta)$
- Then the same function will map the ML estimate $\widehat{\theta}_{\text{ML}} \longmapsto h\left(\widehat{\theta}_{\text{ML}}\right)$