Data-Driven Sampling (Lecture 5)

Expectation-Maximum (EM) Algorithm by an example

Hongwei YUAN

University of Macau

Maximum log likelihood estimator (log-MLE)

Definition

Let $p(x, \theta)$ be a probability distribution with an unknown parameter θ. Let $X_1, ..., X_n$ be n independent random variables with distribution $p(x, \theta)$. The **log likelihood function** based on $X_1, ..., X_n$ is defined as

$$l_n(\theta) = \log L_n(\theta) = \sum_{i=1}^{n} \log p(X_i, \theta).$$

The **maximum likelihood estimator (MLE)** of θ based on $X_1, ..., X_n$ can also be defined as

$$\hat{\theta}_n = argmax_{\theta \in \Theta} l_n(\theta).$$

For a continuous distribution, we replace $p(x, \theta)$ by the pdf $f(x, \theta)$ in the definition.

Example (Class Exercise)

method 3

$l_n(\theta) := \sum_{i=1}^{n} \log f(X_i;\theta)$ $\log f(x) = \log\theta - \theta x$

$= n \cdot \log\theta - \theta \cdot (X_1 + X_2 + \cdots + X_n)$.

$\hat{\theta}_n := \arg\max_{\theta} l_n(\theta)$

$l_n'(\theta) = 0 \Rightarrow \frac{n}{\theta} - (X_1 + X_2 + \cdots + X_n) = 0$

$\Rightarrow \theta = \frac{n}{X_1 + \cdots + X_n}$

$\frac{1}{\theta} = \bar{X} := \frac{X_1 + \cdots + X_n}{n}$

Let $X_1, \ldots, X_n$ be a sequence of random variables from exponential distribution $Exp(\theta)$. (The pdf of $Exp(\theta)$ is $f(x;\theta) = \theta e^{-\theta x}$ for $x \geq 0$ and $f(x;\theta) = 0$ for $x < 0$.)

- Find the log likelihood function $l_n(\theta)$ based on $X_1, \ldots, X_n$.
- Find the MLE $\hat{\theta}_n$ of θ.
- Assignment 1: When $\theta = 1$, write a python code which includes: (1) sample n (e.g. $n = 1000$) Exp(1) distributed random numbers, (2) find the MLE $\hat{\theta}_n$.

File Edit View Navigate Code Refactor Run Tools VCS Window Help Lecture 5.py - Lecture 5.py

python project > Lecture 5.py

Lecture 9.py | Bandits and Thompson sampling.py | MATH2005.py | Lecture3.py | Lecture 5.py | Lecture 4.py

```python
#Python of MLE for Exponential distribution
import numpy as np
from scipy.optimize import minimize_scalar
import matplotlib.pyplot as plt
n=10000
X=np.random.exponential(1,n)
plt.hist(X,bins=100,density=True)
plt.title('Density histogram of Exp(1)')
plt.xlabel('x')
plt.ylabel('Density')
plt.show(block=True)

def L(x): #Define -log MLE function
    return -n*np.log(x)+x*sum(X)
res=minimize_scalar(L) #find minimizer
print(res.x)
#Stochastic Search
size=1000
x=np.random.uniform(0.1,2,size)
mp=x[0]
for i in range(size):
    if L(x[i])<L(mp):
        mp=x[i]
print(mp)

#MLE algorithm without considering the missing data
import numpy as np
```

$x \in R^m$

$\|Ax-b\|$

method 1

method 2

Python Console

```
PyDev console: starting.

Python 3.10.13 | packaged by Anaconda, Inc. | (main, Sep 11 2023, 13:24:38) [MSC v.1916 64 bit (AM
>>>
```

Special Variables

Version Control | Python Packages | TODO | Problems | Terminal | Python Console | Services

Localized PyCharm 2022.2.3 is available // Switch and restart // Don't ask again (12 minutes ago)

MLE in practice

- In real world applications, we do **NOT** have random variables $X_1, ..., X_n$ BUT observed data $x_1, ..., x_n$. Naturally we replace $X_1, ..., X_n$ by $x_1, ..., x_n$ in the (log) likelihood function.
- **Example (patients recovery time):** n patients recovered from deceases by a new medical treatment, their recovery times are recorded as $x_1, ..., x_n$, which approximately satisfy an exponential distribution Exp(θ) with an unknown $\theta > 0$.
- To estimate the θ, we need to use a log likelihood function based the observed data $x_1, ..., x_n$ rather than random variables $X_1, ..., X_n$, i.e.,

$$l_n(\theta) = n \log \theta - \theta(x_1 + ... + x_n),$$

  and maximize $l_n(\theta)$.

Example (Class Exercise)

Suppose that $x_1, ..., x_n$ are observed from $N(\theta, 1)$ with θ unknown (the pdf is $f(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-\mu)^2}{2\sigma^2}}, \quad -\infty < y < \infty$),

- write down the log likelihood function $l_n(\theta)$ based on the observed data $x_1, ..., x_n$ and find the associated MLE $\hat{\theta}_n$.
- Assignment 2: When $\theta = 0$, write a python code which includes: (1) sample n (e.g. $n = 1000$) $N(0, 1)$ distributed random numbers, (2) find the MLE $\hat{\theta}_n$.

$$l_n(\theta) = \sum_{i=1}^{n} \log f(x_i; \mu=\theta; \sigma=1) = \sum_{i=1}^{n} -\frac{1}{2}\log 2\pi - \frac{1}{2}(x_i-\theta)^2 = -\frac{1}{2}\sum_{i=1}^{n} \log 2\pi + (x_i-\theta)^2$$

$$\hat{\theta}_n = \arg\max_{\theta} l_n(\theta) \Leftrightarrow 0 = l_n'(\theta) = \sum_{i=1}^{n}(x_i-\theta) = \sum_{i=1}^{n} x_i - n\theta \quad \Rightarrow \hat{\theta}_n = \frac{\sum_{i=1}^{n} x_i}{n}$$

Missing data of MLE in practice: patients recovery time example

- Suppose that 80 patients used by a new medical treatment, after 40 days of their treatment, the hospital have a follow-up check to obtain patients' recovery times, e.g. (unit is day)

$$x_1 = 10, x_2 = 30, ..., x_{70} = 35, x_{71} > 40, ..., x_{80} > 40.$$

  here '> 40' means that the patient has not recovered when doing a follow-up check.

- A simple way is to remove the data '> 40' and use the exact data to get an MLE. However, this removal will make us lose useful information.

Generalized likelihood function for missing data

Definition

Let $p(x, \theta)$ be a probability distribution with an unknown parameter θ. Let $x_1, ..., x_m$ be m observed data, and there are $n - m$ missed data with some partial information (e.g. we only know these missed data are all larger than a number a). The **generalized log likelihood function** based on $x_1, ..., x_m$ and partial information of $(n - m)$ missing data is defined as

$$\tilde{l}_n(\theta) = \left[\sum_{i=1}^{m} \log p(x_i, \theta)\right] + c(x_{m+1}, ..., x_n, \theta).$$

where $c(x_{m+1}, ..., x_n, \theta)$ is the likelihood based on the partial information of $n - m$ missing data.

Generalized MLE for missing data

Definition

The **maximum likelihood estimator (MLE)** of θ based on $x_1, ..., x_m$ and partial information of missing data $x_{m+1}, ..., x_n$ is defined as

$$\hat{\theta}_n = argmax_{\theta \in \Theta} \tilde{l}_n(\theta).$$

Patients's recovery from a new medical treatment

Example: For 80 patients using a new medical treatment, they are expected to recover from it according to an exponential distribution Exp(θ) with some unknown $\theta > 0$. A follow-up check found that

$$x_1 = 10, x_2 = 30, ..., x_{70} = 35, x_{71} > 40, ..., x_{80} > 40.$$

The solution:

- The first 70 data are known, so we use

$$\prod_{i=1}^{70} \theta e^{-\theta x_i}.$$

Patients's recovery from a new medical treatment

The solution (contiued):

- The remaining 10 data are all larger than 40, all we known is that each occurs with probability

$$P(X > 40) = \int_{40}^{\infty} \theta e^{-\theta t} \mathrm{d}t = e^{-40\theta},$$

hence we use this partial information as

$$e^{-40\theta} \ldots e^{-40\theta} = e^{-400\theta}.$$

- The generalized likelihood function is $\tilde{L}_n(\theta) = (\prod_{i=1}^{70} \theta e^{-\theta x_i}) e^{-400\theta}$.
- The generalized log likelihood is

$$\tilde{l}_n(\theta) = 70 \log \theta - \theta(x_1 + \ldots + x_{70}) - 400\theta.$$

Patients's recovery from a new medical treatment

Class Exercise:

- Find the maximizer $\widetilde{\theta}_n$ of $\tilde{l}_n(\theta)$, that is $\tilde{l}_n(\widetilde{\theta}_n) = \max_\theta \tilde{l}_n(\theta)$.
- Find the MLE $\widehat{\theta}_n$ only with the fully observed data $x_1, ..., x_{70}$, that is, find $\widehat{\theta}_n$ such that $l_n(\widehat{\theta}_n) = \max_\theta l_n(\theta)$ where $l_n(\theta) = 70 \log \theta - \theta(x_1 + ... + x_{70})$.
- Compare $\widetilde{\theta}_n$ with $\widehat{\theta}_n$.

$$\tilde{l}_n(\theta) = 70 \log \theta - \theta \cdot (x_1 + \cdots + x_{70}) - 400\theta$$

$$0 = \tilde{l}'(\theta) = \frac{70}{\theta} - \sum_{i=1}^{70} x_i - 400 \Rightarrow \frac{1}{\tilde{\theta}_n} = \frac{\sum_{i=1}^{70} x_i + 400}{70}$$

$$> 40$$

$$l_n(\theta) = 70 \log \theta - \theta \cdot (x_1 + \cdots + x_{70})$$

$$0 = l_n'(\theta) = \frac{70}{\theta} - \sum_{i=1}^{70} x_i \Rightarrow \frac{1}{\hat{\theta}_n} = \frac{\sum_{i=1}^{70} x_i}{70} < \frac{1}{\tilde{\theta}_n}$$

Motivation of EM algorithm

- In many applications, the partial information $c(x_{m+1}, ..., x_n, \theta)$ cannot be written down explicitly and thus one cannot find MLE by maximizing $\tilde{l}_n(\theta)$
- Fortunately, we can use expectation-maximum algorithm.

A simple example

Censored data may come from experiments where some large observations exceeding a threshold a are just recorded as **'larger than** a**'**. Suppose that $x_1, ..., x_m$ are observed from $N(\theta, 1)$ with θ unknown, and that the remaining $y_{m+1}, ..., y_n$ are all larger than a.

- The generalized likelihood function are

$$\tilde{l}_n(\theta) = \sum_{i=1}^{m} \log \varphi(x_i - \theta) + (n-m) \log[1 - \Phi(a - \theta)].$$

  where $\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ and $\Phi(x) = \int_{-\infty}^{x} \varphi(t) \mathrm{d}t$.

- Because we do not have an explicit formula for Φ, it is not easy to find the maximizer of $\tilde{l}_n$.

$$0 = \tilde{l}_n'(\theta) = -\sum_{i=1}^{m} \frac{\varphi'(x_i-\theta)}{\varphi(x_i-\theta)} + (n-m)\frac{\Phi'(a-\theta)}{1-\Phi(a-\theta)}$$

$$= \sum_{i=1}^{m} \frac{\cancel{\varphi(x_i-\theta)}\cdot(x_i-\theta)}{\cancel{\varphi(x_i-\theta)}} + (n-m)\frac{\varphi(a-\theta)}{1-\Phi(a-\theta)}$$

$$\Phi'(x) = \varphi(x)$$

$$\varphi'(x) = \varphi(x)\cdot(-x)$$

$\Rightarrow \hat{\theta}_n$ satisfies $\underline{m\cdot(\bar{x}-\theta) + (n-m)\dfrac{\varphi(a-\theta)}{1-\Phi(a-\theta)} = 0}$

$$\bar{x} := \frac{1}{m}\sum_{i=1}^{m} x_i$$

1. Stochastic search

randomly sample from Uniform distribution.

$\theta_1, \cdots, \theta_{1000}$

evaluate $\tilde{l}_n(\theta_1), \cdots, \tilde{l}_n(\theta_{1000})$

$$\hat{\theta}_n := \underset{\theta \in \{\theta_1, \cdots, \theta_{1000}\}}{\arg\max} \{\tilde{l}_n(\theta_1), \cdots, \tilde{l}_n(\theta_{1000})\}$$

2.

EM algorithm: an crucial conditional probability

- For the $n-m$ partially observed data, we take them as independent random variables :

$$Y_{m+1}, ..., Y_n.$$

- However, based on the partially observation on these data, we have more information about Y_k than it was drawn from the distribution $p(x,\theta)$.
- We will compute the distribution of Y_k conditioned on the partial observation, this conditional distribution will play a crucial rule in EM algorithm

EM algorithm

$$Q(\tilde{\theta}|\theta, x_1, \cdots, x_m) := \mathbb{E}_\theta\left(l_n(\tilde{\theta}, x_1, \cdots, x_m, Y_{m+1}, \cdots, Y_n) \middle| Y_{m+1} > a, Y_{m+2} > a, \cdots, Y_n > a \right)$$

$$Y_i \sim p(x, \theta) \Rightarrow f(Y_i | Y_i > a) = \frac{\varphi(y)}{\int_a^\infty \varphi(t)\,dt}$$

$$\tilde{l}_n(\theta) = \sum_{i=1}^m \log \varphi(x_i - \theta) + (n-m) \cdot \log(1 - \Phi(a - \theta)).$$

$$0 = \tilde{l}_n'(\theta) \Rightarrow \tilde{\theta}_n \qquad \mathbb{E}(Y_{m+1} | Y_{m+1} > a)$$

For a given θ (e.g. $\theta = 0.5$), define

$$Q(\tilde{\theta}|\theta, x_1, ..., x_m) = \mathbb{E}_\theta[l_n(\tilde{\theta}, x_1, ..., x_m, Y_{m+1}, ..., Y_n) | Y_{m+1} > a, \cdots, Y_n > a],$$

where $l_n(\tilde{\theta}, x_1, ..., x_m, Y_{m+1}, ..., Y_n) = \sum_{i=1}^m \log p(x_i, \tilde{\theta}) + \sum_{i=m+1}^n \log p(Y_i, \tilde{\theta})$

$\mathbb{E}_\theta$ is the expectation with respect to $Y_{m+1}, ..., Y_n$ based on the partial information.

EM algorithm:

- Pick a starting value $\hat{\theta}_0$
- Repeat
  - E-step: compute $Q(\tilde{\theta}|\hat{\theta}_k, x_1, ..., x_m) = \mathbb{E}_{\hat{\theta}_k}[l_n(\tilde{\theta}, x_1, ..., x_n, Y_{m+1}, ..., Y_n)]$.
  - M-step: maximize $Q(\tilde{\theta}|\hat{\theta}_k, x_1, ..., x_m)$ about $\tilde{\theta}$ to obtain $\hat{\theta}_{k+1}$, i.e.

$$\hat{\theta}_{k+1} = argmax_{\theta \in \Theta} Q(\tilde{\theta}|\hat{\theta}_k, x_1, ..., x_m).$$

EM algorithm for the simple example above

(1) Let θ be given (e.g. $\theta = 0.5$), then the densitiy function is $f_\theta(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(y-\theta)^2}{2}}$.

(2) (Class Exercise) For a random variable Y with a density $f_\theta(y)$, if we further know $Y \geq a$ for a given number a, show that the probability density of Y given the condition $Y \geq a$ is

$$g_\theta(y) = \begin{cases} 0, & y < a, \\ \frac{f_\theta(y)}{\int_a^\infty f_\theta(t)\mathrm{d}t}, & y \geq a. \end{cases}$$

(3) Assignment 2: Show that $Q(\tilde{\theta}|\theta, x_1, ..., x_m) = \underbrace{-\frac{1}{2}\sum_{i=1}^m (x_i - \tilde{\theta})^2 - \frac{1}{2}\sum_{i=m+1}^n \mathbb{E}_\theta[(Y_i - \tilde{\theta})^2] - n/2\log(2\pi)}$, where $\mathbb{E}_\theta$ is the expectation with respect to the probability density g_θ.

$$Q(\tilde{\theta} \mid \theta, x_1, \cdots, x_m) := E_\theta\left(l_n(\tilde{\theta}, x_1, \ldots, x_m, Y_{m+1}, \ldots, Y_n) \mid Y_{m+1} > a, Y_{m+2} > a, \cdots, Y_n > a \right)$$

$$Y_i \sim f(x, \theta) \Rightarrow f(Y_i \mid Y_i > a) = \frac{\varphi(y)}{\int_a^\infty \varphi(t)\,dt}$$

$\hat{\theta}_{k+1}$

$$\tilde{l}_n(\theta) = \sum_{i=1}^{m} \log \varphi(x_i - \theta) + (n-m) \cdot \log(1 - \Phi(a-\theta))$$

$$0 = \tilde{l}_n'(\theta) \Rightarrow \tilde{\theta}_n$$

$\hat{\theta}_0, \hat{\theta}_1, \hat{\theta}_2, \cdots$

$E(Y_{m+1} \mid Y_{m+1}$

$$P(Y=y \mid Y \geq a) = \frac{P(Y=y \text{ and } Y \geq a)}{P(Y \geq a)} = \begin{cases} 0, & \text{if } y \leq a \\ \frac{P(Y=y)}{P(Y \geq a)}, & \text{if } y > a \end{cases}$$

$$\frac{P(Y=y)}{P(Y \geq a)} = \frac{f_\theta(y)}{\int_a^\infty f_\theta(y)\,dy}$$

$$P(Y_{m+1} = y \mid Y_{m+1} > a) = \frac{P(Y_{m+1} = y \text{ and } Y_{m+1} > a)}{P(Y_{m+1} > a)} = \begin{cases} 0, & y \leq a \\ \frac{P(Y_{m+1}=y)}{P(Y_{m+1}>a)}, & y > a \end{cases}$$

1.) Stochastic search

randomly sample

$\theta_1, \cdots, \theta_{100}$

evaluate $\tilde{l}_n(\theta_i)$

$\hat{\theta} := \arg\max \{ l$

$$E_\theta\left((Y_i - \tilde{\theta})^2 \mid Y_i > a \right) = E_\theta\left((Y_i^2 - 2\tilde{\theta} \cdot Y_i + \tilde{\theta}^2) \mid Y_i > a \right)$$

$$= E_\theta(Y_i^2 \mid Y_i > a) - 2\tilde{\theta} E_\theta(Y_i \mid Y_i > a) + \tilde{\theta}^2$$

$$= \int_{-\infty}^{\infty} (y - \tilde{\theta})^2 g_\theta(y)\,dy$$

$$= \int_a^\infty (y - \tilde{\theta})^2 \cdot \frac{f_\theta(y)}{\int_a^\infty f_\theta(t)\,dt}\,dy$$

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

$(-1)\cdot(-1) = 1$

$$\frac{d}{d\tilde{\theta}} Q(\tilde{\theta}) = -\frac{1}{2}\sum_{i=m+1}^{n}\left(-2E_\theta(Y_i \mid Y_i > a) + 2\tilde{\theta}\right) + \frac{1}{2} \cdot 2 \cdot \sum_{i=1}^{m} (x_i - \tilde{\theta}) = 0$$

$$\sum_{i=m+1}^{n} E_\theta(Y_i \mid Y_i > a) - (n-m)\tilde{\theta} + \sum_{i=1}^{m} x_i - n\tilde{\theta} = 0$$

$$\Rightarrow \tilde{\theta} = \frac{1}{n}\left(\sum_{i=1}^{m} x_i + \sum_{i=m+1}^{n} E_\theta(Y_i \mid Y_i > a) \right)$$

$$\Phi(x) = \int_{-\infty}^{x} \varphi(t)\,dt$$

1.) Stoch

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

$$\int_{a-\theta}^{\infty} x \cdot \varphi(x)\,dx = \frac{1}{\sqrt{2\pi}} \int_{a-\theta}^{\infty} e^{-\frac{x^2}{2}}\, d\left(\frac{x^2}{2}\right)$$

random

$$\int_{a-\theta}^{\infty} z \cdot \varphi(z)\,dz$$

$$= \frac{1}{\sqrt{2\pi}}\left(-e^{-\frac{x^2}{2}}\right)\Big|_{a-\theta}^{\infty}$$

θ_1

$$= \frac{1}{\sqrt{2\pi}} e^{-\frac{(a-\theta)^2}{2}}$$

$$E_\theta(Y_i \mid Y_i > a) = \int_{-\infty}^{\infty} y \cdot g_\theta(y)\,dy$$

$$= \int_a^\infty \frac{(y - \theta + \theta) \cdot \varphi(y-\theta)}{\int_a^\infty \varphi(t-\theta)\,dt}\,dy = \frac{\varphi(a-\theta)}{1 - \Phi(a-\theta)} + \theta \cancel{\frac{\int_a^\infty \varphi(y-\theta)\,dy}{\int_a^\infty \varphi(t-\theta)\,dt}}$$

$$= \theta + \frac{\varphi(a-\theta)}{1 - \Phi(a-\theta)}$$

ev

θ_i

$\varphi(a-\theta)$

1-1

$\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_j, \cdots$

Show: $\hat{\theta}_j \to \theta^*$ θ^* is the solution of (

$\lim_{j\to\infty} \theta_j = \theta^*$

$$\theta^* = \frac{m}{n}\bar{x} + \frac{n-m}{n}\left(\theta^* + \frac{\varphi(a-\theta^*)}{1-\Phi(a-\theta^*)}\right)$$

$$\Leftrightarrow \frac{m}{\cancel{n}}\theta^* = \frac{m}{\cancel{n}}\bar{x} + \frac{n-m}{\cancel{n}}\left(\frac{\varphi(a-\theta^*)}{1-\Phi(a-\theta^*)}\right)$$

$$\Leftrightarrow 0 = m(\bar{x}-\theta^*) + (n-m)\frac{\varphi(a-\theta^*)}{1-\Phi(a-\theta^*)}$$

$\lim_{j\to\infty} \theta_j = \theta^*$

$$\theta_{j+1} = m\bar{x} - (m-1)\theta_j + (n-m)\frac{\varphi(a-\theta_j)}{1-\boxed{\Phi(a-\theta_j)}}$$

high complexity
count this is executed by how many times

$\tilde{l}_n(\theta) = 0 \Updownarrow$

$$\Rightarrow \hat{\theta}_n \text{ satisfies } \underline{m\cdot(\bar{x}-\theta) + (n-m)\frac{\varphi(a-\theta)}{1-\Phi(a-\theta)} = 0}$$

$\bar{x} := \frac{1}{m}\sum_{i=1}^{m} x_i$

$\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_j, \cdots$

Show: $\hat{\theta}_j \to \theta^*$ θ^* is the solution of

$\lim_{j\to\infty} \theta_j = \theta^*$

$$\theta^* = \frac{m}{n}\bar{x} + \frac{n-m}{n}\left(\theta^* + \frac{\varphi(a-\theta^*)}{1-\Phi(a-\theta^*)}\right)$$

$$\Leftrightarrow \frac{m}{\cancel{n}}\theta^* = \frac{m}{\cancel{n}}\bar{x} + \frac{n-m}{\cancel{n}}\left(\frac{\varphi(a-\theta^*)}{1-\Phi(a-\theta^*)}\right)$$

$$\Leftrightarrow 0 = m(\bar{x}-\theta^*) + (n-m)\frac{\varphi(a-\theta^*)}{1-\Phi(a-\theta^*)}$$

$\lim_{j\to\infty} \theta_j = \theta^*$

$$\theta_{j+1} = m\bar{x} - (m-1)\theta_j + (n-m)\frac{\varphi(a-\theta_j)}{1-\Phi(a-\theta_j)}$$

θ_{1000} 次

$\tilde{l}_n(\theta) = 0 \Updownarrow$

$$\Rightarrow \hat{\theta}_n \text{ satisfies } m\cdot(\bar{x}-\theta) + (n-m)\frac{\varphi(a-\theta)}{1-\boxed{\Phi(a-\theta)}} = 0$$

$\bar{x} := \frac{1}{m}\sum_{i=1}^{m} x_i$

$\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_j, \cdots \boxed{\hat{\theta}_{1000}}$

Show: $\hat{\theta}_j \to \theta^*$ θ^* is the solution of

1.) Stochastic Search

1) randomly sample from Uniform distribution,

$\theta_1, \cdots, \theta_{1000}$

2) evaluate $\tilde{l}_n(\theta_1), \cdots, \hat{l}_n(\theta_{1000})$

$(-1)\cdot(-1) = 1$

$+ \frac{1}{2}\cdot 2 \cdot \sum_{i=1}^{m}(x_i - \tilde{\theta}) = 0$

$$\hat{\theta}_n := \underset{\theta\in\{\theta_1,\cdots,\theta_{1000}\}}{\arg\max} \{\hat{l}_n(\theta_1), \cdots, \hat{l}_n(\theta_{1000})\}$$

2. $Q(\tilde{\theta}\mid\theta, x_1, \cdots, x_m)$

$$= E_\theta\left(l_n(\tilde{\theta}, x_1, \cdots, x_m, \underbrace{Y_{m+1}, \cdots, Y_n}_{\sim f(x,\theta)}) \mid Y_{m+1} > a, \cdots, Y_n > a\right).$$

EM algorithm for the simple example above

(4) Maximizing $\tilde{\theta}$: differentiating $Q(\tilde{\theta}|\theta, x_1, ..., x_m)$ about $\tilde{\theta}$ and letting derivative equal to zero.

(Class Exercise) Show that

$$\begin{aligned}\hat{\theta} &= \frac{x_1 + ... + x_m}{n} + \frac{1}{n}\mathbb{E}_\theta[Y_{m+1} + ... + Y_n] \\ &= \frac{x_1 + ... + x_m}{n} + \frac{n-m}{n}\left(\theta + \frac{\varphi(a-\theta)}{1-\Phi(a-\theta)}\right),\end{aligned}$$

and that $\mathbb{E}_\theta[Y_{m+1}] = \frac{\int_a^\infty y\varphi(y-\theta)\mathrm{d}y}{\int_a^\infty \varphi(y-\theta)\mathrm{d}y} = \theta + \frac{\varphi(a-\theta)}{1-\Phi(a-\theta)}$, where $\varphi(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$ and $\Phi(x) = \int_{-\infty}^x \varphi(t)\mathrm{d}t$.

(5) EM algorithm

$$\theta_{j+1} = \frac{m}{n}\bar{x} + \frac{n-m}{n}\left(\theta_j + \frac{\varphi(a-\theta_j)}{1-\Phi(a-\theta_j)}\right),$$

for $j = 0, 1, 2, ...,$ where $\bar{x} = \frac{x_1 + ... + x_m}{m}$.

MLE algorithm without considering the missing data

```
import numpy as np
from scipy.stats import binom
import matplotlib.pyplot as plt
from scipy.optimize import minimize_scalar
# plt.show()
data_1=[]
data_2=[]
theta,size,bound=2,1000,2.5
X=np.random.normal(theta,1,size)
for i in range(size):
    if X[i]<=bound:
        data_1.append(X[i])
    else: data_2.append(X[i])
data_1=np.array(data_1)
data_2=np.array(data_2)
print(data_1)
print(data_2)
m_1=np.size(data_1)
m_2=np.size(data_2)
print(m_1)
print(m_2)
# Computing MLE without missing data
def l(x):
    return sum((data_1-x)**2/2)
mle_missing=minimize_scalar(l)      # find minimizer
mle_missing.x
print(mle_missing.x)
```

EM algorithm with considering the missing data

```
##The below is EM algorithm for the data_1 and data_2
theta_em0=-1 initial
delta=10 end iteration
data_3=[] empty list of Ym,....,Ym+k,...
loop_number=0
while delta>0.005:
    Y=np.random.normal(theta_em0, 1, size)
    for i in range(size):
        if Y[i]>bound: data_3.append(Y[i])
    data_3=np.array(data_3)
    m_3=np.size(data_3)
    loop_number=loop_number+1
    def l_em(x):
        return m_2*sum((data_3-x)**2/(2*m_3))+l(x)
    mle_em=minimize_scalar(l_em)
    theta_em=mle_em.x
    delta=theta_em-theta_em0
    theta_em0=theta_em
    data_3=[]
    continue

print(loop_number)
print(theta_em0)
```