Xi'an Jiaotong-Liverpool University

Pattern Recognition

Lecture 08(b). Parametric methods: MLE & MAP Practice

Dr. Shanshan ZHAO

School of AI and Advanced Computing
Xi'an Jiaotong-Liverpool University

Academic Year 2023-2024

Table of Contents

Notations

- X : The dataset observed
- x : the random variable, i.e., the feature vector
- x : the univariant , or a random variable in the feature vector
- θ : the parameters unknown in $p(\mathsf{x})$
- N : Number of samples
- $p(\theta|X)$ or $p(X|\theta)$: we consider θ and X as two random variables, this is to denote the dependence between variables
- $p(\mathsf{x}_k;\theta)$: The semicolon means that it is the pdf with respect to x_k, (x_k is the argument of function p), the parameter of it is θ .

Random variable *VS* Parameter

- Both Random variable and Parameter vary with some conditions.
- A 'variable' is something you measure when collecting data
- A 'parameter' is the link between variables

$p(x;\theta)$ VS $p(x|\theta)$

- $p(x;\theta)$: It is to denote a function p, the argument is x, the parameter of function is θ
- $p(x|\theta)$: It is to represent a conditional probability (density) function
- $L(\theta|D)$: The vertical bar might also be used when describing the likelihood
- Basically, vertical bar is to demonstrate the conditional relationship between two variables; semicolon to distinguish the argument and the parameter.

ML *VS* MAP estimate

ML estimate

In ML, we use the likelihood function

$$L = p(X;\theta) = \prod_{k=1}^{N} p(x_k;\theta) \qquad (1)$$

It is proportional to the conditional probability (or density) $p(X|\theta)$. ML estimates θ : the likelihood function takes its maximum value, that is,

$$\hat{\theta}_{ML} = arg \max_{\theta} \prod_{k=1}^{N} p(x_k;\theta) \equiv \max_{\theta} p(X|\theta) \qquad (2)$$

MAP estimate

$$\hat{\theta}_{MAP} = \max_{\theta}[p(X|\theta)p(\theta)] \qquad (3)$$

which is equivalent to

$$\hat{\theta}_{MAP} = \max_{\theta}[\ln p(X|\theta) + \ln p(\theta)] \qquad (4)$$

Frequentist *VS* Bayesian

- `https://www.youtube.com/watch?v=r76oDIvwETI`
- `https://www.youtube.com/watch?v=7-Ud4nyHO_Q`

ML *VS* MAP estimate

- Maximum likelihood is a special case of Maximum A Posterior estimation. To be specific, MLE is what you get when you do MAP estimation using a uniform prior.
- Both methods come about when we want to answer a question of the form: "What is the probability of scenario Y given some data, X, i.e. $P(Y|X)$.

A question of this form is commonly answered using Bayes' Law.

$$\underbrace{P(Y|X)}_{\text{posterior}} = \frac{\overbrace{P(X|Y)}^{\text{likelihood}}\overbrace{P(Y)}^{\text{prior}}}{\underbrace{P(X)}_{\text{probability of seeing the data}}}.$$

ML *VS* MAP estimate

- **MLE** If we're doing Maximum Likelihood Estimation, we do not consider prior information (another way of saying "we have a uniform prior") . In this case, the above equation reduces to

$$P(\theta|X) \propto P(X|\theta) \tag{5}$$

  In this scenario, we can fit a statistical model to correctly predict the posterior, $P(\theta|X)$, by maximizing the likelihood, $P(X|\theta)$. Hence "Maximum Likelihood Estimation."
- **MAP** If we know something about the probability of θ, we can incorporate it into the equation in the form of the prior, $P(\theta)$. In This case, Bayes' laws has it's original form.
  We then find the posterior by taking into account the likelihood and our prior belief about . Hence "Maximum A Posterior".

ML *VS* MAP estimate *example*

Let's say you have an apple, and you want to know its weight. Unfortunately, all you have is a broken scale.

ML *VS* MAP estimate *example*

(a)

- For the sake of this example, lets say you know the scale returns the weight of the object with an error of +/- a standard deviation of $10g$. We can describe this mathematically as:

$$measurement = weight + error \quad (6)$$

$$p(x; \mu) = \mathcal{N}(\mu, 10^2) \quad (7)$$

- Let's also say we can weigh the apple as many times as we want, so we'll weigh it 100 times.
- Notice that here the 'weight' is the 'parameter' μ that we are going to estimate.
- The 'measurement' corresponds to the data 'x'.

code

Task 1: generage some measurement samples

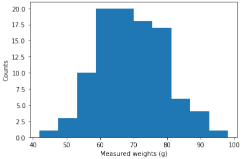
which follows the $\mathcal{N}(70, 10)$

```
# generate evenly distributed samples that follow Normal distibution with defined mean and variance
mu, sigma = 70, 10 # mean and standard deviation
samples = np.random.normal(mu, sigma, 1000)
```

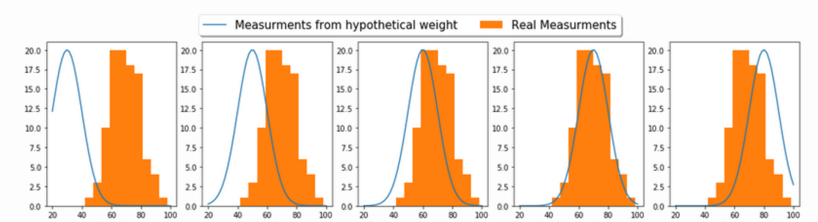
```
# randomly choose 100 samples
# TODO
measurements = random.sample(samples.tolist(),100)
```

ML *VS* MAP estimate *example*

We can look at our measurements by plotting them with a histogram



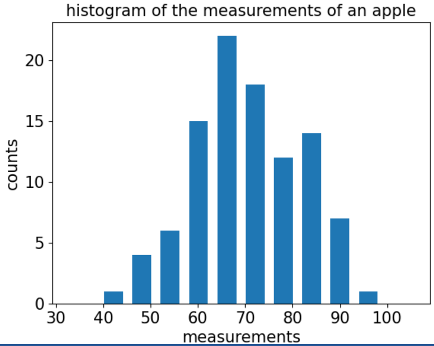
An intuitive way to show how to find the value of the 'weight' that can fit the data best.



code

```
# plot histogram
hist,bin_edges = np.histogram(measurements)
binWid=(bin_edges[1]-bin_edges[0])/2
```

```
plt.figure()
plt.bar(bin_edges[:-1]+binWid, hist, width = 4)
plt.xlim(min(bin_edges)-10, max(bin_edges)+10)
plt.xlabel('measurements',fontsize=15)
plt.ylabel('counts',fontsize=15)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.ylabel('counts',fontsize=15)
plt.title('histogram of the measurements of an apple',fontsize=15)
plt.show()
```



code

MLE

Task 2: define the likelihood function

Our goal is to find the maximum likelihood estimate of μ.

For random variable x, the pdf is

$p(x;\mu) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(\frac{(x-\mu)^2}{2\sigma^2})$

The likelihood

$\mathcal{L}(X;\mu) = \prod_{i=1}^{N} p(x_i;\mu)$ (N=100)

log likelihood function: $l(\mu) = \ln \mathcal{L}(X;\mu) = -\frac{N}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{k=1}^{N}(x_k-\mu)^2$

```
# define a pdf function of x
# TODO:

def fun_LL (X, mu ,sigma=10, N=100):
    X = np.array(X)
    l = -N/2*np.log(2*math.pi*sigma**2) - 1/(2*sigma**2) * sum((X-mu)**2)
    return l
```

code

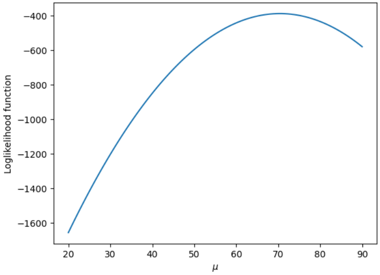
Task 3: Plot the likelihood function

```
mu = np.linspace(20,90, 100)
X = measurements

value_LL = [fun_LL(X, mu_i) for mu_i in mu] # this is nested list
value_LL = np.array(value_LL)

plt.plot(mu, value_LL)
plt.xlabel('$\mu$')
plt.ylabel('Loglikelihood function')
```

```
Text(0, 0.5, 'Loglikelihood function')
```




```
# TODO
# find the postion where the loglikelihood function reaches its maximum value
ind = np.where(value_LL == max(value_LL))          # hint : use np.where
print(mu[ind])
```

```
[70.2020202]
```

ML *VS* MAP estimate *example*

We also know that the ML estimation of a Gaussian is the average of the samples

$$\mu = \frac{1}{N}\sum_{i}^{N} x_i = 70.20 \tag{8}$$

$$SE = \frac{\sigma}{\sqrt{N}} = 10/\sqrt{100} = 1 \tag{9}$$

where, SE is the standard error of the samples in statistics. The weight of the apple is (70.20 +/- 1.) g

ML *VS* MAP estimate *example*

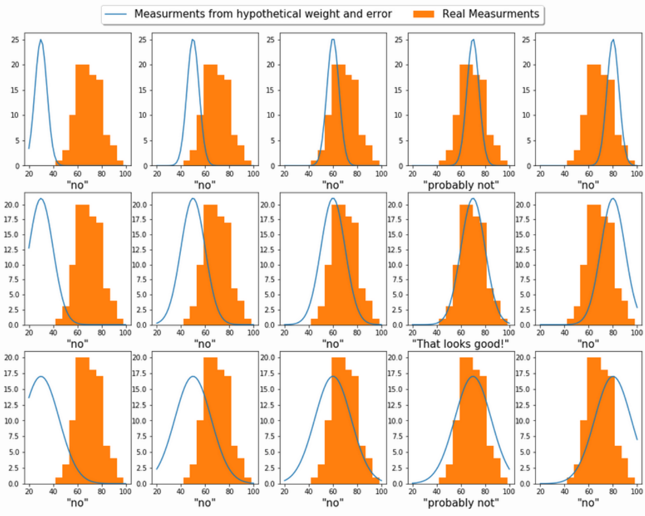
(b)
Now lets say we don't know the error of the scale. We know that its additive random normal, but we don't know what the standard deviation is

$$measurement = weight + error \tag{10}$$
$$\tag{11}$$

we want to find the mostly likely weight of the apple and the most likely error of the scale

$$P(\mu, \sigma|X) \propto P(X|\mu, \sigma) \tag{12}$$

ML *VS* MAP estimate *example*



code

Task 4: Formulate the problem (b): both μ and σ are unknown

plot the density/likelihood/posterior_probability function with respected to the paramters projected o

```
def get_log_likelihood_grid(measurments):
    log_liklelihood = [
        [
            norm(weight_guess, error_guess).logpdf(measurments).sum()
            for weight_guess in WEIGHT_GUESSES
        ]
        for error_guess in ERROR_GUESSES
    ]
    return np.asarray(log_liklelihood)
```

```
def get_mle(measurments):
    log_likelihood = get_log_likelihood_grid(measurments)
    idx_w = np.argwhere(log_likelihood == log_likelihood.max())[0][1]
    idx_e = np.argwhere(log_likelihood == log_likelihood.max())[0][0]
    return WEIGHT_GUESSES[idx_w], ERROR_GUESSES[idx_e]
```

```
WEIGHT_GUESSES = np.linspace(20, 90, 100)
ERROR_GUESSES = np.linspace(5, 15, 100)
```

```
LL_grid =get_log_likelihood_grid(measurements)
L_grid=np.exp(LL_grid) #just for better visualization
```

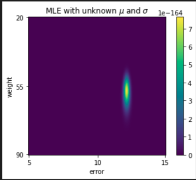
code

```
#for the plots
posX = [0, int(len(ERROR_GUESSES)/2), int(len(ERROR_GUESSES)-1)]
posY = [0, int(len(WEIGHT_GUESSES)/2), int(len(WEIGHT_GUESSES)-1)]

labels_X = ERROR_GUESSES[posX]
labels_Y = WEIGHT_GUESSES[posY]
labels_X = [int(labels_X[i]) for i in range(len(labels_X))]
labels_Y = [int(labels_Y[i]) for i in range(len(labels_Y))]

fig, ax = plt.subplots()
ax.set_xticks(posX)
ax.set_xticklabels(labels_X)
ax.set_yticks(posY)
ax.set_yticklabels(labels_Y)
ax.set_xlabel('error')
ax.set_ylabel('weight')
ax.set_title('MLE with unknown $\mu$ and $\sigma$')
plt.imshow(L_grid)
plt.colorbar()
```

```
<matplotlib.colorbar.Colorbar at 0x7f4a194718e0>
```



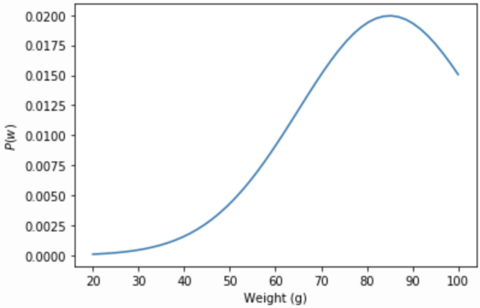

```
# print(f"Maximum Likelihood estimate: {get_mle(measurements):.3f} g")
print(f"Maximum Likelihood estimate: {get_mle(measurements):} ")
```

```
Maximum Likelihood estimate: (70.20202020202021, 10.353535353535353)
```

ML *VS* MAP estimate *example*

(c) We have prior on the weight:

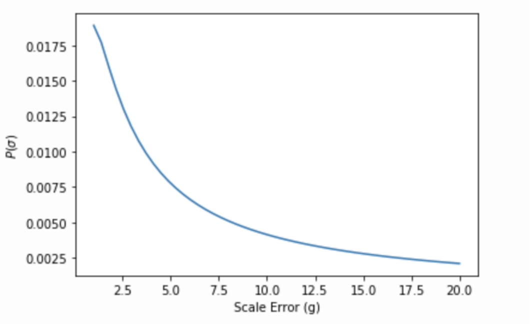
$$P(\mu) = \mathcal{N}(85, 40) \tag{13}$$



ML *VS* MAP estimate *example*

We have prior on the error:

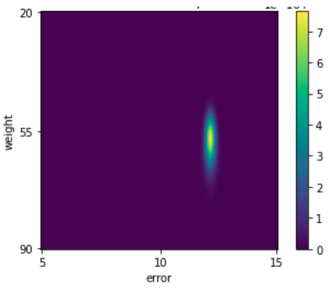
$$P(\sigma) = Inv[Gamma(.05)] \tag{14}$$



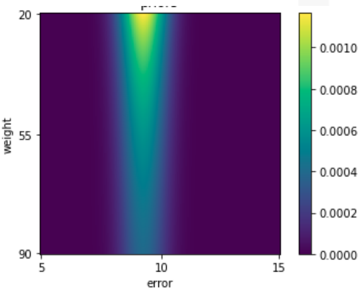
ML *VS* MAP estimate *example*

$$P(\mu,\sigma|X) \propto P(X|\mu,\sigma)P(\mu,\sigma) \tag{15}$$

$$P(\mu,\sigma) = P(\mu)P(\sigma) \tag{16}$$



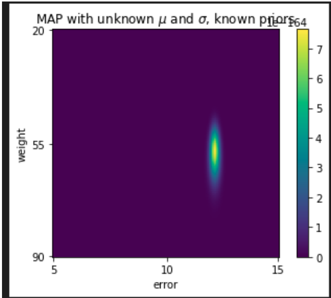
(a)



(b)

ML *VS* MAP estimate *example*

The weight of the apple is (69.49 +/- 1.35) g
(you may get a different value or figure in the exercise)




```
print(f"Maximum A Posterior estimate: {get_map(measurements)} ")
```

```
Maximum A Posterior estimate: (69.4949494949495, 10.353535353535353)
```

Appendix : *not mandatory*

Conditional probability *VS* Likelihood *VS* Likelihood function

- Likelihood not a probability, but is **proportional to a probability**.
- The likelihood of a hypothesis (H) given some data (D) is proportional to the probability of obtaining D given that H is true, multiplied by an arbitrary positive constant (K).
  In other words, $L(H|D) = K \times P(D|H)$.
  - **L(H|D):** likelihood
  - **P(D|H):** conditional probability
  - **p(D;H) or L(D;H) or L(D):** (likelihood) function p with respect to D. In other words, D is the argument of function p. H is the pameter of p.
- Since a likelihood isn't actually a probability it doesn't obey various rules of probability. For example, likelihood need not sum to 1.

https://alexanderetz.com/2015/04/15/understanding-bayes-a-look-at-the-likelihood/

Conditional probability *VS* Likelihood in Bayes' theorem

Assume θ is continuous variable, X is the data, i.e., the observations. The Bayes' theorem can be written in two ways:

1

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int_\theta p(X|\theta)p(\theta)} \tag{17}$$

Where $p(X|\theta)$ is the conditional probability of X given θ, $p(\theta|X)$ is the posterior, and the $p(\theta)$ is the prior.

2

$$p(\theta|X) = \frac{L(\theta|X)p(\theta)}{\int_\theta L(\theta|X)p(\theta)} \tag{18}$$

Where $L(\theta|X)$ is the likelihood. $p(\theta|X)$ is the posterior, and the $p(\theta)$ is the prior.

They are equivalent due to,

$$p(X|\theta) \propto L(\theta|X) \tag{19}$$

https://stats.stackexchange.com/questions/37406/likelihood-vs-conditional-distribution-for-bayesian-analy

Thank You !

$\mathscr{Q}$ & $\mathscr{A}$