

Lecture 11 - Maximum Likelihood Estimation (MLE) & Maximum A Posteriori (MAP)

```
In [1]: import numpy as np
import scipy.stats as stats

import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('bmh')
```

Last class, we introduced the **Bayesian interpretation** of a supervised learning algorithm, and formulated two approaches for search for parameters of the model:

1. **Maximum Likelihood Estimation (MLE)**
2. **Maximum A Posteriori (MAP)**

In our problem, the hypothesis are the *unknown* **(hyper-)parameters** \mathbf{w} .

- In Bayesian statistics (MAP), we are then trying to find the \mathbf{w} 's that maximizing the posterior probability.
- In classical statistics (MLE), on the other hand, we are only computing the probability of some hypothesis (the *null hypothesis*).

Maximum Likelihood Estimation (MLE)

(Frequentist approach)

$$\arg_{\mathbf{w}} \max P(\mathbf{t}|\mathbf{w})$$

In **Maximum Likelihood Estimation** (also referred to as **MLE** or **ML**) we want to *find the set of parameters* that **maximize** the data likelihood $P(\mathbf{t}|\mathbf{w})$. We want to find the *optimal* set of parameters under some assumed distribution such that the data is most likely.

Maximum A Posteriori (MAP)

(Bayesian approach)

$$\begin{aligned} \arg_{\mathbf{w}} \max P(\mathbf{t}|\mathbf{w})P(\mathbf{w}) \\ \propto \arg_{\mathbf{w}} \max P(\mathbf{w}|\mathbf{t}) \end{aligned} \tag{1}$$

In **Maximum A Posteriori** (also referred as **MAP**) we want to *find the set of parameters* that **maximize** the posteriori probability $P(\mathbf{w}|\mathbf{t})$. We want to find the *optimal* set of parameters under some assumed distribution such that the parameters are most likely to have been drawn off of given some prior beliefs.

```
In [ ]: x = np.linspace(-4,4,1000)
Gaussian = np.exp(-x**2/2)/np.sqrt(2*np.pi) #Gaussian with zero-mean and unit-variance
Laplacian = np.exp(-np.abs(x))/(2) #Laplacian with zero-mean and Lambda=1

plt.figure(figsize=(8,5))
plt.plot(x, Gaussian, '--', label='Gaussian, $G(0,1)$')
plt.plot(x, Laplacian, label='Laplacian, $L(0,1)$')
plt.legend(loc='best', fontsize=15)
plt.xlabel('$w$', size=15)
plt.ylabel('Prior Probability\n$P(w)$', size=15);
```

For the Least Squares objective function in our regression problem, we showed that:

- If we do not use a regularization term, then we are performing MLE. We are maximizing the data likelihood which takes the form of a Gaussian distribution.
- If we use a regularization term, then we are performing MAP. We are maximizing the posterior distribution, which is equivalent at maximizing the data likelihood times the prior probability on the parameters.
 - If we use *ridge regularizer*, both data likelihood and prior distribution take the form of a Gaussian.
 - If we use *lasso regularizer*, data likelihood takes the form of a Gaussian distribution and the prior takes the form of a Laplacian distribution.

Furthermore, this Bayesian interpretation allows us to think of a problem in a different way.

- **What is the distribution form of the error values for our model?**
- **Why is it so common to assume Gaussian error distribution?** The reason why we always assume a Gaussian error distribution goes back an important result described by the Central Limit Theorem. The Central Limit Theorem says that whenever a measurement is subject to a very large number of very small errors, the probability distribution for the total error is driven toward the Gaussian distribution. This is true regardless of the form of the original probability distributions of the individual errors.

Example

Problem: Suppose I flip a coin 3 times and observe the event H-H-H. What is the probability of flipping Heads (H) on the next coin flip?

Now, let's consider heads=1 and tails=0, so our sample space is $S = \{1, 0\}$. The probability of heads is equal to some *unknown* value μ , then:

$$P(x = 1|\mu) = \mu \quad (3)$$

$$P(x = 0|\mu) = 1 - \mu \quad (4)$$

We can compute the data likelihood as:

$$P(x|\mu) = \mu^x(1 - \mu)^{1-x} = \begin{cases} \mu & \text{if } x = 1 \\ 1 - \mu & \text{if } x = 0 \end{cases}$$

- This is the **Bernoulli distribution**. The mean and variance of the Bernoulli distribution are:
 $E[x] = \mu$ and $E[(x - E[x])^2] = \mu(1 - \mu)$.
- So, for every outcome of the event E , we will model it using a Bernoulli distribution, and each outcome is pairwise **conditionally independent**. Therefore, we have the event E contains i.i.d. outcomes.

Method 1: Maximum Likelihood Estimator

For simplicity of calculation, assume that the event contains outcomes: $E = x_1 \cap x_2 \cap \dots \cap x_N$, where $x_i = \{0, 1\}$ (0 for Tails and 1 for Heads). Then, for an experiment with N samples, we can write the **data likelihood** as:

$$P(E|\mu) = P(x_1 \cap x_2 \cap \dots \cap x_N|\mu) \quad (5)$$

$$= P(x_1|\mu)P(x_2|\mu) \dots P(x_N|\mu) \quad (6)$$

$$= \prod_{n=1}^N P(x_n|\mu) \quad (7)$$

$$= \prod_{n=1}^N \mu^{x_n} (1 - \mu)^{1-x_n} \quad (8)$$

- Now, we are interested in finding the value of μ given some data set E .

We now optimize the data likelihood. What trick can we use?

$$\arg_{\mu} \max P(E|\mu) = \arg_{\mu} \max \ln(P(E|\mu))$$

because the $\ln(\bullet)$ is a monotonic function.

Where

$$\ln(P(E|\mu)) = \sum_{n=1}^N (x_n \ln(\mu) + (1 - x_n) \ln(1 - \mu))$$

So now we can take the derivative of this function wrt to μ and equal it to zero:

$$\frac{\partial \ln(P(E|\mu))}{\partial \mu} = 0$$

$$(1 - \mu) \sum_{n=1}^N x_n - \mu \left(N - \sum_{n=1}^N x_n \right) = 0 \quad (9)$$

$$\sum_{n=1}^N x_n - \mu \sum_{n=1}^N x_n - \mu N + \mu \sum_{n=1}^N x_n = 0 \quad (10)$$

$$\sum_{n=1}^N x_n - \mu N = 0 \quad (11)$$

$$\mu = \frac{1}{N} \sum_{n=1}^N x_n \quad (12)$$

So the MLE estimation of the probability of seeing heads in the next coin flip is equal to **relative frequency** of outcome heads.

- Suppose you flipped the coin only once, and saw Tails. The probability of flipping Heads according to MLE would be 0.
- MLE is **purely data driven**! This is sufficient *when* we have lots and lots of data.

Method 2: Maximum A Posteriori

In the MAP estimation of μ , we are instead optimizing the posterior probability:

$$\arg_{\mu} \max P(\mu|E) \quad (13)$$

$$= \arg_{\mu} \max \frac{P(E|\mu)P(\mu)}{P(E)} \quad (14)$$

$$\propto \arg_{\mu} \max P(E|\mu)P(\mu), P(E) \text{ is some constant value} \quad (15)$$

We have defined the data likelihood $P(E|\mu)$, we now need to choose a **prior distribution** $P(\mu)$.

- This prior distribution will *encode* any prior knowledge we have about the hidden state of the problem, in this case, the type of coin that was used.

Let's say our **prior distribution** is a Beta Distribution. A **Beta Distribution** takes the form:

$$\text{Beta}(X|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1 - x)^{\beta-1}$$

where $\Gamma(x) = (x - 1)!$ and $\alpha, \beta > 0$.

The mean and variance of the Beta distribution are: $E[x] = \frac{\alpha}{\alpha + \beta}$ and

$$E[(x - E[x])^2] = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}.$$

- Let's see what that looks like:

In [19]:

```
import math

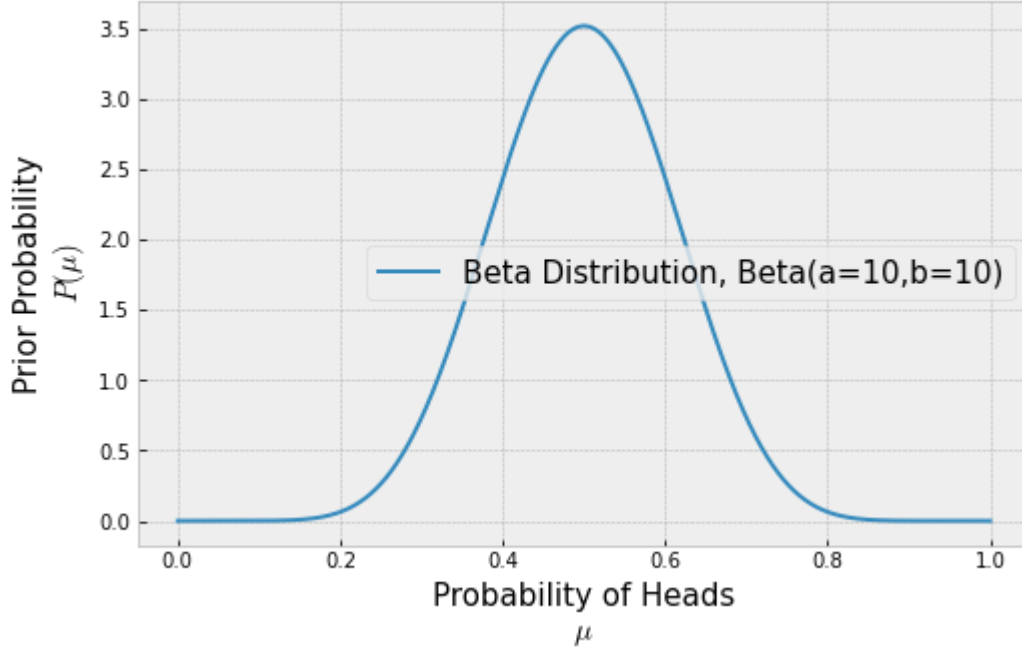
a = 10
b = 10
```

```

x = np.arange(0,1,0.0001)
Beta = (math.gamma(a+b)/(math.gamma(a)*math.gamma(b)))*x**(a-1)*(1-x)**(b-1)

plt.figure(figsize=(8,5))
plt.plot(x, Beta, label='Beta Distribution, Beta(a='+str(a)+' ,b='+str(b)+' )')
plt.legend(loc='best',fontsize=15)
plt.xlabel('Probability of Heads\n$\mu$',fontsize=15)
plt.ylabel('Prior Probability\n$P(\mu)$',fontsize=15);

```



Using the Beta Distribution as our prior, we have:

$$P(\mu|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \mu^{\alpha-1} (1 - \mu)^{\beta-1} \quad (16)$$

$$\propto \mu^{\alpha-1} (1 - \mu)^{\beta-1} \quad (17)$$

Let:

- m the number of heads
- l the number of tails
- $N = m + l$ the total number of coin flips

We can write our **posterior probability** as:

$$P(\mu|E) = \frac{P(E|\mu)P(\mu)}{P(E)} \quad (18)$$

$$\propto P(E|\mu)P(\mu) \quad (19)$$

$$= \left(\prod_{n=1}^N \mu^{x_n} (1 - \mu)^{1-x_n} \right) \mu^{\alpha-1} (1 - \mu)^{\beta-1} \quad (20)$$

$$= \mu^m (1 - \mu)^l \mu^{\alpha-1} (1 - \mu)^{\beta-1} \quad (21)$$

$$= \mu^{m+\alpha-1} (1 - \mu)^{l+\beta-1} \quad (22)$$

- The posterior probability has the same shape as the data likelihood.

- This is a special case called **Conjugate Prior Relationship**, which happens when the posterior has the same form as the prior.

We can now optimize our posterior probability, and we will apply the same trick:

$$\arg_{\mu} \max P(\mu|E) = \arg_{\mu} \max \ln(P(\mu|E))$$

where

$$\ln(P(\mu|E)) = (m + \alpha - 1) \ln(\mu) + (l + \beta - 1) \ln(1 - \mu)$$

We can now *optimize* our posterior probability:

$$\frac{\partial \ln(P(\mu|E))}{\partial \mu} = 0 \quad (23)$$

$$\frac{m + \alpha - 1}{\mu} + \frac{l + \beta - 1}{1 - \mu} = 0 \quad (24)$$

$$\mu = \frac{m + \alpha - 1}{m + l + \alpha + \beta - 2} \quad (25)$$

This is our estimation of the probability of heads using MAP!

- Our estimation for the probability of heads, μ , is going to depend on α and β introduced by the prior distribution. We saw that they control the level of certainty as well as the center value.
- With only a few samples, the prior will play a bigger role in the decision, but eventually the data takes over the prior.

Let's run a simulation to compare MAP and MLE estimators.

```
In [12]: trueMU = 0.5 # 0.5 for a fair coin
Nflips = 10
a = 10
b = 10

Outcomes = []
for i in range(Nflips):
    Outcomes += [stats.bernoulli(trueMU).rvs(1)[0]]
    print(Outcomes)
print('MLE aka Frequentist Probability of Heads = ', np.sum(Outcomes)/len(Outcomes))
print('MAP aka Bayesian Probability of Heads = ', (np.sum(Outcomes)+a-1)/(len(Outco
input('Press enter to flip the coin again...\n'))
```

```
[1]
MLE aka Frequentist Probability of Heads =  1.0
MAP aka Bayesian Probability of Heads =  0.5263157894736842
Press enter to flip the coin again...
```

```
[1, 1]
MLE aka Frequentist Probability of Heads =  1.0
MAP aka Bayesian Probability of Heads =  0.55
Press enter to flip the coin again...
```

```
[1, 1, 0]
```

```
MLE aka Frequentist Probability of Heads = 0.6666666666666666
MAP aka Bayesian Probability of Heads = 0.5238095238095238
Press enter to flip the coin again...
```

```
[1, 1, 0, 0]
MLE aka Frequentist Probability of Heads = 0.5
MAP aka Bayesian Probability of Heads = 0.5
Press enter to flip the coin again...
```

```
[1, 1, 0, 0, 0]
MLE aka Frequentist Probability of Heads = 0.4
MAP aka Bayesian Probability of Heads = 0.4782608695652174
Press enter to flip the coin again...
```

```
[1, 1, 0, 0, 0, 0]
MLE aka Frequentist Probability of Heads = 0.3333333333333333
MAP aka Bayesian Probability of Heads = 0.4583333333333333
Press enter to flip the coin again...
```

```
[1, 1, 0, 0, 0, 0, 0]
MLE aka Frequentist Probability of Heads = 0.2857142857142857
MAP aka Bayesian Probability of Heads = 0.44
Press enter to flip the coin again...
```

```
[1, 1, 0, 0, 0, 0, 0, 0]
MLE aka Frequentist Probability of Heads = 0.25
MAP aka Bayesian Probability of Heads = 0.4230769230769231
Press enter to flip the coin again...
```

```
[1, 1, 0, 0, 0, 0, 0, 0, 1]
MLE aka Frequentist Probability of Heads = 0.3333333333333333
MAP aka Bayesian Probability of Heads = 0.4444444444444444
Press enter to flip the coin again...
```

```
[1, 1, 0, 0, 0, 0, 0, 0, 1, 1]
MLE aka Frequentist Probability of Heads = 0.4
MAP aka Bayesian Probability of Heads = 0.4642857142857143
Press enter to flip the coin again...
```

In []:

Maximum Likelihood Estimation (MLE)

(Frequentist approach)

$$\arg_{\mathbf{w}} \max P(\mathbf{t}|\mathbf{w})$$

In **Maximum Likelihood Estimation** we find the set of parameters that **maximize** the data likelihood $P(\mathbf{t}|\mathbf{w})$. We find the *optimal* set of parameters under some assumed distribution such that the data is most likely.

- MLE focuses on maximizing the data likelihood, which *usually* provides a pretty good estimate
- A common trick to maximize the data likelihood is to maximize the log likelihood

- MLE is purely data driven
- MLE works best when we have lots and lots of data
- MLE will likely overfit when we have small amounts of data or, at least, becomes unreliable
- It estimates relative frequency for our model parameters. Therefore it needs incredibly large amounts of data (infinite!) to estimate the true likelihood parameters
 - This is a problem when we want to make inferences and/or predictions outside the range of what the training data has learned

Maximum A Posteriori (MAP)

(Bayesian approach)

$$\arg_{\mathbf{w}} \max P(\mathbf{t}|\mathbf{w})P(\mathbf{w}) \quad (26)$$

$$\propto \arg_{\mathbf{w}} \max P(\mathbf{w}|\mathbf{t}) \quad (27)$$

In **Maximum A Posteriori** we *find the set of parameters* that **maximize** the the posterior probability $P(\mathbf{w}|\mathbf{t})$. We find the *optimal* set of parameters under some assumed distribution such that the parameters are most likely to have been drawn off of.

- MAP focuses on maximizing the posterior probability - data likelihood with a prior
- A common trick to maximize the posterior probability is to maximize the log likelihood
- MAP is data driven
- MAP is mostly driven by the prior beliefs
- MAP works great with small amounts of data *if* our prior was chosen well
- We need to assume and select a distribution for our prior beliefs
 - A wrong choice of prior distribution can impact negatively our model estimation
- When we have lots and lots of data, the data likelihood will take over and the posterior will depend less and less on the prior

Conjugate Priors

- Two distributions have a **conjugate prior** relationship when the form of the posterior is the same as the form of the prior.
 - For example, consider the data likelihood $P(X|\mu) \sim \mathcal{N}(\mu, \sigma^2)$ and the prior distribution $P(\mu) \sim \mathcal{N}(\mu_0, \sigma_0^2)$. The posterior probability will also be Gaussian distributed

$$P(\mu|X) \sim \mathcal{N} \left(\frac{\sum_{i=1}^N x_i \sigma_0^2 + \mu_0 \sigma^2}{N \sigma_0^2 + \sigma^2}, \left(\frac{N}{\sigma^2} + \frac{1}{\sigma_0^2} \right)^{-1} \right)$$

- There are many conjugate prior relationships, e.g., Bernoulli-Beta, Gaussian-Gaussian, Gaussian-Inverse Wishart, Multinomial-Dirichlet.
- Conjugate prior relationships play an important role for online **updating** our prior distribution.
- In an online model estimation scenario, where we the posterior has the same form as the prior, we can use the posterior as our new prior. This new prior is now data informative and will update it's parameters based on (1) our initial choice, and (2) the data.