hw6

March 19, 2024

Name: Kefan Zheng

StudentId: 9086175008

Email: kzheng58@wisc.edu

1 Problem 6.1

$$\hat{p^*}_{ML} = \arg\max_p \ \mathbb{P}(X \mid p) = \arg\max_p \ p^{1^T X}(1-p)^{N-1^T X}$$

Take log:

$$\hat{p^*}_{ML} = \arg\max_p \ \log(\mathbb{P}(X \mid p)) = \arg\max_p \ 1^T X \log(p) + (N - 1^T X)\log(1-p)$$

Take the derivative:

$$\frac{\partial \log(\mathbb{P}(X \mid p))}{\partial p} = \frac{1^T X}{p} - \frac{N - 1^T X}{1-p}$$

Set to zero:

$$\frac{\partial \log(\mathbb{P}(X \mid p))}{\partial p} = 0 \tag{1}$$

$$\frac{1^T X}{p} - \frac{N - 1^T X}{1-p} = 0 \tag{2}$$

$$\frac{1^T X(1-p) - p(N - 1^T X))}{p(1-p)} = 0 \tag{3}$$

$$\frac{1^T X - pN}{p(1-p)} = 0 \tag{4}$$

$$p = \frac{1^T X}{N} \tag{5}$$

So

$$\hat{p^*}_{ML} = \frac{1^T X}{N}$$

2 Problem 6.2

$$\hat{p^*}_{MAP} := \arg\max_p \ \ \mathbb{P}(p \mid X)$$

Rewrite using Bayes Rule:

$$\hat{p^*}_{MAP} = \arg\max_p \ \ \frac{\mathbb{P}(X \mid p)\mathbb{P}(p)}{\mathbb{P}(X)} = \arg\max_p \ \ \mathbb{P}(X \mid p)\mathbb{P}(p)$$

Model the prior of $\mathbb{P}(p)$ as $Beta(\alpha, \beta)$ with parameters $\alpha > \beta > 1$:

$$\mathbb{P}(p) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}$$

So

$$\begin{aligned}
\hat{p^*}_{MAP} &= \arg\max_p \ \ \mathbb{P}(p \mid X) && (6) \\
&= \arg\max_p \ \ \mathbb{P}(X \mid p)\mathbb{P}(p) && (7) \\
&= \arg\max_p \ \ \left(p^{1^T X}(1-p)^{N-1^T X}\right)\left(\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}\right) && (8) \\
& && (9)
\end{aligned}$$

Take log:

$$\begin{aligned}
\hat{p^*}_{MAP} &= \arg\max_p \ \ \log\left(\mathbb{P}(X \mid p)\mathbb{P}(p)\right) && (10) \\
&= \arg\max_p \ \ \log\left(\left(p^{1^T X}(1-p)^{N-1^T X}\right)\left(\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}\right)\right) && (11) \\
&= \arg\max_p \ \ \log\left(p^{1^T X}(1-p)^{N-1^T X}\right) + \log\left(\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}\right) && (12) \\
&= \arg\max_p \ \ 1^T X \log(p) + (N - 1^T X)\log(1-p) + \log(\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}) + (\alpha-1)\log(p) + (\beta-1)\log(1-p) && (13)
\end{aligned}$$

Take the derivative:

$$\begin{aligned}
\frac{\partial \log(\mathbb{P}(X \mid p)\mathbb{P}(p))}{\partial p} &= \frac{1^T X}{p} - \frac{N - 1^T X}{1-p} + 0 + \frac{\alpha-1}{p} - \frac{\beta-1}{1-p} && (14) \\
&= \frac{1^T X + \alpha - 1}{p} - \frac{N - 1^T X + \beta - 1}{1-p} && (15) \\
&= \frac{1^T X + \alpha - 1 - p(\alpha + N + \beta - 2)}{p(1-p)} && (16)
\end{aligned}$$

Set to zero:

$$\frac{\partial \log(\mathbb{P}(X \mid p)\mathbb{P}(p))}{\partial p} = 0 \tag{17}$$

$$\frac{1^T X + \alpha - 1 - p(\alpha + N + \beta - 2)}{p(1-p)} = 0 \tag{18}$$

$$p(\alpha + N + \beta - 2) = 1^T X + \alpha - 1 \tag{19}$$

$$p = \frac{1^T X + \alpha - 1}{\alpha + N + \beta - 2} \tag{20}$$

So

$$\hat{p^*}_{MAP} = \frac{1^T X + \alpha - 1}{\alpha + N + \beta - 2}$$

3 Problem 6.3 (a)

[1]:
```
import numpy as np

def mle_estimate(X):
    N = len(X)
    ones_vector = np.ones(N)
    return (ones_vector @ X) / N

def map_estimate(X, alpha, beta):
    N = len(X)
    ones_vector = np.ones(N)
    return (ones_vector @ X + alpha -1) / (N + beta + alpha - 2)

p_star = 0.99
p_hat = float("inf")
N = 0
while abs(p_star - p_hat) >= 0.01:
    N += 1
    X = np.random.binomial(1, p_star, N)
    p_hat = mle_estimate(X)

print("Number of samples required: ", N)
```

```
Number of samples required:  52
```

4 Problem 6.3 (b)

[2]:
```
alpha = 7
beta = 2
p_star = 0.99
p_hat = float("inf")
N = 0
```

```
while abs(p_star - p_hat) >= 0.01:
    N += 1
    X = np.random.binomial(1, p_star, N)
    p_hat = map_estimate(X, alpha, beta)

print("Number of samples required: ", N)
```

```
Number of samples required:  44
```

5 Problem 6.4 (a)

[3]:
```
p_star = 0.01
p_hat = float("inf")
N = 0
while abs(p_star - p_hat) >= 0.01:
    N += 1
    X = np.random.binomial(1, p_star, N)
    p_hat = mle_estimate(X)

print("Number of samples required: ", N)
```

```
Number of samples required:  55
```

6 Problem 6.4 (b)

[4]:
```
alpha = 7
beta = 2
p_star = 0.01
p_hat = float("inf")
N = 0
while abs(p_star - p_hat) >= 0.01:
    N += 1
    X = np.random.binomial(1, p_star, N)
    p_hat = map_estimate(X, alpha, beta)

print("Number of samples required: ", N)
```

```
Number of samples required:  315
```

7 Problem 6.5

It depends on the specific scenario! When the prior information is certain, I prefer **MAP**, because MAP is more accurate than MLE when the number of samples is the same; And when there is a lack of prior information, I prefer **MLE** because MLE only focuses on the observed data and is applicable to most cases.
1. **MLE Advantages:** 1. Solid math foundation and easy to understand. 2. Wider application scope because it only focuses on observation samples and prior information is not necessary. 2.

MLE Disadvantages: 1. Prior information is not used if it exists. 2. Easy to overfit when the number of samples is small.
3. **MAP Advantages:** 1. Prior information is used to make estimation more accurate. 2. Performance is better on small sample data with prior information. 3. Solid math foundation
4. **MAP Disadvantages:** 1. Not all situations have prior information. 2. Improper prior information can mislead results.

[ ]:
```

```