CSE 6363 - HW01

Bardia Mojra

February 20, 2021

Exercise 1

MLE and MAP
Poisson distribution (PMF):

$$P_\lambda(k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

Part a: MLE of Poisson distribution

$$P_{MLE}(\lambda \mid k_1, k_2, ..., k_n) = \prod_{i=1}^{n} f_K(k_i \mid \lambda)\,; \quad where \;\; f_K = P_\lambda(k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

$$\implies P_{MLE}(\lambda \mid k_1, k_2, ..., k_n) = \prod_{i=1}^{n} \frac{(e^{-\lambda}\ \lambda^{k_i})}{k_i!}$$

For optimization, we use the log function and we end up with log-likelihood:

$$P_{MLE}(\lambda \mid k_1, k_2, ..., k_n) = log(\prod_{i=1}^{n} \frac{(e^{-\lambda}\ \lambda^{k_i})}{k_i!})$$

$$\implies \sum_{i=1}^{n} ln((e^{-\lambda})(\frac{1}{k_i!})(\lambda^{k_i})) \implies \sum_{i=1}^{n}[(-\lambda) - ln(k_i!) + k_i ln(\lambda)]$$

$$\implies ln\ \mathcal{L} = P_{MLE}(\lambda \mid k_1, k_2, ..., k_n) = -n\lambda - \sum_{i=1}^{n}[ln(k_i!)] + ln(\lambda)\sum_{i=1}^{n} k_i$$

Thus, one can generalize the following case:

$$\frac{\partial ln\mathcal{L}}{\partial \hat{\lambda}} = -n + \frac{\sum_{i=0}^{n} k_i}{\hat{\lambda}} \implies -n + \frac{\sum_{i=0}^{n} k_i}{\hat{\lambda}} = 0 \implies \hat{\lambda_n} = \frac{1}{n}\sum_{i=1}^{n} k_i$$

And since λ is equal to the expected value (mean) and variance we can estimate λ as:

$$\hat{\lambda_n} = \frac{1}{n}\sum_{i=1}^{n} k_i = E[K] = Var[K]$$

Part b: Calculate $P_{MLE}(\lambda \mid D)$, where $D = \{2, 5, 0, 3, 1, 3\}$:

$$P_{MLE}(\lambda \mid D) = \hat{\lambda_n} = \frac{1}{n}\sum_{i=1}^{n} k_i \Longrightarrow P_{MLE}(\lambda \mid D) = \frac{2+5+0+3+1+3}{6} = \frac{14}{6}$$

$$\Longrightarrow P_{MLE}(\lambda \mid D) = \hat{\lambda} = 2.33$$

Part c: Derive an optimization for a MAP using conjugate prior and the Gamma distribution. The Gamma distribution is given as:

$$P_{\alpha,\beta}(\lambda) = \frac{\beta^{\alpha}}{\Gamma(\alpha)}\lambda^{\alpha-1}\ e^{-\beta\lambda}\ ;\quad where\ \ \alpha = 2,\ \ \beta = 1$$

For this part, we use the Bayesian theorem to calculate the posterior distribution considering the prior density of λ, $P(\lambda)$, and likelihood of data being a match for a given expected value, $P(D \mid \lambda)$.

$$P(\lambda \mid D) \propto P(\lambda)P(D \mid \lambda)$$

Derive likelihood for Poisson distribution and prepare for Bayes' equation:

$$P(D \mid \lambda) = \prod_{i=1}^{n} \frac{e^{\lambda}\lambda^{k_i}}{k_i!} = \frac{\prod_{i=1}^{n} e^{\lambda} \prod_{i=1}^{n} \lambda^{k_i}}{\prod_{i=1}^{n} k_i!} = \frac{e^{n\lambda}\lambda^{\sum_{i=1}^{n} k_i}}{\prod_{i=1}^{n} k_i!}$$

And we are given:

$$P(\lambda \mid \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)}\lambda^{\alpha-1}\ e^{-\beta\lambda}\ ;\quad where\ \ \alpha = 2,\ \ \beta = 1,\ \ \lambda > 0,$$

So we can substitute and also make n negative for $e^{n\lambda}$:

$$P(\lambda \mid D) \propto (\frac{\beta^{\alpha}}{\Gamma(\alpha)}\lambda^{\alpha-1}\ e^{-\beta\lambda})(\frac{e^{-n\lambda}\lambda^{\sum_{i=1}^{n} k_i}}{\prod_{i=1}^{n} k_i!})$$

$$\Longrightarrow P(\lambda \mid D) \propto (\frac{\beta^{\alpha}}{\Gamma(\alpha)\ \prod_{i=1}^{n} k_i!})(\lambda^{\alpha-1+\sum_{i=1}^{n} k_i}\ e^{-\lambda(n+\beta)})\ ;\quad where\ \ \alpha = 2,\ \ \beta = 1,\ \ k_i \in D$$

With the last move, now we have the first parenthesis remain constant for a given data and distribution ($D\ and\ \lambda$) so proportional probability would be the second parenthesis.

So for posterior distribution, we would have, which looks like a β distribution therefore we can also define estimations for its parameters:

$$\Longrightarrow P(\lambda \mid D) \propto (\lambda^{\alpha-1+\sum_{i=1}^{n} k_i}\ e^{-\lambda(\beta+n)})$$

$$\Longrightarrow P(\lambda \mid D) \propto (\lambda^{\hat{\alpha}-1}\ .\ e^{-\lambda\hat{\beta}})\ ;\quad \hat{\alpha} = \alpha + \sum_{i=1}^{n} k_i\ ,\quad \hat{\beta} = \beta + n$$

For $\hat{\lambda} = 2.33$ and $\alpha = 2,\ \beta = 1$, we would have:

$$P_{MAP}(\lambda \mid D) \propto (\lambda^{\hat{\alpha}-1} \,.\, e^{-\lambda\hat{\beta}}) = (\lambda \mid \alpha + \sum_{i=1}^{n} k_i \,, \beta + n)\ ;\ where\ \ \sum_{i=1}^{n} k_i = 14$$

$$P_{MAP}(\lambda \mid D) \propto (2.33^{(2-1+14)} \,.\, e^{-(2.33)(1+6)}) = 2.\bar{3}^{15} \,.\, e^{-7(2.\bar{3})}$$
$$\Longrightarrow P_{MAP}(\lambda \mid D, \alpha, \beta) \approx 0.02\bar{6}$$

Exercise 2

K Nearest Neighbor
The following data provides, height, weight, age and gender.

D = {
((170, 57, 32), W),
((190, 95, 28), M),
((150, 45, 35), W),
((168, 65, 29), M),
((175, 78, 26), M),
((185, 90, 32), M),
((171, 65, 28), W),
((155, 48, 31), W),
((165, 60, 27), W),
((182, 80, 30), M),
((175, 69, 28), W),
((178, 80, 27), M),
((160, 50, 31), W),
((170, 72, 30), M),
}

a. Using Cartesian distance as the similarity measure. Show predictions for the following test data. For values of K for 1, 3, and 5. Include distance calculation, neighbor selection, and predictions.

$$D_{test} = \{(162, 53, 28), (168, 75, 32), (175, 70, 30), (180, 85, 29)\}$$

For distance calculation, neighbor selection for k of 1, 3, and 5 please refer to **`simple_knn.out`**.

b. For algorithm implementation please refer to **`simple_knn.py`**. The following command could be used to run the KNN algorithm on predefined data set: **`python simple_knn.py`**.

c. For distance calculation, neighbor selection for k of 1, 3, and 5 please refer to **`simple_knn.out`**.

Exercise 3

Gaussian Naive Bayes Classification
a,b,c. Using Gaussian Naive Bayes and data from problem 2 learn Gaussian distribution for each feature, i.e. for the following:

$$p(height|W), p(height|M), p(weight|W), p(weight|M), p(age|W), p(age|M)$$

For algorithm implementation please refer to **`naiveBayes.py`**. The following command could be used to run the naive Bayes algorithm on predefined data set: **`python naiveBayes.py`**.

For a detailed output, please refer to **`naiveBayes.out`**

d. Compare the two classifiers and discuss their comparative performance.
The performance varies bases on training configuration, I could say 5 nearest neighbors performed similar to Naive Bayes but for lower K's the performance decreased. Also Considering small size of the problem, there were not much room for variational solutions. KNN sorts training dataset based on a similarity score for a given test datum and takes a vote among the nearest neighbors to make a prediction. Naive Bayes, computes conditional probabilities based on statistical priori and uses the highest Bayesian score as the most likely estimate.