BIM3008-Assignment2

Junyang Deng (120090791)

November 6, 2022

1. Write the log likelihood for a multinomial sample and show equation (4.6).

$$\hat{p}_i = \frac{\sum_t x_i^t}{N}$$

Answer:
In a multinomial sample, the outcome of a random event is one of K mutually exclusive and exhaustive states, each of which has a probability of occuring p_i with $\sum_{i=1}^{K} p_i$. Let $x_1, x_2, ..., x_K$ be indicators where $x_i = 1$ if the outcome is state i and 0 otherwise.
In one experiment,

$$P(x_1, x_2, \ldots, x_K) = \prod_{i=1}^{K} p_i^{x_i}$$

We do N such independent experiments with outcomes $\mathcal{X} = \{\boldsymbol{x}^t\}_{t=1}^{N}$ where $\sum_i x_i^t = 1$ and

$$x_i^t = \begin{cases} 1 & \text{if experiment } t \text{ chooses state } i \\ 0 & \text{otherwise} \end{cases}$$

We can derive the log likelihood function as follows,

$$\begin{aligned} L(x_1, ..., x_N; p) &= \log \prod_{i=1}^{K} \prod_{t=1}^{N} p_i^{x_i{}^t} \\ &= \sum_{i}^{K} \sum_{t}^{N} x_i^t \log p_i \end{aligned}$$

We add a constraint $\sum_i p_i = 1$ as a Lagrange term into the log likelihood function and maximize the it as below.

$$\begin{aligned} L(p_i) &= \sum_i \sum_t x_i^t \log p_i + \lambda \left(1 - \sum_i p_i\right) \\ \frac{\partial L(p_i)}{\partial p_i} &= \sum_t \frac{x_i^t}{p_i} - \lambda = 0 \\ \lambda &= \sum_t \frac{x_i^t}{p_i} \Rightarrow p_i \lambda = \sum_t x_i^t \\ \sum_i p_i \lambda &= \sum_i \sum_t x_i^t \Rightarrow \lambda = \sum_t \sum_i x_i^t \\ p_i &= \frac{\sum_t x_i^t}{\sum_t \sum_i x_i^t} = \frac{\sum_t x_i^t}{N}, \text{ since } \sum_i x_i^t = 1 \end{aligned}$$

2. Write the code that generates a normal sample with given μ and σ, and the code that calculates m and s from the sample. Do the same using the Bayes' estimator assuming a prior distribution for μ.

Answer:

3. Assume a linear model and then add 0-mean Gaussian noise to generate a sample. Divide your sample into two as training and validation sets. Use linear regression using the training half. Compute error on the validation set. Do the same for polynomials of degrees 2 and 3 as well.

Answer:
The following result is computed by Python using numpy and sklearn packages. 10000 data points were generated with a linear model $y = 4x + 3$. Gaussian random noise ($N \sim (0, 10)$) is then added to y. Among these data, 60% are used for training, and 40% are used for validation. After training, errors are computed on both training and validation data by $h(x) = \sum(y - y_{pred})^2$.

	Training Error	Validation Error
Linear	96.983	99.662
Degree=2	96.975	99.891
Degree=3	96.950	100.812

We can observe from data that as degree increases, training error decreases, while validation error increases. With higher degree, the model can fit data better, but the risk of overfitting will also increase.

4. When the training set is small, the contribution of variance to error may be more than that of bias and in such a case, we may prefer a simple model even though we know that it is too simple for the task. Can you give an example?

   Answer:

5. Generate a sample from a multivariate normal density $N(\mu, \Sigma)$, calculate m and S, and compare them with μ and Σ. Check how your estimates change as the sample size changes.

   Answer:
   I generated samples that follow multivariate normal using the *numpy* package in Python. The parameters I used to generate samples are

$$\mu = [5, 2], \Sigma = \begin{bmatrix} 6 & -3 \\ -3 & 3.5 \end{bmatrix}$$

   After generation, the sample mean m is computed by $\frac{1}{n}\sum_{i=1}^{n} x_i$ using the np.mean function, the sample covariance is computed by $\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$ using the np.cov function. When $n = 10$:

$$m = [6.33294155, 1.23506303], S = \begin{bmatrix} 2.93776277 & -0.22573906 \\ -0.22573906 & 1.33529572 \end{bmatrix}$$

   When $n = 100$:

$$m = [5.5272423, 1.67562988], S = \begin{bmatrix} 6.07333567 & -3.33817829 \\ -3.33817829 & 4.14136322 \end{bmatrix}$$

   When $n = 500$:

$$m = [5.14566333, 1.93026976], S = \begin{bmatrix} 6.03793336 & -2.80178162 \\ -2.80178162 & 3.51947524 \end{bmatrix}$$

   When $n = 1000$:

$$m = [4.96310198, 2.03890537], S = \begin{bmatrix} 5.97037428 & -2.8075266 \\ -2.8075266 & 3.13549034 \end{bmatrix}$$

   When $n = 5000$:

$$m = [5.01694065, 2.00090362], S = \begin{bmatrix} 6.08991548 & -2.9627677 \\ -2.9627677 & 3.44442192 \end{bmatrix}$$

   We can see clearly that as n increases, the estimate values m, S become closer to μ and Σ.

6. Generate samples from two multivariate normal densities $N \sim (\mu_i, \Sigma_i)$, i = 1, 2, and calculate the Bayes' optimal discriminant for the four cases in table 5.1.

   Answer:

7. In figure 6.11, we see a synthetic two-dimensional data where LDA does a better job than PCA. Draw a similar dataset where PCA and LDA find the same good direction. Draw another where neither PCA nor LDA find a good direction.

   Answer: