Machine Learning - Assignment 3

Brian Pulfer

November 2019

1

1.1 a

The distribution p(x) is:

$$p(x) = \sum_{k=0}^{K} \pi_k P(x, \lambda_k) = \sum_{k=0}^{K} \pi_k e^{-\lambda_k} \frac{\lambda_k^x}{x!}$$

1.2 b

Responsabilities γ_{nk} can be written as

$$\gamma_{nk} = \frac{\pi_k P(x_n, \lambda_k)}{\sum_j \pi_j P(x_n, \lambda_j)} = \frac{\pi_k e^{-\lambda_k} \frac{\lambda_k^{x_n}}{x_n!}}{\sum_j \pi_j e^{-\lambda_j} \frac{\lambda_j^{x_n}}{x_n!}}$$

1.3 c

The expectation maximization algorithm can be run assuming a mixture of Poisson model by updating at each step the values of λ_k and the latent variables π_k.

$$\lambda_k^{new} = \frac{\sum_n \gamma_{nk} x_n}{\sum_n \gamma(z_{nk})}$$

$$\pi_k^{new} = \frac{\sum_n \gamma(z_{nk})}{\sum_k \sum_n \gamma(z_{nk})}$$

2

2.1 a

$$S = \{Urn_1, Urn_2\}$$
$$O = \{Blue, Red, Yellow\}$$
$$\pi = [0.5\ 0.5]$$
$$A = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{3}{4} & \frac{1}{4} \end{bmatrix}$$
$$B = \begin{bmatrix} \frac{5}{11} & \frac{2}{11} & \frac{4}{11} \\ \frac{3}{10} & \frac{4}{10} & \frac{3}{10} \end{bmatrix}$$

2.2 b

By using dynamic programming we can fill a T by N table where T is the number of time steps, N is the number of possible states and each entry (t,n) of the table is the probability of being in state n at time step t.

The first row of the table (when t=0) is going to be:

$$\left[\begin{array}{lcc} t & p_t(Urn_1) & p_t(Urn_2) \\ \hline 1 & \frac{1}{2}\frac{4}{11} & \frac{1}{2}\frac{3}{10} \end{array}\right]$$

Since both Urns have same probability of being picked as first urn ($\frac{1}{2}$) and we have respectively probabilities $\frac{4}{11}$ and $\frac{3}{10}$ to pick a yellow ball.

The probability of being in Urn_2 at time step 2 is now the sum of the probability of being in Urn_1 at time step 1 multiplied by the probability of passing to from Urn_1 to Urn_2 multiplied by the probability of picking a red ball from Urn_2, plus the very same thing but considering that at time step 1 we were in Urn_2.

The same calculation can be done for $p_2(Urn_1)$. This leads to the second row of our table.

$$p_2(Urn_1) = \frac{1}{2} * \frac{4}{11} * \frac{1}{2} * \frac{2}{11} + \frac{1}{2} * \frac{3}{10} * \frac{3}{4} * \frac{2}{11} = \frac{179}{4'840}$$
$$p_2(Urn_2) = \frac{1}{2} * \frac{4}{11} * \frac{1}{2} * \frac{4}{10} + \frac{1}{2} * \frac{3}{10} * \frac{1}{4} * \frac{4}{10} = \frac{113}{2'200}$$
$$\left[\begin{array}{lcc} t & p_t(Urn_1) & p_t(Urn_2) \\ \hline 1 & \frac{2}{11} & \frac{3}{20} \\ \\ 2 & \frac{179}{4'840} & \frac{113}{2'200} \end{array}\right]$$

We can compute $p_3(Urn_1)$ and $p_3(Urn_2)$ in the same manner. We finally obtain:

$$\begin{bmatrix} t & p_t(Urn_1) & p_t(Urn_2) \\ \hline 1 & \frac{2}{11} & \frac{3}{20} \\ \\ 2 & \frac{179}{4'840} & \frac{113}{2'200} \\ \\ 3 & \frac{5'519}{212'960} & \frac{9'099}{968'000} \end{bmatrix}$$

Since we now that the order of picked balls really was yellow red and blue, then this means that we can convert each value of the table such that the sum of every row is equal to one and the rapport between values is kept.

This leads to such a table:

$$\begin{bmatrix} t & p_t(Urn_1) & p_t(Urn_2) \\ \hline 1 & \frac{40}{73} & \frac{33}{73} \\ \\ 2 & \frac{895}{2'138} & \frac{1243}{2'138} \\ \\ 3 & \frac{275'950}{376'039} & \frac{100'089}{376'039} \end{bmatrix}$$

Finally, calculating the probability of transitioning from Urn_1 to Urn_2 and then back to Urn_1 (knowing that we picked yellow, red and blue balls), is simply:

$$\frac{40}{73} * \frac{1'243}{2'138} * \frac{275'950}{376'039} \approx 0.233$$

2.3 c

For this problem, we can once more use dynamic programming to compute $\delta_t(i)$, which is the maximum probability of a path of length t ending up in i-th state following a sequence of observations.

Following the recursive definitions:

$$\delta_1(i) = \pi_i * b_i(O_1)$$

$$\delta_{t+1}(j) = max_i(\delta_t(i) * a_{ij} * b_j(O_{t+1}))$$

We can fill a new table:

$$\begin{bmatrix} t & \delta_t(Urn_1) & \delta_t(Urn_2) \\ \hline 1 & \frac{1}{11} & \frac{2}{10} \\ \\ 2 & \frac{3}{55} & \frac{3}{200} \\ \\ 3 & \frac{3}{242} & \frac{9}{1100} \end{bmatrix}$$

Finally, by considering the maximum value of each row ($\frac{2}{10}$, $\frac{3}{55}$ and $\frac{3}{242}$) and the respective states, we can determine that the most probable sequence of states is: Urn_2, Urn_1 and Urn_1.

3

See the code attached.

4

4.1 a

We will assume that the model f(x)=c adapts the parameter c accordingly to the data points it has been trained with. More specifically, we'll assume that this parameter is going to be the mean of the y's of the training points.

When leaving out the first data point (-1, 0) the validation error for the model f(x)=c, where c is the mean of the training data points (0.5), is:

$$error_1 = (0 - c)^2 = 0.25$$

When leaving out the second data point (0,1) instead, the model f(x)=c becomes f(x)=0. We have that the error is:

$$error_2 = (1 - c)^2 = 1$$

And finally, when leaving out the third data point (1,0), the model's constant c=0.5. Validation error is:

$$error_3 = (0 - c)^2 = 0.25$$

The final model's validation error is the average of the 3 previously calculated errors:

$$error_{total} = \frac{0.25 + 1 + 0.25}{3} = 0.5$$

4.2 b

For the model f(x) = ax+b, we will assume that the model's parameters a and b are such that the line interpolates the 2 training points.

When leaving out the first data point (-1, 0), the validation error for the model f(x)=ax+b (a=-1 , b=1) is:

$$error_1 = (0 - (-1a + b))^2 = a^2 - 2ab + b^2 = 4$$

When leaving out the second data point (0,1) instead, we have that the model function is f(x)=0x+0 and the error is:

$$error_2 = (1 - (a0 + b))^2 = 1 - 2b + b^2 = 1$$

And finally, when leaving out the third data point (1,0), the model function is f(x)=1x+1 and validation error is:

$$error_3 = (0 - (1a + b))^2 = a^2 + 2ab + b^2 = 4$$

The final model's validation error is the average of the 3 previously calculated errors:

$$error_{total} = \frac{4 + 1 + 4}{3} = 3$$