Machine Learning
Università della Svizzera Italiana
Faculty of Informatics
November 27, 2019

Assignment 3: Mixture Models, Hidden Markov Models and Cross-Validation

Giorgia Adorni (giorgia.adorni@usi.ch)

1. Poisson distribution is given with

$$\mathcal{P}(x;\lambda) = e^{-\lambda}\frac{\lambda^x}{x!}$$

for $x = 0, 1, \ldots$ (non-negative integers) and $\lambda > 0$. Suppose you are given $\lambda_1, \ldots, \lambda_K$ and $\pi_1, \ldots, \pi_K$, $\pi_k > 0$, $\sum_k \pi_k = 1$ and the following generating process: sample $k \in \{1, \ldots, K\}$ with probability π_k and then sample x from $\mathcal{P}(x;\lambda_k)$.

(a) What is the distribution $\mathrm{p}(x)$ under this generating process?

(b) Write the expression for responsibilities γ_{nk}.

(c) Write the expression for M-step of expectation maximisation algorithm assuming mixture of Poissons model.

Solution:

(a) Given the random variables K and X such that their distributions are $\mathrm{P}(K = k) = \pi_k$ and $\mathrm{P}(X = x|K = k) = \mathcal{P}(x;\lambda_k)$, the probability of sampling a certain x is given by the sum, over all the possible distributions, of the probability of choosing a particular Poisson distribution multiplied by the probability of sampling x given that distribution:

$$\mathrm{p}(x) = \sum_k \mathrm{P}(K = k)\mathrm{P}(X = x|K = k) = \sum_k \pi_k\, \mathcal{P}(x, \lambda_k) = \sum_k \pi_k\, e^{-\lambda_k}\frac{\lambda_k^x}{x!} \tag{1}$$

(b) During the E-step, the responsibilities of each data point for each class γ_{nk}, that are the posterior probabilities, can be computed using Bayes with the following expression:

$$\gamma_{nk} = \frac{\pi_k\, \mathcal{P}(x_n, \lambda_k)}{\sum_j \pi_j\, \mathcal{P}(x_n, \lambda_j)} = \frac{\pi_k\, e^{-\lambda_k}\frac{\lambda_k^{x_n}}{x_n!}}{\sum_j \pi_j\, e^{-\lambda_j}\frac{\lambda_j^{x_n}}{x_n!}} \tag{2}$$

(c) The M-step consists in re-estimate the class parameters using new responsibilities provided by the E-step. With a Poisson distribution, it is necessary to compute only λ_k and π_k as follows:

$$\lambda_k^{\text{new}} = \frac{1}{N_k}\sum_n \gamma_{nk} x_n \tag{3}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \tag{4}$$

where $N_k = \sum_n \gamma_{nk}$, hence $\sum_k N_k = N$.

2. Your friend has two urns, labelled 1 and 2. Urn 1 contains 5 blue, 2 red and 4 yellow balls. Urn two contains 3 blue, 4 red and 3 yellow balls. She covers your eyes with a tape. Then, she chooses one urn at random with equal probability. You pick one ball from that urn, she tells you its colour and then you return the ball to the urn you picked it from (you don't know which one as your eyes are covered). Your friend switches from urn 1 to urn 2 with probability 1/2 and from urn 2 to urn 1 with probability 3/4. You pick one ball again, she tells you its colour and the process repeats.

(a) Describe the system as Hidden Markov Model. What are S, O, π, A, B?

(b) What is the probability that initial urn was urn 1, then urn 2 and urn 1 again given that you picked yellow, red and blue balls respectively. Use dynamic programming! *Hint*: use Bayes formula!

(c) What is the most probable sequence of urns given that you picked red, yellow and blue. Use Viterbi Algorithm!

Solution:

(a)

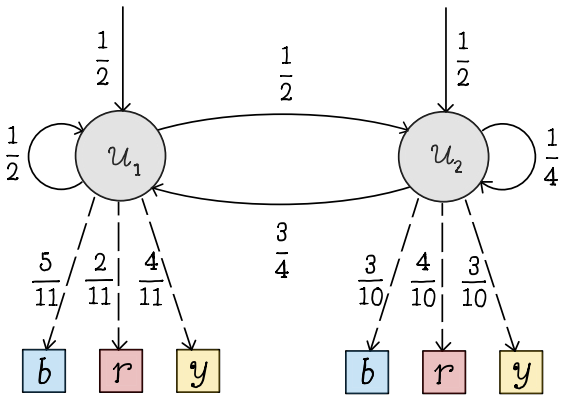

Figure 1: Representation of the described system as Hidden Markov Model.

- Set of N states $S = \{s_1, \ldots, s_N\}$: in this case, the states are the two urns, so

$$S = \{u_1, u_2\}.$$

- The starting state probabilities are contained in the vector $\pi = \{\pi_1, \ldots, \pi_N\}$. In this example, since each urn is chosen at random with equal probability:

$$\pi = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}.$$

- State transition probabilities are given in matrix A, where the probability of a transitioning from state i to state j is $a_{ij} = \mathrm{P}(q_{t+1} = s_j | q_t = s_i)$, with $i \geq 1$ and $j \leq N$. In this example, since the probability of switches from urn 1 to urn 2 is 1/2 and from urn 2 to urn 1 is 3/3, then

$$A = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{3}{4} & \frac{1}{4} \end{bmatrix}.$$

- The M observations $O = \{v_1, \dots, v_m\}$ in the space. In this example, the possible observations correspond to picking a blue, red or yellow ball

$$O = \{b, r, y\}.$$

- Observation probabilities are give in the matrix B. In particular, the probability of observing m is given by $b_i(m) = \mathrm{P}(O_t = v_m | q_t = s_i)$, with $1 \leq i \leq N$ and $1 \leq j \leq M$

$$B = \begin{bmatrix} \frac{5}{11} & \frac{2}{11} & \frac{4}{11} \\ \frac{3}{10} & \frac{4}{10} & \frac{3}{10} \end{bmatrix}.$$

(b) It is possible to compute the required probability relying on Bayes formula as follows:

$$\mathrm{p}(q_{0:2} = u_1, u_2, u_1 | O_{0:2} = y, r, b) = \frac{\mathrm{p}(q_{0:2} = u_1, u_2, u_1)\mathrm{p}(O_{0:2} = y, r, b | q_{0:2} = u_1, u_2, u_1)}{\mathrm{p}(O_{0:2} = y, r, b)}$$

The numerator of the fraction can be rewritten as:

$$\cdots = \mathrm{p}(u_1)\mathrm{p}(y|u_1)\mathrm{p}(u_2|u_1)\mathrm{p}(r|u_2)\mathrm{p}(u_1|u_2)\mathrm{p}(b|u_1) = \; = \frac{1}{2} \cdot \frac{4}{11} \cdot \frac{1}{2} \cdot \frac{4}{10} \cdot \frac{3}{4} \cdot \frac{5}{11} = \frac{3}{242}$$

The denominator of the fraction is computed applying the Forward Algorithm. In particular, it is necessary to compute α_t for each possible state and for each observation as follows:

$$\alpha_{t+1}(j) = \sum_i \alpha_t(i) \cdot a_{ij} \cdot b_j(O_{t+1})$$

with $\alpha_0(i) = \pi_i \cdot b_i(O_0)$.

At the end,

$$\mathrm{p}(O) = \sum_i \alpha_t(i).$$

Given the first observation, it is possible to compute:

$$\alpha_0(u_1) = \pi_{u_1} \cdot b_{u_1}(y) = \frac{1}{2} \cdot \frac{4}{11} = \frac{2}{11}$$
$$\alpha_0(u_2) = \pi_{u_2} \cdot b_{u_2}(y) = \frac{1}{2} \cdot \frac{3}{10} = \frac{3}{20}$$

Now the probability can be updated according to the new observation:

$$\begin{aligned}\alpha_1(u_1) &= \sum_{q_t \in S} \alpha_0(q_t) \cdot a_{q_t u_1} \cdot b_{u_1}(r) = \\ &= \left(\frac{2}{11} \cdot \frac{1}{2} \cdot \frac{2}{11}\right) + \left(\frac{3}{20} \cdot \frac{3}{4} \cdot \frac{2}{11}\right) = \\ &= \frac{2}{121} + \frac{9}{440} = \frac{179}{4840}\end{aligned}$$

$$\begin{aligned}\alpha_1(u_2) &= \sum_{q_t \in S} \alpha_0(q_t) \cdot a_{q_t u_2} \cdot b_{u_2}(r) = \\ &= \left(\frac{2}{11} \cdot \frac{1}{2} \cdot \frac{4}{10}\right) + \left(\frac{3}{20} \cdot \frac{1}{4} \cdot \frac{4}{10}\right) = \\ &= \frac{2}{5} + \frac{3}{200} = \frac{113}{2200}\end{aligned}$$

Given the last observation, the probability is updated as follows:

$$\begin{aligned}\alpha_2(u_1) &= \sum_{q_t \in S} \alpha_1(q_t) \cdot a_{q_t u_1} \cdot b_{u_1}(b) = \\ &= \left(\frac{179}{4840} \cdot \frac{1}{2} \cdot \frac{5}{11}\right) + \left(\frac{113}{2200} \cdot \frac{3}{4} \cdot \frac{5}{11}\right) = \\ &= \frac{179}{21296} + \frac{339}{19360} = \frac{5519}{212960}\end{aligned}$$

$$\begin{aligned}\alpha_2(u_2) &= \sum_{q_t \in S} \alpha_1(q_t) \cdot a_{q_t u_2} \cdot b_{u_2}(b) = \\ &= \left(\frac{179}{4840} \cdot \frac{1}{2} \cdot \frac{3}{10}\right) + \left(\frac{113}{2200} \cdot \frac{1}{4} \cdot \frac{3}{10}\right) = \\ &= \frac{537}{96800} + \frac{339}{88000} = \frac{9099}{968000}\end{aligned}$$

Now,

$$\mathrm{p}(O_{0:2} = y, r, b) = \alpha_2(u_1) + \alpha_2(u_2) = \frac{5519}{212960} + \frac{9099}{968000} = \frac{376039}{10648000}$$

Finally,

$$\mathrm{p}(q_{0:2} = u_1, u_2, u_1 | O_{0:2} = y, r, b) = \frac{3}{242} \cdot \frac{10648000}{376039} = \frac{132000}{376039} \approx 0.35$$

(c)

$$\begin{aligned}\texttt{Viterbi(r, y, b)} &= \arg\max_{q_0, q_1, q_2} \mathrm{p}(q_0, q_1, q_2 | O_0 = r, O_1 = y, O_2 = b) = \\ &\overset{bayes}{=} \arg\max_{q_0, q_1, q_2} \frac{\mathrm{p}(q_0, q_1, q_2, O_0 = r, O_1 = y, O_2 = b)}{\mathrm{p}(O_0 = r, O_1 = y, O_2 = b)} = \\ &= arg\max_{q_0, q_1, q_2} \mathrm{p}(q_0, q_1, q_2, O_0 = r, O_1 = y, O_2 = b)\end{aligned}$$

Defining recursively the following variable it is easier to compute the algorithm

$$\begin{aligned}\delta_t(i) &= \max_{q_0,\dots,q_{t-1}} \mathrm{p}(q_0,\dots,q_{t-1},q_t=s_i,O_0,\dots,O_t)\\ &\downarrow\\ \delta_0(i) &= \pi_i \cdot b_i(O_0)\\ \delta_{t+1}(j) &= \max_i(\delta_t(i)\cdot a_{ij}\cdot b_j(o_{t+1}))\end{aligned}$$

In the first step is computed the probability of being in state q_0 as follow:

$$\begin{aligned}\delta_0(u_1) &= \pi_{u_1} b_{u_1}(r) = \frac{1}{2}\cdot\frac{2}{11} = \frac{1}{11}\\ \delta_0(u_2) &= \pi_{u_2} b_{u_2}(r) = \frac{1}{2}\cdot\frac{4}{10} = \frac{1}{5}\end{aligned}$$

In the second step the probability of being in state q_1 is computed given the previous state q_0 and the new observation as follows:

$$\begin{aligned}\delta_1(u_1) &= \max\{\delta_0(u_1)\cdot a_{u_1u_1}\cdot b_{u_1}(y), \delta_0(u_2)\cdot a_{u_2u_1}\cdot b_{u_1}(y)\}\\ &= \max\{\frac{1}{11}\cdot\frac{1}{2}\cdot\frac{4}{11}, \frac{1}{5}\cdot\frac{3}{4}\cdot\frac{4}{11}\} = \max\{\frac{2}{121}, \frac{3}{55}\} = \frac{3}{55}\\ \delta_1(u_2) &= \max\{\delta_0(u_1)\cdot a_{u_1u_2}\cdot b_{u_2}(y), \delta_0(u_2)\cdot a_{u_2u_2}\cdot b_{u_2}(y)\}\\ &= \max\{\frac{1}{11}\cdot\frac{1}{2}\cdot\frac{3}{10}, \frac{1}{5}\cdot\frac{1}{4}\cdot\frac{3}{10}\} = \max\{\frac{3}{220}, \frac{3}{200}\} = \frac{3}{200}\end{aligned}$$

Finally the probability of being in state q_2 is computed given the previous most probable state and the new observation as follows:

$$\begin{aligned}\delta_2(u_1) &= \max\{\delta_1(u_1)\cdot a_{u_1u_1}\cdot b_{u_1}(b), \delta_1(u_2)\cdot a_{u_2u_1}\cdot b_{u_1}(b)\}\\ &= \max\{\frac{3}{55}\cdot\frac{1}{2}\cdot\frac{5}{11}, \frac{3}{200}\cdot\frac{3}{4}\cdot\frac{5}{11}\} = \max\{\frac{3}{242}, \frac{9}{1760}\} = \frac{3}{242}\\ \delta_2(u_2) &= \max\{\delta_1(u_1)\cdot a_{u_1u_2}\cdot b_{u_2}(b), \delta_1(u_2)\cdot a_{u_2u_2}\cdot b_{u_2}(b)\}\\ &= \max\{\frac{3}{55}\cdot\frac{1}{2}\cdot\frac{3}{10}, \frac{3}{200}\cdot\frac{1}{4}\cdot\frac{3}{10}\} = \max\{\frac{9}{1100}, \frac{9}{8000}\} = \frac{9}{1100}\end{aligned}$$

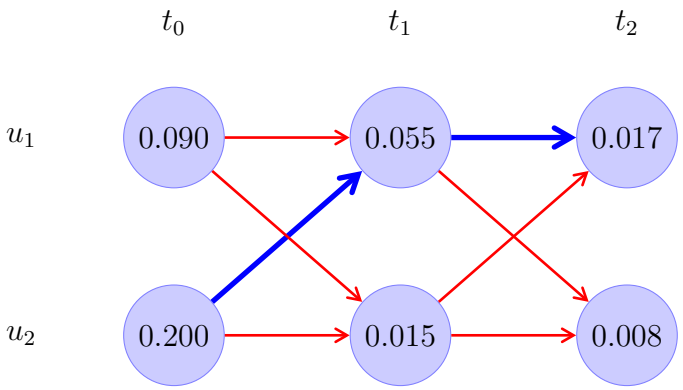

Figure 2: Trellis of the most probable path.

Hence, the most probable sequence of states is u_2, u_1, u_1.

3. Implement a Python function `state_probability(pi, A, s, t)` that takes initial state distribution π, transition matrix A, state s and time t and outputs the probability of state s at time t. You can assume that the set of state is $S = \{0, 1, \ldots, N-1\}$.
You can use file `ex3.py` provided online.
Solution:

```python
import numpy as np


def state_probability(pi, A, s, T):
    """
    :param pi:        initial state distribution
    :param A:         transition matrix
    :param s:         state
    :param T:         time
    :return prob[s]: the probability of state s at time t
    """
    prob = np.array(pi)
    for i in range(T):
        prob = np.matmul(prob, A)
    return prob[s]


pi = np.array([0.6, 0.3, 0.1])
A = np.array([[0.7, 0.2, 0.1], [0.4, 0.4, 0.2], [0.2, 0.3, 0.5]])
s = 2
T = 2

prob = state_probability(pi, A, s, T)
print(prob)
```

4. You are given three data points $(x, y) : (-1, 0), (0, 1), (1, 0)$. We are using squared error loss function $\ell(y, \hat{y}) = (y - \hat{y})^2$.

(a) What is leave-one-out cross-validation error of constant model $f(x) = c$?

(b) What is leave-one-out cross-validation error of linear model $f(x) = ax + b$?

Solution:

(a) In each step, the model is trained on $n - 1$ data-points and validated on the other. First of all, the train set loss is computed:

$$
\begin{aligned}
l_1(c_1) &= (0 - c_1)^2 + (1 - c_1)^2 = 2{c_1}^2 - 2c_1 + 1 \\
l_2(c_2) &= (0 - c_2)^2 + (0 - c_2)^2 = 2{c_2}^2 \\
l_3(c_3) &= (1 - c_3)^2 + (0 - c_3)^2 = 2{c_3}^2 - 2c_3 + 1
\end{aligned}
$$

Since we want to find the value of c that minimise the squared error, it is necessary to compute the derivative of the loss l with respect to c:

$$\frac{\partial l_1}{\partial c_1} = 4c_1 - 2 = 0 \qquad c_1 = \frac{1}{2}$$
$$\frac{\partial l_2}{\partial c_2} = 4c_2 = 0 \qquad c_2 = 0$$
$$\frac{\partial l_3}{\partial c_3} = 4c_3 - 2 = 0 \qquad c_3 = \frac{1}{2}$$

Now, it is possible to validate the model on the other data-point:

$$e_1(c_1) = \Big(0 - c_1\Big)^2 = \Big(0 - \frac{1}{2}\Big)^2 = \frac{1}{4}$$
$$e_2(c_2) = \Big(0 - c_2\Big)^2 = \Big(1 - 0\Big)^2 = 1$$
$$e_3(c_3) = \Big(0 - c_3\Big)^2 = \Big(0 - \frac{1}{2}\Big)^2 = \frac{1}{4}$$

The final error is computed as the mean of the validation errors:

$$E = \frac{e_1 + e_2 + e_3}{3} = \Big(\frac{1}{4} + 1 + \frac{1}{4}\Big)\frac{1}{3} = \frac{3}{2} \cdot \frac{1}{3} = \frac{1}{2} = 0.5$$

(b) As in the previous exercise, in each step, the model is trained on $n-1$ data-points and validated on the other. First of all, the train set loss is computed:

$$l_1(a_1x_1 + b_1) = \big(0 - (-a_1 + b_1)\big)^2 + \big(1 - b_1\big)^2 = a_1^2 - 2a_1b_1 + 2b_1^2 - 2b_1 + 1$$
$$l_2(a_2x_2 + b_2) = \big(0 - (-a_2 + b_2)\big)^2 + \big(0 - (a_2 + b_2)\big)^2 = 2a_2^2 + 2b_2^2$$
$$l_3(a_3x_3 + b_3) = \big(1 - b_1\big)^2 + \big(0 - (a_2 + b_2)\big)^2 = a_3^2 + 2a_3b_3 + 2b_3^2 - 2b_3 + 1$$

Since we want to find the values of a and b that minimise the squared error, it is necessary to compute the derivative of the loss l with respect to a and b:

$$\frac{\partial l_1}{\partial a_1} = 2a_1 - 2b_1 = 0 \qquad a_1 = b_1$$
$$\frac{\partial l_1}{\partial b_1} = -2a_1 + 4b_1 - 2 = 0 \qquad b_1 = \frac{1}{2}(a_1 + 1)$$

$$a_1 = \frac{1}{2}(a_1 + 1) \qquad \rightarrow \quad a_1 = 1$$
$$b_1 = \frac{1}{2}(1 + 1) \qquad \rightarrow \quad b_1 = 1$$

$$\frac{\partial l_2}{\partial a_2} = 4a_2 = 0 \qquad \rightarrow \quad a_2 = 0$$
$$\frac{\partial l_2}{\partial b_2} = 4b_2 = 0 \qquad \rightarrow \quad b_2 = 0$$

$$\frac{\partial l_3}{\partial a_3} = 2a_3 + 2b_3 = 0 \qquad a_3 = -b_3$$
$$\frac{\partial l_3}{\partial b_3} = 2a_3 + 4b_3 - 2 = 0 \qquad b_3 = \frac{1}{2} - \frac{a_3}{2}$$

$$a_3 = -\left(\frac{1}{2} - \frac{a_3}{2}\right) \quad \rightarrow \quad a_3 = -1$$
$$b_3 = \frac{1}{2} - \left(-\frac{1}{2}\right) \quad \rightarrow \quad b_3 = 1$$

Now, it is possible to validate the model on the other data-point:

$$a_1x_1 + b_1 = 1 \cdot 1 + 1 = 2$$
$$a_2x_2 + b_2 = 0 \cdot 0 + 0 = 0$$
$$a_3x_3 + b_3 = -1 \cdot -1 + 1 = 2$$

$$e_1(a_1x_1 + b_1) = \big(0 - (a_1x_1 + b_1)\big)^2 = (0-2)^2 = 4$$
$$e_2(a_2x_2 + b_2) = \big(1 - (a_2x_2 + b_2)\big)^2 = (1-0)^2 = 1$$
$$e_3(a_3x_3 + b_3) = \big(0 - (a_3x_3 + b_3)\big)^2 = (0-2)^2 = 4$$

The final error is computed as the mean of the validation errors:

$$E = \frac{e_1 + e_2 + e_3}{3} = \frac{4+1+4}{3} = \frac{9}{3} = 3$$