Problem 1: Teacher-mood-model

Your school teacher gave three different types of daily homework assignments:

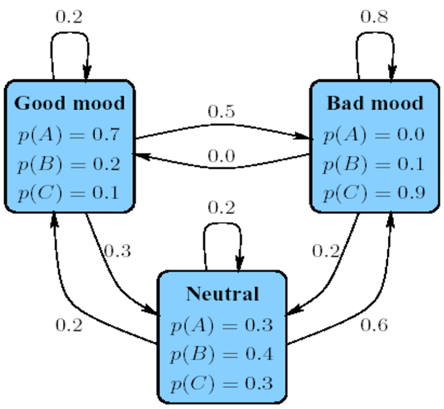
- A: took about 5 minutes to complete
- B: took about 1 hour to complete
- C: took about 3 hours to complete

Your teacher did not reveal openly his mood to you daily, but you kne that your teacher was either in a bad, neutral, or a good mood for a whole day.

Mood changes occurred only overnight.

Model parameters:

- Observation $\Sigma = \{A, B, C\}$
- Set of states $S = \{\,good, neutral, bad\}$
- Transition probabilities between any two states a_{ij}
- Emission probabilities within each state $b_i(x)$



One week, your teacher gave the following homework assignments:

Monday	Tuesday	Wednesday	Thursday	Friday
A	C	B	A	C

Questions

What did his mood curve look like most likely that week?

- Searching for the most probable path – Viterbi algorithm

Solution

Empty table

	A	C	B	A	C
good					
neutral					
bad					

Assume three states appear by the same possibility at first.

Initialization: $V_1^j = b_j(x_1)p(q_1 = j)$

$V_1^{good} = \frac{b_{good}(A)}{\#states} = 0.7/3 = 0.23$

$V_1^{neutral} = \frac{b_{neutral}(A)}{\#states} = 0.3/3 = 0.1$

$V_1^{bad} = \frac{b_{bad}(A)}{\#states} = 0/3 = 0$

	A	C	B	A	C
good	$0.2\dot{3}$				
neutral	0.1				
bad	0				

$V_2^j = b_j(C)\max(V_1^i a_{ij})$

$V_2^{good} = 0.1max(0.2\dot{3} * 0.2, 0.1 * 0.2, 0 * 0) = 0.004\dot{6}(i = 1)$

$V_2^{neutral} = 0.3max(0.2\dot{3} * 0.3, 0.1 * 0.2, 0 * 0.2) = 0.020\dot{9}(i = 1)$

$V_2^{bad} = 0.9max(0.2\dot{3} * 0.5, 0.1 * 0.6, 0 * 0.8) = 0.105(i = 1)$

	A	C	B	A	C
good	$0.2\dot{3}$	$0.004\dot{6}$			
neutral	0.1	$0.020\dot{9}$			
bad	0	0.105			

$V_3^j = b_j(B)\max(V_2^i a_{ij})$

$V_3^{good} = 0.2max(0.004\dot{6} * 0.2, 0.020\dot{9} * 0.2, 0.105 * 0) = 0.000828(i = 2)$

$V_3^{neutral} = 0.4max(0.004\dot{6} * 0.3, 0.020\dot{9} * 0.2, 0.105 * 0.2) = 0.0084(i = 3)$

$V_3^{bad} = 0.1max(0.004\dot{6} * 0.5, 0.020\dot{9} * 0.6, 0.105 * 0.8) = 0.0084(i = 3)$

	A	C	B	A	C
good	0.23	$0.004\dot{6}$	0.000828		
neutral	0.1	$0.020\dot{9}$	0.0084		
bad	0	0.105	0.0084		

$V_4^j = b_j(A)\max(V_3^i a_{ij})$

$V_4^{good} = 0.7max(0.00084 * 0.2, 0.0084 * 0.2, 0.0084 * 0) = 0.001176(i = 2)$

$V_4^{neutral} = 0.3max(0.0084 * 0.3, 0.0084 * 0.2, 0.0084 * 0.2) = 0.000504(i = 2)$

$V_4^{bad} = 0max(0.00084 * 0.5, 0.0084 * 0.6, 0.0084 * 0.8) = 0(i = 3)$

	A	C	B	A	C
good	0.23	$0.004\dot{6}$	0.000828	0.001176	
neutral	0.1	$0.020\dot{9}$	0.0084	0.000504	
bad	0	0.105	0.0084	0	

$V_5^j = b_j(C)\max(V_4^i a_{ij})$

$V_5^{good} = 0.1max(0.001176 * 0.2, 0.000504 * 0.2, 0 * 0) = 0.00002352(i = 1)$

$V_5^{neutral} = 0.3max(0.001176 * 0.3, 0.000504 * 0.2, 0 * 0.2) = 0.00010584(i = 1)$

$V_5^{bad} = 0max(0.001176 * 0.5, 0.000504 * 0.6, 0 * 0.8) = 0.0005292(i = 1)$

	A	C	B	A	C
good	0.23	$0.004\dot{6}$	0.000828	0.001176	0.00002352
neutral	0.1	$0.020\dot{9}$	0.0084	0.000504	0.00010584
bad	0	0.105	0.0084	0	0.0005292

At last,We choose the ***bad-C*** at last state. So the path is as folows(**BOLD**):

	A	C	B	A	C
good	**0.23**	$0.004\dot{6}$	0.000828	**0.001176**	0.00002352
neutral	0.1	$0.020\dot{9}$	**0.0084**	0.000504	0.00010584
bad	0	**0.105**	0.0084	0	**0.0005292**

(*It is noted that I use the programming to compute the result, and the numeric precision is higher so that the numbers are little different from the "PPT".*)

Problem 2:

In this problem you will derive the EM algorithm for a *one-dimensional* Laplacian mixture model. You are given n observations $x_1, \ldots, x_n \in \mathbb{R}$ and we want to fit a mixture of m Laplacians, which has the following density

$$f(x) = \sum_{j=1}^{m} \pi_j f_L(x; \mu_j, \beta_j),$$

where $f_L(x; \mu_j, \beta_j) = \frac{1}{2\beta_j} e^{-\frac{1}{\beta_j}|x-\mu_j|}$, and the mixture weights π_j are a convex combination, i.e. $\pi_j \geq 0$ and $\sum_{j=1}^{m} \pi_j = 1$. For simplicity, assume that the scale parameters $\beta_j > 0$ are known beforehand and thus *fixed*.

- Introduce latent variables so that we can apply the EM procedure.
- Analogously to the previous question, write down the steps of the EM procedure for this model. If some updates cannot be written analytically, give an approach on how to compute them.

  (*Hint: Recall a property of functions that makes them easy to optimize.*)

Solution

For each data point x_i , we introduce a latent variable $Y_i \in \{1, 2, \ldots, m\}$ denoting the component that point belongs to.

Let $\gamma_j(x_i) = P(y_i = j \mid x_i)$

$P(y_i = j \mid x_i) = \frac{P(x_i|y_i=j)P(y_i=j)}{\sum_{l=1}^{m} P(x_i|y_i=l)P(y_i=l)} = \frac{\pi_j f_L(x_i;\mu_j,\beta_j)}{\sum_{l=1}^{m} \pi_l f_L(x_i;\mu_l,\beta_l)}$

We want to maximum: The total probability of N data point can be expressed as the product of the probabilities of each data point, which is known as the likelihood function

$$P(x) = \prod_{i}^{N} f(x_i)$$

This can first be simplified by solving for logarithms, turning the product into a sum.

And then introducing unobserved data items that can identify the components that “generated” each data item, we can simplify the log-likelihood of for Laplacian mixtures, as follows:

$$L(x) = \ln P(x) = \sum_{i}^{n} \ln(f(x_i)) = \sum_{i}^{n} \ln(\sum_{j=1}^{m} \frac{P(x_i, y_i = j)}{P(y_i = j|x_i)} P(y_i = j|x_i))$$

According to **Jensen inequality**,Let $f(u) = \ln u$,$u(y_i|x_i) = \frac{P(x_i,y_i=j))}{P(y_i=j|x_i)}$,So $f[E(x)] \geq E(f(x))]$.

$$L(x) \geq \sum_{i}^{n} \sum_{j=1}^{m} (P(y_i = j|x_i)) \ln(\frac{P(x_i, y_i = j)}{P(y_i = j|x_i)})$$

Jensen’s Inequality: equality holds when $f(x) = \ln(\frac{P(x_i,y_i=j)}{P(y_i=j|x_i)})$ is an affine function. This is achieved for $\ln(\frac{P(x_i,y_i=j)}{P(y_i=j|x_i)}) = \ln\left(\sum_{l=1}^{m} P(x_i \mid y_i = l) P(y_i = l)\right)$

So,the equality holds.

$$L(x) = \sum_{i}^{n} \sum_{j=1}^{m} (P(y_i = j|x_i)) \ln(\frac{P(x_i, y_i = j)}{P(y_i = j|x_i)})$$

$$= \sum_{i}^{n} \sum_{j=1}^{m} (P(y_i = j|x_i) \ln P(x_i, y_i = j)) - \sum_{i}^{n} \sum_{j=1}^{m} (P(y_i = j|x_i) \ln P(y_i = j|x_i))$$

And the part to the right of the minus sign is the posterior probability, which does not require optimization.

So we want to maximum the follows:

$$
\begin{aligned}
L(x) = \sum_{i=1}^{n} \sum_{j=1}^{m} \gamma_j(x_i) \ln P(x_i, y_i = j) &= \sum_{i=1}^{n} \sum_{j=1}^{m} \gamma_j(x_i) \ln \pi_j f_L(x_i; \mu_j, \beta_j) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{m} \gamma_j(x_i) \left(\ln \pi_j - \frac{1}{\beta_j} |x_i - \mu_j| \right) + \text{ const.}
\end{aligned}
$$

- Subject to maximising the likelihood function, we first estimate the parameters μ_j

$$
\frac{\partial L(x)}{\partial \mu_j} = 0 = \sum_{i}^{N} \frac{\pi_j f_L(x_i; \mu_j, \beta_j)}{\sum_{l=1}^{m} \pi_l f_L(x_i; \mu_l, \beta_l)} \frac{1}{\beta_j} \frac{x_i - \mu_j}{|x_i - \mu_j|} = \sum_{i}^{N} \gamma_j(x_i) \frac{x_i - \mu_j}{|x_i - \mu_j|}
$$

$$
\therefore \mu_j = \frac{\sum_i^N \gamma_j(x_i) \frac{x_i}{|x_i - \mu_j|}}{\sum_i^N \gamma_j(x_i) \frac{1}{|x_i - \mu_j|}}
$$

So we can maximize $L(x)$ with respect to μ_j .we have to solve m separate optimization problems, one for each μ_j.

- β_j is known beforehand and fixed.
- We add a Lagrange multiplier λ to make sure that $\sum_{j=1}^{m} \pi_j = 1$ and obtain the Lagrangian

$$
\mathcal{L}(\pi, \mu, \lambda) = \sum_{i=1}^{n} \sum_{j=1}^{m} \gamma_j(x_i) \left(\log \pi_j - \frac{1}{\beta_j} |x_i - \mu_j| \right) + \lambda \left(\sum_{j=1}^{m} \pi_j - 1 \right).
$$

Exactly as in the previous problem, by setting the gradient with respect to π_j to zero, we obtain

$$
\frac{\partial}{\partial_{\pi_j}} \mathcal{L}(\pi, \mu, \lambda) = \sum_{i=1}^{n} \gamma_j(x_i) / \pi_j + \lambda = 0 \Longrightarrow \pi_j = \frac{\sum_{i=1}^{n} \gamma_j(x_i)}{-\lambda}.
$$

So we can maximize $L(x)$ with respect to π_j .

EM steps:

1. Initialization: $X = x_1, x_2, \ldots, x_n$
2. E-steps: for every x_i,compute
   $\gamma_j(x_i) = P(y_i = j \mid x_i) = \frac{P(x_i|y_i=j)P(y_i=j)}{\sum_{l=1}^{m} P(x_i|y_i=l)P(y_i=l)} = \frac{\pi_j f_L(x_i;\mu_j,\beta_j)}{\sum_{l=1}^{m} \pi_l f_L(x_i;\mu_l,\beta_l)}$
3. M-steps: iterates to renew the parameters μ_j and π_j ,according to the formulations **as mentioned above**.
4. Compute $L(x)$ .Repeat 2,3,4 until the ***algorithm convergence***,meaning the parameters' estimates do not change significantly with further iterations.