Part 1-1 we know the posterior dist is:

$$O = \sum_{i=1}^{N}\sum_{k=1}^{K} r_k^{(i)} \left[\log P(z^{(i)}=k) + \log p(x^{(i)}|z^{(i)}=k)\right] + \log p(\pi) + \log p(\theta)$$

$$\sim \sum_i \sum_k r_k^{(i)} \left[\log \pi_k + \log \prod_j \theta_{k,j}^{x_j^{(i)}} (1-\theta_{k,j})^{1-x_j^{(i)}}\right] + \log \prod_k \pi_k^{\alpha-1} +$$

$$\log \prod_k \prod_j \theta_{k,j}^{a-1} (1-\theta_{k,j})^{b-1}$$

↑ this is because $\theta_{k,j}$ are independent (defined in the question)

We want to maximize the posterior, therefore:

$$\pi_k = \underset{\pi_k}{\operatorname{argmax}}\ O(\pi, \theta)$$

under $\sum_k \pi_k = 1$

So we use lagrange multiplier to solve it.

$$O' = O + \lambda\left(1 - \sum_k \pi_k\right)$$

$$\frac{\partial O'}{\partial \pi_k} = \sum_i \frac{r_k^{(i)}}{\pi_k} + \frac{\alpha-1}{\pi_k} - \lambda = 0$$

$$\Rightarrow \frac{1}{\pi_k}\left(\sum_i r_k^{(i)} + \alpha - 1\right) = \lambda \Rightarrow \pi_k = \frac{\sum_i r_k^{(i)} + \alpha - 1}{\lambda}$$

$$\sum_k \pi_k = 1 \Rightarrow \frac{\sum_k \sum_i r_k^{(i)} + K\alpha - K}{\lambda} = 1$$

$$\Rightarrow \lambda = \sum_k \sum_i r_k^{(i)} + K\alpha - K$$

$$\Rightarrow \boxed{\pi_k = \frac{\sum_i r_k^{(i)} + \alpha - 1}{\sum_k \sum_i r_k^{(i)} + K\alpha - K}}$$

For inferring $\theta_{k,j}$, we don't need to use lagrang multiplier because of $\theta_{k,j}(1-\theta_{k,j})$, this is already bound.

So,

$$\frac{\partial Q}{\partial \theta_{k,j}} = \sum_i r_k^{(i)} \left(\frac{x_j^{(i)}}{\theta_{k,j}} - \frac{1-x_j^{(i)}}{1-\theta_{k,j}} \right) + \frac{a-1}{\theta_{k,j}} - \frac{b-1}{1-\theta_{k,j}} = 0.$$

$$\Rightarrow \frac{\gamma_{kj}+a-1}{\theta_{k,j}} - \frac{\gamma'_{kj}+b-1}{1-\theta_{k,j}} = 0 \quad \text{where } \gamma_{kj} = \sum_i x_j^{(i)} r_k^{(i)}$$

$$\gamma'_{kj} = \sum_i (1-x_j^{(i)}) r_k^{(i)}$$

$$\Rightarrow (1-\theta_{k,j})(\gamma_{kj}+a-1) - \theta_{k,j}(\gamma'_{kj}+b-1) = 0$$

$$\Rightarrow \theta_{k,j}(\gamma_{kj}+\gamma'_{kj}+b-1+\gamma_{kj}+a-1) = \gamma_{kj}+a-1$$

$$\Rightarrow \boxed{\theta_{k,j} = \frac{\gamma_{kj}+a-1}{\gamma_{kj}+\gamma'_{kj}+a+b-2} = \frac{\gamma_{kj}+a-1}{\sum_i r_k^{(i)}+a+b-2}}$$

therefore we update π_k and θ_{kj} with the latest responibility coefficients $r_k^{(i)}$.

$$\boxed{\begin{aligned} \pi_k &\leftarrow \frac{\sum_i r_k^{(i)} + \alpha - 1}{\sum_k \sum_i r_k^{(i)} + \alpha - 1} \\ \theta_{k,j} &\leftarrow \frac{\sum_i x_j^{(i)} r_k^{(i)} + a - 1}{\sum_i r_k^{(i)} + a + b - 2} \end{aligned}}$$

CSC411 Fall 2018: Homework 6

Niloufar Afsariardchi

November 22, 2018

1 Part 1-2

This is the output of the `mixture.print_part_1_values()`:

```
   ('pi[0]', 0.085000000000000006)
('pi[1]', 0.13)
('theta[0, 239]', 0.64271062271062318)
('theta[3, 298]', 0.46573612495845823)
```

Part 2-1

from Bayes rule:

$$Pr(z=k|x_{ob}) = \frac{P(x_{ob}|z=k)P(z=k)}{P(x_{ob})} \leftarrow \text{for a single image } x_{obj}$$

$$= \frac{P(x_{ob}|z=k)P(z=k)}{\sum_{K} P(x_{ob}|z=k)P(z=k)}$$

$$= \frac{\prod_{j=1}^{D} \theta_{k,j}^{x_j m_j} (1-\theta_{k,j})^{(1-x_j)m_j} \pi_k}{\sum_{K=1}^{K} \prod_{j=1}^{D} \pi_k \theta_{k,j}^{x_j m_j} (1-\theta_{k,j})^{(1-x_j)m_j}}$$

For a set of images:

$$Pr(z=k|X_{ob}) = \frac{\prod_{i=1}^{N} \prod_{j=1}^{D} \pi_k \theta_{k,j}^{x_j^{(i)} m_j^{(i)}} (1-\theta_{k,j})^{(1-x_j^{(i)})m_j^{(i)}}}{\prod_{i=1}^{N} \sum_{K=1}^{K} \pi_k \theta_{k,j}^{x_j^{(i)} m_j^{(i)}} (1-\theta_{k,j})^{(1-x_j^{(i)})m_j^{(i)}}}$$

because images are independent

Note that denominator is constant and could be ignored for other parts.

2 Part2-2

```
('R[0, 2]', 0.17488951492117288)
('R[1, 0]', 0.68853767610922922)
('P[0, 183]', 0.65161519981310367)
('P[2, 628]', 0.47408017249133011)
```

3 Part3-1

With uniform distribution, if a pixel j is off in the all training images, we would set $\theta_{k,j} = 0$ for all values of k. This, however, makes the probability of the test image that has that pixel on 0. This can be easily seen in Equation (2) of the handout, if $\theta_{k,j} = 0$ then $P(\mathbf{x}^{(i)}|z = k) = 0$ for all ks. If we assign zero probability to an image, we would fail to classify the image or make prediction about on-observed pixels as the log likelihood would approach minus infinity. This shows that it is critical to place an appropriate prior (or equivalently do Laplace smoothing) on the model parameters.

4 Part3-2

In EM, we consider soft class assignment, which mean we assign a probability to the responsibility of each class. This is particularly important when there are ambiguous examples in the training set. The given labels act as the hard assignments that do not consider the probability assigned to similar digits, so even though there is more information we are not using information on similar digits and ambiguous cases. On the other hand, EM maximizes the posterior distribution in a heuristic manner, and thus optimize the likelihood function by using extra information inferred from soft assignments.

5 Part3-3

The `log_likelihood` function defined in the code has two terms: $\log \pi$ and $\log p(\mathbf{x}|z)$. Therefore if the log likelihood of the images of digit 1 is higher than that of the digit 8, this does not necessarily means that the number of 1s was higher in the training set but rather it can be due to the second term $\log p(\mathbf{x}|z)$. If the higher log likelihood is due higher number of images then this must be reflected in in $\log \pi$. In this case, it seems the higher log-likelihood of 1 is due to the fact that it's easier and more straightforward to detect digit 1 because there is less ambiguity assigned to writing digit 1 compared to digit 8 that could have be disturbed due to handwriting style, hence $\log p(\mathbf{x}|z)$ is higher for digit 1 compared digit 8, resulting in higher log likelihood value.