

HW6

LINO LASTELLA

1001237654

Part 1

1.

$$\frac{\partial}{\partial \pi_k} \left[\sum_{i=1}^N \sum_{k=1}^K n_k^{(i)} \left[\log p(z^{(i)}=k) + \log p(x^{(i)} | z^{(i)}=k) \right] + \log p(\pi) + \log p(\theta) \right] =$$

$$= \frac{\partial}{\partial \pi_k} \left[\sum_{i=1}^N \sum_{k=1}^K n_k^{(i)} \left[\log \pi_k + \log \left(\prod_{j=1}^D \theta_{k,j}^{x_j^{(i)}} (1-\theta_{k,j})^{1-x_j^{(i)}} \right) \right] + \log \left(\prod_{k=1}^K \pi_k^{d_k-1} \right) + \log \left(\prod_{k=1}^K \prod_{j=1}^D \theta_{k,j}^{d_k-1} (1-\theta_{k,j})^{b-1} \right) \right]$$

$$= \sum_{i=1}^N n_k^{(i)} \cdot \frac{1}{\pi_k} + \frac{\partial}{\partial \pi_k} \left[\sum_{k=1}^K (d_k-1) \log \pi_k \right]$$

$$= \sum_{i=1}^N n_k^{(i)} \cdot \frac{1}{\pi_k} + \frac{d_k-1}{\pi_k} \quad (\text{Partial w.r.t. } \pi_k)$$

Similarly,

$$\frac{\partial}{\partial \theta_{k,j}} \left[\sum_{i=1}^N \sum_{k=1}^K n_k^{(i)} \left[\log p(z^{(i)}=k) + \log p(x^{(i)} | z^{(i)}=k) \right] + \log p(\pi) + \log p(\phi) \right] = \left\{ \text{Same first step as before} \right\}$$

$$= \frac{\partial}{\partial \theta_{k,j}} \left[\sum_{i=1}^N \sum_{k=1}^K n_k^{(i)} \sum_{j=1}^D \left[x_j^{(i)} \log \theta_{k,j} + (1-x_j^{(i)}) \log (1-\theta_{k,j}) \right] + \sum_{k=1}^K \sum_{j=1}^D \left[(d-1) \log \theta_{k,j} + (b-1) \log (1-\theta_{k,j}) \right] \right] =$$

$$= \sum_{i=1}^N n_k^{(i)} \left[\frac{x_j^{(i)}}{\theta_{k,j}} + \frac{x_j^{(i)} - 1}{1 - \theta_{k,j}} \right] + \frac{d-1}{\theta_{k,j}} + \frac{1-b}{1-\theta_{k,j}}$$

(Partial w.r.t. $\theta_{k,j}$)

In the second case we can just set the partial equal to zero and solve for $\theta_{k,j}$.

$$0 = \frac{\sum_{i=1}^N n_k^{(i)} x_j^{(i)}}{\theta_{k,j}} + \frac{\sum_{i=1}^N (x_j^{(i)} - 1) n_k^{(i)}}{1 - \theta_{k,j}} + \frac{d-1}{\theta_{k,j}} + \frac{1-b}{1-\theta_{k,j}}$$

$$\rightarrow (1 - \theta_{k,j}) \left[\sum_{i=1}^N n_k^{(i)} x_j^{(i)} + d-1 \right] = \theta_{k,j} \left(-1+b + \sum_{i=1}^N n_k^{(i)} (1-x_j^{(i)}) \right)$$

$$\rightarrow 0 = \theta_{k,j} \left[(-1+b + \sum_{i=1}^N n_k^{(i)} (1-x_j^{(i)})) + \sum_{i=1}^N n_k^{(i)} x_j^{(i)} + d-1 \right] - \sum_{i=1}^N n_k^{(i)} x_j^{(i)} + 1-d$$

$$\begin{aligned} \rightarrow \theta_{k,j} &= \frac{\sum_{i=1}^N R_k^{(i)} X_j^{(i)} - 1 + d}{\left[\sum_{i=1}^N R_k^{(i)} X_j^{(i)} + \sum_{i=1}^N R_k^{(i)} \right] (d+1) + \sum_{i=1}^N R_k^{(i)} X_j^{(i)} + d-1} \\ &= \frac{d-1 + \sum_{i=1}^N R_k^{(i)} X_j^{(i)}}{d+b-2 + \sum_{i=1}^N R_k^{(i)}} \end{aligned}$$

On the other hand, for π_k we need to apply Lagrange multiplier theorem:

$$\frac{\sum_{i=1}^N R_k^{(i)}}{\pi_k} + \frac{d_k-1}{\pi_k} + \frac{\partial}{\partial \pi_k} \left[\lambda \sum_{k=1}^K (\pi_k - 1) \right] = 0$$

$$\rightarrow \frac{\sum_{i=1}^N R_k^{(i)} + \alpha - 1}{\pi_k} = -\lambda \quad \left[\text{Since all } d_k \text{'s are equal, all them } \alpha \right].$$

$$\rightarrow \pi_k = \frac{\sum_{i=1}^N R_k^{(i)} + \alpha - 1}{-\lambda}$$

$$\text{Since } \sum_{k'=1}^K \pi_{k'} = 1, \quad \sum_{k'=1}^K \sum_{i=1}^N R_{k'}^{(i)} + \alpha - 1 = \lambda$$

$$\text{therefore, } \pi_k = \frac{\sum_{i=1}^N R_k^{(i)} + \alpha - 1}{\sum_{k'=1}^K \left[\sum_{i=1}^N R_{k'}^{(i)} + \alpha - 1 \right]}$$

Part 2

1.

$$P_n(z=k | x_{\text{obs}}) = \frac{P(x_{\text{obs}} | z=k) P(z=k)}{P(x_{\text{obs}})} \quad [\text{Bayes' rule}]$$

$$= \frac{P(x_{\text{obs}} | z=k) P(z=k)}{\sum_{k'=1}^K P(x_{\text{obs}} | z=k') P(z=k')} \quad [\text{Law of total pr.}]$$

$$= \frac{\pi_k \cdot \prod_{j=1}^D \theta_{k,j}^{x_j^{(i)}} (1 - \theta_{k,j})^{1-x_j^{(i)}}}{\sum_{k'=1}^K \left[\prod_{j=1}^D \theta_{k',j}^{x_j^{(i)}} (1 - \theta_{k',j})^{1-x_j^{(i)}} \cdot \pi_{k'} \right]}$$

Note: $x_j^{(i)}$ should be $x_{\text{obs},j}^{(i)}$ everywhere.

Part 3.

1.

If instead of a Beta prior we used a Uniform prior, the MAP learning algorithm would have the problem that it might assign zero prob. to images in the test set because that image might not have appeared consistently in the training set. For instance, if a pixel is always zero in the training set, but 1 in the test set, then the algorithm would not be able to identify the proper cluster to maximize in the M-step.

2.

I think model Part 2 performs better because the partial observations give a closer approximation of the clusters for each step

3.

No, the model is just outlining that predicting a 1 is far easier than predicting an 8. If we sample from its distribution we won't get more 1's than 8's because they are all equally likely to be sampled.

C:\> Command Prompt - py

Microsoft Windows [Version 10.0.17134.407]

(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\linol>cd C:\Users\linol\OneDrive\Desktop\University\4th year\CSC411H1\Assignments\HW6\code_and_data

C:\Users\linol\OneDrive\Desktop\University\4th year\CSC411H1\Assignments\HW6\code_and_data>py

Python 3.5.4 (v3.5.4:3f56838, Aug 8 2017, 02:17:05) [MSC v.1900 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import mixture
```

```
>>> mixture.print_part_1_values()
```

```
pi[0] 0.085
```

```
pi[1] 0.13
```

```
theta[0, 239] 0.6427106227106232
```

```
theta[3, 298] 0.46573612495845823
```

```
>>>
```

— □ ×

