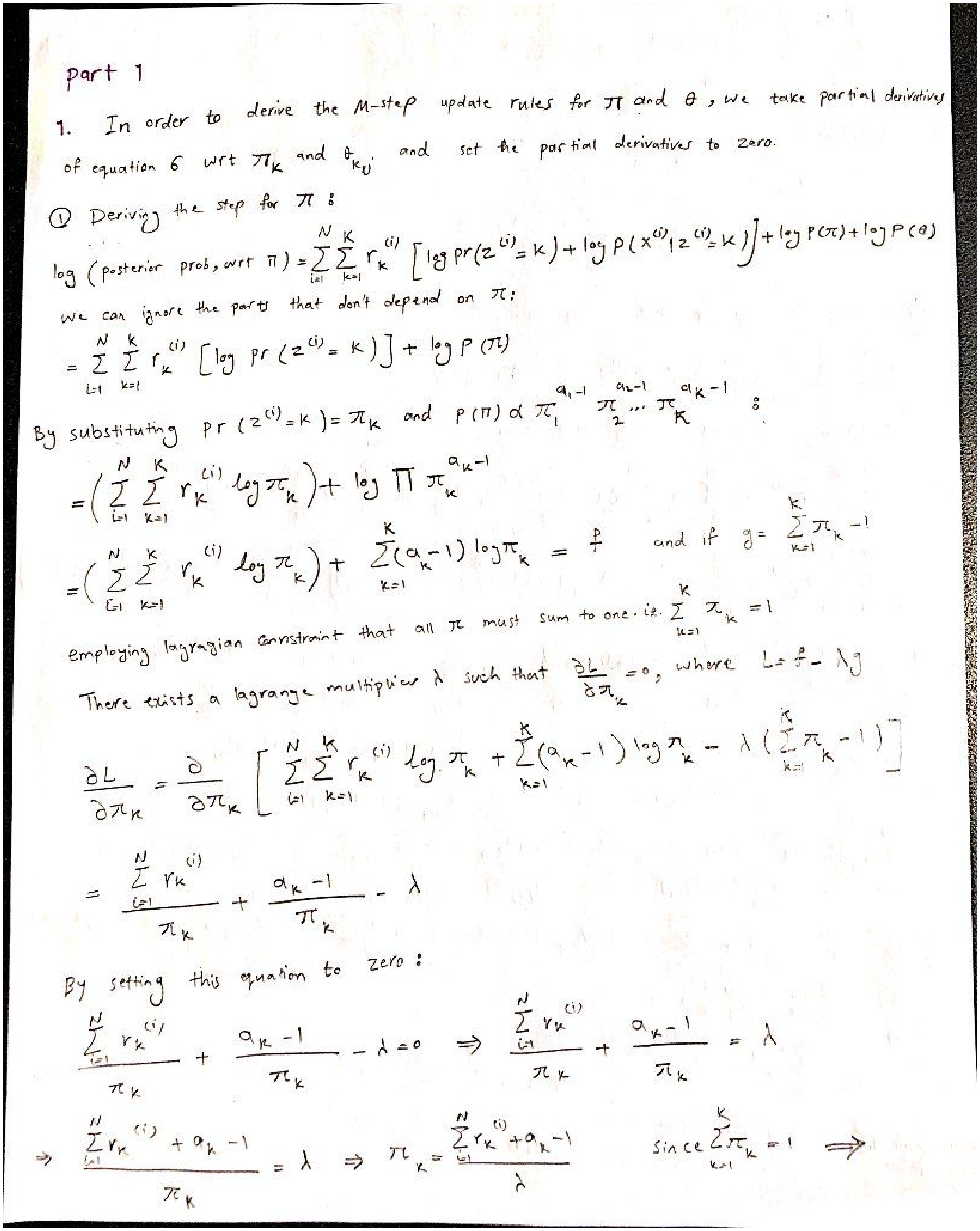
Part 1.



part 1

1. In order to derive the M-step update rules for π and θ, we take partial derivatives of equation 6 wrt π_k and θ_{kj} and set the partial derivatives to zero.

① Deriving the step for π:

$$\log(\text{posterior prob, wrt } \pi) = \sum_{i=1}^{N}\sum_{k=1}^{K} r_k^{(i)} \left[\log \Pr(z^{(i)}=k) + \log p(x^{(i)} | z^{(i)}=k)\right] + \log p(\pi) + \log p(\theta)$$

we can ignore the parts that don't depend on π:

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

By substituting $\Pr(z^{(i)}=k) = \pi_k$ and $p(\pi) \propto \pi_1^{a_1-1}\pi_2^{a_2-1}\dots\pi_K^{a_K-1}$:

$$= \left(\sum_{i=1}^{N}\sum_{k=1}^{K} r_k^{(i)} \log \pi_k\right) + \log \prod \pi_k^{a_k-1}$$

$$= \left(\sum_{i=1}^{N}\sum_{k=1}^{K} r_k^{(i)} \log \pi_k\right) + \sum_{k=1}^{K}(a_k-1)\log\pi_k = f$$

and if $g = \sum_{k=1}^{K}\pi_k - 1$

employing lagragian constraint that all π must sum to one. i.e. $\sum_{k=1}^{K}\pi_k = 1$

There exists a lagrange multiplier λ such that $\frac{\partial L}{\partial \pi_k} = 0$, where $L = f - \lambda g$

$$\frac{\partial L}{\partial \pi_k} = \frac{\partial}{\partial \pi_k}\left[\sum_{i=1}^{N}\sum_{k=1}^{K} r_k^{(i)} \log \pi_k + \sum_{k=1}^{K}(a_k-1)\log\pi_k - \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right)\right]$$

$$= \frac{\sum_{i=1}^{N} r_k^{(i)}}{\pi_k} + \frac{a_k-1}{\pi_k} - \lambda$$

By setting this equation to zero:

$$\frac{\sum_{i=1}^{N} r_k^{(i)}}{\pi_k} + \frac{a_k-1}{\pi_k} - \lambda = 0 \Rightarrow \frac{\sum_{i=1}^{N} r_k^{(i)}}{\pi_k} + \frac{a_k-1}{\pi_k} = \lambda$$

$$\Rightarrow \frac{\sum_{i=1}^{N} r_k^{(i)} + a_k - 1}{\pi_k} = \lambda \Rightarrow \pi_k = \frac{\sum_{i=1}^{N} r_k^{(i)} + a_k - 1}{\lambda}$$

since $\sum_{k=1}^{K}\pi_k = 1 \Rightarrow$

$$\Rightarrow \sum_{k=1}^{N} \frac{\sum_{i=1}^{N} r_k^{(i)} + a_k - 1}{\lambda} = 1 \Rightarrow \lambda = \sum_{k=1}^{K} \left[\sum_{i=1}^{N} r_k^{(i)} + a_k - 1 \right] = \sum_{k=1}^{K}\sum_{i=1}^{N} r_k^{(i)} + \sum_{k=1}^{K} a_k - K$$

Therefore :

$$\pi_k \leftarrow \frac{a_k - 1 + \sum_{i=1}^{N} r_k^{(i)}}{\sum_{k'=1}^{K} \sum_{i=1}^{N} r_{k'}^{(i)} + \sum_{k'=1}^{K} a_{k'} - K}$$

---

② Deriving the step for θ : by ignoring the parts that don't depend on θ :

$$L = \sum_{i=1}^{N}\sum_{k=1}^{K} r_k^{(i)} \log p(x^{(i)} | z^{(i)} = k) + \log p(\theta)$$

Substituting $p(x^{(i)} | z^{(i)} = k)$ and $p(\theta)$:

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} r_k^{(i)} \log \prod_{j=1}^{D} \theta_{k,j}^{x_j^{(i)}} (1-\theta_{k,j})^{1-x_j^{(i)}} + \log \prod_{k=1}^{K}\prod_{j=1}^{D} \theta_{k,j}^{a-1} (1-\theta_{k,j})^{b-1}$$

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

$$\frac{\partial L}{\partial \theta_{k,j}} = \sum_{i=1}^{K} r_k^{(i)} \left(\frac{x_j^{(i)}}{\theta_{k,j}} + \frac{(-1)(1-x_j^{(i)})}{1-\theta_{k,j}} \right) + \frac{a-1}{\theta_{k,j}} + \frac{(-1)(b-1)}{1-\theta_{k,j}}$$

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

$$= \sum_{i=1}^{N} r_k^{(i)} \frac{x_j^{(i)} - \theta_{k,j}}{\theta_{k,j}(1-\theta_{k,j})} + \frac{(2-a-b)\theta_{k,j} + a - 1}{\theta_{k,j}(1-\theta_{k,j})} = 0$$ by setting this to zero :

$$\sum_{i=1}^{N} r_k^{(i)} (x_j^{(i)} - \theta_{k,j}) + (2-a-b)\theta_{k,j} + a - 1 = 0 \Rightarrow \sum_{i=1}^{N} r_k^{(i)} x_j^{(i)} - \sum_{i=1}^{N} r_k^{(i)} \theta_{k,j} + (2-a-b)\theta_{k,j} + a - 1 = 0$$

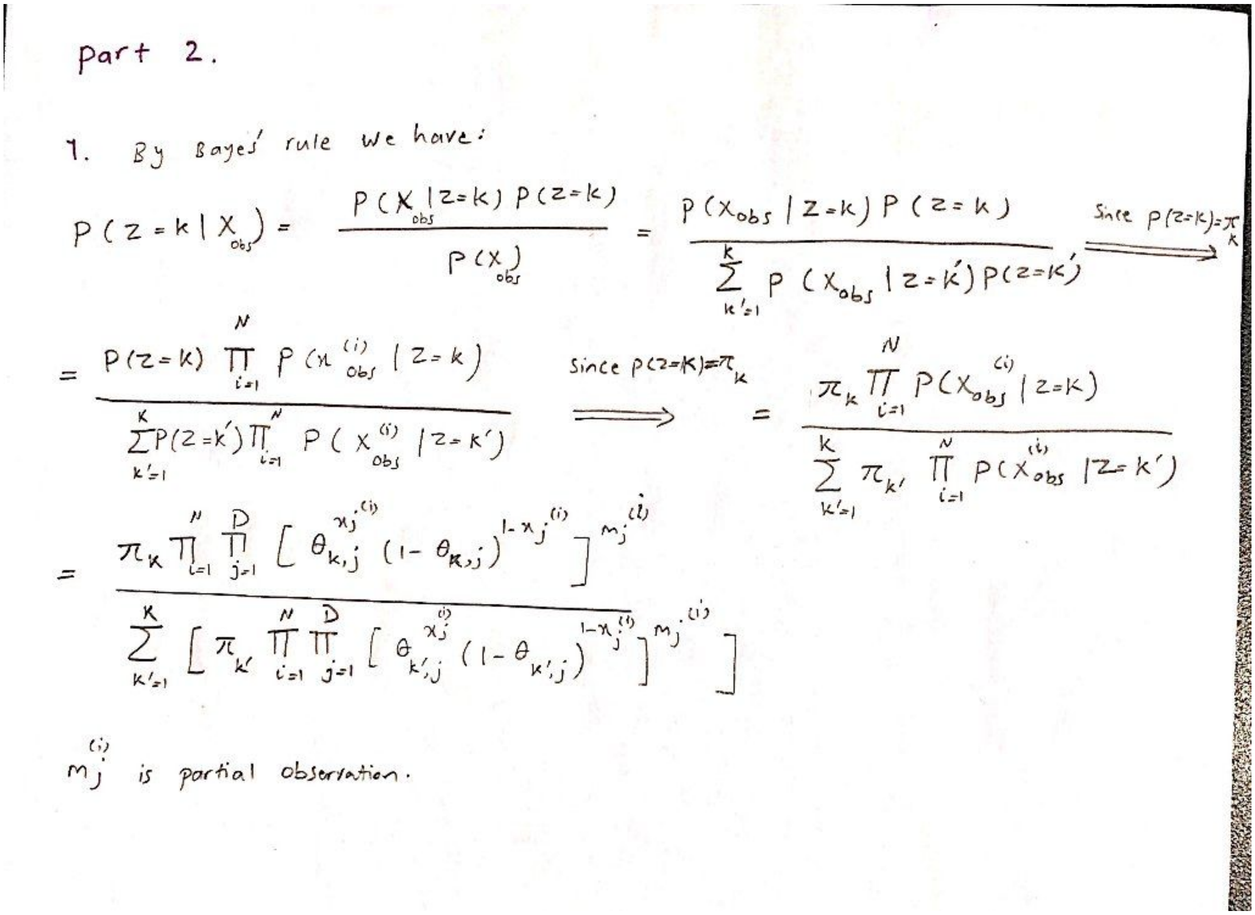
$$\theta_{k,j} = \frac{\sum_{i=1}^{N} r_k^{(i)} x_j^{(i)} + a - 1}{\sum_{i=1}^{N} r_k^{(i)} + a + b - 2}$$

Therefore : $$\theta_{k,j} \leftarrow \frac{a - 1 + \sum_{i=1}^{N} r_k^{(i)} x_j^{(i)}}{a + b - 2 + \sum_{i=1}^{N} r_k^{(i)}}$$

2.

pi[0] 0.085
pi[1] 0.13
theta[0, 239] 0.6427106227106232
theta[3, 298] 0.46573612495845823

Part 2.



part 2.

1. By Bayes' rule we have:

$$P(z=k \mid X_{obs}) = \frac{P(X_{obs} \mid z=k)\, P(z=k)}{P(X_{obs})} = \frac{P(X_{obs} \mid z=k)\, P(z=k)}{\sum_{k'=1}^{K} P(X_{obs} \mid z=k')\, P(z=k')} \xrightarrow{\text{Since } P(z=k)=\pi_k}$$

$$= \frac{P(z=k) \prod_{i=1}^{N} P(x_{obs}^{(i)} \mid z=k)}{\sum_{k'=1}^{K} P(z=k') \prod_{i=1}^{N} P(x_{obs}^{(i)} \mid z=k')} \xrightarrow{\text{Since } P(z=k)=\pi_k} = \frac{\pi_k \prod_{i=1}^{N} P(X_{obs}^{(i)} \mid z=k)}{\sum_{k'=1}^{K} \pi_{k'} \prod_{i=1}^{N} P(X_{obs}^{(i)} \mid z=k')}$$

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

$m_j^{(i)}$ is partial observation.

2.

R[0, 2] 0.17488951492117288
R[1, 0] 0.6885376761092292
P[0, 183] 0.6516151998131037
P[2, 628] 0.4740801724913301

Part 3.

1.

Based on the derivation of $\theta_{k,j}$ from part 1 question 1, we have:

$$\theta_{k,j} \leftarrow \frac{a - 1 + \sum_{i=1}^{N} r^{(i)}_{k} \; x^{(i)}_{j}}{a + b - 2 + \sum_{i=1}^{N} r^{(i)}_{k}}$$

Therefore, if we set a = b = 1 then we have:

$$\theta_{k,j} \leftarrow \frac{\sum_{i=1}^{N} r^{(i)}_{k} \; x^{(i)}_{j}}{\sum_{i=1}^{N} r^{(i)}_{k}}$$

Therefore if pixel $x^{(i)}_{j}$ is always zero in the training set, then we would have $\theta_{k,j} \leftarrow 0$ in the training process which means that this pixel in the test image would have a probability of zero although it has a value of 1. So if we use uniform distribution, the MAP learning algorithm would cause the problem that it might assign zero probability to images in the test set, which implies we may end with a bad model trained and would not produce good estimation.

2.

In part one, we only have 10 classes to work with, whereas in part 2 we have 100 classes. The additional freedom provided by these 100 components allows us to capture the idiosyncrasies within a class, meaning that variation within each component is reduced. 10 classes are definitely not enough to train a good model in this case since people have distinct handwriting habits, e.g the mean of all the 9's might not actually look like any particular 9.

3.

No, 1s are not more common. This does indicate, however, that 1's show less variation.
If you sample from the distribution, you will get 1 and 8 an equal amount of times since the distribution should be equally likely for both of them. The model is only saying that it is easier to pick a 1 than an 8, not that it is more likely.
The larger the value of log-probability, the more confidence the model is on its prediction. Thus, both models are more confident in their predictions on 1 than their predictions on 8. This makes

sense since 1 is a number that is very similar for its top half and its bottom half part, but for 8, only given top half part, it may be classified as other digit.