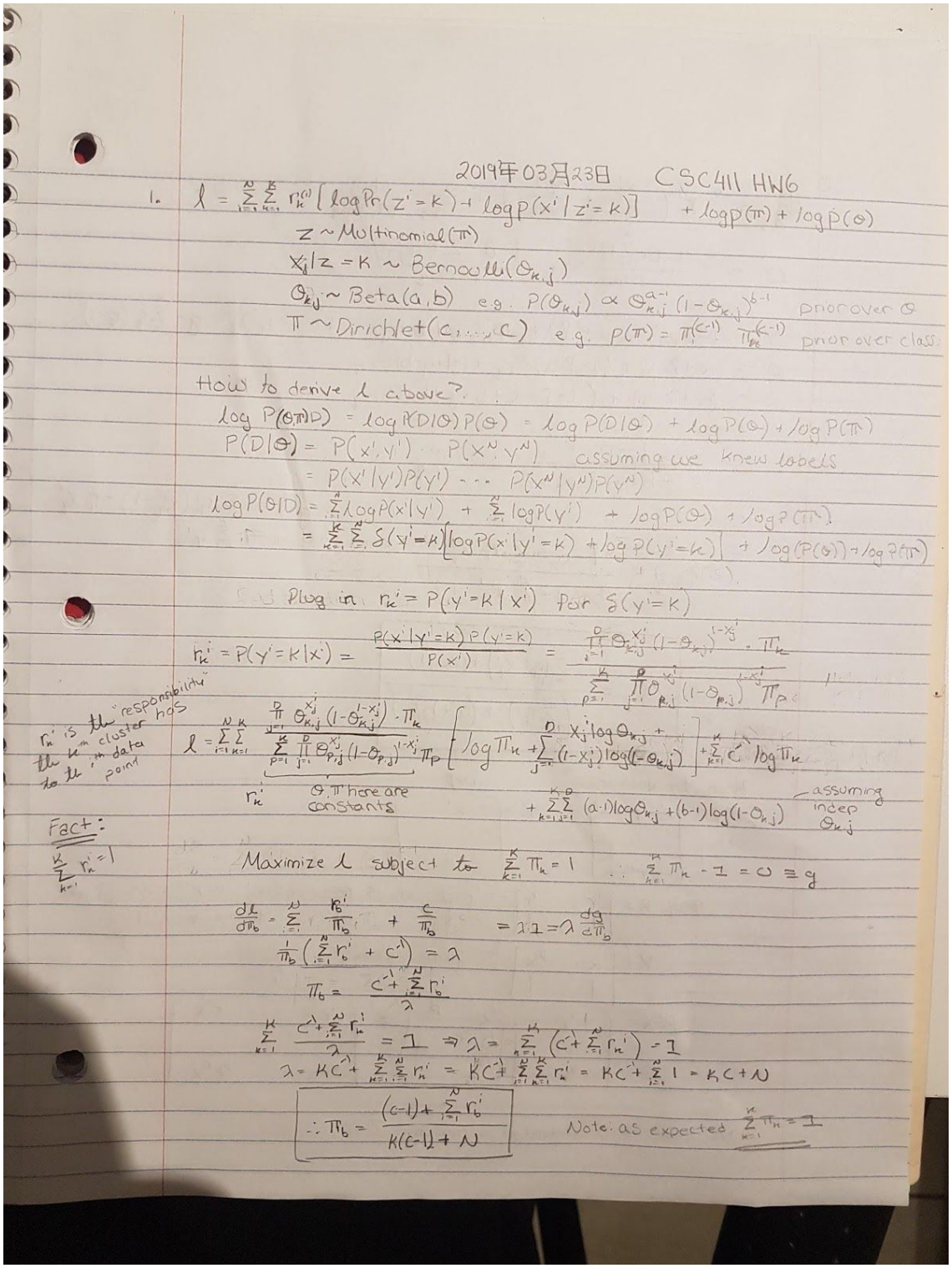
Part 1

Question 1



$$\frac{dl}{d\theta_{p,q}} = \sum_{i=1}^{N} r_p^i \left(\frac{x_q^i}{\theta_{p,q}} + \frac{x_q^i - 1}{1-\theta_{p,q}} \right) + \frac{a-1}{\theta_{p,q}} - \frac{b-1}{1-\theta_{p,q}}$$

$$0 = \frac{a-1+\sum_{i=1}^{N} r_p^i x_q^i}{\theta_{p,q}} + \frac{1-b+\sum_{i=1}^{N} r_p^i (x_q^i - 1)}{1-\theta_{p,q}}$$

$$0 = (1-\theta_{p,q})(a-1+f) + \theta_{p,q}(1-b+g), \quad f = \sum_{i=1}^{N} r_p^i x_q^i \quad g = \sum_{i=1}^{N} r_p^i (x_q^i - 1)$$

$$0 = (a+f-1) + (1-f-a)\theta_{p,q} + (+1+g-b)\theta_{p,q}$$

$$0 = (a+f-1) + (1-f-a-b+g+1)\theta_{p,q}$$

$$\frac{1-a-f}{-b+g-f-a+2} = \theta_{p,q}$$

$$g - f = \sum_{i=1}^{N} r_p^i (x_q^i - 1) - r_p^i x_q^i = -\sum_{i=1}^{N} r_p^i$$

$$\theta_{p,q} = \frac{1-a-\sum_{i=1}^{N} r_p^i x_q^i}{-(b+a) - \sum_{i=1}^{N} r_p^i + 2}$$

$$\boxed{\theta_{p,q} = \frac{\sum_{i=1}^{N}(r_p^i x_q^i) + (a-1)}{\left(\sum_{i=1}^{N} r_p^i\right) + (a+b) - 2}}$$

class (p), feature (q)

How to vectorize?

$$\underset{N\times K}{R} = \begin{bmatrix} — r^1 — \\ \vdots \\ — r^N — \end{bmatrix} \qquad \underset{N\times D}{X} = \begin{bmatrix} — x^1 — \\ \vdots \\ — x^N — \end{bmatrix}$$

Say $R = \begin{bmatrix} {}^1r & \cdots & {}^Kr \\ | & & | \end{bmatrix}$ — each col. is responsibilities of K^{th} class of all training examples

$X = \begin{bmatrix} {}^1x & \cdots & {}^Dx \\ | & & | \end{bmatrix}$ — same idea.

$${}^pr'{}^qx$$

$$R'X = \begin{bmatrix} — {}^1r — \\ \vdots \\ — {}^Kr — \end{bmatrix} \begin{bmatrix} {}^1x & \cdots & {}^Dx \\ | & & | \end{bmatrix} = \begin{bmatrix} {}^1r'x & \cdots & {}^1r^Dx \\ \vdots & & \\ {}^Kr'x & \cdots & {}^Kr^Dx \end{bmatrix}$$

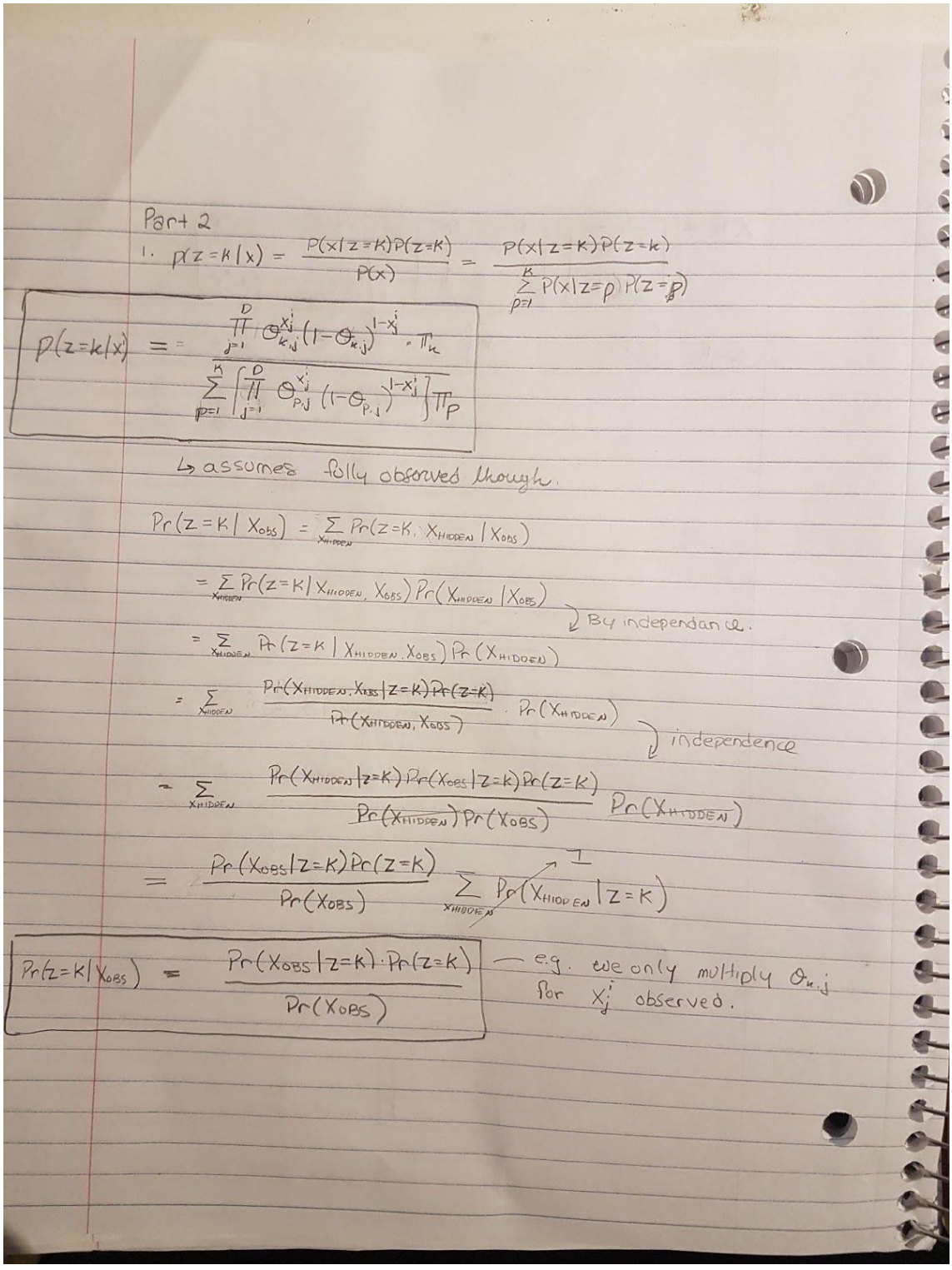
Then divide each element of a row by $\sum_{i=1}^{N} r_p^i$.

Question 2

Part 1 values:
('pi[0]', 0.084999999999999923)
('pi[1]', 0.12999999999999987)
('theta[0, 239]', 0.64271062271062329)
('theta[3, 298]', 0.46573612495845823)

Part 2

Question 1



Part 2

1. $p(z=k|x) = \frac{P(x|z=k)P(z=k)}{P(x)} = \frac{P(x|z=k)P(z=k)}{\sum_{\rho=1}^{K} P(x|z=\rho)P(z=\rho)}$

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

↳ assumes fully observed though.

$Pr(z=K|X_{obs}) = \sum_{X_{HIDDEN}} Pr(z=K, X_{HIDDEN}|X_{OBS})$

$= \sum_{X_{HIDDEN}} Pr(z=K|X_{HIDDEN}, X_{OBS}) Pr(X_{HIDDEN}|X_{OBS})$ ↘ By independance.

$= \sum_{X_{HIDDEN}} Pr(z=K|X_{HIDDEN}, X_{OBS}) Pr(X_{HIDDEN})$

$= \sum_{X_{HIDDEN}} \frac{Pr(X_{HIDDEN}, X_{OBS}|z=K)Pr(z=K)}{Pr(X_{HIDDEN}, X_{OBS})} \cdot Pr(X_{HIDDEN})$ ↘ independence

$= \sum_{X_{HIDDEN}} \frac{Pr(X_{HIDDEN}|z=K)Pr(X_{OBS}|z=K)Pr(z=K)}{Pr(X_{HIDDEN})Pr(X_{OBS})} Pr(X_{HIDDEN})$

$= \frac{Pr(X_{OBS}|z=K)Pr(z=K)}{Pr(X_{OBS})} \sum_{X_{HIDDEN}} Pr(X_{HIDDEN}|z=K)$ ↗ 1

$$Pr(z=K|X_{OBS}) = \frac{Pr(X_{OBS}|z=K)\cdot Pr(z=K)}{Pr(X_{OBS})}$$

— e.g. we only multiply $\theta_{k,j}$ for X_j^i observed.

Question 2

Posterior Predictive Mean

Given Posterior: $p(z=k \mid x_{obs}, \hat{\theta})$. Define $x_u := x_{unknown}$

$$p(x_u \mid x_{obs}, \hat{\theta}) = \sum_{k=1}^{K} p(x_u, z=k \mid x_{obs}, \hat{\theta})$$
$$= \sum_{k=1}^{K} p(x_u \mid x_{obs}, z=k, \hat{\theta})\, p(z=k \mid x_{obs}, \hat{\theta})$$

↓

By our model, $z=k$ totally gives distribution

$$\therefore \sum_{k=1}^{K} p(x_u \mid z=k, \hat{\theta})\, p(z=k \mid x_{obs}, \hat{\theta})$$

$$E[x_u] = 0 \cdot p(x_u \mid x_{obs}, \hat{\theta}) + 1 \cdot p(x_u \mid x_{obs}, \hat{\theta})$$

→ abuse of notation a bit. we are considering a specific pixel. here. say feature d

$$\therefore E[x_u] = \sum_{k=1}^{K} p(x_u = 1 \mid z=k, \hat{\theta})\, p(z=k \mid x_{obs}, \hat{\theta})$$
$$= \sum_{k=1}^{K} \hat{\theta}_{k,d} \cdot p(z=k \mid x_{obs}, \hat{\theta})$$

e.g. Average the θ of the classes weighted by the ~~pro~~ posterior

Vectorization:

$$\theta = \begin{bmatrix} - \theta_0 - \\ \vdots \\ - \theta_K - \end{bmatrix} \qquad \text{posterior} = \begin{bmatrix} P(y^1=0) & \cdots & P(y^1=K) \\ & \vdots & \\ P(y^N=0) & \cdots & P(y^N=K) \end{bmatrix}$$

→ Need weighted avg of rows of this

$$\theta^T \begin{bmatrix} P(y^1=0) \\ \vdots \\ P(y^1=K) \end{bmatrix} = E[x_u \text{ for } x^1] = \begin{bmatrix} 1 \\ \theta_0 \\ 1 \end{bmatrix} P(y^1=0) + \cdots + \begin{bmatrix} 1 \\ \theta_K \\ 1 \end{bmatrix} P(y^1=K)$$

Question 3

Part 2 values:
('R[0, 2]', 0.1748895149211743)
('R[1, 0]', 0.68853767610923089)
('P[0, 183]', 0.65161519981310789)
('P[2, 628]', 0.47408017249133272)

Part 3

Output of average log probability by digit class for model trained by labels:

Training set

Average log-probability of a 0 image: -201.136
Average log-probability of a 1 image: -97.755
Average log-probability of a 2 image: -201.701
Average log-probability of a 3 image: -184.880
Average log-probability of a 4 image: -172.018
Average log-probability of a 5 image: -191.928
Average log-probability of a 6 image: -177.253
Average log-probability of a 7 image: -156.895
Average log-probability of a 8 image: -187.626
Average log-probability of a 9 image: -162.041

Test set

Average log-probability of a 0 image: -199.743
Average log-probability of a 1 image: -96.973
Average log-probability of a 2 image: -199.290
Average log-probability of a 3 image: -182.329
Average log-probability of a 4 image: -171.179
Average log-probability of a 5 image: -190.813
Average log-probability of a 6 image: -181.537
Average log-probability of a 7 image: -154.226
Average log-probability of a 8 image: -187.226
Average log-probability of a 9 image: -159.568

Part 3

1. Recall: $\theta_{p,q} \leftarrow \frac{\left(\sum_{i=1}^{n} r_p^i x_q^i\right) + a - 1}{\left(\sum_{i=1}^{n} r_p^i\right) + a + b - 2}$ if $x_q^i = 0 \ \forall i$, $\theta_{p,q} \leftarrow 0$.

if $a = b = 1$, $\theta_{p,q} \leftarrow \frac{\sum r_p^i x_q^i}{\sum_{i=1}^{n} r_p^i}$

By our indep. assumption:

$$P(x \mid z = k) = \prod_{j=1}^{D} \theta_{k,j}^{x_j^i} (1 - \theta_{k,j})^{x_j^i}$$

if $x_q^i = 1$, then it is clear $\forall k$, $P(x \mid z = k) = 0$.

Conceptually, based on the data, algo is 100% sure that for all classes, that pixel MUST be off. So it predicts 0 prob if by chance it is on.

2. The part 1 model only has 10 clusters for each label so it averages and misses out on variations in writing. This will be detrimental to similar looking numbers. e.g: 7, 7, 9, 8, 9

In contrast, part 2 has 100 clusters. It can form clusters to differentiate 7, 7, 9, 8, 9

3. The average log probability for each digit class is a report of how confidently the model predicts the correct class for examples in the class by averaging over all the examples in a class.

It is clear then that the model does NOT believe 1's are more common than 8's. It means it is easier to differentiates 1's than 8's. This makes sense because 7, 9, 3 for example look like 8.

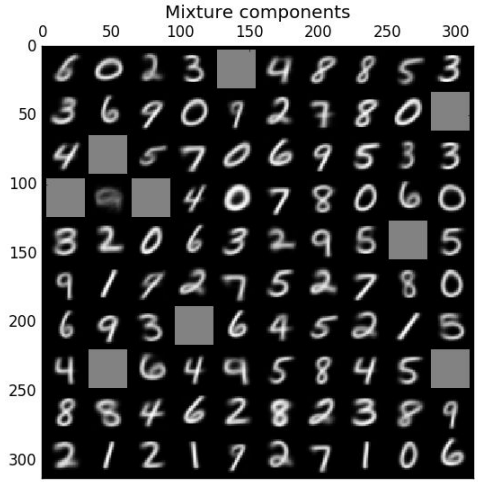
The relative frequency of sampling a class is actually given by π. Thus, from the info given, it is unknown if one would sample more 1s or 8s.

Extra

Train_from_labels output:
('Training log-likelihood:', -172.07854196938328)
('Test log-likelihood:', -170.97538133747267)

Train_with_em output:
('Final training log-likelihood:', -138.14790916090735)
('Final test log-likelihood:', -138.52928834184047)

EM Mixtures:



Model Predictions (top half was observed):