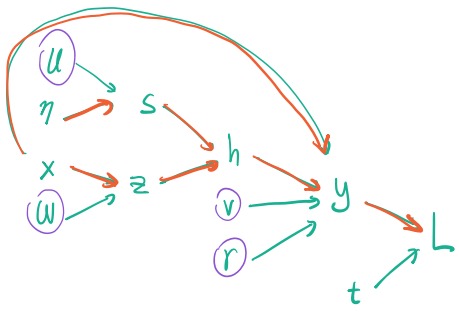
1. a)



b) $\frac{d\sigma}{dx} = \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right) = \frac{d}{dx}\left((1+e^{-x})^{-1}\right)$

$= -(1+e^{-x})^{-2} \cdot (-e^{-x})$

$= e^{-x} \cdot (1+e^{-x})^{-2}$ Note: $1-\sigma(x) = 1-\frac{1}{1+e^{-x}}$ $= \frac{e^{-x}}{1+e^{-x}}$

$= \frac{e^{-x}}{1+e^{-x}} \cdot \frac{1}{1+e^{-x}}$

$= (1-\sigma(x)) \cdot \sigma(x)$

c) $z_i = \sum_j w_{ij} x_j$, $s_i = \sum_j u_{ij} \eta_j$, $h_i = z_i \cdot s_i$

1. $\bar{L} = 1$ (Consider every variable except t)
2. $\bar{y} = \bar{L}\left(\frac{t}{y} - \frac{1-t}{1-y}\right)$
3. $\bar{h}_i = \bar{y}\left(\frac{dy}{dh_i}\right)$

$= \sigma'(v^T h + r^T x) \cdot \frac{d}{dh_i}(v^T h + r^T x)$

$\sigma'(v^T h + r^T x) = \sigma(v^T h + r^T x)(1-\sigma(v^T h + r^T x))$ (let this term be denoted by m)

$\frac{d}{dh_i}(v^T h + r^T x) = \sum_k v_k h_k + \sum_c r_c x_c$

$= m \cdot v_i$

$= \bar{y} \cdot m \cdot v_i$

4. $\bar{v}_i = \bar{y} \cdot \frac{dy}{dv_i} = \bar{y} \cdot m \cdot \frac{d}{dv_i}\left(\sum_k v_k h_k + \sum_c r_c x_c\right)$

$= \bar{y} \cdot m \cdot h_i$

5. $\bar{r}_i = \bar{y} \cdot \frac{dy}{dr_i} = \bar{y} \cdot m \cdot \frac{d}{dr_i}\left(\sum_k v_k h_k + \sum_c r_c x_c\right)$

$= \bar{y} \cdot m \cdot x_i$

6. $\bar{s}_i = \bar{h}_i \cdot \frac{dh_i}{ds_i} = \bar{h}_i \cdot z_i$

7. $\bar{z}_i = \bar{h}_i \cdot \frac{dh_i}{dz_i} = \bar{h}_i \cdot s_i$

8. $\bar{u}_{ij} = \bar{s}_i \cdot \eta_j$

9. $\bar{w}_{ij} = \bar{z}_i \cdot x_j$

10. $\bar{\eta}_j = \sum_i \bar{s}_i u_{ij}$

11. $\bar{x}_j = \sum_i \bar{z}_i w_{ij} + \bar{y} \cdot \frac{dy}{dx_j}$

$= \sum_i \bar{z}_i w_{ij} + \bar{y} \cdot m \cdot \frac{d}{dx_j}\left(\sum_k v_k h_k + \sum_c r_c x_c\right)$

$= \sum_i \bar{z}_i w_{ij} + \bar{y} \cdot m \cdot r_j$

2.

The starter code will download the dataset and parse it for you: Each training sample $(\mathbf{t}^{(i)}, \mathbf{x}^{(i)})$ is composed of a vectorized binary image $\mathbf{x}^{(i)} \in \{0,1\}^{784}$, and 1-of-10 encoded class label $\mathbf{t}^{(i)}$. i.e. $t_c^{(i)} = 1$ means image i belongs to class c.

Given parameters $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$, Naïve Bayes defines the joint probability of each data point $\mathbf{x}$ and its class label c as follows:

prob. x is labeled w/ c, given θ & π

$$p(\mathbf{x}, c \,|\, \boldsymbol{\theta}, \boldsymbol{\pi}) = p(c \,|\, \boldsymbol{\theta}, \boldsymbol{\pi}) p(\mathbf{x} \,|\, c, \boldsymbol{\theta}, \boldsymbol{\pi}) = p(c \,|\, \boldsymbol{\pi}) \prod_{j=1}^{784} p(x_j \,|\, c, \theta_{jc}).$$

prob. of jth pixel is black, given c & θ

where $p(c \,|\, \boldsymbol{\pi}) = \pi_c$ and $p(x_j = 1 \,|\, c, \boldsymbol{\theta}) = \theta_{jc}$. Here, $\boldsymbol{\theta}$ is a matrix of probabilities for each pixel and each class, so its dimensions are 784×10, and $\boldsymbol{\pi}$ is a vector with one entry for each class. (Note that in the lecture, we simplified notation and didn't write the probabilities conditioned on the parameters, i.e. $p(c|\boldsymbol{\pi})$ is written as $p(c)$ in lecture slides).

π_i = prob that target has the ith label

For binary data ($x_j \in \{0,1\}$), we can write the Bernoulli likelihood as

$$p(x_j \,|\, c, \theta_{jc}) = \theta_{jc}^{x_j} (1 - \theta_{jc})^{(1-x_j)}, \tag{1}$$

which is just a way of expressing $p(x_j = 1|c, \theta_{jc}) = \theta_{jc}$ and $p(x_j = 0|c, \theta_{jc}) = 1 - \theta_{jc}$ in a compact form.

For the prior $p(\mathbf{t} \,|\, \boldsymbol{\pi})$, we use a categorical distribution (generalization of Bernoulli distribution to multi-class case),

prob. of class label given set of probabilities → How many targets have label j

$$p(t_c = 1 \,|\, \boldsymbol{\pi}) = p(c \,|\, \boldsymbol{\pi}) = \pi_c \quad \text{or equivalently} \quad p(\mathbf{t} \,|\, \boldsymbol{\pi}) = \Pi_{j=0}^{9} \pi_j^{t_j} \quad \text{where} \quad \sum_{i=0}^{9} \pi_i = 1,$$

↳ prob. of target having label j

where $p(c \,|\, \boldsymbol{\pi})$ and $p(\mathbf{t} \,|\, \boldsymbol{\pi})$ can be used interchangeably. You will fit the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$ using MLE and MAP techniques. In both cases, your fitting procedure can be written as a few simple matrix multiplication operations.

a) MLE:

1. Write down the likelihood objective:

$$L(\theta; x_1, \cdots, x_N) = \prod_{i=1}^{N} L(\theta; x_i)$$

2. Transform to log likelihood:

$$l(\theta; x_1, \cdots, x_N) = \sum_{i=1}^{N} \log L(\theta; x_i)$$

3. Compute the critical point:

$$\frac{\partial l}{\partial \theta} = 0$$

$$\begin{aligned}
\ell(\boldsymbol{\theta}) &= \sum_{i=1}^{N} \log p(c^{(i)}, \mathbf{x}^{(i)}) = \sum_{i=1}^{N} \log \left\{ p(\mathbf{x}^{(i)}|c^{(i)}) p(c^{(i)}) \right\} \\
&= \sum_{i=1}^{N} \log \left\{ p(c^{(i)}) \prod_{j=1}^{D} p(x_j^{(i)} \,|\, c^{(i)}) \right\} \\
&= \sum_{i=1}^{N} \left[\log p(c^{(i)}) + \sum_{j=1}^{D} \log p(x_j^{(i)} \,|\, c^{(i)}) \right] \\
&= \underbrace{\sum_{i=1}^{N} \log p(c^{(i)})}_{\substack{\text{Log-likelihood} \\ \text{of labels}}} + \sum_{j=1}^{D} \underbrace{\sum_{i=1}^{N} \log p(x_j^{(i)} \,|\, c^{(i)})}_{\substack{\text{Log-likelihood} \\ \text{for feature } x_j}}
\end{aligned}$$

$$\ell(\theta) = \sum_{i=1}^{N} \log \left\{ p(x^{(i)} | c^{(i)}) \cdot p(c^{(i)}) \right\}$$

$$= \sum_{i=1}^{N} \log \left\{ \prod_{j=1}^{784} p(x_j^{(i)} | c, \theta_{jc}) \cdot p(c^{(i)}) \right\}$$

$$= \sum_{i=1}^{N} \log \left\{ \prod_{j=1}^{784} \theta_{jc}^{x_j^{(i)}} (1-\theta_{jc})^{1-x_j^{(i)}} \cdot p(c^{(i)}) \right\}$$

$$= \sum_{i=1}^{N} \left[\log p(c^{(i)}) + \sum_{j=1}^{784} \log \left(\theta_{jc}^{x_j^{(i)}} (1-\theta_{jc})^{1-x_j^{(i)}} \right) \right]$$

$$= \underbrace{\sum_{i=1}^{N} \log p(c^{(i)})}_{\text{①}} + \underbrace{\sum_{i=1}^{N} \sum_{j=1}^{784} x_j^{(i)} \log(\theta_{jc}) + (1-x_j^{(i)}) \log(1-\theta_{jc})}_{\text{②}}$$

① log likelihood of labels

② log likelihood for pixel x_j (let this be ℓ_θ)

① : $$\sum_{i=1}^{N} \log p(c^{(i)}) = \sum_{i=1}^{N} \log \left(\prod_{j=0}^{9} \pi_j^{t_j^{(i)}} \right)$$

$$= \sum_{i=1}^{N} \sum_{j=0}^{9} t_j^{(i)} \cdot \log(\pi_j)$$

$$= \sum_{i=1}^{N} \left(\sum_{j=0}^{8} t_j^{(i)} \log(\pi_j) + t_9^{(i)} \log \left(1 - \sum_{j=0}^{8} \pi_j \right) \right)$$

(let = ℓ_π)

then $$\frac{d\ell_\pi}{d\pi_j} = \sum_{i=1}^{N} \left(\frac{t_j^{(i)}}{\pi_j} + \frac{t_9^{(i)}}{1 - \sum_{j=0}^{8} \pi_j} \cdot (-1) \right)$$

$$= \sum_{i=1}^{N} \left(\frac{t_j^{(i)}}{\pi_j} - \frac{t_9^{(i)}}{\pi_9} \right) = 0$$

$$\Rightarrow \sum_{i=1}^{N} \frac{t_j^{(i)}}{\pi_j} = \sum_{i=1}^{N} \frac{t_9^{(i)}}{\pi_9}$$

$$\Rightarrow \sum_{i=1}^{N} \frac{t_j^{(i)}}{t_9^{(i)}} = \frac{\hat{\pi}_j}{\hat{\pi}_9}$$

Then sum up all j:

$$\sum_{j=0}^{9} \frac{\hat{\pi}_j}{\hat{\pi}_9} = \frac{\hat{\pi}_0 + \ldots + \hat{\pi}_9}{\hat{\pi}_9} = \frac{1}{\hat{\pi}_9}$$

$$\hookrightarrow = \sum_{j=0}^{9} \sum_{i=1}^{N} \frac{t_j^{(i)}}{t_9^{(i)}}$$

$$= \sum_{i=1}^{N} \frac{t_0^{(i)} + \ldots + t_9^{(i)}}{t_9^{(i)}}$$, where each training sample has exactly 1 class

$$= \frac{N}{\sum_{i=1}^{N} t_9^{(i)}}$$

So $\hat{\pi}_9 = \frac{\sum_{i=1}^{N} t_9^{(i)}}{N}$, repeating this process for every π_j, (i.e. instead of isolating π_9, isolate π_k for $k \in \{0, \ldots, 8\}$, and we get:

$$\hat{\pi}_j = \frac{\sum_{i=1}^{N} t_j^{(i)}}{N}$$

→ # of samples classified as label j

→ Total # of samples

identity function

② $$\frac{d\ell_\theta}{d\theta_{jc}} = \sum_{i=1}^{N} \mathbb{I}(c^{(i)} = c) \cdot \left(\frac{x_j^{(i)}}{\theta_{jc}} + \frac{1 - x_j^{(i)}}{1 - \theta_{jc}} (-1) \right) = 0$$

$$\Rightarrow \sum_{i=1}^{N} \underbrace{\mathbb{I}(c^{(i)} = c)}_{= t_c^{(i)}} \cdot \frac{x_j^{(i)}}{\theta_{jc}} = \sum_{i=1}^{N} \underbrace{\mathbb{I}(c^{(i)} = c)}_{= t_c^{(i)}} \cdot \frac{1 - x_j^{(i)}}{1 - \theta_{jc}}$$

$$\Rightarrow \sum_{i=1}^{N} t_c^{(i)} \cdot \frac{x_j^{(i)} - x_j^{(i)} \theta_{jc}}{\theta_{jc}} = \sum_{i=1}^{N} t_c^{(i)} \left(1 - x_j^{(i)} \right)$$

$$\Rightarrow \sum_{i=1}^{N} t_c^{(i)} \left(\frac{x_j^{(i)}}{\theta_{jc}} - x_j^{(i)} \right) = \sum_{i=1}^{N} t_c^{(i)} \left(1 - x_j^{(i)} \right)$$

$$\Rightarrow \frac{\sum_{i=1}^{N} t_c^{(i)} x_j^{(i)}}{\theta_{jc}} = \sum_{i=1}^{N} t_c^{(i)}$$

$$\Rightarrow \hat{\theta}_{jc} = \frac{\sum_{i=1}^{N} t_c^{(i)} x_j^{(i)}}{\sum_{i=1}^{N} t_c^{(i)}}$$

→ Sum of binary values on pixel j for all samples classified as c

→ total # of samples classified as c

b) Note that $p(t|x,\theta,\pi) \cdot p(x|\theta,\pi) = p(t,x|\theta,\pi)$

then $p(t|x,\theta,\pi) = \frac{p(t,x|\theta,\pi)}{p(x|\theta,\pi)} = \sum_{i=0}^{9} p(x,t_i|\theta,\pi)$ by law of total prob.

$p(t,x|\theta,\pi)$: calculated as in a) for $N=1$

$$p(x|\theta,\pi) := \sum_{i=0}^{9}\left[p(t_i=1|\pi) \cdot \prod_{j=1}^{784} p(x_j|t_i=1,\theta_{jc})\right]$$

$$= \sum_{i=0}^{9}\left(\pi_i^{t_i} \cdot \prod_{j=1}^{784} \theta_{jt_i}^{x_j}(1-\theta_{jt_i})^{1-x_j}\right)$$

Then $\log p(x|\theta,\pi) = \log\left(\sum_{i=0}^{9} \pi_i^{t_i} \cdot \prod_{j=1}^{784} \theta_{jt_i}^{x_j}(1-\theta_{jt_i})^{1-x_j}\right)$

$$\therefore \text{Ans} = \sum_{j=0}^{9} t_j \cdot \log(\pi_j) + \sum_{j=1}^{784} x_j^{(i)} \log(\theta_{jc}) + (1-x_j^{(i)}) \log(1-\theta_{jc})$$

$$- \log\Big(\underbrace{\sum_{i=0}^{9} \pi_i^{t_i} \prod_{j=1}^{784} \theta_{jt_i}^{x_j}(1-\theta_{jt_i})^{1-x_j}}_{= e^{\log(\pi_i^{t_i} \prod_{j=1}^{784} \theta_{jt_i}^{x_j}(1-\theta_{jt_i})^{1-x_j})}}\Big)$$

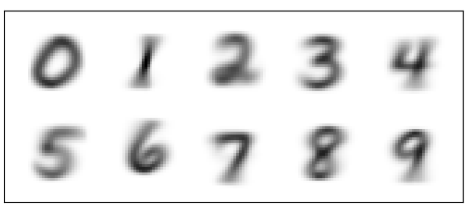
$$= e^{t_i \log(\pi_i) + \sum_{j=1}^{784} x_j \log \theta_{jt_i} + (1-x_j)\log(1-\theta_{jt_i})}$$

c)

```
Average log-likelihood for MLE is  nan
```

I think this is due to some classes having a likelihood of 0, causing us to take log(0), which results in nan. Hence, the avg. log-likelihood would also be nan.

d) For MLE estimator:



e) β-distribution:

$$p(\theta; a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \theta^{a-1}(1-\theta)^{b-1}$$

then $p(\theta_{jc}; 3, 3) \propto \theta_{jc}^2 (1-\theta_{jc})^2$

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_{\text{MAP}} &= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathcal{D}) \\
&= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}) p(\mathcal{D} \mid \boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) + \log p(\mathcal{D} \mid \boldsymbol{\theta})
\end{aligned}$$

$$\Rightarrow \hat{\theta}_{MAP} \propto \arg\max_{\theta} \underbrace{\log\left(\theta_{jc}^{2}(1-\theta_{jc})^{2}\right)}_{= 2\log\theta_{jc} + 2\log(1-\theta_{jc})} + \underbrace{\log\left(p(c|\pi)\prod_{j=1}^{784} p(x_j|c,\theta_{jc})\right)}_{= \log\left[\prod_{i=1}^{N} p(c^{(i)}|\pi)\cdot\prod_{j=1}^{784} p(x_j^{(i)}|c,\theta_{jc})\right]}$$

$$= \sum_{i=1}^{N} \log \pi_{c^{(i)}} + \sum_{j=1}^{784} x_j \log(\theta_{jc}) + (1-x_j)\log(1-\theta_{jc})$$

↳ Let the above eqn be denoted by ℓ, then:

$$\frac{d\ell}{d\theta} = 0 \Rightarrow \frac{2}{\theta_{jc}} - \frac{2}{1-\theta_{jc}} + \underbrace{\sum_{i=1}^{N} t_c^{(i)} \cdot \left(\frac{x_j^{(i)}}{\theta_{jc}} + \frac{1-x_j^{(i)}}{1-\theta_{jc}}(-1)\right)}_{\text{from part a)}} = 0$$

$$\Rightarrow \frac{2}{\theta_{jc}} + \sum_{i=1}^{N} t_c^{(i)} \frac{x_j^{(i)}}{\theta_{jc}} = \frac{2}{1-\theta_{jc}} + \sum_{i=1}^{N} t_c^{(i)} \frac{1-x_j^{(i)}}{1-\theta_{jc}}$$

$$\Rightarrow \frac{2(1-\theta_{jc})}{\theta_{jc}} + \sum_{i=1}^{N} \frac{t_c^{(i)} x_j^{(i)} (1-\theta_{jc})}{\theta_{jc}} = 2 + \sum_{i=1}^{N} t_c^{(i)} (1-x_j^{(i)})$$

$$\Rightarrow \frac{2}{\theta_{jc}} - 2 + \sum_{i=1}^{N} \frac{t_c^{(i)} x_j^{(i)}}{\theta_{jc}} - t_c^{(i)} x_j^{(i)} = 2 + \sum_{i=1}^{N} t_c^{(i)} (1-x_j^{(i)})$$

$$\Rightarrow \frac{2}{\theta_{jc}} + \sum_{i=1}^{N} \frac{t_c^{(i)} x_j^{(i)}}{\theta_{jc}} = 4 + \underbrace{\sum_{i=1}^{N} t_c^{(i)} (1-x_j^{(i)}) + t_c^{(i)} x_j^{(i)}}_{= \sum_{i=1}^{N} t_c^{(i)}}$$

$$\Rightarrow \hat{\theta}_{MAP} = \frac{2 + \sum_{i=1}^{N} t_c^{(i)} x_j^{(i)}}{4 + \sum_{i=1}^{N} t_c^{(i)}}$$

Notice that for $\hat{\theta}_{MLE} = \frac{A}{B}$, our $\hat{\theta}_{MAP} = \frac{A+2}{B+4}$

$$= \frac{A+a-1}{B+a+b-2}$$

Formula

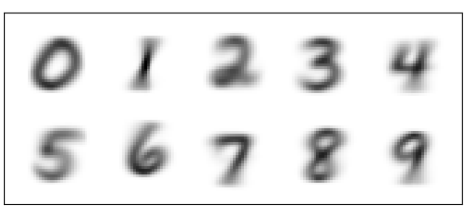
$\hat{\theta}_{\mathrm{ML}}$	$\frac{N_H}{N_H+N_T}$
$\mathbb{E}[\theta\|\mathcal{D}]$	$\frac{N_H+a}{N_H+N_T+a+b}$
$\hat{\theta}_{\mathrm{MAP}}$	$\frac{N_H+a-1}{N_H+N_T+a+b-2}$

$\hat{\pi}_{MAP}$ is the same as $\hat{\pi}_{MLE}$, that is: $\hat{\pi}_{MAP} = \frac{\sum_{i=1}^{N} t_j^{(i)}}{N}$

f)

```
Average log-likelihood for MAP is  -3.357063137860285
Training accuracy for MAP is  0.8352166666666667
Test accuracy for MAP is  0.816
```

g) MAP estimator:



h) Naive Bayes assumes that features are indep. of each other, which, is the case if the features are selected well enough.

However in this case, the pixels in a picture may fall in a pattern, that is, 2 pixels may be dependent, i.e.

$p(x_i = 1 \mid x_j = 1) \neq p(x_i = 1)$

3. a) $p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}$

Formula:

$$p(\mathbf{x}|\boldsymbol{\theta}) = \prod_{k=1}^{K} \theta_k^{x_k}.$$

$= \prod_{i=1}^{N} \prod_{k=1}^{K} \theta_k^{x_k^{(i)}}$ $\quad = \theta_1^{\alpha_1 - 1} \cdots \theta_K^{\alpha_K - 1}$

$\propto p(D|\theta)\,p(\theta)$

$\propto \left(\prod_{i=1}^{N} \prod_{k=1}^{K} \theta_k^{x_k^{(i)}} \right) \left(\theta_1^{\alpha_1 - 1} \cdots \theta_K^{\alpha_K - 1} \right)$

$= \prod_{i=1}^{N} \left(\theta_1^{x_1^{(i)}} \cdots \theta_K^{x_K^{(i)}} \right)$

$= \theta_1^{x_1^{(1)} + \ldots + x_1^{(N)}} \cdots \theta_K^{x_K^{(1)} + \ldots + x_K^{(N)}}$

($= N_1$, $= N_K$)

$= \theta_1^{\left(\sum_{i=1}^{N} x_1^{(i)}\right) + \alpha_1 - 1} \cdots \theta_K^{\left(\sum_{i=1}^{N} x_K^{(i)}\right) + \alpha_K - 1}$

Then let $b_j = \sum_{i=1}^{N} x_j^{(i)} + \alpha_j$:

$= \theta_1^{b_1 - 1} \cdots \theta_K^{b_K - 1}$

↳ This is also a Dirichlet distribution

∴ Prior & Likihood have the same functional form

⇒ Dirichlet distribution is a conjugate prior

b)

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathcal{D})$$
$$= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}) p(\mathcal{D} \mid \boldsymbol{\theta})$$
$$= \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) + \log p(\mathcal{D} \mid \boldsymbol{\theta})$$

$$\Rightarrow \hat{\theta}_{MAP} = \arg\max_{\theta} \theta_1^{b_1-1} \dots \theta_K^{b_K-1}$$

$$= \arg\max_{\theta} \theta_1^{N_1+a_1-1} \dots \theta_K^{N_K+a_K-1}$$

$$= \arg\max_{\theta} \sum_{k=1}^{K} \underbrace{\log(\theta_k^{N_k+a_k-1})}_{= (N_k+a_k-1)\log(\theta_k)}$$

$$= \arg\max_{\theta} \left(\sum_{k=2}^{K} (N_k+a_k-1)\log(\theta_k) \right) + (N_1+a_1-1)\log(\underbrace{\theta_1}_{= 1-\sum_{k=2}^{N}\theta_k})$$

Let f denote the above function & set $\frac{df}{d\theta_k} = 0$:

$$\Rightarrow \frac{N_k+a_k-1}{\theta_k} + \frac{N_1+a_1-1}{1-\sum_{k=2}^{K}\theta_k}(-1) = 0$$

$$\Rightarrow \frac{N_k+a_k-1}{\theta_k} = \frac{N_1+a_1-1}{\theta_1}$$

$$\Rightarrow \frac{\hat{\theta}_k}{\hat{\theta}_1} = \frac{N_k+a_k-1}{N_1+a_1-1}$$

Then $\sum_{k=1}^{K} \frac{\hat{\theta}_k}{\hat{\theta}_1} = \frac{1}{\hat{\theta}_1} = \sum_{k=1}^{K} \frac{N_k+a_k-1}{N_1+a_1-1}$

$$\therefore \hat{\theta}_1 = \frac{N_1 + \alpha_1 - 1}{\sum_{k=1}^{K} N_k + \alpha_k - 1}$$

Repeat this process by replacing $\hat{\theta}_1$ w/ $\hat{\theta}_j$, then we have:

$$\hat{\theta}_j = \frac{N_j + \alpha_j - 1}{\sum_{k=1}^{K} N_k + \alpha_k - 1}$$

c) $p(\mathbf{x}^{(N+1)}|\mathcal{D}) = \int p(\mathbf{x}^{(N+1)}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D})\, d\boldsymbol{\theta}$

$p(x|\theta) = \prod_{k=1}^{K} \theta_k^{x_k}$, so $p(x^{(N+1)}|\theta) = \prod_{k=1}^{K} \theta_k^{x_k^{(N+1)}}$

$\Rightarrow p(x^{(N+1)} = k|\theta) = p(x_k^{(N+1)} = 1 \mid \theta) = \theta_k$

$\Rightarrow p(x_k^{(N+1)} = 1 \mid D) = \int \theta_k \cdot p(\theta|D)\, d\theta$

$= E[\theta_k|D]$

by 3a), where $\theta_k|D \sim \text{Dirichlet}(b_1, \ldots, b_K)$, for $b_j = N_j + \alpha_j$, so by hint 2:

$$E[\theta_k|D] = \frac{b_k}{\sum_{k'} b_{k'}}$$

Hint 1:

$$p(x^{(N+1)} < K) = \sum_{k=1}^{K-1} p(x^{(N+1)} = k)$$

$$\therefore\ p(x^{(N+1)} < K \mid D) = \sum_{k=1}^{K-1} p(x^{(N+1)} = k \mid D)$$

$$= \sum_{k=1}^{K-1} p(x_k^{(N+1)} = 1 \mid D)$$

$$= \sum_{k=1}^{K-1} b_k \Big/ \sum_{k'} b_{k'}$$

4. a) # Multivariate Mean and Covariance

Mean

$$\boldsymbol{\mu} = \mathbb{E}[\mathbf{x}] = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_d \end{pmatrix}$$

Covariance

$$\boldsymbol{\Sigma} = \mathrm{Cov}(\mathbf{x}) = \mathbb{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top] = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1D} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{D1} & \sigma_{D2} & \cdots & \sigma_D^2 \end{pmatrix}$$

how different dimensions interact.

4. **[5pts] Gaussian Discriminant Analysis.** For this question you will build classifiers to label images of handwritten digits. Each image is 8 by 8 pixels and is represented as a vector of dimension 64 by listing all the pixel values in raster scan order. The images are grayscale and the pixel values are between 0 and 1. The labels y are $0, 1, 2, \ldots, 9$ corresponding to which character was written in the image. There are 700 training cases and 400 test cases for each digit; they can be found in the `data` directory in the starter code.

each image: $\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{64} \end{pmatrix}$ for $x_i \in \{0,1\}$

A skeleton (`q4.py`) is is provided for each question that you should use to structure your code. Starter code to help you load the data is provided (`data.py`). Note: the `get_digits_by_label` function in `data.py` returns the subset of digits that belong to a given class.

Using maximum likelihood, fit a set of 10 class-conditional Gaussians with a separate, full covariance matrix for each class. Remember that the conditional multivariate Gaussian probability density is given by,

$$p(\mathbf{x} \mid y = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_\mathbf{k}) = (2\pi)^{-d/2} |\boldsymbol{\Sigma}_\mathbf{k}|^{-1/2} \exp\left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_\mathbf{k}^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right\} \tag{2}$$

log " $= -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log(|\Sigma_k|) + \left(-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right)$

You should take $p(y = k) = \frac{1}{10}$. You will compute parameters μ_{kj} and Σ_k for $k \in (0...9), j \in (1...64)$. You should implement the covariance computation yourself (i.e. without the aid of `np.cov`). *Hint: To ensure numerical stability you may have to add a small multiple of the identity to each covariance matrix. For this assignment you should add* $0.01\mathbf{I}$ *to each matrix.*

a)

```
train avg conditional log-likelihood: -0.12462443666862937
test avg conditional log-likelihood: -0.19667320325525464
```

b)

```
train accuracy: 0.9814285714285714
test accuracy: 0.97275
```

c) For diagonal covariance matrices:

```
train avg conditional log-likelihood (diagonal covariances): -1.230765422272791
test avg conditional log-likelihood (diagonal covariances): -1.287260366755839
train accuracy (diag): 0.85
test accuracy (diag): 0.84
```

The avg. log-likelihoods are lower if the covariance matrices are diagonal, indicating that the model is less certain to assign to the correct class label.

As a result, we get comparatively lower accuracies on both train & test data.