Contribution Report

Discussion and Problem Solving for Questions 1, 2, and 3:
Equal Contribution

Solutions Write-up for Questions 1,2, and 3:
David Wu

Proof-reading for Questions 1,2, and 3:
Shu Wang

HW 4

1.a. Note that $p(x|\theta) = \prod_{j=1}^{k-1} \theta_j^{x_j} \cdot (1 - \sum_{j=1}^{k-1} \theta_j)^{x_j}$ (from hint 2)

$L(\theta) = p(x^{(1)} \dots x^{(n)} | \theta) = \prod_{i=1}^{n} p(x^{(i)}|\theta)$. Then:

$$\ell(\theta) = \sum_{i=1}^{n} \left[\sum_{j=1}^{k-1} x_j^{(i)} \log(\theta_i) + x_k^{(i)} \log(1 - \sum_{j=1}^{k-1} \theta_j) \right]$$

For $\alpha < K$, $\frac{\partial \ell}{d\theta_\alpha} = \sum_{i=1}^{n} \left[\frac{x_\alpha^{(i)}}{\theta_\alpha} + x_k^{(i)} \frac{-1}{1 - \sum \theta_j} \right] = \frac{N_\alpha}{\theta_\alpha} - \frac{N_k}{1 - \sum_{j=1}^{k-1} \theta_j}$

Setting the derivative to 0 and rearranging gives:

$\frac{\theta_\alpha}{N_\alpha} = \frac{1 - \sum \theta_j}{N_K} = -\frac{\theta_\alpha}{N_K} + \frac{1 - \sum_{j \neq \alpha, k} \theta_j}{N_K}$, so $\frac{N_\alpha + N_k}{N_\alpha N_k} \hat{\theta}_\alpha = \frac{1}{N_k} \left[1 - \sum_{j \neq \alpha, k} \theta_j \right]$.

This means that $\hat{\theta}_\alpha = \frac{N_\alpha}{N_\alpha + N_k} \left[1 - \sum_{j \neq \alpha, k} \theta_j \right]$ for all $1 \leq \alpha \leq k-1$. (✲)

Claim: $\hat{\theta}_\alpha = \frac{N_\alpha}{N}$ solves the above equation

Indeed: $1 - \sum_{j \neq \alpha, k} \hat{\theta}_j = 1 - \sum_{j \neq \alpha, k} \frac{N_j}{N} = \frac{1}{N} \left[N - \sum_{j \neq \alpha, k} N_j \right] = \frac{N_\alpha + N_k}{N}$. (1)

Substituting (1) into (✲) gives:

1. a. $\hat{\theta}_\alpha = \frac{N_\alpha}{N_\alpha + N_K} \left[\frac{N_\alpha + N_K}{N} \right] = \frac{N_\alpha}{N}$

Thus $\boxed{\hat{\theta}_K = \frac{N_K}{N}}$ is a solution. From hint 2, ℓ is concave, so the critical point corresponds to a global max and $\hat{\theta}_K$ is the MLE.

b $p(\theta | D) \propto p(\theta)\, p(D | \theta)$ [from lecture]

$\propto [\theta_1^{a_1 - 1} \cdots \theta_K^{a_K - 1}] \cdot \prod_j \theta_j^{N_j}$ [see appendix A (next page)]

$= \theta_1^{N_1 + a_1 - 1} \cdots \theta_K^{N_K + a_K - 1}$

The probability density is a Dirichlet distribution defined over θ and with parameters $N_1 + a_1 - 1, \ldots N_K + a_K - 1$

Appendix ☆ : Proof $p(D|\theta) = \prod_j \theta_j^{N_j}$

$$p(D|\theta) = \prod_{i=1}^{N} p(x|\theta)$$

$$= \prod_{i=1}^{N} \prod_{j=1}^{K} \theta_j^{x_j^{(i)}} = \prod_{j=1}^{K} \prod_{i=1}^{N} \theta_j^{x_j^{(i)}} = \prod_{j=1}^{K} \theta_j^{\Sigma x_j^{(i)}} = \prod_{j=1}^{K} \theta_j^{N_j}$$

c. Note that $\theta_k = 1 - \sum_j^{k-1} \theta_j$, so $p(\theta | D) \propto \prod_{i=1}^{k-1} \theta_i^{N_i + a_i - 1} \cdot \left(1 - \sum_j^{k-1} \theta_j\right)^{N_k + a_k - 1}$

$$\log p(\theta | D) = \text{Const} + \sum_{i=1}^{k-1} (N_i + a_i - 1) \log(\theta_i) + (N_k + a_k - 1)\left(1 - \sum_j^{k-1} \theta_j\right)$$

$$\frac{\partial}{\partial \theta_\alpha} p(\theta | D) = \frac{N_\alpha + a_\alpha - 1}{\theta_\alpha} + \frac{N_k + \alpha_k - 1}{1 - \sum \theta_k}$$

This has the same form as part a, except N_α is replaced with

$\bar{N}_\alpha := N_\alpha + a_\alpha - 1$. Similarly $\bar{N} := \sum_i \bar{N}_i = \sum_i (N_i + a_i - 1) = N + \sum_i a_i - k$

Substituting these values into part a:

$$\boxed{\hat{\theta}_{i,map} = (N_i + a_i - 1) \Big/ \left(N + \sum_{j=1}^{k} a_j - k\right).}$$

d. $P(x_j^{(N+1)} = 1) = \theta_j$ and $p(\theta | D)$ follows Dirchlet $(N_1 + a_1 - 1 \ldots N_k + a_k - 1)$

$$p(x_j^{(N+1)} | D) = \int P(x_j^{(N+1)} = 1 | D)\, p(\theta | D)\, d\theta = \int \theta_j\, p(\theta | D)\, d\theta = E(\theta_j)$$

Using $\bar{N}_\alpha$, $\bar{N}$ from part c, $\boxed{p(x_j^{(N+1)} = 1) = \dfrac{N_j + a_j - 1}{N + \sum_{j=1}^{k} a_j - k}}$

2.a. Using the law of total probability:

$$p(x) = \sum_k p(t=k)\, p(x \mid t=k)$$

$$= \sum_k \alpha_k \left(\prod_{d=1}^{D} 2\pi \sigma_d^2 \right)^{-\frac{1}{2}} \exp\left\{ - \sum_{d=1}^{D} \frac{1}{2\sigma_d^2} (x_d - \mu_{kd})^2 \right\}$$

$$= \left(\prod_{d=1}^{D} 2\pi \sigma_d^2 \right)^{-\frac{1}{2}} \left[\sum_k \alpha_k \exp\left\{ - \sum_{d=1}^{D} \frac{1}{2\sigma_d^2} (x_d - \mu_{kd})^2 \right\} \right]$$

From Baye's Rule:

$$p(t=k \mid x) = \frac{p(t=k)\, p(x \mid t=k)}{p(x)}$$

$$= \frac{\alpha_k \cancel{\left(\prod_{d=1}^{D} 2\pi \sigma_d^2 \right)^{-\frac{1}{2}}} \exp\left\{ - \sum_{d=1}^{D} \frac{1}{2\sigma_d^2} (x_d - \mu_{kd})^2 \right\}}{\cancel{\left(\prod_{d=1}^{D} 2\pi \sigma^2 \right)^{-\frac{1}{2}}} \left[\sum_k \alpha_k \exp\left\{ - \sum_{d=1}^{D} \frac{1}{2\sigma_d^2} (x_d - \mu_{kd})^2 \right\} \right]}$$

$$= \frac{\alpha_k \exp\left\{ - \sum_{d=1}^{D} \frac{1}{2\sigma_d^2} (x_d - \mu_{kd})^2 \right\}}{\sum_{k=1}^{K} \alpha_k \exp\left\{ - \sum_{d=1}^{D} \frac{1}{2\sigma_d^2} (x_d - \mu_{kd})^2 \right\}}$$

2b.

$$l(\theta) = \log \prod_{i=1}^{N} p(t^{(i)}, x^{(i)})$$

$$= \sum_{i=1}^{N} \log p(x^{(i)} \mid t^{(i)})\, p(t^{(i)})$$

$$= \sum_{i=1}^{N} \Big[\log \alpha_{t^{(i)}} + \log \underbrace{\Big(\prod_{d=1}^{D} 2\pi\sigma_d^2 \Big)^{-\frac{1}{2}}} \underbrace{\exp\Big\{ - \sum_{d=1}^{D} \frac{1}{2\sigma_d^2} \big(x_d^{(i)} - \mu_{t^{(i)}d}\big)^2}\, \Big]$$

$$= \Big(\sum_{i=1}^{N} \log \alpha_{t^{(i)}} \Big) + \sum_{i=1}^{N} \Big[-\frac{1}{2} \sum_{d=1}^{D} \big[\log 2\pi + \log \sigma_d^2 \big] - \sum_{d=1}^{D} \frac{1}{2\sigma_d^2} \big(x_d^{(i)} - \mu_{t^{(i)}d}\big)^2 \Big]$$

$$= \Big(\sum_{i=1}^{N} \log \alpha_{t^{(i)}} \Big) + \big(-\tfrac{1}{2}\big) ND \log 2\pi - \frac{N}{2} \sum_{d=1}^{D} \log \sigma_d^2 - \sum_{i=1}^{N} \sum_{d=1}^{D} \frac{1}{2\sigma_d^2} \big(x_d^{(i)} - \mu_{t^{(i)}d}\big)^2$$

2.c. From part b,

$$\ell(\theta) = \text{Const.} + \left(\sum_{i=1}^{N} \log \alpha_{t^{(i)}}\right) - \frac{N}{2}\sum_{d=1}^{D} \log \sigma_d^2 - \sum_{i=1}^{N}\sum_{d=1}^{D} \frac{1}{2\sigma_d^2}\left(x_d^{(i)} - \mu_{t^{(i)}d}\right)^2$$

$$\frac{\partial \ell}{\partial \mu_{kd}} = -\sum_{i=1}^{N_k} \frac{1}{2\sigma_d^2}\, 2\left(\tilde{x}_d^{(i)} - \mu_{kd}\right)(-1)$$

Here we sum over the $N_k \geq 1$ data points in class k, and $\tilde{x}^{(i)}$ is an observation in class k.

Setting $\frac{\partial \ell}{\partial \mu_{kd}} = 0$, gives:

$$\sum_{i=1}^{N_k} \frac{1}{\sigma_d^2}\left(\tilde{x}_d^{(i)} - \mu_{kd}\right) = 0 \quad \Rightarrow \quad \sum_{i=1}^{N_k} \tilde{x}_d^{(i)} - \mu_{kd} = 0 \text{, so}$$

$$\hat{\mu}_{kd} = \frac{1}{N_k}\sum_{i=1}^{N_k} \tilde{x}_d^{(i)} \text{, so} \quad \boxed{\hat{\mu}_{kd} = \frac{1}{N_k}\sum_{i=1}^{N} I\left(t^{(i)} = k\right) x_d^{(i)}}$$

$$\frac{\partial \ell}{\partial \sigma_d^2} = -\frac{N}{2}\frac{1}{\sigma_d^2} + \sum_{i=1}^{N} \frac{1}{2}\left(\sigma_d^2\right)^{-2}\left(x_d^{(i)} - \mu_{t^{(i)}d}\right)^2$$

Setting to 0 gives:

$$-\frac{N}{2} + \sum_{i=1}^{N} \frac{1}{2\sigma_d^2}\left(x_d^{(i)} - \mu_{t^{(i)}d}\right)^2 = 0$$

$$\frac{1}{\sigma_d^2}\sum_{i=1}^{N}\left(x_d^{(i)} - \mu_{t^{(i)}d}\right)^2 = N \text{, so}$$

$$\boxed{\hat{\sigma}_d^2 = \frac{1}{N}\sum_{i=1}^{N}\left(x_d^{(i)} - \hat{\mu}_{t^{(i)}d}\right)^2}$$

Question 3

a.

```
def compute_mean_mles(train_data, train_labels):
    # Initialize array to store means
    means = np.zeros((10, 64))
    # == YOUR CODE GOES HERE ==

    for k in range(10):
        indicator_k = np.where(train_labels == k, 1, 0)
        N_k = np.dot(indicator_k.T, indicator_k)
        sum = np.dot(train_data.T, indicator_k)

        means[k, :] = sum/N_k

    # ====
    return means


def compute_sigma_mles(train_data, train_labels):

    covariances = np.zeros((10, 64, 64))
    # == YOUR CODE GOES HERE ==
    means = compute_mean_mles(train_data, train_labels)

    for k in range(10):
        indicator_k = np.where(train_labels == k, 1, 0)
        N_k = np.dot(indicator_k.T, indicator_k)

        temp = train_data - means[k,:]
        #Set values not is class k to 0
        temp = np.where(train_labels == k, temp.T, 0).T

        #temp is N x 64
        covariances[k,:,:] = np.dot(temp.T, temp)/N_k + 0.01*np.identity(64)

    # ====
    return covariances
```

```python
def generative_likelihood(digits, means, covariances):

    N = digits.shape[0]
    likelihoods = np.zeros((N, 10))
    # == YOUR CODE GOES HERE ==

    for k in range(10):
        normalizer = -32 * np.log(2*np.pi) - 0.5 * np.log(np.linalg.det(covariances[k,:,:]))

        temp = -1/2*np.dot(digits - means[k,:], np.linalg.inv(covariances[k,:,:]))
        temp = np.dot(temp, (digits - means[k,:]).T)

        likelihoods[:, k] = normalizer + np.diag(temp)

    # ====
    return likelihoods


def conditional_likelihood(digits, means, covariances):

    p_x_given_t = np.exp(generative_likelihood(digits, means, covariances))
    p_t = np.full((10, 1), 1/10)

    #log p(t|x) = log p(x|t) + log p(t) - log p(x)
    # N x 1 matrix with p(x)
    p_x = np.dot(p_x_given_t, p_t)

    likelihoods = np.log(p_x_given_t) + np.log(p_t.T) - np.log(p_x)

    return likelihoods
    # ====

def classify_data(digits, means, covariances):

    # == YOUR CODE GOES HERE ==
    p_t_given_x = conditional_likelihood(digits, means, covariances)

    pred = np.argmax(p_t_given_x, axis = 1)
    # ====
    return pred
```

b. Running the code, we get the following output:

```
Train average conditional log-likelihood:  -0.12462443666862984
Test average conditional log-likelihood:  -0.19667320325525503
Train posterior accuracy:  0.9814285714285714
Test posterior accuracy:  0.97275
```