1 Mixture Models

(a)

$$
\begin{aligned}
p(x_1, x_2, \dots x_N | \{\lambda_k\}, \{\pi_k\}) &= \prod_{i=1}^{N} p(x_i | \{\lambda_k\}, \{\pi_k\}) \\
&= \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k p(x_i | \lambda_k) \\
&= \prod_{i=1}^{N} \sum_{k=1}^{K} \left(\pi_k \frac{1}{x_i!} \lambda_k^{x_i} exp(-\lambda_k) \right)
\end{aligned}
$$

(b)

$$
\begin{aligned}
\log p(x_1, x_2, \dots x_N | \{\lambda_k\}, \{\pi_k\}) &= \log \prod_{i=1}^{N} \sum_{k=1}^{K} \left(\pi_k \frac{1}{x_i!} \lambda_k^{x_i} exp(-\lambda_k) \right) \\
&= \sum_{i=1}^{N} \log \sum_{k=1}^{K} \left(\pi_k \frac{1}{x_i!} \lambda_k^{x_i} exp(-\lambda_k) \right) \\
&= \sum_{i=1}^{N} \log \frac{1}{x_i!} \sum_{k=1}^{K} (\pi_k \lambda_k^{x_i} exp(-\lambda_k)) \\
&= \sum_{i=1}^{N} \left(\log \left(\sum_{k=1}^{K} (\pi_k \lambda_k^{x_i} exp(-\lambda_k)) \right) - \log x_i! \right)
\end{aligned}
$$

(c)

$$
\begin{aligned}
r_{nk} &= p(z_{nk} = 1 | x_n, \{\lambda_k\}, \{\pi_k\}) \\
&= \frac{p(z_{nk} = 1) p(x_n | z_{nk} = 1, \lambda_k)}{\sum_{j=1}^{N} p(z_{nj} = 1) p(x_n | z_{nj} = 1, \{\lambda_j\})} \\
&= \frac{\pi_k \cancel{\frac{1}{x_n!}} \lambda_k^{x_n} exp(-\lambda_k)}{\sum_{j=1}^{N} \pi_j \cancel{\frac{1}{x_n!}} \lambda_j^{x_n} exp(-\lambda_j)} \\
&= \frac{\pi_k \lambda_k^{x_n} exp(-\lambda_k)}{\sum_{j=1}^{N} \pi_j \lambda_j^{x_n} exp(-\lambda_j)}
\end{aligned}
$$

(d)

Note: N_k is the number of datapoints belonging to cluster k.

$$
\begin{aligned}
\frac{\partial \log p(x_1, x_2, \dots x_N | \{\lambda_k\}, \{\pi_k\})}{\partial \lambda_k} &= \frac{\partial \left[\sum_{i=1}^{N} \left(\log \left(\sum_{j=1}^{K} (\pi_j \lambda_j^{x_i} exp(-\lambda_j)) \right) - \log x_i! \right) \right]}{\partial \lambda_k} \\
&= \sum_{i=1}^{N} \frac{\left(\pi_k x_i \lambda_k^{x_i - 1} exp(-\lambda_k) \right) + \left(- \left(\pi_k \lambda_k^{x_i} exp(-\lambda_k) \right) \right)}{\sum_{j=1}^{K} \left(\pi_j \lambda_j^{x_i} exp(-\lambda_j) \right)} \\
&= \sum_{i=1}^{N} \frac{(\pi_k \lambda_k^{x_i} exp(-\lambda_k)) (\lambda_k^{-1} x_i - 1)}{\sum_{j=1}^{K} \left(\pi_j \lambda_j^{x_i} exp(-\lambda_j) \right)} \\
&= \sum_{i=1}^{N} r_{ik} (x_i \lambda_k^{-1} - 1) = 0
\end{aligned}
$$

$$\sum_{i=1}^{N} r_{ik} x_i \lambda_k^{-1} - \sum_{i=1}^{N} r_{ik} = 0$$

$$\lambda_k = \frac{\sum_{i=1}^{N} r_{ik} x_i}{\sum_{i=1}^{N} r_{ik}} = \frac{\sum_{i=1}^{N} r_{ik} x_i}{N_k}$$

(e)

Note: Because $\sum_{k=1}^{N} N_k$ means summing over number of datapoints belonging to every cluster, $\sum_{k=1}^{N} N_k = N$ where N is the total number of datapoints. First, we shall build Lagrangian:

$$\begin{aligned} L(x_1, \dots, x_N, \{\lambda_k\}, \{\pi_k\}, \alpha) &= \log p(x_1, \dots, x_N | \{\lambda_k\}, \{\pi_k\}) + \alpha(\sum_{j=1}^{K} \pi_j - 1) \\ &= \sum_{i=1}^{N} \left(\log \left(\sum_{j=1}^{K} \left(\pi_j \lambda_j^{x_i} exp(-\lambda_j) \right) \right) - \log x_i! \right) + \alpha(\sum_{j=1}^{K} \pi_j - 1) \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial \pi_k} L(x_1, \dots, x_N, \{\lambda_k\}, \{\pi_k\}, \alpha) &= \frac{\partial}{\partial \pi_k} \left(\sum_{i=1}^{N} \left(\log \left(\sum_{j=1}^{K} \left(\pi_j \lambda_j^{x_i} exp(-\lambda_j) \right) \right) - \log x_i! \right) + \alpha(\sum_{j=1}^{K} \pi_j - 1) \right) \\ &= \sum_{i=1}^{N} \frac{\lambda_k^{x_i} exp(-\lambda_k)}{\sum_{j=1}^{K} \left(\pi_j \lambda_j^{x_i} exp(-\lambda_j) \right)} + \alpha = 0 \end{aligned}$$

Because left part is zero, we can multiply both parts by π_k:

$$\sum_{i=1}^{N} \frac{\lambda_k^{x_i} exp(-\lambda_k)}{\sum_{j=1}^{K} \left(\pi_j \lambda_j^{x_i} exp(-\lambda_j) \right)} \pi_k + \alpha \pi_k = 0$$

$$\sum_{i=1}^{N} r_{ik} + \alpha \pi_k = 0$$

$$N_k + \alpha \pi_k = 0$$

This doesn't give an answer for every k, however, because this should be true for every $\pi_k, k \in (1, \dots K)$, we can sum over all equalities and it gets us the following equation:

$$\begin{aligned} \sum_{k=1}^{K} N_k + \sum_{k=1}^{K} \alpha \pi_k &= 0 \\ N + \alpha \sum_{k=1}^{K} c \pi_k &= \{\sum_{k=1}^{K} \pi_k = 1\} \\ &= N + \alpha = 0 \\ \alpha &= -N \\ &\Downarrow \\ N_k + (-N) \pi_k &= 0 \\ \pi_k &= \frac{N_k}{N} \end{aligned}$$

(f)

$$\begin{aligned}
\log p(\mathbf{x}, \{\pi_k\}, \{\lambda_k\}|a, b, \alpha, K) &= log\left[p(\mathbf{x}|\{\pi_k\}, \{\lambda_k\})p(\{\pi_k\}|\alpha, K)p(\{\lambda_k\}|a, b)\right] \\
&= log\left[\left(\prod_{n=1}^{N}\sum_{k=1}^{K}\frac{\pi_k}{x_n!}\lambda_k^{x_n}exp(-\lambda_k)\right)\left(\frac{\Gamma(\cancel{K}\cdot\frac{\alpha}{\cancel{K}})}{\Gamma(\frac{\alpha}{K})^K}\prod_{k=1}^{K}\pi_k^{\frac{\alpha}{K}-1}\right)\left(\prod_{k=1}^{K}\frac{b^a}{\Gamma(a)}\lambda_k^{a-1}exp(-b\lambda_k)\right)\right] \\
&= \sum_{n=1}^{N}\left[log\sum_{k=1}^{K}(\pi_k\lambda_k^{x_n}exp(-\lambda_k)) - log(x_n!)\right] + log\Gamma(\alpha) - Klog\Gamma(\frac{\alpha}{K}) + \sum_{k=1}^{K}(\frac{\alpha}{K}-1)log(\pi_k) \\
&+ \sum_{k=1}^{K}\left[a\cdot logb - log\Gamma(a) + (a-1)log\lambda_k - b\lambda_k\right] = \sum_{n=1}^{N}log\sum_{k=1}^{K}\pi_k\lambda_k^{x_n}exp(-\lambda_k) \\
&+ (\frac{\alpha}{K}-1)\sum_{k=1}^{K}\left[log(\pi_k) + (a-1)log\lambda_k - b\lambda_k\right] + C,
\end{aligned}$$

where $C = \sum_{n=1}^{N} -(log(x_n!)) + log\Gamma(\alpha) - Klog\Gamma(\frac{\alpha}{K}) + K\left[a\cdot logb - log\Gamma(a)\right]$

(g)

$$\begin{aligned}
\frac{\partial\log p(\mathbf{x}, \{\pi_k\}, \{\lambda_k\}|a, b, \alpha, K)}{\partial\lambda_k} &= \frac{\partial\left[\sum_{n=1}^{N}log\sum_{k=1}^{K}\pi_k\lambda_k^{x_n}exp(-\lambda_k) + (\frac{\alpha}{K}-1)\sum_{k=1}^{K}log(\pi_k) + (a-1)log\lambda_k - b\lambda_k + C\right]}{\partial\lambda_k} \\
&= \sum_{n=1}^{N}r_{nk}(\frac{x_n}{\lambda_k}-1) + \frac{a-1}{\lambda_k} - b \\
&= \frac{1}{\lambda_k}\sum_{n=1}^{N}r_{nk}x_n - \sum_{n=1}^{N}r_{nk} + \frac{a-1}{\lambda_k} - b \\
&= \frac{1}{\lambda_k}((\sum_{n=1}^{N}r_{nk}x_n) + a - 1) - N_k - b = 0 \\
\lambda_k &= \frac{(\sum_{n=1}^{N}r_{nk}x_n) + a - 1}{N_k + b}
\end{aligned}$$

(h)

First, we shall build Lagrangian:

$$\begin{aligned}
L(x_1, \dots, x_N, \{\lambda_k\}, \{\pi_k\}, \mu) &= \log p(\mathbf{x}, \{\pi_k\}, \{\lambda_k\}|a, b, \alpha, K) + \mu(\sum_{j=1}^{K}\pi_j - 1) \\
&= \sum_{n=1}^{N}log\sum_{k=1}^{K}\pi_k\lambda_k^{x_n}exp(-\lambda_k) + (\frac{\alpha}{K}-1)\sum_{k=1}^{K}log(\pi_k) + (a-1)log\lambda_k - b\lambda_k + C + \mu(\sum_{j=1}^{K}\pi_j - 1)
\end{aligned}$$

$$\frac{\partial L(x_1, \dots, x_N, \{\lambda_k\}, \{\pi_k\}, \mu)}{\partial\pi_k} = \sum_{n=1}^{N}\frac{\lambda_k^{x_n}exp(-\lambda_k)}{\sum_{j=1}^{K}\pi_j\lambda_j^{x_n}exp(-\lambda_j)} + \frac{(\frac{\alpha}{K}-1)}{\pi_k} + \mu = 0$$

Because left part is zero, we can multiply both parts by π_k:

$$\begin{aligned}
\sum_{n=1}^{N}\frac{\lambda_k^{x_n}exp(-\lambda_k)}{\sum_{j=1}^{K}\pi_j\lambda_j^{x_n}exp(-\lambda_j)}\pi_k + (\frac{\alpha}{K}-1) + \mu\pi_k &= 0 \\
\sum_{n=1}^{N}r_{nk} + (\frac{\alpha}{K}-1) + \mu\pi_k &= 0 \\
N_k + (\frac{\alpha}{K}-1) + \mu\pi_k &= 0
\end{aligned}$$

As in the previous example, this doesn't give an answer for every k, however, because this should be true for every $\pi_k, k \in (1, \dots K)$, we can sum over all equalities and it gets us the following equation:

$$\sum_{k=1}^{N} \Big(N_k + \frac{\alpha}{K} - 1 + \mu\pi_k \Big) = 0$$
$$N + \alpha - K + \mu \sum_{k=1}^{N} \pi_k = \{\sum_{k=1}^{N} \pi_k = 1\}$$
$$= N + \alpha - K + \mu = 0$$
$$\mu = -N + K - \alpha$$
$$\Downarrow$$
$$N_k + \pi_k(-N + K - \alpha) + \frac{\alpha}{K} - 1 = 0$$
$$\pi_k = \frac{-(N_k + \frac{\alpha}{K} - 1)}{-N + K - \alpha} = \frac{1 - N_k - \frac{\alpha}{K}}{K - N - \alpha}$$

(i)

Note: for convenience matters, in the following pseudocode, log-likelihood $\log p(x_1, x_2, \dots x_N | \{\lambda_k\}, \{\pi_k\})$ is shortened to $logp$, thus, more formally, $logp = \log p(x_1, x_2, \dots x_N | \{\lambda_k\}, \{\pi_k\})^{(\tau)}$.
For ML solution, we would have following algorithm:

Algorithm 1 *

EM Algorithm for Maximum Likelihood
  Choose threshold ϵ
  $\tau := 0$
  For every $n \in (1, \dots N), k \in (1, \dots K)$ initialize responsibilities: $r_{nk} := \frac{1}{K}$
  Perform M-Step:
    For every $k \in (1, \dots K)$ calculate $\lambda_k := \frac{\sum_{i=1}^{N} r_{ik} x_i}{N_k}$
    For every $k \in (1, \dots K)$ calculate $\pi_k := \frac{N_k}{N}$
  Calculate $logp^{(\tau)} := \sum_{i=1}^{N} \Big(\log \Big(\sum_{k=1}^{K} (\pi_k \lambda_k^{x_i} exp(-\lambda_k)) \Big) - \log x_i! \Big)$
  do
    Perform E-Step:
      For every $n \in (1, \dots N), k \in (1, \dots K)$ calculate responsibilities: $r_{nk} := \frac{\pi_k \lambda_k^{x_n} exp(-\lambda_k)}{\sum_{j=1}^{N} \pi_j \lambda_j^{x_n} exp(-\lambda_j)}$
    Perform M-Step:
      For every $k \in (1, \dots K)$ calculate $\lambda_k := \frac{\sum_{i=1}^{N} r_{ik} x_i}{N_k}$
      For every $k \in (1, \dots K)$ calculate $\pi_k := \frac{N_k}{N}$
    $\tau := \tau + 1$
    Calculate $logp^{(\tau)} := \sum_{i=1}^{N} \Big(\log \Big(\sum_{k=1}^{K} (\pi_k \lambda_k^{x_i} exp(-\lambda_k)) \Big) - \log x_i! \Big)$
  while $|logp^{\tau} - logp^{\tau-1}| > \epsilon$
  return $\{\pi_k\}$, $\{\lambda_k\}$, $\{r_{nk}\}$

Note: for convenience matters, in the following pseudocode, log-likelihood $\log p(\mathbf{x}, \{\pi_k\}, \{\lambda_k\}|a, b, \alpha, K)$ is shortened to $logp$, thus, more formally,
$logp = \log p(\mathbf{x}, \{\pi_k\}, \{\lambda_k\}|a, b, \alpha, K) = \sum_{n=1}^{N} log \sum_{k=1}^{K} \pi_k \lambda_k^{x_n} exp(-\lambda_k) - log(x_n!) + log\Gamma(\alpha) - Klog\Gamma(\frac{\alpha}{K}) + \sum_{k=1}^{K}(\frac{\alpha}{K} - 1)log(\pi_k) + \sum_{k=1}^{K} a \cdot logb - log\Gamma(a) + (a-1)log\lambda_k - b\lambda_k$.

Algorithm 2 *

EM Algorithm for MAP

Choose threshold ϵ
$\tau := 0$
For every $n \in (1, \dots N), k \in (1, \dots K)$ initialize responsibilities: $r_{nk} := \frac{1}{K}$
Perform M-Step:
For every $k \in (1, \dots K)$ calculate $\lambda_k := \frac{(\sum_{n=1}^{N} r_{nk} x_n) + a - 1}{N_k + b}$
For every $k \in (1, \dots K)$ calculate $\pi_k := \frac{1 - N_k - \frac{\alpha}{K}}{K - N - \alpha}$
Calculate $logp^{(\tau)}$
do
Perform E-Step:
For every $n \in (1, \dots N), k \in (1, \dots K)$ calculate responsibilities: $r_{nk} := \frac{\pi_k \lambda_k^{x_n} exp(-\lambda_k)}{\sum_{j=1}^{N} \pi_j \lambda_j^{x_n} exp(-\lambda_j)}$
Perform M-Step:
For every $k \in (1, \dots K)$ calculate $\lambda_k := \frac{(\sum_{n=1}^{N} r_{nk} x_n) + a - 1}{N_k + b}$
For every $k \in (1, \dots K)$ calculate $\pi_k := \frac{1 - N_k - \frac{\alpha}{K}}{K - N - \alpha}$
$\tau := \tau + 1$
Calculate $logp^{(\tau)}$
while $|logp^{\tau} - logp^{\tau - 1}| > \epsilon$
return $\{\pi_k\}$, $\{\lambda_k\}$, $\{r_{nk}\}$

2 PCA

(a)

To center $\mathbf{x}_n$, we would need data mean:

$$\overline{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n$$

$$\hat{\mathbf{x}}_n = \mathbf{x}_n - \overline{\mathbf{x}} = \mathbf{x}_n - \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n$$

(b)

Average of $\hat{\mathbf{x}}_n$ over N vectors is calculated as following:

$$\begin{aligned} E(\hat{\mathbf{X}}) &= \frac{1}{N} \sum_{n=1}^{N} \hat{\mathbf{x}}_n \\ &= \frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}_n - \overline{\mathbf{x}}) \\ &= \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n - \frac{1}{N} \sum_{n=1}^{N} \overline{\mathbf{x}} \\ &= \overline{\mathbf{x}} - \frac{1}{N} N \cdot \overline{\mathbf{x}} = \overline{\mathbf{x}} - \overline{\mathbf{x}} = \mathbf{0} \end{aligned}$$

(c)

$$S = Cov(\hat{\mathbf{X}}, \hat{\mathbf{X}}) = E(\hat{\mathbf{X}}\hat{\mathbf{X}}^T) - E(\hat{\mathbf{X}})E(\hat{\mathbf{X}}^T) = \{E(\hat{\mathbf{X}}) = \mathbf{0} \Rightarrow E(\hat{\mathbf{X}})E(\hat{\mathbf{X}})^T = \mathbf{0}\} =$$
$$= E(\hat{\mathbf{X}}\hat{\mathbf{X}}^T) = \frac{1}{N}\sum_{n=1}^{N} \hat{\mathbf{x}}_n \hat{\mathbf{x}}_n^T = \frac{1}{N}\hat{\mathbf{X}}\hat{\mathbf{X}}^T$$

The previous calculation can be easily seen after rewriting $\hat{\mathbf{X}}$ and $\hat{\mathbf{X}}^T$ in a following form:

$$\hat{\mathbf{X}} = \begin{bmatrix}\mathbf{x}_1 - \overline{\mathbf{x}} & \mathbf{x}_2 - \overline{\mathbf{x}} & \dots & \mathbf{x}_N - \overline{\mathbf{x}}\end{bmatrix}$$
$$\hat{\mathbf{X}}^T = \begin{bmatrix}(\mathbf{x}_1 - \overline{\mathbf{x}})^T \\ (\mathbf{x}_2 - \overline{\mathbf{x}})^T \\ \vdots \\ (\mathbf{x}_N - \overline{\mathbf{x}})^T\end{bmatrix}$$

(d)

Because D is the dimensionality of the data ($\mathbf{x_n}$ has dimensionality of Dx1), and every element in the **S** is a summation of a product between column vector (dimensionality of DxN) and row vector (dimensionality of NxD), which is thus an outer product, and **S** has dimensionality of (DxN)x(NxD) = DxD matrix.

(e)

$$\mathbf{y}_n = L\hat{\mathbf{x}}_n$$
$$E(\mathbf{y}_n) = \frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n = \frac{1}{N}\sum_{n=1}^{N} L\hat{\mathbf{x}}_n = L\frac{1}{N}\sum_{n=1}^{N}\hat{\mathbf{x}}_n$$
$$= \{\text{Using results from subtask (b): } \frac{1}{N}\sum_{n=1}^{N}\hat{\mathbf{x}}_n = \mathbf{0}\} = L \cdot \mathbf{0} = \mathbf{0}$$

Now we would like to find operator **L**, which would allow us to project datapoints into lower-dimensional subspace and have identity covariance matrix. This will make our operator look as following: $\mathbf{L} = \mathbf{P}\mathbf{U}^T$, where **U** is the projection operator into K-dimensional subspace, consisting of k eigenvectors with largest eigenvalues; **P** is the operator, which would allow us to have identity covariance matrix in the subspace. Here we also use the fact, that $S = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$

$$\begin{aligned}
Cov(\mathbf{y}_n, \mathbf{y}_n) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{y}_n\mathbf{y}_n^T &= \frac{1}{N}\sum_{n=1}^{N}(L(\mathbf{x}_n - \overline{\mathbf{x}}_n))(L(\mathbf{x}_n - \overline{\mathbf{x}}_n))^T \\
&= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{P}\mathbf{U}^T(\mathbf{x}_n - \overline{\mathbf{x}}_n))(\mathbf{P}\mathbf{U}^T(\mathbf{x}_n - \overline{\mathbf{x}}_n))^T \\
&= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{P}\mathbf{U}^T(\mathbf{x}_n - \overline{\mathbf{x}}_n))((\mathbf{x}_n - \overline{\mathbf{x}}_n)^T\mathbf{U}\mathbf{P}^T) \\
&= \frac{1}{N}\sum_{n=1}^{N}\mathbf{P}\mathbf{U}^T(\mathbf{x}_n - \overline{\mathbf{x}}_n)(\mathbf{x}_n - \overline{\mathbf{x}}_n)^T\mathbf{U}\mathbf{P}^T \\
&= \mathbf{P}\mathbf{U}^T\left[\frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \overline{\mathbf{x}}_n)(\mathbf{x}_n - \overline{\mathbf{x}}_n)^T\right]\mathbf{U}\mathbf{P}^T \\
&= \mathbf{P}\mathbf{U}^T\mathbf{S}\mathbf{U}\mathbf{P}^T \\
&= \mathbf{P}\mathbf{U}^T\mathbf{U}\Lambda\mathbf{U}^T\mathbf{U}\mathbf{P}^T \\
&= \{\text{Because } \mathbf{U} \text{ is orthogonal matrix: } \mathbf{U}^T\mathbf{U} = \mathbb{1}\} \\
&= \mathbf{P}\mathbf{I}\Lambda\mathbf{I}\mathbf{P}^T \\
&= \mathbf{P}\Lambda\mathbf{P}^T = \mathbb{1}
\end{aligned}$$

Because $\mathbf{\Lambda}$ is a diagonal matrix, to satisfy the condition, we need P to be equal to $\mathbf{\Lambda}^{-\frac{1}{2}}$. Therefore, $P = \mathbf{\Lambda}^{-\frac{1}{2}}$.

$$\mathbf{L} = \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{U}^T$$

So,

$$\mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{\Lambda} \mathbf{\Lambda}^{-\frac{1}{2}} = \mathbb{1}$$

The operation, that we have performed is called whitening.