Homework # 5

Reading (optional)

- Section 9.4 discusses the general theory of EM and covers the material I discussed in the lecture.
- Setion 12.1 discusses PCA.

1. In this problem, we will once again revisit the Hope heights problem of HW 3, but this time we will use the formal notation of EM. Recall, we consider the two component Gaussian mixture model,

$$X = \begin{cases} \mathcal{N}(\mu_1, \sigma_1^2) & \text{with probability } p_1 \\ \mathcal{N}(\mu_2, \sigma_2^2) & \text{with probability } p_2 \end{cases} \tag{1}$$

   where $\mathcal{N}(\mu, \sigma^2)$ is the normal distribution and X models the height of a person when gender is unknown. Let $\theta = (\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, p_1, p_2)$. Let $\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_N$ be the sample heights given in the file.

   (a) Recall that to implement EM we define

$$Q(\theta, \theta') = \sum_{i=1}^{N} E_\theta[\log P(\hat{X}_i, z_i \mid \theta')], \tag{2}$$

   where z_i is either 0 or 1 and determines the mixture $\hat{X}_i$ was sampled from and the θ subscript in the expectation means that we take the expectation with the z_i distributed according to θ. Write down an expression for $Q(\theta, \theta')$ using $r_{i1} = P(z_i = 1 \mid \hat{X}_i, \theta)$, $r_{i2} = P(z_i = 2 \mid \hat{X}_i, \theta)$ and the pdfs of the normals. Then give a formula for r_{i1} and r_{i2} in terms of θ and the $\hat{X}_i$.

   (b) Compute $\text{argmax}_{\theta'} Q(\theta', \theta)$. You should derive an expression for each entry of θ' by solve $\nabla_{\theta'} Q(\theta', \theta) = 0$. Hint: To compute that values of p_1' and p_2' for θ', you can either use a Lagrange multiplier approach, with the constraint $p_1' + p_2' = 1$ or you can simply substitute $p_2' = 1 - p_1'$.

   (c) Compare your updates for the parameters to the heuristic updates of soft EM.

a) $Q(\theta, \theta') = \sum_{i=1}^{N} E_{\theta}\left[\log P(\hat{X}_i, z_i \mid \theta')\right]$

$= \sum_{i=1}^{N} P(z_i = 1 \mid \hat{X}_i, \theta) \log P(\hat{X}_i, z_i = 1 \mid \theta') + P(z_i = 2 \mid \hat{X}_i, \theta) \log P(\hat{X}_i, z_i = 2 \mid \theta')$

$= \sum_{i=1}^{N} r_{i1} \log P(\hat{X}_i, z_i = 1 \mid \theta') + r_{i2} \log P(\hat{X}_i, z_i = 2 \mid \theta')$

$= \sum_{i=1}^{N} r_{i1} \log\left[P(\hat{X}_i \mid z_i = 1, \theta') P(z_i = 1 \mid \theta')\right] + r_{i2}\left[\log P(\hat{X}_i \mid z_i = 2, \theta') P(z_i = 2 \mid \theta')\right]$

$= \sum_{i=1}^{N} r_{i1} \log\left(N(\mu_1', \sigma_1'^2)\big|_{\hat{X}_i} p_1'\right) + r_{i2}\left(N(\mu_2', \sigma_2'^2)\big|_{\hat{X}_i} (1 - p_1')\right)$

$= \sum_{i=1}^{N} r_{i1}\left(\log p_1' + \log \frac{1}{\sqrt{2\pi}\sigma_1'} e^{-\frac{(\hat{X}_i - \mu_1')^2}{2\sigma_1'^2}}\right) + r_{i2}\left(\log(1 - p_1') + \log \frac{1}{\sqrt{2\pi}\sigma_2'} e^{-\frac{(\hat{X}_i - \mu_2')^2}{2\sigma_2'^2}}\right)$

$= \sum_{i=1}^{N} r_{i1}\left(\log p_1' - \log\sqrt{2\pi}\sigma_1' - \frac{(\hat{X}_i - \mu_1')^2}{2\sigma_1'^2}\right) + r_{i2}\left(\log(1 - p_1') - \log\sqrt{2\pi}\sigma_2' - \frac{(\hat{X}_i - \mu_2')^2}{2\sigma_2'^2}\right)$

Now, a formula r_{i1} and r_{i2}. Note that our parameter vector θ consists of $\theta = \langle \mu_1, \mu_2, \sigma_1, \sigma_2, p_1 \rangle$ (θ' is just $\theta' = \langle \mu_1', \mu_2', \sigma_1', \sigma_2', p_1' \rangle$)
Also for simplicity, I am letting $p_2 = 1 - p_1$ since we only have two distributions in the mixture.

$r_{i1} = P(z_i = 1 \mid \hat{X}_i, \theta) = \frac{P(z_i = 1, \hat{X}_i \mid \theta)}{P(\hat{X}_i \mid \theta)} = \frac{P(\hat{X}_i \mid z_i = 1, \theta) P(z_i = 1 \mid \theta)}{P(\hat{X}_i \mid z_i = 1, \theta) P(z_i = 1 \mid \theta) + P(\hat{X}_i \mid z_i = 2, \theta) P(z_i = 2 \mid \theta)}$

$= \frac{N(\mu_1, \sigma_1^2)\big|_{\hat{X}_i} p_1}{N(\mu_1, \sigma_1^2)\big|_{\hat{X}_i} p_1 + N(\mu_2, \sigma_2^2)\big|_{\hat{X}_i} (1 - p_1)}$

By symmetry we have that $r_{i2} = \frac{N(\mu_2, \sigma_2^2)\big|_{\hat{X}_i} (1 - p_1)}{N(\mu_1, \sigma_1^2)\big|_{\hat{X}_i} p_1 + N(\mu_2, \sigma_2^2)\big|_{\hat{X}_i} (1 - p_1)}$

b) We want $\arg\max_{\theta'} Q(\theta, \theta')$. Let's find an expression for each entry of θ' by solving $\nabla_{\theta'} Q(\theta, \theta') = 0$.

Again, note $\theta = \langle \mu_1, \mu_2, \sigma_1^2, \sigma_2^2, p_1 \rangle$.

Also note from a) that

$$Q(\theta, \theta') = \sum_{i=1}^{N} r_{i1} \left(\log p_1' - \log \sqrt{2\pi}\sigma_1' - \frac{(\hat{x}_i - \mu_1')^2}{2\sigma_1'^2} \right) + r_{i2} \left(\log(1-p_1') - \log \sqrt{2\pi}\sigma_2' - \frac{(\hat{x}_i - \mu_2')^2}{2\sigma_2'^2} \right)$$

$$\frac{\partial Q(\theta, \theta')}{\partial \mu_1'} = -\sum_{i=1}^{N} r_{i1} \frac{\partial}{\partial \mu_1} \frac{(\hat{x}_i - \mu_1')^2}{2\sigma_1'^2} = \sum_{i=1}^{N} r_{i1} \frac{(\hat{x}_i - \mu_1')}{\sigma_1'^2} = 0$$

$$\Rightarrow \sum_{i=1}^{N} r_{i1} \hat{x}_i = \sum_{i=1}^{N} r_{i1} \mu_1'$$

$$\Rightarrow \mu_1' \sum_{i=1}^{N} r_{i1} = \sum_{i=1}^{N} r_{i1} \hat{x}_i$$

$$\Rightarrow \mu_1' = \frac{\sum_{i=1}^{N} r_{i1} \hat{x}_i}{\sum_{i=1}^{N} r_{i1}}$$

By symmetry, $\mu_2' = \frac{\sum_{i=1}^{N} r_{i2} \hat{x}_i}{\sum_{i=1}^{N} r_{i2}}$

$$\frac{\partial Q(\theta, \theta')}{\partial \sigma_1'^2} = \sum_{i=1}^{N} r_{i1} \frac{\partial}{\partial \sigma_1'^2} \left(-\log \sqrt{2\pi}\sigma_1' - \frac{(\hat{x}_i - \mu_1')^2}{2\sigma_1'^2} \right)$$

$$= \sum_{i=1}^{N} r_{i1} \frac{\partial}{\partial \sigma_1'^2} \left(-\frac{1}{2} \log 2\pi\sigma_1'^2 - \frac{(\hat{x}_i - \mu_1')^2}{2\sigma_1'^2} \right)$$

$$= \sum_{i=1}^{N} r_{i1} \left(-\frac{1}{2} \frac{2\pi}{2\pi\sigma_1'^2} + \frac{(\hat{x}_i - \mu_1')^2}{2(\sigma_1'^2)^2} \right) = 0$$

$$\Rightarrow \sum_{i=1}^{N} r_{i1} \frac{(\hat{x}_i - \mu_1')^2}{2(\sigma_1'^2)^2} = \sum_{i=1}^{N} r_{i1} \frac{\sigma_1'^2}{\sigma_1'^2}$$

$$\Rightarrow \sum_{i=1}^{N} r_{i1} (\hat{x}_i - \mu_1')^2 = \sum_{i=1}^{N} \sigma_1'^2 r_{i1}$$

$$\Rightarrow \theta_1'^2 = \frac{\sum_{i=1}^{N} r_{i1}(x_i - \mu_1')^2}{\sum_{i=1}^{N} r_{i1}}$$

By symmetry, we have that $\theta_2'^2 = \frac{\sum_{i=1}^{N} r_{i2}(x_i - \mu_2')^2}{\sum_{i=1}^{N} r_{i2}}$

Lastly,

$$\frac{\partial Q(\theta, \theta')}{\partial p_1'} = \frac{\partial}{\partial p_1'} \sum_{i=1}^{N} r_{i1} \log p_1' + r_{i2} \log(1 - p_1')$$

$$= \sum_{i=1}^{N} \frac{r_{i1}}{p_1'} - \frac{r_{i2}}{1 - p_1'} = 0$$

$$\Rightarrow \sum_{i=1}^{N} \frac{r_{i1}}{p_1'} = \sum_{i=1}^{N} \frac{r_{i2}}{1 - p_1'}$$

$$\Rightarrow \frac{1 - p_1'}{p_1'} = \sum_{i=1}^{N} \frac{r_{i2}}{r_{i1}}$$

$$\Rightarrow \frac{1}{p_1'} - 1 = \sum_{i=1}^{N} \frac{r_{i2}}{r_{i1}}$$

$$\Rightarrow \frac{1}{p_1'} = \sum_{i=1}^{N} \frac{r_{i1} + r_{i2}}{r_{i1}}$$

$$\Rightarrow p_1' = \sum_{i=1}^{N} \frac{r_{i1}}{r_{i1} + r_{i2}} = \sum_{i=1}^{N} \frac{r_{i1}}{N}$$

(Note, if we wanted p_2' explicitly, by symmetry: $p_2' = \sum_{i=1}^{N} \frac{r_{i2}}{N}$)

C) They are exactly the same. The soft EM updates were derived in the previous HW.

3. This problem focuses on the computations involved in deriving the PCA. Consider the dataset formed by $X^{(i)} \in \mathbb{R}^n$ for $i = 1, 2, \ldots, N$. Set $\mu = 1/N \sum_{i=1}^{n} X^{(i)}$.

   (a) Let $a, b \in \mathbb{R}^n$ be column vectors. Show in any way you like - by proof, through example, by intuitive explanation - that $(a \cdot b)^2 = a^T M a$ where M is an $n \times n$ matrix given by $M = bb^T$.

Consider $a^T M a$ where $M = bb^T$:

$$a^T M a = a^T bb^T a = (b^T a)^T (b^T a) = (b^T a)(b^T a) = (b \cdot a)(b \cdot a) = (a \cdot b)(a \cdot b) = (a \cdot b)^2 \quad \blacksquare$$

The third equality follows by defn of dot product where $b^T a = b \cdot a$. The fourth equality follows since the dot product is commutative. The remaining follows from properties of $\mathbb{R}$ (note $b^T a = b \cdot a = a \cdot b$ is just a scaler in $\mathbb{R}$).

   (b) Let $\hat{\Sigma}$ be the covariance matrix of the data. Then, by definition

$$\hat{\Sigma}_{jk} = \frac{1}{N} \sum_{i=1}^{N} (X_j^{(i)} - \mu_j)(X_k^{(i)} - \mu_k) \tag{4}$$

   Show that $\hat{\Sigma}$ can also be written in the following two forms

   - Let $\tilde{X}$ be the $N \times n$ matrix with the $X^{(i)} - \mu$ as the rows

$$\hat{\Sigma} = \frac{1}{N} \tilde{X}^T \tilde{X} \tag{5}$$

   - Thinking of the $X^{(i)}$ as column vectors,

$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (X^{(i)} - \mu)(X^{(i)} - \mu)^T \tag{6}$$

Let's first show (5). We will show $\left(\frac{1}{N} \tilde{X}^T \tilde{X}\right)_{jk}$ is indeed $\frac{1}{N} \sum_{i=1}^{N} (X_j^{(i)} - \mu_j)(X_k^{(i)} - \mu_k)$.

Note $\tilde{X}^T \tilde{X} = \begin{bmatrix} X^{(1)} - \mu & X^{(2)} - \mu & \cdots & X^{(N)} - \mu \end{bmatrix} \begin{bmatrix} (X^{(1)} - \mu)^T \\ (X^{(2)} - \mu)^T \\ \vdots \\ (X^{(N)} - \mu)^T \end{bmatrix}$

$$= \begin{bmatrix} X_1^{(1)} - \mu_1 & X_1^{(2)} - \mu_1 & \cdots & X_1^{(N)} - \mu_1 \\ X_2^{(1)} - \mu_2 & X_2^{(2)} - \mu_2 & \cdots & X_2^{(N)} - \mu_2 \\ \vdots & \vdots & \ddots & \vdots \\ X_N^{(1)} - \mu_N & X_N^{(2)} - \mu_N & \cdots & X_N^{(N)} - \mu_N \end{bmatrix} \begin{bmatrix} X_1^{(1)} - \mu_1 & X_2^{(1)} - \mu_2 & \cdots & X_N^{(1)} - \mu_N \\ X_1^{(2)} - \mu_1 & X_2^{(2)} - \mu_2 & \cdots & X_N^{(2)} - \mu_N \\ \vdots & & & \\ X_1^{(N)} - \mu_1 & X_2^{(N)} - \mu_2 & \cdots & X_N^{(N)} - \mu_N \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{i=1}^{N} (X_1^{(i)} - \mu_1)(X_1^{(i)} - \mu_1) & \cdots & \sum_{i=1}^{N} (X_1^{(i)} - \mu_1)(X_N^{(i)} - \mu_N) \\ \sum_{i=1}^{N} (X_2^{(i)} - \mu_2)(X_1^{(i)} - \mu_1) & \cdots & \sum_{i=1}^{N} (X_2^{(i)} - \mu_2)(X_N^{(i)} - \mu_N) \\ \vdots & & \vdots \\ \sum_{i=1}^{N} (X_N^{(i)} - \mu_N)(X_1^{(i)} - \mu_1) & \cdots & \sum_{i=1}^{N} (X_N^{(i)} - \mu_N)(X_N^{(i)} - \mu_N) \end{bmatrix}$$

So $\left(\frac{1}{N} \tilde{X}^T \tilde{X}\right)_{jk} = \frac{1}{N} \left(\tilde{X}^T \tilde{X}\right)_{jk} = \frac{1}{N} \sum_{i=1}^{N} (X_j^{(i)} - \mu_j)(X_k^{(i)} - \mu_k)$ ∎

Let's now show (6). Let's also show that $\left(\frac{1}{N} \sum_{i=1}^{N} (X^{(i)} - \mu)(X^{(i)} - \mu)^T\right)_{jk}$ is $\frac{1}{N} \sum_{i=1}^{N} (X_j^{(i)} - \mu_j)(X_k^{(i)} - \mu_k)$.

Note that $\sum_{i=1}^{N} (X^{(i)} - \mu)(X^{(i)} - \mu)^T$

$$= \sum_{i=1}^{N} \begin{bmatrix} X_1^{(i)} - \mu_1 \\ \vdots \\ X_N^{(i)} - \mu_N \end{bmatrix} \begin{bmatrix} X_1^{(i)} - \mu_1 & \cdots & X_N^{(i)} - \mu_N \end{bmatrix}$$

$$= \sum_{i=1}^{N} \begin{bmatrix} (X_1^{(i)} - \mu_1)(X_1^{(i)} - \mu_1) & \cdots & (X_1^{(i)} - \mu_1)(X_N^{(i)} - \mu_N) \\ \vdots & & \vdots \\ (X_N^{(i)} - \mu_N)(X_1^{(i)} - \mu_1) & \cdots & (X_N^{(i)} - \mu_N)(X_N^{(i)} - \mu_N) \end{bmatrix}$$ by defn of outer product

$$= \begin{bmatrix} \sum_{i=1}^{N} (X_1^{(i)} - \mu_1)(X_1^{(i)} - \mu_1) & \cdots & \sum_{i=1}^{N} (X_1^{(i)} - \mu_1)(X_N^{(i)} - \mu_N) \\ \vdots & & \vdots \\ \sum_{i=1}^{N} (X_N^{(i)} - \mu_N)(X_1^{(i)} - \mu_1) & \cdots & \sum_{i=1}^{N} (X_N^{(i)} - \mu_N)(X_N^{(i)} - \mu_N) \end{bmatrix}$$

$= \tilde{X}^T \tilde{X}$ by work done for (5)

$$\Rightarrow \frac{1}{N} \sum_{i=1}^{N} (X^{(i)} - \mu)(X^{(i)} - \mu)^T = \frac{1}{N} \tilde{X}^T \tilde{X}$$

And so $\left(\frac{1}{N} \sum_{i=1}^{N} (X^{(i)} - \mu)(X^{(i)} - \mu)^T\right)_{jk} = \left(\frac{1}{N} \tilde{X}^T \tilde{X}\right)_{jk} = \frac{1}{N} \sum_{i=1}^{N} (X_j^{(i)} - \mu_j)(X_k^{(i)} - \mu_k)$ ∎

(c) The 1-d PCA involves the parameters μ, $w^{(1)}$ and $c_i \in \mathbb{R}$ for $i = 1, 2, \ldots, N$ that are used to approximate $X^{(i)}$ according to

$$X^{(i)} \approx \mu + c_i w^{(1)}. \tag{7}$$

Derive the values of c_i and $w^{(1)}$ that optimize this approximation. (We did this in class.) Then, compute the mean and variance of the c_i.

We want $\min_{\substack{c_i \\ w^{(1)} \\ \|w^{(1)}\|=1}} f(\mu, w^{(1)}, c_1, \ldots, c_N) = \min \sum_{i=1}^{N} \| \hat{x}^{(i)} - \mu - c_i w^{(1)} \|$

Note: $f(\mu, w^{(1)}, c_1 \cdots, c_N) = \sum_{i=1}^{N} (\hat{x}^{(i)} - \mu - c_i w^{(1)})^T (\hat{x}^{(i)} - \mu - c_i w^{(1)})$

$= \sum_{i=1}^{N} \hat{x}^{(i)T} \hat{x}^{(i)} - 2\hat{x}^{(i)T}\mu - 2c_i \hat{x}^{(i)T} w^{(1)} + \mu^T\mu + 2c_i \mu^T w^{(1)} + c_i^2 w^{(1)T} w^{(1)}$

$\frac{\partial f(\mu, w^{(1)}, c_1 \cdots, c_N)}{\partial c_i} = -2\hat{x}^{(i)T} w^{(1)} - 2\mu^T w^{(1)} + 2c_i w^{(1)T} w^{(1)} = 0$

$\Rightarrow -\hat{x}^{(i)T} w^{(1)} - \mu^T w^{(1)} + c_i w^{(1)T} w^{(1)} = 0$

$\Rightarrow c_i w^{(1)T} w^{(1)} = \hat{x}^{(i)T} w^{(1)} + \mu^T w^{(1)}$

$\Rightarrow c_i = \frac{w^{(1)T}(\hat{x}^{(i)} - \mu)}{w^{(1)T} w^{(1)}} = \frac{w^{(1)T}(\hat{x}^{(i)} - \mu)}{\|w^{(1)}\|^2} = w^{(1)T}(\hat{x}^{(i)} - \mu)$ since $\|w^{(1)}\| = 1$ by our constraint

Need to find $w^{(1)}$:

$$\min_{\substack{c^{(i)} \\ w^{(1)} \\ \|w^{(1)}\|=1}} \sum_{i=1}^{N} \| x^{(i)} - \mu - c_i w^{(1)} \|^2$$

$$\min_{\substack{w^{(1)} \\ \|w^{(1)}\|=1}} \sum_{i=1}^{N} \| x^{(i)} - \mu - ((x^{(i)} - \mu) \cdot w^{(1)}) w^{(1)} \|^2$$

Take the data and subtract mean for ease:

$$\Rightarrow \min_{\substack{w^{(1)} \\ \|w^{(1)}\|=1}} \sum_{i=1}^{N} \| x^{(i)} - (x^{(i)} \cdot w^{(1)}) w^{(1)} \|^2$$

$$= \min_{\substack{w^{(1)} \\ \|w^{(1)}\|^2=1}} \sum_{i=1}^{N} (x^{(i)} - (x^{(i)} \cdot w^{(1)}) w^{(1)}) \cdot (x^{(i)} - (x^{(i)} \cdot w^{(1)}) w^{(1)})$$

$$= \min_{\substack{w^{(1)} \\ \|w^{(1)}\|^2=1}} \sum_{i=1}^{N} (x^{(i)T} x^{(i)} - (x^{(i)T} w^{(1)})^2)$$

$$\propto \min_{\substack{w^{(1)} \\ \|w^{(1)}\|^2=1}} \sum_{i=1}^{N} -(x^{(i)T} w^{(1)})^2$$ since $x^{(i)T} x^{(i)}$ is a constant in the minimization

$\nabla f(x) = \lambda \nabla g(x)$

$$\nabla_{w^{(1)}} \left(-\sum_{i=1}^{N} (\hat{x}^{(i)} \cdot w^{(1)})(\hat{x}^{(i)} \cdot w^{(1)}) \right)$$

$$= -\sum_{i=1}^{N} 2\hat{x}^{(i)} (\hat{x}^{(i)} \cdot w^{(1)})$$

$$= -2\left(\sum_{i=1}^{N} \hat{x}^{(i)} \hat{x}^{(i)T}\right) w^{(1)}$$

$$= -2N \left(\frac{1}{N} \sum_{i=1}^{N} \hat{x}^{(i)} \hat{x}^{(i)T}\right) w^{(1)}$$

$$= -2N \hat{\Sigma} w^{(1)}$$

$g(w^{(1)}) = 1$

$g(w^{(1)}) = w_1^{(1)2} + w_2^{(1)2} + \dots + w_n^{(1)2}$

$\nabla g(w^{(1)}) = 2 w_1^{(1)}$

$$-2N \hat{\Sigma} w^{(1)} = \lambda 2 w_1^{(1)}$$

$$\hat{\Sigma} w^{(1)} = \tilde{\lambda} w_1^{(1)}$$

$\hat{\Sigma}$ is symmetric so it has $q^{(1)}, q^{(2)}, \dots, q^{(n)}$
$\downarrow \quad \downarrow \quad \dots \quad \downarrow$
$\lambda_1 \quad \lambda_2 \quad \lambda_2$

you have n possible solns: $q^{(1)}, q^{(2)}, \dots, q^{(n)}$

and plug into f and choose min: $w^{(1)} = q^{(j)}$

$$-\sum_{i=1}^{N} (x^{(i)} \cdot q^{(j)})^2$$

$$= -\sum_{i=1}^{N} (x^{(i)} \cdot q^{(j)})(x^{(i)} \cdot q^{(j)})$$

$$= -\sum_{i=1}^{N} q^{(j)T} x^{(i)} x^{(i)T} q^{(j)}$$

$$= -q^{(j)T} \sum_{i=1}^{N} x^{(i)} x^{(i)T} q^{(j)}$$

$$= -q^{(j)T} N \hat{\Sigma} q^{(j)}$$

$$= -N q^{(j)T} \lambda_j q^{(j)}$$

$$= -N \lambda_j q^{(j)T} q^{(j)} = -N \lambda_j$$

← most negative when λ_j is greatest so $w^{(1)}$ is the eigenvector of $\hat{\Sigma}$ with the largest eigenvalue.

Now, let's find the mean and variance of the C_i:

Mean:
$$\frac{1}{N}\sum_{i=1}^{N} C_i = \frac{1}{N}\sum_{i=1}^{N} w^{(1)T}(\hat{x}^{(i)} - \mu) = \frac{w^{(1)T}}{N}\sum_{i=1}^{N}(\hat{x}^{(i)} - \mu) = 0 \quad \blacksquare$$

Variance:
$$\frac{1}{N}\sum_{i=1}^{N}(C_i - 0)^2 = \frac{1}{N}\sum_{i=1}^{N} C_i^2 = \frac{1}{N}\sum_{i=1}^{N}\left(w^{(1)T}(\hat{x}^{(i)} - \mu)\right)^2$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left[w^{(1)T}(\hat{x}^{(i)} - \mu)\right]\left[w^{(1)T}(\hat{x}^{(i)} - \mu)\right]$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left[w^{(1)T}(\hat{x}^{(i)} - \mu)\right]\left[(\hat{x}^{(i)} - \mu)^T w^{(1)}\right]$$

$$= \frac{1}{N} w^{(1)T}\sum_{i=1}^{N}(\hat{x}^{(i)} - \mu)(\hat{x}^{(i)} - \mu)^T w^{(1)}$$

$$= w^{(1)T}\hat{\Sigma} w^{(1)} = \lambda_1 \quad \blacksquare$$

2. See the attached section 9.3.3 from Bishop for a definition and discussion of Bernoulli mixture models. Let X represent 10 bits, i.e. $X = (X_1, X_2, \ldots, X_{10})$ where each coordinate of X is either 0 or 1. Assume the following Bernoulli mixture model for the ith coordinate of X, X_i:

$$X_i = \begin{cases} \text{Bernoulli}(\mu_i^{(1)}) & \text{with probability } p_1 \\ \text{Bernoulli}(\mu_i^{(2)}) & \text{with probability } p_2, \end{cases} \tag{3}$$

where $\mu^{(1)}, \mu^{(2)} \in \mathbb{R}^{10}$ with all coordinates in $[0, 1]$. Assume further that the coordinates of X are always sampled from the same mixture, with probabilities p_1 and p_2 for mixture 1 and 2 respectively, but that the Bernoulli draw of each coordinate is independent.

   (a) Write down an EM iteration for this mixture model.

   (b) Attached is the file `noisy_bits.csv` which contains a 500×10 matrix. Each row of the matrix is a sample of X. If you look at an image of the matrix (in R use **image** on the transposed matrix), you will see that there are two patterns, but with some noise added. Use your EM algorithm to fit the mixture model to the data. Does your fit recover the two underlying patterns?

a)

Let $x^{(i)}$ be 10-dim where $X_j^{(i)}$ is distributed as a mixture of 2 bernoulli distributions with parameters $\mu_j^{(1)}$ and $\mu_j^{(2)}$.

So our parameters of interest are $\theta = \langle \mu^{(1)}, \mu^{(2)}, p_1 \rangle$.

For each $x^{(i)}$ let us introduce a latent random variable $z^{(i)} = \begin{bmatrix} z_1^{(i)} \\ z_2^{(i)} \end{bmatrix}$

$\Rightarrow z^{(i)} = \left\{ \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\}$ and $p(z_1^{(i)} = 1) = p_1$, $p(z_2^{(i)} = 1) = 1 - p_1$

Now, let us write down $p(x \mid z, \mu) = \prod_{k=1}^{2} p(x \mid \mu^k)^{z_k} = \prod_{k=1}^{2} \prod_{j=1}^{10} (\mu_j^k)^{x_j} (1-\mu_j^k)^{1-x_j}$

and $p(z \mid p) = \prod_{k=1}^{2} p_k^{z_k}$

And so, the complete data log likelihood is:

$\ln p(x, z \mid \mu, p) = \ln p(x \mid z, \mu)\, p(z \mid p)$

Considering our N data points we have:

$$\ln p(x, z \mid \mu, p) = \ln \prod_{i=1}^{N} p(x^{(i)} \mid z^{(i)}, \mu)\, p(z^{(i)} \mid p)$$

$$= \ln \prod_{i=1}^{N} \prod_{k=1}^{2} p(x^{(i)} \mid z^{(i)}, \mu^k)^{z_k^{(i)}} p_k^{z_k^{(i)}}$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{2} z_k^{(i)} \left[\ln p(x^{(i)} \mid \mu^k) + \ln p_k \right]$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{2} z_k^{(i)} \left[\ln \prod_{j=1}^{10} (\mu_j^k)^{x_j^{(i)}} (1-\mu_j^k)^{1-x_j^{(i)}} + \ln p_k \right]$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{2} z_k^{(i)} \left[\sum_{j=1}^{10} \left(x_j^{(i)} \ln(\mu_j^k) + (1 - x_j^{(i)}) \ln(1-\mu_j^k) \right) + \ln p_k \right]$$

Now, we are interested in:

$$E_z\left[\ln P(x,z\mid\mu,P)\right] = \sum_{i=1}^{N}\sum_{k=1}^{2} E(z_k^{(i)})\left[\sum_{j=1}^{10}\left(x_j^{(i)}\ln(\mu_j^k) + (1-x_j^{(i)})\ln(1-\mu_j^k)\right) + \ln p_k\right]$$

Note that $E(z_k^{(i)}) = 1\cdot P(z_k^{(i)}=1\mid x^{(i)},\mu^k) + 0\cdot P(z_k^{(i)}=0\mid x^{(i)},\mu^k)$

$$= P(z_k^{(i)}=1\mid x^{(i)},\mu^k)$$

By Bayes we have:

$$E(z_k^{(i)}) = P(z_k^{(i)}=1\mid x^{(i)},\mu^k) = \frac{P(x^{(i)}\mid z_k^{(i)}=1,\mu^k)\,P(z_k^{(i)}=1\mid p_k)}{\sum_{l=1}^{2} P(x^{(i)}\mid z_l^{(i)}=1,\mu^l)\,P(z_l^{(i)}=1\mid p_l)}$$

Now, let's set up the EM algorithm:

For iterations $n = 1, 2, \dots$

① E step

Sps we have $\theta^{(n)} = \langle (\mu^{(1)})^{(n)}, (\mu^{(2)})^{(n)}, p_1^{(n)}\rangle$

$$Q(\theta^{(n)},\theta') = E_z(\ln P(x,z\mid\theta') = E_z(\ln(x,z\mid\mu',p')$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{2} P(z_k^{(i)}=1\mid\theta^{(i)})\left[\sum_{j=1}^{10} x_j^{(i)}\ln\mu_j^{k'} + (1-x_j^{(i)})\ln(1-\mu_j^{k'}) + \ln p_k'\right]$$

$$= \sum_{i=1}^{N} P(z_1^{(i)}=1\mid\theta^{(i)})\left[\sum_{j=1}^{10} x_j^{(i)}\ln\mu_j^{(1)'} + (1-x_j^{(i)})\ln(1-\mu_j^{(1)'}) + \ln p_1'\right]$$

$$+ P(z_2^{(i)}=1\mid\theta^{(i)})\left[\sum_{j=1}^{10} x_j^{(i)}\ln\mu_j^{(2)'} + (1-x_j^{(i)})\ln(1-\mu_j^{(2)'}) + \ln(1-p_1')\right]$$

② M step

$$\theta^{(n+1)} = \arg\max_{\theta'} Q(\theta^{(n)},\theta')$$

Now,

$$\frac{\partial Q(\theta^{(n)},\theta')}{\partial \mu_j^{(1)'}} = \sum_{i=1}^{N} P(z_1^{(i)}=1\mid\theta^{(i)})\left(\frac{x_j^{(i)}}{\mu_j^{(1)'}} - \frac{1-x_j^{(i)}}{1-\mu_j^{(1)'}}\right) = 0$$

$$\Rightarrow \sum_{i=1}^{N} P(z_1^{(i)}=1|\theta^{(i)})\left[(1-\mu_j^{(1)\prime})x_j^{(i)} - (1-x_j^{(i)})\mu_j^{(1)\prime}\right] = 0$$

$$\Rightarrow \mu_j^{(1)\prime} = \frac{\sum_{i=1}^{N} P(z_1^{(i)}=1|\theta^{(i)})\, x_j^{(i)}}{\sum_{i=1}^{N} P(z_1^{(i)}=1|\theta^{(i)})}$$

So $\mu^{(1)\prime} = \dfrac{\sum_{i=1}^{N} P(z_1^{(i)}=1|\theta^{(i)})\, x^{(i)}}{\sum_{i=1}^{N} P(z_1^{(i)}=1|\theta^{(i)})}$. By symmetry, $\mu^{(2)\prime} = \dfrac{\sum_{i=1}^{N} P(z_2^{(i)}=1|\theta^{(i)})\, x^{(i)}}{\sum_{i=1}^{N} P(z_2^{(i)}=1|\theta^{(i)})}$

Now,

$$\frac{\partial Q(\theta^{(n)},\theta')}{\partial p_1'} = \sum_{i=1}^{N}\left(\frac{P(z_1^{(i)}=1|\theta^{(i)})}{p_1'} - \frac{P(z_2^{(i)}=1|\theta^{(i)})}{1-p_1'}\right) = 0$$

$$\Rightarrow p_1' = \frac{\sum_{i=1}^{N} P(z_1^{(i)}=1|\theta^{(i)})}{\sum_{i=1}^{N} P(z_1^{(i)}=1|\theta^{(i)}) + \sum_{j=1}^{N}(z_2^{(i)}=1|\theta^{(i)})} = \frac{\sum_{i=1}^{N} P(z_1^{(i)}=1|\theta^{(i)})}{N}$$

And $P(z_1^{(i)}=1|\theta') = \dfrac{P(x^{(i)}|z_1^{(i)}=1,\mu^{l\prime})\,P(z_1^{(i)}=1|p_1')}{\sum_{l=1}^{2} P(x^{(i)}|z_l^{(i)}=1,\mu^{l\prime})\,P(z_l^{(i)}=1|p_l')}$

$$= \frac{p_1' \prod_{j=1}^{10}(\mu_j^{(1)\prime})^{x_j^{(i)}}(1-\mu_j^{(1)\prime})^{1-x_j^{(i)}}}{\sum_{l=1}^{2} P(x^{(i)}|z_l^{(i)}=1,\mu^{l\prime})\,P(z_l^{(i)}=1|p_l')}$$

$$= \frac{p_1' \prod_{j=1}^{10}(\mu_j^{(1)\prime})^{x_j^{(i)}}(1-\mu_j^{(1)\prime})^{1-x_j^{(i)}}}{p_1' \prod_{j=1}^{10}(\mu_j^{(1)\prime})^{x_j^{(i)}}(1-\mu_j^{(1)\prime})^{1-x_j^{(i)}} + (1-p_1')\prod_{j=1}^{10}(\mu_j^{(2)\prime})^{x_j^{(i)}}(1-\mu_j^{(2)\prime})^{1-x_j^{(i)}}}$$

($\Rightarrow P(z_2^{(i)}=1|\theta') = 1 - P(z_1^{(i)}=1|\theta')$) ■

HW5

October 4, 2020

3.b

Attached is the file noisy_bits.csv which contains a 500×10 matrix. Each row of the matrix is a sample of X. If you look at an image of the matrix (in R use image on the transposed matrix), you will see that there are two patterns, but with some noise added. Use your EM algorithm to fit the mixture model to the data. Does your fit recover the two underlying patterns?

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
data = pd.read_csv("noisy_bits.csv")
data.head()
```

[2]:

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	1	1	1	0	1	0	1	0	0	1
1	1	0	0	1	1	0	0	1	0	0
2	1	1	1	1	1	0	0	0	0	0
3	0	1	1	0	0	0	0	0	0	0
4	1	0	1	1	1	0	0	0	0	0

[3]:
```
X = data.to_numpy()
X.shape
```

[3]:
```
(500, 10)
```

[4]:
```
plt.figure(figsize=(10,10))
plt.imshow(X.T)
plt.show()
```

[5]:
```
N, n = X.shape[0], X.shape[1]
```

Let's create a function that gives us $P(Z_1^{(i)} = 1|\theta')$ for any given θ'.

[6]:
```
def p(mu1, mu2, p1):
    num = p1*np.prod(mu1**X*(1-mu1)**(1-X), axis=1)
    den = num + (1-p1)*np.prod(mu2**X*(1-mu2)**(1-X), axis=1)
    return num/den
```

[7]:
```
def logl(mu1, mu2, p1):
    probs = p(mu1, mu2, p1)
    a = probs.dot(X.dot(np.log(mu1))+(1-X).dot(np.log(1-mu1))+np.log(p1))
    b = (1-probs).dot(X.dot(np.log(mu2))+(1-X).dot(np.log(1-mu2))+np.log(1-p1))
    return a+b
```

[8]:
```
def EM(mu1, mu2, p1):
    """EM algorithm

    Args:
        mu1: vector of size (10,)
        mu2: vector of size (10,)
        p1: scalar

    Returns:
        A list with the following elements:
            mu1: vector of size (10,)
            mu2: vector of size (10,)
            p1: scalar


    """
    probs = p(mu1, mu2, p1)
    mu1 = probs.dot(X)/sum(probs)
    mu2 = (1-probs).dot(X)/sum(1-probs)
    p1 = sum(probs)/N

    return [mu1, mu2, p1]


def EM_function(theta, eps=10**-4):
    """Function that runs the EM algorithm

    Args:
        theta: A list with the following elements:
            mu1: vector of size (10,)
            mu2: vector of size (10,)
            p1: scalar
```

```
    Returns:
            1. theta: A list with the following elements
                mu1: vector of size (10,)
                mu2: vector of size (10,)
                p1: scalar
            2. logs: a list with the log likelihood value of each iteration i

    """

    l = np.Inf
    logs = []
    while(abs(l - logl(*EM(*theta))) > eps):
        theta = EM(*theta)
        l = logl(*theta)
        logs.append(l)
    return theta, logs
```

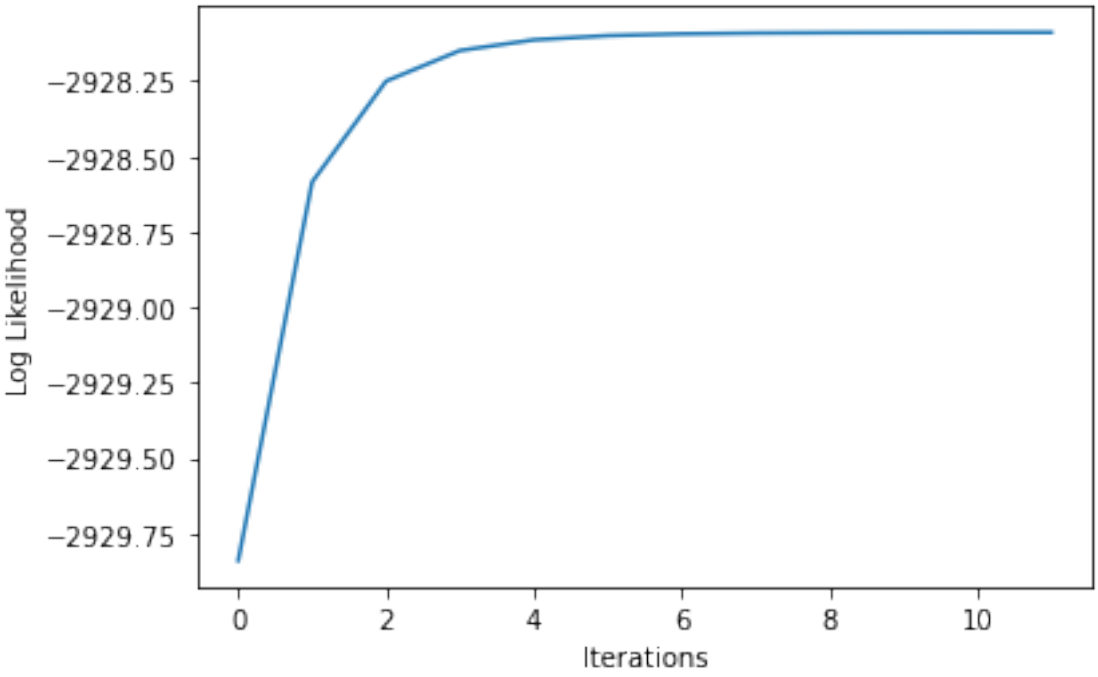
[9]:
```
# Use KMeans to initialize the parameter vector theta
from sklearn.cluster import KMeans

model = KMeans(n_clusters=2)
model.fit(X)
centers = model.cluster_centers_

theta_start = [centers[0], centers[1], 0.2]
```

[10]:
```
theta, logs = EM_function(theta_start)
```

[11]:
```
plt.plot(logs)
plt.xlabel("Iterations")
plt.ylabel("Log Likelihood")
plt.show()
```



μ_1 :

[12]:
```
theta[0]
```

[12]:
```
array([0.80896481, 0.78512506, 0.78081163, 0.76835602, 0.77582801,
       0.2793406 , 0.20532695, 0.20609471, 0.20782407, 0.18556576])
```

μ_2

[13]:
```
theta[1]
```

[13]:
```
array([0.22500748, 0.22105249, 0.21737246, 0.16137866, 0.23862691,
       0.8141801 , 0.79273639, 0.81216121, 0.75381171, 0.79270119])
```

p_1 :

[14]:
```
theta[2]
```

[14]:
```
0.5051610828757628
```

Yes, the trend has been uncovered. From make_noisy_bits.R we can see that the first half of the datapoints (250 datapoints) were created as $< 1, 1, 1, 1, 1, 0, 0, 0, 0, 0 >$ with 80% probability and $< 0, 0, 0, 0, 0, 1, 1, 1, 1, 1 >$ otherwise. Note how μ_1 has approx. 0.8 as the expected mean for the first five coordinates and approx. 0.2 for the last five.

Similarly, the second half (250 datapoints) were created as $< 0, 0, 0, 0, 0, 1, 1, 1, 1, 1 >$ with 80% probability and $< 1, 1, 1, 1, 1, 0, 0, 0, 0, 0 >$ otherwise. Note how μ_2 has approx. 0.2 as the expected

mean for the first five coordinates and approx. 0.8 for the last five.

Finally, note how p_1 is approx. 0.5 – agreeing with the 50/50 split.

[ ]: