Problem Set 4: Unsupervised Learning and Reinforcement Learning

Eitan Joseph Caroline Wang

August 20, 2020

Problem 1

EM for supervised learning
In class we applied EM to the unsupervised learning setting. In particular, we represented $p(x)$ by marginalizing over a latent random variable

$$p(x) = \sum_z p(x, z) = \sum_z p(x|z)p(z)$$

However, EM can also be applied to the supervised learning setting, and in this problem we discuss a "mixture of linear regressors" model; this is an instance of what is often call the Hierarchical Mixture of Experts model. We want to represent $p(y|x)$, $x \in \mathbb{R}^n$ and $y \in \mathbb{R}$, and we do so by again introducing a discrete latent random variable

$$p(y|x) = \sum_z p(y, z|x) = \sum_z p(y|x, z)p(z|x)$$

For simplicity we'll assume that z is binary valued, that p(y—x, z) is a Gaussian density, and that $p(y|x)$ is given by a logistic regression model. More formally

$$\begin{aligned} p(z|x;\phi) =& g(\phi^T x)^z (1 - g(\phi^T x))^{1-z} \\ p(y|x, z = i; \theta_i) =& \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(y - \theta_i^T x)^2}{2\sigma^2}\right) \quad i = 0, 1 \end{aligned}$$

where σ is a known parameter and $\phi, \theta_0, \theta_1 \in \mathbb{R}$ are parameters of the model (here we use the subscript on θ to denote two different parameter vectors, not to index a particular entry in these vectors).
Intuitively, the process behind model can be thought of as follows. Given a data point x, we first determine whether the data point belongs to one of two hidden classes $z = 0$ or $z = 1$, using a logistic regression model. We then determine y as a linear function of x (different linear functions for different values of z) plus Gaussian noise, as in the standard linear regression model. For example, the following data set could be well-represented by the model, but not by standard linear regression.

(a) Suppose x, y, and z are all observed, so that we obtain a training set $(x^{(1)}, y^{(1)}, z^{(1)}), ..., (x^{(m)}, y^{(m)}, z^{(m)})$. Write the log-likelihood of the parameters, and derive the maximum likelihood estimates for ϕ, θ_0, and θ_1. Note that because $p(z|x)$ is a logistic regression model, there will not exist a closed form estimate of ϕ. In this case, derive the gradient and the Hessian of the likelihood with respect to ϕ; in practice, these quantities can be used to numerically compute the ML estimate.

answer:

The log-likelihood can be written as

$$\begin{aligned}
\ell(\phi, \theta_0, \theta_1) =& \sum_{i=1}^{m} \log p(y^{(i)}|x^{(i)}; \phi, \theta_0, \theta_1) \\
=& \sum_{i=1}^{m} \log p(y^{(i)}|x^{(i)}, z^{(i)}; \theta_0, \theta_1) p(z^{(i)}|x^{(i)}; \phi) \\
=& \sum_{i=1}^{m} \mathbb{1}\{z^{(i)} = 0\} \log \left((1 - g(\phi^T x^{(i)})) \left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(y^{(i)} - \theta_k^T x^{(i)})^2}{2\sigma^2} \right) \right) \right) \\
& + \sum_{i=1}^{m} \mathbb{1}\{z^{(i)} = 1\} \log \left((g(\phi^T x^{(i)})) \left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(y^{(i)} - \theta_1^T x^{(i)})^2}{2\sigma^2} \right) \right) \right)
\end{aligned}$$

The maximum likelihood estimation for θ_k can be derived by taking the gradient with respect to θ_k and setting the result to zero.

$$\nabla_{\theta_0} \ell(\phi, \theta_0, \theta_1) \overset{\text{set}}{=} 0$$

Then by extracting the non-relevant constants to θ_k:

$$\begin{aligned}
& \nabla_{\theta_k} \sum_{i=1}^{m} - (y - \theta_k^T x^{(i)})^2 = 0 \\
& \sum_{i=1}^{m} - 2(y^{(i)} - \theta_k^T x^{(i)}) \cdot x^{(i)} = 0 \\
\implies & \sum_{i=1}^{m} x^{(i)} y^{(i)} = \sum_{i=1}^{m} \theta_k^T x^{(i)T} x^{(i)} \\
\implies & X^T \vec{y} = \theta_k^T X^T X \\
\implies & \theta_k = (X^T X)^{-1} X^T \vec{y}
\end{aligned}$$

Therefore the specific estimations for θ_0 and θ_1 are:

$$\begin{aligned}
\theta_0 &= (X_0^T X_0)^{-1} X_0^T \vec{y_0} \\
\theta_1 &= (X_1^T X_1)^{-1} X_1^T \vec{y_1}
\end{aligned}$$

Where X_0, y_0, X_1, y_1 are the associated matrices and vectors derived in each of the two separate summations that combine to equal all of z.

The next step is to find the gradient vector and the Hessian matrix respect with respect to ϕ. We can once again remove the terms not relating to ϕ:

$$\begin{aligned}
\nabla_\phi \ell(\phi, \theta_0, \theta_1) =& \nabla_\phi \sum_{i=1}^{m} \mathbb{1}\{z^{(i)} = 0\} \log(1 - g(\phi^T x^{(i)})) + \sum_{i=1}^{m} \mathbb{1}\{z^{(i)} = 1\} \log((g(\phi^T x^{(i)})) \\
=& \nabla_\phi \sum_{i=1}^{m} (1 - z^{(i)}) \log(1 - g(\phi^T x^{(i)})) + z^{(i)} \log((g(\phi^T x^{(i)}))
\end{aligned}$$

From previous classes, the derivative of a sigmoid is known to be $\frac{\partial}{\partial z}g(z) = g(z)(1-g(z))$, therefore

$$\begin{aligned}
\nabla_\phi \ell(\phi,\theta_0,\theta_1) =& \sum_{i=1}^{m} -\frac{1-z^{(i)}}{(1-g(\phi^T x^{(i)}))}\frac{\partial}{\partial \phi}g(\phi^T x^{(i)})\cdot x^{(i)} + \frac{z^{(i)}}{g(\phi^T x^{(i)})}\frac{\partial}{\partial \phi}g(\phi^T x^{(i)})\cdot x^{(i)} \\
=& \sum_{i=1}^{m} -\frac{1-z^{(i)}}{(1-g(\phi^T x^{(i)}))}g(\phi^T x^{(i)})(1-g(\phi^T x^{(i)}))\cdot x^{(i)} + \frac{z^{(i)}}{g(\phi^T x^{(i)})}g(\phi^T x^{(i)})(1-g(\phi^T x^{(i)}))\cdot x^{(i)} \\
=& \sum_{i=1}^{m} -(1-z^{(i)})g(\phi^T x^{(i)})\cdot x^{(i)} + z^{(i)}(1-g(\phi^T x^{(i)}))\cdot x^{(i)} \\
=& \sum_{i=1}^{m} x^{(i)}(z^{(i)} - g(\phi^T x^{(i)}))
\end{aligned}$$

Finally, this tells us that the gradient is

$$\nabla_\phi \ell(\phi,\theta_0,\theta_1) = X^T(\vec{z}-\vec{g}) \quad \text{where} \quad \vec{g}_i = g(\phi^T x^{(i)})$$

Based on the linear algebra review sheet, the Hessian Matrix can be derived by taking the derivative of the gradient, which is equivalent to looking at each ith entry of the gradient vector, taking the gradient of that entry, and setting that to be the ith column of the Hessian. The ith entry of the gradient vector is

$$x^{(i)T}\cdot(\vec{z}-\vec{g})$$

and after taking the derivative with respect to that ith entry we get

$$\nabla_\phi x^{(i)T}(\vec{z}-\vec{g}) = x^{(i)T}(\vec{g}\cdot(1-\vec{g}))\cdot x^{(i)}$$

Which gives the matrix

$$H = X^T D X \quad \text{where} \quad D_{ii} = g(\phi^T x^{(i)})(1-g(\phi^T x^{(i)}))$$

(b) Now suppose z is a latent (unobserved) random variable. Write the log-likelihood of the parameters, and derive an EM algorithm to maximize the log-likelihood. Clearly specify the E-step and M-step (again, the M-step will require a numerical solution, so find the appropriate gradients and Hessians).

answer:

The log-likelihood can now be written as

$$\begin{aligned}
\ell(\phi,\theta_0,\theta_1) &= \sum_{i=1}^{m}\log p(y^{(i)}|x^{(i)};\phi,\theta_0,\theta_1) \\
&= \sum_{i=1}^{m}\log p(y^{(i)}|x^{(i)},z^{(i)};\theta_0,\theta_1)p(z^{(i)}|x^{(i)};\phi) \\
&= \sum_{i=1}^{m}\log\left(\left(1-g(\phi^T x^{(i)})\right)^{1-z^{(i)}}\left(\frac{1}{\sqrt{2\pi}\sigma}\exp\left(\frac{-(y^{(i)}-\theta_k^T x^{(i)})^2}{2\sigma^2}\right)\right)\right) \\
&+ \log\left(\left(g(\phi^T x^{(i)})^{z^{(i)}}\right)\left(\frac{1}{\sqrt{2\pi}\sigma}\exp\left(\frac{-(y^{(i)}-\theta_1^T x^{(i)})^2}{2\sigma^2}\right)\right)\right)
\end{aligned}$$

Since each $z^{(i)}$ is unobserved they cannot be separated explicitly into two cases - this is because we have no way of knowing how to separate the summation into two groups for each cluster of $z^{(i)}$ values when their values are hidden.

In order to derive the new EM algorithm, instead of explicitly maximizing ℓ we will use the same approach as explained in class which will be to repeatedly construct a lower-bound on ℓ (E-step), and then optimize that lower-bound (M-step).

In order to accomplish this, we first define Q_i to be the distribution over z_i for all i (implying $\sum_z Q_i(z) = 1$ and $Q_i(z) \geq 0$). Next we write the following:

$$\begin{aligned}
\sum_{i=1}^{m} \log p(y^{(i)}|x^{(i)};\phi,\theta_0,\theta_1) &= \sum_{i=1}^{m} \log \sum_{z^{(i)}} Q_i(z^{(i)}) \frac{p(y^{(i)}|x^{(i)},z^{(i)};\theta_0,\theta_1)p(z^{(i)}|x^{(i)};\phi)}{Q_i(z^{(i)})} && (1) \\
&\geq \sum_{i=1}^{m} \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(y^{(i)}|x^{(i)},z^{(i)};\theta_0,\theta_1)p(z^{(i)}|x^{(i)};\phi)}{Q_i(z^{(i)})} && (2)
\end{aligned}$$

Where step (2) utilizes Jensen's Inequality.

We note here that the term

$$\sum_{z^{(i)}} Q_i(z^{(i)}) \frac{p(y^{(i)}|x^{(i)},z^{(i)};\theta_0,\theta_1)p(z^{(i)}|x^{(i)};\phi)}{Q_i(z^{(i)})}$$

is just the expectation over the quantity

$$\frac{p(y^{(i)}|x^{(i)},z^{(i)};\theta_0,\theta_1)p(z^{(i)}|x^{(i)};\phi)}{Q_i(z^{(i)})}$$

according to the distribution Q_i. By Jensen's Inequality we know that

$$f\left(\mathop{\mathbb{E}}_{z^{(i)}\sim Q_i}\left[\frac{p(y^{(i)}|x^{(i)},z^{(i)};\theta_0,\theta_1)p(z^{(i)}|x^{(i)};\phi)}{Q_i(z^{(i)})}\right]\right) \geq \mathop{\mathbb{E}}_{z^{(i)}\sim Q_i}\left[f\left(\frac{p(y^{(i)}|x^{(i)},z^{(i)};\theta_0,\theta_1)p(z^{(i)}|x^{(i)};\phi)}{Q_i(z^{(i)})}\right)\right]$$

Since (2) holds true for any set of distributions Q_i we can choose a specific distribution for Q_i which makes the inequality hold with equality at ϕ, θ_0, θ_1. Since Jensen's Inequality holds with equality over a constant valued random variable it suffices to find $Q_i(z^{(i)}) = c \times p(y^{(i)}|x^{(i)},z^{(i)};\theta_0,\theta_1)p(z^{(i)}|x^{(i)};\phi)$ for some $c \in \mathbb{R}, c \neq 0$. We can solve for $Q_i(z^{(i)})$ using many iterations of Bayes' Theorem as follows:

$$\begin{aligned}
Q_i(z^{(i)}) &= \frac{p(y^{(i)}|x^{(i)},z^{(i)};\theta_0,\theta_1)p(z^{(i)}|x^{(i)};\phi)}{\sum_{z^{(i)}} p(y^{(i)}|x^{(i)},z^{(i)};\theta_0,\theta_1)p(z^{(i)}|x^{(i)};\phi)} \\
&= \frac{p(y^{(i)}|x^{(i)},z^{(i)};\theta_0,\theta_1)p(z^{(i)}|x^{(i)};\phi)}{p(y^{(i)}|x^{(i)};\theta_0,\theta_1)} \\
&= \frac{p(y^{(i)}|x^{(i)},z^{(i)};\theta_0,\theta_1)p(z^{(i)},x^{(i)};\phi)}{p(y^{(i)}|x^{(i)};\theta_0,\theta_1)p(x^{(i)};\phi)} \\
&= \frac{p(y^{(i)},x^{(i)},z^{(i)};\phi,\theta_0,\theta_1)}{p(y^{(i)},x^{(i)};\phi,\theta_0,\theta_1)} \\
&= p(z^{(i)}|x^{(i)},y^{(i)};\phi,\theta_0,\theta_1)
\end{aligned}$$

With this solved we can now write the following EM algorithm

(E-step) For each i, j set

$$w_j^{(i)} := p(z^{(i)} = j | x^{(i)}, y^{(i)}; \phi, \theta_j)$$

(M-step) Update the parameters of

$$\sum_{i=1}^{m} \sum_j w_j^{(i)} \log \frac{p(y^{(i)}|x^{(i)}, z^{(i)}; \theta_j) p(z^{(i)}|x^{(i)}; \phi)}{w_j^{(i)}}$$

In order to maximize the equation in the M-step with respect to the parameters ϕ, θ_0, θ_1 we need to take the gradient with respect to each parameter and set the resulting expression to zero. We can first generalize the θs as one gradient and solve

$$\nabla_{\theta_k} \sum_{i=1}^{m} \sum_j w_j^{(i)} \log \frac{p(y^{(i)}|x^{(i)}, z^{(i)}; \theta_j) p(z^{(i)}|x^{(i)}; \phi)}{w_j^{(i)}} \overset{set}{=} 0$$

After dividing out all the non variable multipliers and taking the gradient we are left with the equation

$$\begin{aligned}
&\sum_{i=1}^{m} -2(y^{(i)} - \theta_k^T x^{(i)}) x^{(i)T} w_k^{(i)} = 0 \\
\implies &\sum_{i=1}^{m} x^{(i)T} w_k^{(i)} y^{(i)} - x^{(i)T} w_k^{(i)} \theta_k^T x^{(i)} = 0 \\
\implies &\sum_{i=1}^{m} x^{(i)T} w_k^{(i)} \theta_k^T x^{(i)} = \sum_{i=1}^{m} x^{(i)T} w_k^{(i)} y^{(i)} \\
\implies &\sum_{i=1}^{m} x^{(i)T} w_k^{(i)} \theta_k^T x^{(i)} = \sum_{i=1}^{m} x^{(i)T} w_k^{(i)} y^{(i)} \\
\implies &\theta_k^T \sum_{i=1}^{m} x^{(i)T} w_k^{(i)} x^{(i)} = \sum_{i=1}^{m} x^{(i)T} w_k^{(i)} y^{(i)} \\
\implies &\theta_k^T = \sum_{i=1}^{m} (x^{(i)T} w_k^{(i)} x^{(i)})^{-1} x^{(i)T} w_k^{(i)} y^{(i)} \\
\implies &\theta_k = (X^T W X)^{-1} X^T W y \quad \text{where} \quad W_{i,i} = w_k^{(i)}
\end{aligned}$$

Finally we must derive the gradient vector and Hessian matrix

$$\begin{aligned}
&\nabla_\phi \sum_{i=1}^{m} \sum_j w_j^{(i)} \log \frac{p(y^{(i)}|x^{(i)}, z^{(i)}; \theta_j) p(z^{(i)}|x^{(i)}; \phi)}{w_j^{(i)}} \\
=&\nabla_\phi \sum_{i=1}^{m} w_0^{(i)} \log(1 - g(\phi^T x^{(i)})) + w_1^{(i)} \log((g(\phi^T x^{(i)}))
\end{aligned}$$

Since $w_0^{(i)} = 1 - w_1^{(i)}$ we have

$$\begin{aligned}
&\nabla_\phi \sum_{i=1}^{m} (1 - w_1^{(i)}) \log(1 - g(\phi^T x^{(i)})) + w_1^{(i)} \log((g(\phi^T x^{(i)})) \\
=&\sum_i x^{(i)} (w_1^{(i)} - g(\phi^T x^{(i)}))
\end{aligned}$$

The matrix representation turns out to be

$$\nabla_\phi = X^T(\vec{w} - \vec{g}) \quad \text{where} \quad \vec{g}_i = g(\phi^T x^{(i)})$$

Finding the Hessian for ϕ in part (b), yields the same result as part (a)

$$H = X^T D X \quad \text{where} \quad D_{i,i} = g(\phi^T x^{(i)})(1 - g(\phi^T x^{(i)}))$$

Problem 2

Factor Analysis and PCA
In this problem we look at the relationship between two unsupervised learning algorithms we discussed in class: Factor Analysis and Principle Component Analysis.
Consider the following joint distribution over (x, z) where $z \in \mathbb{R}^k$ is a latent random variable

$$\begin{aligned} z &\sim \mathcal{N}(0, I) \\ x|z &\sim \mathcal{N}(Uz, \sigma^2 I) \end{aligned}$$

where $U \in \mathbb{R}^k$ is a model parameters and σ^2 is assumed to be a known constant. This model is often called Probabilistic PCA. Note that this is nearly identical to the factor analysis model except we assume that the variance of $x|z$ is a known scaled identity matrix rather than the diagonal parameter matrix Φ, and we do not add an additional μ term to the mean (though this last difference is just for simplicity of presentation). However, as we will see, it turns out that as $\sigma^2 \to 0$, this model is equivalent to PCA.
For simplicity, you can assume for the remainder of the problem that $k = 1$, i.e., that U is a column vector in $\mathbb{R}^n$.

(a) Use the rules for manipulating Gaussian distributions to determine the joint distribution over (x, z) and the conditional distribution of $z|x$. [Hint: for later parts of this problem, it will help significantly if you simplify your solution for the conditional distribution using the identity we first mentioned in problem set 1: $(I + BA)^{-1}B = B(I + AB)^{-1}$.]

answer:

According to the definition of x, the expectation of x is

$$E[x] = E[Uz + \epsilon] = UE[z] + E[\epsilon] = 0$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ indicates the added covariance noise. We can now obtain the four different components of the matrix Σ using the substitution $x = Uz + \epsilon$

$$\begin{aligned} \Sigma_{zz} &= E[zz^T] = I \\ \Sigma_{xz} &= E[(Uz + \epsilon)z^T] = (UE[zz^T] + E[\epsilon z^T]) = U \\ \Sigma_{zx} &= E[z(Uz + \epsilon)^T] = U^T E[zz^T] + E[z\epsilon^T] = U^T \\ \Sigma_{xx} &= E[(Uz + \epsilon)(Uz + \epsilon)^T] = UU^T E[zz^T] + UE[z\epsilon^T] + U^T E[\epsilon z^T] + E[\epsilon\epsilon^T] = UU^T + \sigma^2 I \end{aligned}$$

Therefore our join distribution is written

$$\begin{bmatrix} z \\ x \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} I & U^T \\ U & UU^T + \sigma^2 I \end{bmatrix} \right)$$

Then we can apply the formula to define the conditional distribution $z|x \sim \mathcal{N}(\mu_{z|x}, \Sigma_{z|x})$ such that

$$\begin{aligned}
\mu_{z|x} =& \mu_z + \Sigma_{zx}\Sigma_{xx}^{-1}x = U^T(UU^T + \sigma^2 I)^{-1}x = \frac{U^T x}{UU^T + \sigma^2} \\
\Sigma_{z|x} =& \Sigma_{zz} - \Sigma_{zx}\Sigma_{xx}^{-1}\Sigma_{xz} = I - U^T(UU^T + \sigma^2 I)^{-1}U = 1 - \frac{U^T U}{UU^T + \sigma^2}
\end{aligned}$$

In conclusion,

$$z|x \quad \sim \quad \mathcal{N}\left(\frac{U^T x}{UU^T + \sigma^2}, 1 - \frac{U^T U}{UU^T + \sigma^2}\right)$$

(b) Using these distributions, derive an EM algorithm for the model. Clearly state the E-step and the M-step of the algorithm.

answer:

For the E-step, we need to compute $Q_i(z^{(i)})$ where

$$Q_i(z^{(i)}) = p(z^{(i)}|x^{(i)}; U)$$

For the M-step we need to maximize the equation

$$\begin{aligned}
&\sum_{i=1}^{m} \int_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)} : U)}{Q_i(z^{(i)})} \\
=&\sum_{i=1}^{m} \mathop{\mathbb{E}}_{z^{(i)} \sim Q_i} [\log p(x^{(i)}|z^{(i)}; U) + \log p(z^{(i)}) - \log(Q_i(z^{(i)}))]
\end{aligned}$$

Then we take the gradient with respect to U and set it to 0 in order to maximize the equation above.

$$\begin{aligned}
&\sum_{i=1}^{m} \mathop{\mathbb{E}}_{z^{(i)} \sim Q_i} [\log p(x^{(i)}|z^{(i)}; U)] \\
=&\sum_{i=1}^{m} \mathop{\mathbb{E}}_{z^{(i)} \sim Q_i} \left[\log \frac{1}{\sqrt{2\pi^n}|\sigma^2 I|} \exp\left(-\frac{1}{2\sigma^2}(x^{(i)} - Uz^{(i)})^T(x^{(i)} - Uz^{(i)})\right)\right] \\
=&\sum_{i=1}^{m} \mathop{\mathbb{E}}_{z^{(i)} \sim Q_i} \left[\frac{1}{2}\log|\sigma^2 I| - \frac{n}{2}\log(2\pi) - \frac{1}{2\sigma^2}(x^{(i)} - Uz^{(i)})^T(x^{(i)} - Uz^{(i)})\right]
\end{aligned}$$

We can drop the terms not dependent on U before taking the gradient.

$$
\begin{aligned}
&\nabla_U \sum_{i=1}^{m} \mathop{\mathbb{E}}_{z^{(i)}\sim Q_i}\left[\frac{1}{2\sigma^2}(x^{(i)} - Uz^{(i)})^T(x^{(i)} - Uz^{(i)})\right] \\
&= -\frac{1}{2\sigma^2}\sum_{i=1}^{m} \nabla_U \mathop{\mathbb{E}}_{z^{(i)}\sim Q_i}\left[\operatorname{tr} {z^{(i)}}^T U^T U z^{(i)} - 2\operatorname{tr} {z^{(i)}}^T U x^{(i)}\right] \\
&= -\frac{1}{2\sigma^2}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{z^{(i)}\sim Q_i}\left[2z^{(i)}{z^{(i)}}^T U^T - 2{z^{(i)}}^T x^{(i)}\right] \\
&= -\frac{1}{\sigma^2}\sum_{i=1}^{m}\left[U \mathop{\mathbb{E}}_{z^{(i)}\sim Q_i}\left[z^{(i)}{z^{(i)}}^T\right] - x^{(i)} \mathop{\mathbb{E}}_{z^{(i)}\sim Q_i}\left[{z^{(i)}}^T\right]\right] \overset{\text{set}}{=} 0 \\
\Longrightarrow U &= \left(\sum_{i=1}^{m} x^{(i)} \mathop{\mathbb{E}}_{z^{(i)}\sim Q_i}\left[{z^{(i)}}^T\right]\right)\left(\sum_{i=1}^{m} \mathop{\mathbb{E}}_{z^{(i)}\sim Q_i}\left[z^{(i)}{z^{(i)}}^T\right]\right)^{-1}
\end{aligned}
$$

From the definition of Q_i being Gaussian we know that

$$
\begin{aligned}
\mathop{\mathbb{E}}_{z^{(i)}\sim Q_i}\left[{z^{(i)}}^T\right] &= \mu^T_{z^{(i)}|x^{(i)}} \\
\mathop{\mathbb{E}}_{z^{(i)}\sim Q_i}\left[z^{(i)}{z^{(i)}}^T\right] &= \mu_{z^{(i)}|x^{(i)}}\mu^T_{z^{(i)}|x^{(i)}}
\end{aligned}
$$

Therefore, the final update of the M-step is

$$
U = \left(\sum_{i=1}^{m} x^{(i)}\mu^T_{z^{(i)}|x^{(i)}}\right)\left(\sum_{i=1}^{m}\mu_{z^{(i)}|x^{(i)}}\mu^T_{z^{(i)}|x^{(i)}}\right)^{-1}
$$

(c) As $\sigma^2 \to 0$, show that if the EM algorithm convergences to a parameter vector U^* (and such convergence is guaranteed by the argument presented in class), then U^* must be an eigenvector of the sample covariance matrix $\Sigma = \frac{1}{m}\sum x^{(i)}{x^{(i)}}^T$ — i.e., U^* must satisfy

$$
\lambda U^* = \Sigma U^*
$$

answer:

The first thing to notice is that as $\sigma^2 \to 0$ the E-step only needs to compute the means and not the variances, since $\Sigma_{z|x} \to 0$ as well. If we let $w \in \mathbb{R}^m$ be the vector that contains all the means such that $w_i = \mu_{z^{(i)}|x^{(i)}}$ then we can compute the new E-step to be

$$
\begin{aligned}
w_i = \mu_{z^{(i)}|x^{(i)}} &= U^T(\sigma^2 I - UU^T)^{-1}(Uz^{(i)} + \epsilon) && (3) \\
&= U^T(UU^T)^{-1}x^{(i)} && (4) \\
&= \frac{U^T x^{(i)}}{UU^T} && (5) \\
&= \frac{{x^{(i)}}^T U}{U^T U} && (6)
\end{aligned}
$$

Where step (6) uses the fact that step (5) yields a Real. When this is written in matrix form we see that

$$
w = \frac{XU}{U^T U}
$$

Next we need to derive the new M-step. We can use the result from (b) in order to obtain the new update rule

$$U = \left(\sum_{i=1}^{m} x^{(i)} \mathop{\mathbb{E}}_{z^{(i)} \sim Q_i} \left[z^{(i)T}\right]\right) \left(\sum_{i=1}^{m} \mathop{\mathbb{E}}_{z^{(i)} \sim Q_i} \left[z^{(i)} z^{(i)T}\right]\right)^{-1} \tag{7}$$

$$= \left(\sum_{i=1}^{m} x^{(i)} \mu^T_{z^{(i)}|x^{(i)}}\right) \left(\sum_{i=1}^{m} \mu_{z^{(i)}|x^{(i)}} \mu^T_{z^{(i)}|x^{(i)}}\right)^{-1} \tag{8}$$

$$= \left(\sum_{i=1}^{m} x^{(i)} w_i\right) \left(\sum_{i=1}^{m} w_i w_i\right)^{-1} \tag{9}$$

Where step (8) is derived from the definition of Q_i being Guassian with mean $\mu_{z^{(i)}|x^{(i)}}$ and variance $\Sigma_{z^{(i)}|x^{(i)}}$ yielding

$$\mathop{\mathbb{E}}_{z^{(i)} \sim Q_i} \left[z^{(i)T}\right] = \mu^T_{z^{(i)}|x^{(i)}}$$

$$\mathop{\mathbb{E}}_{z^{(i)} \sim Q_i} \left[z^{(i)} z^{(i)T}\right] = \mu_{z^{(i)}|x^{(i)}} \mu^T_{z^{(i)}|x^{(i)}} + \Sigma_{z^{(i)}|x^{(i)}}$$

Now to write expression (9) in matrix form we see that

$$\left(\sum_{i=1}^{m} w_i w_i\right)^{-1} = \frac{1}{w^T w}$$

however to write the numerator in matrix form will be a little more difficult. We want to represent that each column of X from $i \in 1, ..., m$ gets multiplied by its the ith entry of w and then summed together. We cannot simply write this as $X^T w$ so we will instead have to construct a new diagonal matrix W where $W_{i,i} = w_i$. Then, by multiplying XW we get

$$XW = \begin{bmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \dots & x^{(m)} \\ | & | & & | \end{bmatrix}_{n \times m} \begin{bmatrix} w_1 & & & \\ & w_2 & & \\ & & \ddots & \\ & & & w_m \end{bmatrix}_{m \times m}$$

$$= \begin{bmatrix} | & | & & | \\ w_1 x^{(1)} & w_2 x^{(2)} & \dots & w_m x^{(m)} \\ | & | & & | \end{bmatrix}_{n \times m}$$

Next we just need to sum up each column of XW which we can accomplish by constructing new vector $\vec{1} \in \mathbb{R}^m$ such that $\vec{1}_i = 1$ for all $i \in 1, ..., m$ and transforming it according to XW as follows

$$XW\vec{1} = \begin{bmatrix} | & | & & | \\ w_1 x^{(1)} & w_2 x^{(2)} & \dots & w_m x^{(m)} \\ | & | & & | \end{bmatrix}_{n \times m} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{m \times 1}$$

$$= \left(\sum_{i=1}^{m} x^{(i)} w_i\right)$$

Finally we can write the expression for U in matrix form

$$U = \frac{XW\vec{1}}{w^T w}$$

We know that U has converged when its value has not changed after the E-step, so we can substitute our value for w in the E-step and set it equal to our original U.

$$\begin{aligned}
&U = \frac{XW\vec{1}}{\frac{U^TX^TXU}{(U^TU)^2}} \\
&\implies (U^TX^TXU)U = (U^TU)^2XW\vec{1} \\
&\implies X^TXU = U^TUXW\vec{1} \\
&\implies X^TXU = UU^TXW\vec{1} = U^TXW\vec{1}U \\
&\implies \Sigma U = \lambda U
\end{aligned}$$

where we know that $\Sigma = X^TX$ and we take $\lambda = U^TXW\vec{1} \in \mathbb{R}$ and therefore proving the desired result.

Problem 4

Convergence of Policy Iteration
In this problem we show that the Policy Iteration algorithm, described in the lecture notes, is guaranteed to find the optimal policy for an MDP. First, define B^π to be the Bellman operator for policy π, defined as follows: if $V' = B(V)$, then

$$V'(s) = R(s) + \gamma \sum_{s' \in \mathcal{S}} P_{s,\pi(s)}(s')V(s')$$

(a) Prove that if $V_1(s) \leq V_2(s)$ for all $s \in \mathcal{S}$, then $B(V_1)(s) \leq B(V_2)(s)$ for all $s \in \mathcal{S}$.

answer:

Given that $V_1(s) \leq V_2(s)$ for all $s \in \mathcal{S}$, we can also write the inequality

$$\begin{aligned}
&R(s) + V_1(s') \leq R(s) + V_2(s') \\
\implies &R(s) + \gamma \sum_{s' \in \mathcal{S}} P_{s,\pi(s)}(s')V_1(s') \leq R(s) + \gamma \sum_{s' \in \mathcal{S}} P_{s,\pi(s)}(s')V_2(s')
\end{aligned}$$

However, it turns out that

$$\begin{aligned}
R(s) + \gamma \sum_{s' \in \mathcal{S}} P_{s,\pi(s)}(s')V_1(s') = B(V_1)(s) \\
R(s) + \gamma \sum_{s' \in \mathcal{S}} P_{s,\pi(s)}(s')V_2(s') = B(V_2)(s)
\end{aligned}$$

So we have shown that $B(V_1)(s) \leq B(V_2)(s)$ for all $s \in \mathcal{S}$

(b) Prove that for any V,

$$||B^\pi(V) - V^\pi||_\infty \leq \gamma||V - V^\pi||_\infty$$

where $||V||_\infty = \max_{s \in \mathcal{S}} |V(s)|$. Intuitively, this means that applying the Bellman operator B^π to any value function V, brings that value function "closer" to the value function for π, V^π. This also means that applying B^π repeatedly (an infinite number of times):

$$B^\pi(B^\pi(B^\pi \ldots B^\pi(V) \ldots)))$$

will result in the value function V^π.

answer:

We can first observe that

$$\begin{aligned} ||B^\pi(V) - V^\pi||_\infty &= \max_{s\in\mathcal{S}} \big| R(s) + \gamma \sum_{s'\in\mathcal{S}} P_{s,\pi(s)}(s')V(s') - R(S) - \gamma \sum_{s'\in\mathcal{S}} P_{s,\pi(s)}(s')V^\pi(s') \big| \\ &= \gamma \max_{s\in\mathcal{S}} \big| \sum_{s'\in\mathcal{S}} P_{s,\pi(s)}(s')(V(s') - V^\pi(s')) \big| \end{aligned}$$

Then since $P_{s,a}$ is a probability distribution it must be true that

$$P_{s,\pi(s)}(s') \geq 0 \tag{10}$$

$$\sum_{s'\in\mathcal{S}} P_{s,\pi(s)}(s') = 1 \tag{11}$$

we can apply the fact that for any $a, x \in \mathbb{R}^n$, if $\sum_i a_i = 1$, and $a_i \geq 0$, then $\sum_i a_i x_i \leq \max_i x_i$ to (10) and (11) to write

$$\gamma \max_{s\in\mathcal{S}} \big| \sum_{s'\in\mathcal{S}} P_{s,\pi(s)}(s')(V(s') - V^\pi(s')) \big| \leq \gamma \max_{s'\in\mathcal{S}} |V(s') - V^\pi(s')|$$

Which is, by definition, the same inequality as

$$||B^\pi(V) - V^\pi||_\infty \leq \gamma ||V - V^\pi||_\infty$$

(c) Now suppose that we have some policy π, and use Policy Iteration to choose a new policy $\pi\prime$ according to

$$\pi'(s) = \arg\max_{a\in A} \sum_{s'\in S} P_{s,a}(s')V^\pi(s')$$

Show that this policy will never perform worse that the previous one — i.e., show that for all $s \in \mathcal{S}$, $V^\pi(s) \leq V^{\pi'}(s)$.

answer:

We know that

$$\begin{aligned} V^\pi(s) &= R(s) + \gamma \sum_{s'\in\mathcal{S}} P_{s,\pi(s)}(s')V^\pi(s') \\ &\leq \\ B^{\pi'}(V^\pi)(s) &= R(S) + \gamma \sum_{s'\in\mathcal{S}} P_{s,\pi'(s)}(s')V^\pi(s') \end{aligned}$$

due to the fact that

$$\gamma \sum_{s'\in\mathcal{S}} P_{s,\pi(s)}(s')V^\pi(s') \leq \gamma \max_{a\in A} \sum_{s'\in\mathcal{S}} P_{s,a}(s')V^\pi(s') = \gamma \sum_{s'\in\mathcal{S}} P_{s,\pi'(s)}(s')V^\pi(s')$$

We can further expand this to show

$$\begin{aligned}
B^{\pi'}(V^{\pi})(s) &= R(S) + \gamma \sum_{s' \in \mathcal{S}} P_{s,\pi'(s)}(s') V^{\pi}(s') \\
&= R(S) + \gamma \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P_{s,a}(s') V^{\pi}(s') \\
&= R(S) + \gamma \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P_{s,a}(s')(R(s') + \gamma \sum_{s'' \in \mathcal{S}} P_{s',\pi(s')}(s'') V^{\pi}(s'')) \\
&\leq R(S) + \gamma \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P_{s,a}(s')(R(s') + \gamma \max_{a' \in \mathcal{A}} \sum_{s'' \in \mathcal{S}} P_{s',a'}(s'') V^{\pi}(s'')) \\
&= B^{\pi'}(B^{\pi'}(V^{\pi}))(s)
\end{aligned}$$

Using part (b) we also know that

$$||B^{\pi}(V) - V^{\pi}||_{\infty} \leq \gamma ||V - V^{\pi}||_{\infty}$$

implies that applying B^{π} repeatedly (an infinite number of times) results in the convergent identity

$$B^{\pi'}(B^{\pi'}(B^{\pi'} \ldots B^{\pi'}(V) \ldots))) = V^{\pi'}$$

which finally proves the desired result that $V^{\pi}(s) \leq V^{\pi'}(s)$ for all $s \in \mathcal{S}$.

(d) Use the proceeding exercises to show that policy iteration will eventually converge (i.e., produce a policy $\pi' = \pi$). Furthermore, show that it must converge to the optimal policy π^*. For the later part, you may use the property that if some value function satisfies

$$V(s) = R(s) + \gamma \max_{a \in A} \sum_{s' \in \mathcal{S}} P_{s,a}(s') V(s')$$

then $V = V^*$.

answer:

Given that $|S| + |A| < \infty$ (i.e. that the number of states and the number of actions are both finite), we let π_n be the resulting policy after the nth iteration of the Policy Iteration, and let

$$\lim_{n \longrightarrow \infty} \pi_n = \pi_{\infty}$$

We know from part (c) that $V^{\pi}(s) \leq V^{\pi'}(s)$ for all s, where π' is the resulting policy after one iteration of Policy Iteration, therefore we have that

$$V^{\pi_n}(s) \leq V^{\pi_{\infty}} \quad \forall s \in \mathcal{S}, \quad \forall n < \infty$$

However, since there are only $k = |A|^{|S|}$ possibly policies it must be the case that

$$\lim_{n \longrightarrow \infty} \pi_n = \pi_k$$

Therefore Policy Iteration converges to π_k and further,

$$\begin{aligned}
V^{\pi_k}(s) =& R(s) + \gamma \sum_{s' \in \mathcal{S}} P_{s,\pi_{|A|^{|S|}}}(s') V^{\pi_k} \\
=& R(s) + \gamma \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P_{s,a}(s') V^{\pi_k}
\end{aligned}$$

which proves that $V^{\pi_k} = V^*$