1. **[5 points] EM for Probabilistic PCA.** In lecture, we covered the EM algorithm applied to mixture of Gaussians models. In this question, we’ll look at another interesting example of EM but where the latent variables are continuous: probabilistic PCA. This is a model very similar in spirit to PCA: we have data in a high-dimensional space, and we’d like to summarize it with a lower-dimensional representation. Unlike ordinary PCA, we formulate the problem in terms of a probabilistic model. We assume the latent code vector $z$ is drawn from a standard Gaussian distribution $\mathcal{N}(0, I)$, and that the observations are drawn from a spherical Gaussian whose mean is a linear function of $z$. We’ll consider the slightly simplified case of scalar-valued $z$ (i.e. only one principal component). The probabilistic model is given by:

$$
\begin{align*}
  z &\sim \mathcal{N}(0, 1) \\
  x \mid z &\sim \mathcal{N}(zu, \sigma^2 I),
\end{align*}
$$

where $\sigma^2$ is the noise variance (which we assume to be fixed) and $u$ is a parameter vector (which, intuitively, should correspond to the top principal component). Note that the observation model can be written in terms of coordinates:

$$
x_j \mid z \sim \mathcal{N}(zu_j, \sigma^2).
$$

We have a set of observations $\{x^{(i)}\}_{i=1}^N$, and $z$ is a latent variable, analogous to the mixture component in a mixture-of-Gaussians model.

In this question, you’ll derive both the E-step and the M-step for the EM algorithm.

(a) **E-step (2 points).** In this step, your job is to calculate the statistics of the posterior distribution $q(z) = p(z \mid x)$ which you’ll need for the M-step. In particular, your job is to find formulas for the (univariate) statistics:

$$
\begin{align*}
  m &= \mathbb{E}[z \mid x] = \\
  s &= \mathbb{E}[z^2 \mid x] =
\end{align*}
$$

**Tips:**

- First determine the conditional distribution $p(z \mid x)$ using the Gaussian conditioning formulas from the Appendix. To help you check your work: $p(z \mid x)$ is a univariate Gaussian distribution whose mean is a linear function of $x$, and whose variance does not depend on $x$. 

---

• Once you’ve determined the conditional distribution (and hence the posterior mean and variance), use the fact that \( \text{Var}(Y) = \mathbb{E}[Y^2] - \mathbb{E}[Y]^2 \) for any random variable \( Y \).

(b) **M-step (3 points).** In this step, we need to re-estimate the parameters, which consist of the vector \( \mathbf{u} \). (Recall that we’re treating \( \sigma \) as fixed.) Your job is to derive a formula for \( \mathbf{u}_{\text{new}} \) that maximizes the expected log-likelihood, i.e.,

\[
\mathbf{u}_{\text{new}} \leftarrow \arg \max_{\mathbf{u}} \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{q(z(i))} [\log p(z(i), x(i))].
\]

(Recall that \( q(z) \) is the distribution computed in part (a).) This is the new estimate obtained by the EM procedure, and will be used again in the next iteration of the E-step. Your answer should be given in terms of the \( m(i) \) and \( s(i) \) from the previous part. (I.e., you don’t need to expand out the formulas for \( m(i) \) and \( s(i) \) in this step, because if you were implementing this algorithm, you’d use the values \( m(i) \) and \( s(i) \) that you previously computed.)

**Tips:**
- First expand out \( \log p(z(i), x(i)) \). You’ll find that a lot of the terms don’t depend on \( \mathbf{u} \) and can therefore be dropped.
- Apply linearity of expectation. You should wind up with terms proportional to \( \mathbb{E}_{q(z(i))}[z(i)] \) and \( \mathbb{E}_{q(z(i))}[z(i)^2] \). Replace these expectations with \( m(i) \) and \( s(i) \). You should get an equation that does not mention \( z(i) \). (If you don’t wind up with terms of this form, then see if there’s some way you can simplify \( \log p(z(i), x(i)) \).
- In order to find the maximum likelihood parameter \( \mathbf{u}_{\text{new}} \), you need to determine the gradient with respect to \( \mathbf{u} \), set it to zero, and solve for \( \mathbf{u}_{\text{new}} \).

2. **[2 points] Contraction Maps.** In lecture, we showed that the optimal Bellman backup operator is a contraction map, and hence that value iteration converges to the optimal \( Q \)-function \( Q^* \). Now consider the problem of *policy evaluation*, i.e. finding the \( Q \)-function \( Q^\pi \) for a given (stochastic) policy \( \pi \). Since \( Q^\pi \) is characterized by the fixed-point equation \( T^\pi Q^\pi = Q^\pi \), we can repeatedly apply the update

\[
Q_{k+1} \leftarrow T^\pi Q_k,
\]

which can be written out in full as:

\[
Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s'} \mathbb{P}(s'|a, s) \sum_{a'} \pi(a'|s') Q_k(s', a').
\]

Show that the Bellman backup operator \( T^\pi \) is a contraction map in the \( \| \cdot \|_\infty \) norm. Your proof will probably look very similar to the one from Slide 30 of Lecture 10, but be sure to justify each step.

3. **[3 points] Q-Learning.** In lecture, we made the claim that Q-learning only converges to the optimal \( Q \)-function if the agent follows an exploration-encouraging strategy such as \( \varepsilon \)-greedy. Your job is to give a counterexample to show that exploration is necessary. I.e., you will show that Q-learning might get stuck with a suboptimal \( Q \)-function if it always chooses \( \pi(s) = \arg \max_a Q(s, a) \).
Consider an MDP with two states $s_1$ and $s_2$, and two actions, \textbf{Stay} and \textbf{Switch}. The environment is deterministic. If the agent chooses \textbf{Stay}, then it stays in the current state (i.e. $S_{t+1} = S_t$), while if it chooses \textbf{Switch}, it switches to the other state (i.e., if it’s in $s_1$, it transitions to $s_2$, and vice versa). The reward function is given by:

$$r(S, A) = \begin{cases} 
1 & \text{if } S = s_1 \\
2 & \text{if } S = s_2 
\end{cases}$$

The discount factor is $\gamma = 0.9$.

(a) \textbf{(1 point)} Determine the optimal policy and the Q-function for the optimal policy. You should give the Q-function as a table. You don’t need to show your work or justify your answer for this part.

(b) \textbf{(2 points)} Now suppose we apply Q-learning, except that instead of the $\epsilon$-greedy policy, the agent follows the greedy policy which always chooses $\pi(s) = \arg \max_a Q(s, a)$. Assume the agent starts in state $S_0 = s_1$. Give an example of a Q-function that is in equilibrium (i.e. it will never change after the Q-learning update rule is applied), but which results in a suboptimal policy. (You should specify the Q-function as a table.) Justify your answer.
Appendix: Some Properties of Gaussians

Consider a multivariate Gaussian random variable $z$ with the mean $\mu$ and the covariance matrix $\Sigma$. I.e.,

$$p(z) = \mathcal{N}(z | \mu, \Sigma).$$

Now consider another Gaussian random variable $x$, whose mean is an affine function of $z$ (in the form to be clear soon), and its covariance $S$ is independent of $z$. The conditional distribution of $x$ given $z$ is

$$p(x | z) = \mathcal{N}(x | Az + b, S).$$

Here the matrix $A$ and the vector $b$ are of appropriate dimensions.

In some problems, we are interested in knowing the distribution of $z$ given $x$, or the marginal distribution of $x$. One can apply Bayes’ rule to find the conditional distribution $p(z | x)$. After some calculations, we can obtain the following useful formulae:

$$p(x) = \mathcal{N}(x | A\mu + b, A\Sigma A^\top + S)$$

$$p(z | x) = \mathcal{N}(z | C(A^\top S^{-1}(x - b) + \Sigma^{-1}\mu), C)$$

with

$$C = (\Sigma^{-1} + A^\top S^{-1}A)^{-1}.$$

You may also find it helpful to read Section 2.3 of Bishop.