1.

Let u be a unit-length vector and $\alpha \in \mathbb{R}$. Then

$$\|x - \alpha u\|^2 = \|x\|^2 - 2\alpha \langle x, u \rangle + \alpha^2$$

For fixed x, u this is a quadratic expression in α, that takes its minimum iff $\alpha = \langle x, u \rangle$

Therefore:

$$f_u(x) = \underset{v \in V}{\operatorname{argmin}} \|x - v\|^2 = \langle x, u \rangle u.$$

Hense:

$$\sum_{i=1}^{n} \|x^{(i)} - f_u(x^{(i)})\|^2 = \sum_{i=1}^{n} \|x^{(i)}\|^2 - 2\langle x^{(i)}, f_u(x^{(i)}) \rangle + \|f_u(x^{(i)})\|^2$$

$$= \sum_{i=1}^{n} \|x^{(i)}\|^2 - 2\langle x^{(i)}, \langle x^{(i)}, u \rangle u \rangle + \|\langle x^{(i)}, u \rangle u\|^2$$

$$= \sum_{i=1}^{n} \|x^{(i)}\|^2 - 2\langle x^{(i)}, u\rangle^2 + \langle x^{(i)}, u\rangle^2$$

$$= \sum_{i=1}^{n} \|x^{(i)}\|^2 - \langle x^{(i)}, u\rangle^2$$

We conclude that:

$$\arg\min_{u: u^T u = 1} \sum_{i=1}^{n} \|x^{(i)} - f_u(x^{(i)})\|^2 = \arg\max_{u: u^T u = 1} \sum_{i=1}^{n} \langle x^{(i)}, u\rangle^2,$$

where the right side is exactly the definition of the first principal component, i.e the unit vector u that maximizes the variance of the projection of the data.

2. a

As we know from the lecture notes that

$$J(Q, \theta) := \sum_{i=1}^{n} \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})},$$

where Q is any set of distributions Q_i, we have the inequality

$$l_{unsup}(\theta) \geq J(Q, \theta),$$

in which equality holds iff

$$Q_i(z^{(i)}) = \phi(z^{(i)} | x^{(i)}; \theta).$$

Using J we can rewrite $\theta^{(t+1)}$ as:

$$\theta^{(t+1)} = \arg\max_{\theta} \left[J(Q^{(t)}, \theta) + \alpha l_{sup}(\theta) \right].$$

Therefore

$$l_{semi\text{-}sup}(\theta^{(t)}) = l_{unsup}(\theta^{(t)}) + \alpha l_{sup}(\theta^{(t)})$$

$$= J(Q^{(t)}, \theta^{(t)}) + \alpha l_{sup}(\theta^{(t)})$$

$$\leq J(Q^{(t)}, \theta^{(t+1)}) + \alpha l_{sup}(\theta^{(t+1)})$$

$$\leq l_{unsup}(\theta^{(t+1)}) + \alpha l_{sup}(\theta^{(t+1)})$$

$$= l_{semi\text{-}sup}(\theta^{(t+1)})$$

Hence, $l_{semi\text{-}sup}(\theta^{(t+1)}) \geq l_{semi\text{-}sup}(\theta^{(t)})$ and the algorithm will converge monotonically.

2.b.

In the E-step, we need to re-estimate the latent variables $z^{(i)}$'s, $i = 1, \dots, n$. We have:

$$w_j^{(i)} = p(z^{(i)} = j \mid x^{(i)}; \theta)$$

$$= \frac{p(z^{(i)} = j; \theta)\, p(x^{(i)} \mid z^{(i)} = j; \theta)}{\sum_{l=1}^{k} p(z^{(i)} = l; \theta)\, p(x^{(i)} \mid z^{(i)} = l, \theta)}$$

$$= \frac{\frac{1}{(2\pi)^{d/2} |\Sigma_j|^{1/2}} \exp\left\{-\frac{1}{2}(x^{(i)} - \mu_j)^T \Sigma_j^{-1} (x^{(i)} - \mu_j)\right\} \phi_j}{\sum_{l=1}^{k} \frac{1}{(2\pi)^{d/2} |\Sigma_l|^{1/2}} \exp\left\{-\frac{1}{2}(x^{(i)} - \mu_l)^T \Sigma_l^{-1} (x^{(i)} - \mu_l)\right\} \phi_l}.$$

2. C.

In the M-step, we re-estimate the model parameters μ_j's, Σ_j's and ϕ_j's, $j \in \{1, \dots, k\}$ so to maximize the log-likelihood function:

$$\sum_{i=1}^{n} \sum_{j=1}^{k} w_j^{(i)} \log \frac{p(x^{(i)}, z^{(i)} = j; \theta)}{w_j^{(i)}} + \alpha \sum_{i=1}^{\tilde{n}} \log p(\tilde{x}^{(i)}, \tilde{z}^{(i)}; \theta)$$

which is the same as maximizing

$$\sum_{i=1}^{n} \sum_{j=1}^{k} w_j^{(i)} \log p(x^{(i)}, z^{(i)} = j; \theta) + \alpha \sum_{i=1}^{\tilde{n}} \sum_{j=1}^{k} 1\{\tilde{z}^{(i)} = j\} \log p(\tilde{x}^{(i)}, \tilde{z}^{(i)}, \theta)$$

If we append the labeled dataset to the, we get the entire training set of $n + \tilde{n}$ examples $(x^{(i)}, z^{(i)})$ of which the first n are unlabeled and the rest $\tilde{n}$ index by

$\{n+1, \dots, n+\tilde{n}\}$ are labeled. Also denote for labeled examples $w_j^{(i)} = \alpha \cdot 1\{z^{(i)} = j\}$, $i \in \{n, \dots, n+\tilde{n}\}$. So the objective can be re-written as

$$\sum_{i=1}^{n} \sum_{j=1}^{k} w_j^{(i)} \log p(x^{(i)}, z^{(i)} = j; \theta) + \sum_{i=n+1}^{n+\tilde{n}} \sum_{j=1}^{k} w_j^{(i)} \log p(x^{(i)}, z^{(i)} = j; \theta)$$

$$= \sum_{i=1}^{n+\tilde{n}} \sum_{j=1}^{k} w_j^{(i)} \log p(x^{(i)}, z^{(i)} = j; \theta)$$

This is the same as the objective we need to maximize in the M-step of the classical GMM model (excluding additive constants) is equivalent to the one mentioned. By following the approach presented in the lecture notes, we can derive the update rule in the following manner:

$$\phi_j = \frac{1}{n+\alpha\tilde{n}} \sum_{i=1}^{n+\tilde{n}} w_j^{(i)}$$

$$\mu_j = \frac{\sum_{i=1}^{n+\tilde{n}} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{n+\tilde{n}} w_j^{(i)}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{n+\tilde{n}} w_j^{(i)} (x^{(i)} - \mu_j)^T (x^{(i)} - \mu_j)}{\sum_{i=1}^{n+\tilde{n}} w_j^{(i)}}$$

Note that $w_j^{(i)}$ is computed as conditional probability in the E-step for $i \in \{1, \ldots, n\}$ and is set to $\alpha \cdot 1\{z^{(i)} = j\}$ for $i \in \{n+1, \ldots, n+\tilde{n}\}$.

2. f

i.

The unsupervised model takes longer to converge.

ii.

Due to having no observed data to base its classifications on, it is to be expected that the unsupervised model does not label any data point consistently. The semi-supervised model does not have this problem, because it knows the correct labels of at least some examples.
But furthermore the unsupervised model even fails to create comparable partitions of the dataset, while the semi-

supervised model seems to always end up with the same partition.

iii.

The unsupervised model seems to have trouble to identify the high-variance distribution and to distinguish two of the low-variance distributions. The semi-supervised model does seem to find reasonable assignments everytime.

3.a.

In ICA, we want to maximize as a function of W the following objective:

$$\ell(W) = \sum_{i=1}^{n} \log p_x(x^{(i)})$$

$$= \sum_{i=1}^{n} \log\left(p_s(Wx^{(i)})|W|\right)$$

$$= \sum_{i=1}^{n} \log\left(\frac{1}{(2\pi)^{d/2}} \exp\left\{-\frac{1}{2}(Wx^{(i)})^T(Wx^{(i)})\right\}|W|\right)$$

$$= \sum_{i=1}^{n} \left(-\frac{d}{2}\log(2\pi) - \frac{1}{2}(x^{(i)})^T W^T W x^{(i)} + \log|W|\right)$$

To maximize this objective, we will compute its gradient and set to 0. We have:

$$\nabla_W \ell(W) = \sum_{i=1}^{n} \left(-\frac{1}{2} \nabla_W x^{(i)T} W^T W x^{(i)} + \nabla_W \log |W| \right)$$

$$= \sum_{i=1}^{n} \left(-W x^{(i)} x^{(i)T} + (W^{-1})^T \right)$$

$$= -W \left(\sum_{i=1}^{n} x^{(i)} x^{(i)T} \right) + n (W^{-1})^T$$

$$= -W X^T X + n (W^{-1})^T$$

Set this equal to 0, we can get: $W^T W = \left(\frac{1}{n} X^T X\right)^{-1}$, assuming that the right-hand side is invertible. Let $Y = \left(\frac{1}{n} X^T X\right)^{-1}$,

then Y is positive semi-definite. We can decompose $W = U\Sigma V^T$ where U, V are orthogonal and Σ is diagonal. Then, we have

$$W^T W = (V\Sigma U^T)(U\Sigma V^T) = V\Sigma(U^T U)\Sigma V^T = V\Sigma^2 V^T = Y$$

So, we can compute the eigendecomposition of Y to recover Σ^*, V^* and reconstruct $W = U\Sigma^* V^*$ with an arbitrary orthogonal matrix U. This arbitrary rotational component U can be determined from the data X which leads to ambiguity. The ICA can not recover the original sources.

3.6.

In general we have

$$(d_W \log g'(w_j^T x^{(i)}))V = \frac{g''(w_j^T x^{(i)})}{g'(w_j^T x^{(i)})} e_j^T V x^{(i)}$$

$$= \frac{g''(w_j^T x^{(i)})}{g'(w_j^T x^{(i)})} \operatorname{tr}\left((x^{(i)})^T e_j^T V\right)$$

and therefore

$$\nabla_W \log g'(w_j^T x^{(i)}) = \frac{g''(w_j^T x^{(i)})}{g'(w_j^T x^{(i)})} \left((x^{(i)})^T e_j^T\right)^T$$

$$= \frac{g''(w_j^T x^{(i)})}{g'(w_j^T x^{(i)})} e_j (x^{(i)})^T$$

So we can write the gradient of the log-likelihood (for a single example, i.e. $m=1$) as:

$$\nabla_W \ell(W) = (W^T)^{-1} + \sum_{j=1}^{n} \frac{g'(w_j^T x^{(i)})}{g''(w_j^T x^{(i)})} e_j (x^{(i)})^T$$

$$= (W^T)^{-1} + \begin{bmatrix} \frac{g'(w_1^T x^{(i)})}{g''(w_1^T x^{(i)})} \\ \vdots \\ \frac{g'(w_n^T x^{(i)})}{g''(w_n^T x^{(i)})} \end{bmatrix} (x^{(i)})^T$$

In particular for a Laplacian density $g'(s) = f_L(s) = \frac{1}{2}\exp(-|s|)$ we get the gradient update rule

$$\nabla_W \ell(W) = W + \alpha\left((W^T)^{-1} - \begin{bmatrix} \operatorname{sgn}(w_1^T x^{(i)}) \\ \vdots \\ \operatorname{sgn}(w_n^T x^{(i)}) \end{bmatrix} (x^{(i)})^T\right).$$

4.a.

Let $X \in \mathbb{R}^{n \times d}$ be a matrix with $n < d$. If XX^T is invertible, then β achieves zero cost in Eq (1) iff:

$$\beta = X^T(XX^T)^{-1}y + \zeta$$

for some ζ in the subspace orthogonal to all the data (that is, for some ζ such that $\zeta^T x^{(i)} = 0$, $\forall 1 \leq i \leq n$).

Let β be a global minimizer of Eq (1). Then we have

$J(\beta) = 0$

This means that

$X\beta - y = 0$ or $\beta = X^T y$.

Now, let ζ be any vector such that $\zeta^T x^{(i)} = 0$ $\forall 1 \leq i \leq n$.

Then we have

$$(\beta+\zeta)^T x^{(i)} = \beta^T x^{(i)} + \zeta^T x^{(i)} = 0$$

for all $1 \leq i \leq n$. This means that $\beta+\zeta$ is orthogonal to all the data.

Let β be any vector of the form:

$$\beta = X^T (XX^T)^{-1} y + \zeta$$

for some ζ in the subspace orthogonal to all the data

Then we have

$$J(\beta) = \frac{1}{2n} \|X\beta - y\|_2^2$$

Since ζ is orthogonal to all the data, we have

$$\|X\beta - y\|_2^2 = \|X^T (XX^T)^{-1} y\|_2^2$$

Now, we know that XX^T is invertible, so we can write

$$X^T(XX^T)^{-1}y = X^T(XX^T)^{-1}(XX^T)\beta = y$$

This means that

$$\|X\beta - y\|_2^2 = \|y\|_2^2 = 0$$

Therefore, β is a global minimizer of $E_\beta(1)$. And we have shown that β achieves zero cost in $E_\beta(1)$ iff

$$\beta = X^T(XX^T)^{-1}y + \zeta$$

for some ζ in the subspace orthogonal to all the data. This implies that there is infinite number of β's such that $E_\beta(1)$ is minimized.

4.b.

Let $X \in \mathbb{R}^{n \times d}$ be a matrix with $n < d$. If XX^T is invertible, then for any β such that $J(\beta) = 0$, we have

$$\|\beta\|_2 \leq \|\beta\|_2$$

where $\beta = X^T(XX^T)^{-1}y$.

Let β be any vector of the form

$$\beta = X^T(XX^T)^{-1}y + \zeta$$

for some ζ in the subspace orthogonal to all data.

Then we have

$$\|\beta\|_2^2 = \|\beta\|_2^2 + \|\zeta\|_2^2$$

To see this we have

$$\|\beta\|_2^2 = (\beta)^T\beta = (X^T(XX^T)^{-1}y + \zeta)^T(X^T(XX^T)^{-1}y + \zeta)$$

Expanding we get

$$\|\beta\|_2^2 = X^T(XX^T)^{-1}y\,(XX^T)^{-1}X\beta + \zeta^T\zeta$$

Since ζ is orthogonal to all the data, we have $\zeta^T x^{(i)} = 0$ $\forall\, 1 \leq i \leq n$. This means that

$$X^T(XX^T)^{-1}y\,(XX^T)^{-1}X\beta = X^T(XX^T)^{-1}y\,(XX^T)^{-1}Xy = X^T(XX^T)^{-1}\|y\|_2^2$$

Therefore we have

$$\|\beta\|_2^2 = \|p\|_2^2 + \|\zeta\|_2^2$$

Let β be any vector such that $f(\beta) = 0$. Then we get

$$\|\beta\|_2^2 = \|p\|_2^2 + \|\zeta\|_2^2$$

for some ζ in the subspace orthogonal to all the data

Since $\|\zeta\|_2^2 \geq 0$, we have

$$\|\beta\|_2^2 \geq \|p\|_2^2$$

This means that p is minimum norm solution. And we have shown that for any β such that $f(\beta) = 0$, we have

$$\|p\|_2 \leq \|\beta\|_2.$$

4. d

Let $X \in \mathbb{R}^{n \times d}$ be a matrix with $n < d$. If XX^T is invertible, then the gradient descent algorithm with zero initialization always converges to the minimum norm solution.

Let $\beta(0) = 0$. Then we have

$$\beta(1) = -\eta/n \, X^T(X\beta(0) - y) = \frac{\eta}{n} X^T y$$

Since XX^T invertible, we can write

$$\beta(1) = \frac{\eta}{n} X^T y = \frac{\eta}{n} X^T (XX^T)^{-1} y = \frac{\eta}{n} X^T (XX^T)^{-1} (XX^T) y = \frac{\eta}{n} y$$

This shows that $\beta(1)$ is a linear combination of $\{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$.

Assume that $\beta^{(t)}$ is a linear combination of $\{x^{(1)}, \ldots x^{(n)}\}$ for

some $t \geq 1$. Then we have

$$\beta^{(t+1)} = \beta^{(t)} - \frac{\eta}{n} X^T (X\beta^{(t)} - y) = \beta^{(t)} - \frac{\eta}{n} X^T (X\beta^{(t)} - X^T v^{(t)})$$
$$= \beta^{(t)} - \frac{\eta}{n} X^T (X(\beta^{(t)} - v^{(t)})).$$

Since $\beta^{(t)}$ is a linear combination of $\{x^{(1)}, x^{(2)}, \ldots x^{(n)}\}$, so is $\beta^{(t)} - v^{(t)}$. This means that $\beta^{(t+1)}$ is also a linear combinat. of $\{x^{(1)}, x^{(2)} \ldots x^{(n)}\}$.

By the principle of mathematical induction, we have shown that $\beta^{(t)}$ is a linear combination of $\{x^{(1)}, x^{(2)} \ldots x^{(n)}\}$ for all $t \geq 0$.

Now let $\hat{\beta}$ be any solution to the gradient descent with zero initialization that converges to $J(\hat{\beta}) = 0$

Then we have
$\hat{\beta} = \beta^{(t)}$ for some $t \geq 0$.
Since $\beta^{(t)}$ is a linear combination of $\{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$, so is $\hat{\beta}$. This means that $\hat{\beta}$ is in the subspace orthogonal to all the data. But we know that the minimum norm solution is also in the subspace orthogonal to all the data. Therefore, $\hat{\beta}$ must be the minimum norm solution.

We have shown that the gradient descent algorithm with zero initialization always converges to the minimum norm solution.

4.e.

To prove that there are infinitely many solutions with zero cost in the quadratically parameterized model, let's first establish that $y^{(i)}$ is linear in $x^{(i)}$. We can use the parameter β^* from sub-question (a) to represent $y^{(i)}$ as follows

$$y^{(i)} = (x^{(i)})^T((\theta^*)^{\odot 2} - (\phi^*)^{\odot 2})$$
$$= (x^{(i)})^T(\beta^*)^T$$
$$= (\beta^*)^T x^{(i)}$$

This shows that $y^{(i)}$ is linear in $x^{(i)}$ with parameter β^*

Now, lets consider the cost function $J(\beta)$ for linear model:

$$J(\beta) = \frac{1}{2n} \|X\beta - y\|_2^2$$

We can rewrite the quadratically parameterized model $f_{\theta,\phi}(x)$ in terms of β as follows:

$$f_{\theta,\phi}(x) = x^T(\theta^{\odot 2} - \phi^{\odot 2})$$
$$= (\beta^*)^T x$$

Comparing this with the linear model, we can see that $\theta(\beta)_k$ and $\phi(\beta)_k$ can be considered as solutions with zero cost, where k represents the different possible solutions.

Therefore, by mapping the parameters θ and ϕ to parameter β in the linear model, we sheeen that there are infinitely many solutions with zero cost in the quadratically param. models.

4.9

. The models with the smaller initialization parameters ($\alpha = 0.01$, and $\alpha = 0.03$) achieve lower training errors. This suggests that smaller initialization parameters can help with overfitting.

. Model with the smallest initialization parameter achieves the best validation error. This suggests that smaller initialization parameters can help to improve generalization performance.

4.1.

Typically, SGD can find solutions that generalize better than GD due to the noise introduced by the mini-batches. This stochasticity allowes the optimizer to explore different areas of the parameter space and potentially avoid getting stuck at in poor local optima.

But in our case GD finds better solution than SGD because our data set is small. And GD with the smallest initialization parameter generalizes better.