18-698 / 42-632
Neural Signal Processing
Problem Set 6 Solutions

1.)

a.)

$$P(z_n = k|\boldsymbol{x_n}) = \frac{P(\boldsymbol{x_n}, z_n = k)}{P(\boldsymbol{x_n})} = \frac{P(\boldsymbol{x_n}|z_n = k)P(z_n = k)}{\sum_{j=1}^{K} P(\boldsymbol{x_n}, z_n = j)}$$
$$= \frac{P(\boldsymbol{x_n}|z_n = k)P(z_n = k)}{\sum_{j=1}^{K} P(\boldsymbol{x_n}|z_n = j)P(z_n = j)} = \frac{\mathcal{N}(\boldsymbol{x_n}|\boldsymbol{\mu_k}, \Sigma_k)\pi_k}{\sum_{j=1}^{K} \mathcal{N}(\boldsymbol{x_n}|\boldsymbol{\mu_j}, \Sigma_j)\pi_j}$$

b.)

$$Q(\theta) = \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma_{nk}[\ln P(\boldsymbol{x_n}|z_n = k, \theta) + \ln P(z_n = k|\theta)]$$
$$= \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma_{nk}[\ln \mathcal{N}(\boldsymbol{x_n}|\boldsymbol{\mu_k}, \Sigma_k) + \ln \pi_k]$$

Notice that $\ln \mathcal{N}(\boldsymbol{x_n}|\boldsymbol{\mu_k}, \Sigma_k) = \ln\left\{(2\pi)^{-\frac{31}{2}}|\Sigma_k|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{x_n} - \boldsymbol{\mu_k})^T\Sigma^{-1}(\boldsymbol{x_n} - \boldsymbol{\mu_k})\right]\right\}$

$$= \ln\left[(2\pi)^{-\frac{31}{2}}\right] + \ln\left(|\Sigma_k|^{-\frac{1}{2}}\right) - \frac{1}{2}(\boldsymbol{x_n} - \boldsymbol{\mu_k})^T\Sigma^{-1}(\boldsymbol{x_n} - \boldsymbol{\mu_k})$$
$$= -\frac{31}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_k| - \frac{1}{2}tr[\Sigma^{-1}(\boldsymbol{x_n} - \boldsymbol{\mu_k})(\boldsymbol{x_n} - \boldsymbol{\mu_k})^T]$$

since $tr(scalar) = scalar$ and $tr(ABC \dots N) = tr(BC \dots NA)$.

Using the properties on the following line, we get the next two results.

$$\frac{d}{d\boldsymbol{x}}(\boldsymbol{x}^T A\boldsymbol{x}) = (A + A^T)\boldsymbol{x}, \frac{d}{dX}tr(AX^{-1}B) = -(X^{-1}BAX^{-1})^T, \frac{d}{dX}\ln|X| = X^{-T}$$

$$\frac{d}{d\boldsymbol{\mu_k}}\ln \mathcal{N}(\boldsymbol{x_n}|\boldsymbol{\mu_k}, \Sigma_k) = \Sigma_k^{-1}(\boldsymbol{x_n} - \boldsymbol{\mu_k})$$
$$\frac{d}{d\Sigma_k}\ln \mathcal{N}(\boldsymbol{x_n}|\boldsymbol{\mu_k}, \Sigma_k) = -\frac{1}{2}\Sigma_k^{-1} + \frac{1}{2}\Sigma_k^{-1}(\boldsymbol{x_n} - \boldsymbol{\mu_k})(\boldsymbol{x_n} - \boldsymbol{\mu_k})^T\Sigma_k^{-1}$$

So now we can take derivatives of $Q(\theta)$ with respect to the parameters to find the optimal.

$$\frac{dQ(\theta)}{d\boldsymbol{\mu_k}} = \sum_{n=1}^{N} \gamma_{nk} \Sigma_k^{-1} (\boldsymbol{x_n} - \boldsymbol{\mu_k}) \; set = 0$$

$$\Rightarrow \sum_{n=1}^{N} \gamma_{nk} (\boldsymbol{x_n} - \boldsymbol{\mu_k^*}) = 0 \Rightarrow \boldsymbol{\mu_k^*} = \frac{\sum_{n=1}^{N} \gamma_{nk} \boldsymbol{x_n}}{\sum_{n=1}^{N} \gamma_{nk}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \boldsymbol{x_n}$$

$$\frac{dQ(\theta)}{d\Sigma_k} = \sum_{(n=1)}^{N} \gamma_{nk} \left[-\frac{1}{2} \Sigma_k^{-1} + \frac{1}{2} \Sigma_k^{-1} (\boldsymbol{x_n} - \boldsymbol{\mu_k})(\boldsymbol{x_n} - \boldsymbol{\mu_k})^T \Sigma_k^{-1} \right] set = 0$$

$$\Rightarrow \sum_{n=1}^{N} \gamma_{nk} (\Sigma_k^*)^{-1} = \sum_{n=1}^{N} \gamma_{nk} (\Sigma_k^*)^{-1} (\boldsymbol{x_n} - \boldsymbol{\mu_k})(\boldsymbol{x_n} - \boldsymbol{\mu_k})^T (\Sigma_k^*)^{-1}$$

$$\Rightarrow \sum_{n=1}^{N} \gamma_{nk} \Sigma_k^* = \sum_{n=1}^{N} \gamma_{nk} (\boldsymbol{x_n} - \boldsymbol{\mu_k})(\boldsymbol{x_n} - \boldsymbol{\mu_k})^T$$

$$\Rightarrow \Sigma_k^* = \frac{\sum_{n=1}^{N} \gamma_{nk} (\boldsymbol{x_n} - \boldsymbol{\mu_k})(\boldsymbol{x_n} - \boldsymbol{\mu_k})^T}{\sum_{n=1}^{N} \gamma_{nk}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} (\boldsymbol{x_n} - \boldsymbol{\mu_k})(\boldsymbol{x_n} - \boldsymbol{\mu_k})^T$$

To find the optimal π_k, incorporate the constraint that $\sum_{k=1}^{K} \pi_k = 1$ as a Lagrangian multiplier.

$$Q_1(\theta) = Q(\theta) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

$$\frac{dQ_1(\theta)}{d\pi_k} = \sum_{n=1}^{N} \frac{\gamma_{nk}}{\pi_k} + \lambda \; set = 0 \Rightarrow \pi_k^* = -\frac{1}{\lambda} \sum_{n=1}^{N} \gamma_{nk}$$

Plugging into the constraint that $\sum_{k=1}^{K} \pi_k^* = 1$, we get:

$$-\frac{1}{\lambda} \sum_{k=1}^{K} \left(\sum_{n=1}^{N} \gamma_{nk} \right) - 1 = 0 \Rightarrow \lambda = -N \Rightarrow \pi_k^* = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk} = \frac{N_k}{N}$$

To summarize, we find the following optimal parameters:

$$\boldsymbol{\mu_k^*} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \boldsymbol{x_n}, \Sigma_k^* = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} (\boldsymbol{x_n} - \boldsymbol{\mu_k})(\boldsymbol{x_n} - \boldsymbol{\mu_k})^T, \pi_k^* = \frac{N_k}{N}$$

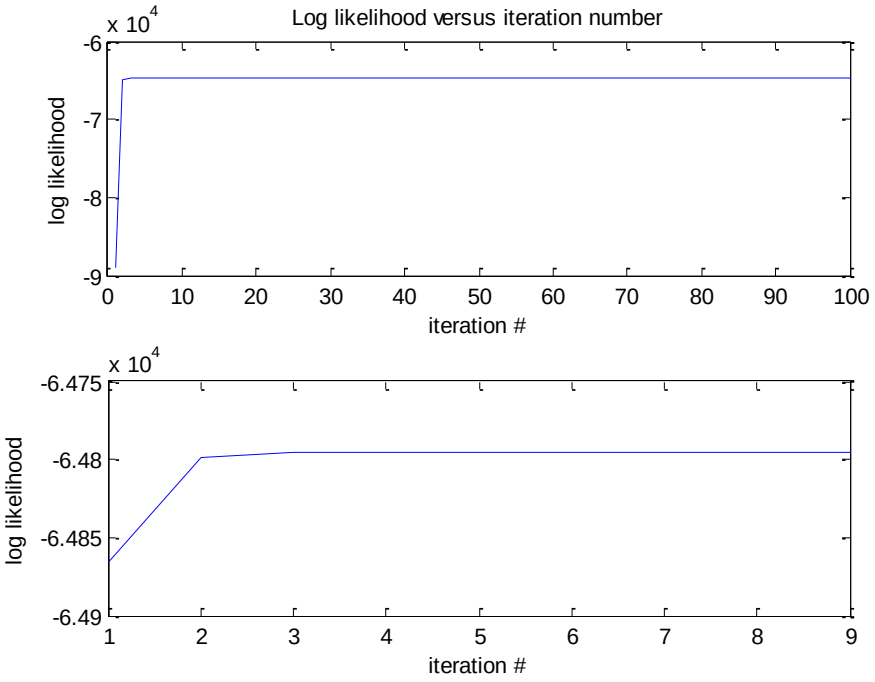
2.)

```
clear all
close all
load ps6_data
f_0 = 30000;
K = 3;
[D,N] = size(Spikes);
t_spike = (0:D-1)/f_0;
InitParams = InitParams1;
for k=1:K
    InitParams.Sigma(:,:,k) = InitParams.Sigma(:,:,1);
end
[MU,SIGMA,PI,GAMMA,LL] = func_GMM(InitParams,Spikes);
[maxGAMMA,c] = max(GAMMA);
subplot(2,1,1)
plot(LL)
title('Log likelihood versus iteration number')
xlabel('iteration #')
ylabel('log likelihood')
subplot(2,1,2)
plot(LL(2:10))
xlabel('iteration #')
ylabel('log likelihood')
figure

for k=1:K
subplot(K,1,k)
hold on
plot(t_spike,Spikes(:,c==k),'k');
plot(t_spike,MU(:,k),'r-','linewidth',2)
plot(t_spike,MU(:,k)+sqrt(diag(SIGMA(:,:,k))),'r--','linewidth',1.5)
plot(t_spike,MU(:,k)-sqrt(diag(SIGMA(:,:,k))),'r--','linewidth',1.5)
ylim([min(min(Spikes)) max(max(Spikes))])
xlabel('time (seconds)')
ylabel('potential (mV)');
title(sprintf('Cluster %i voltage versus time',k));
end
saveas(gcf,'ps6_sol_fig2.pdf');
```

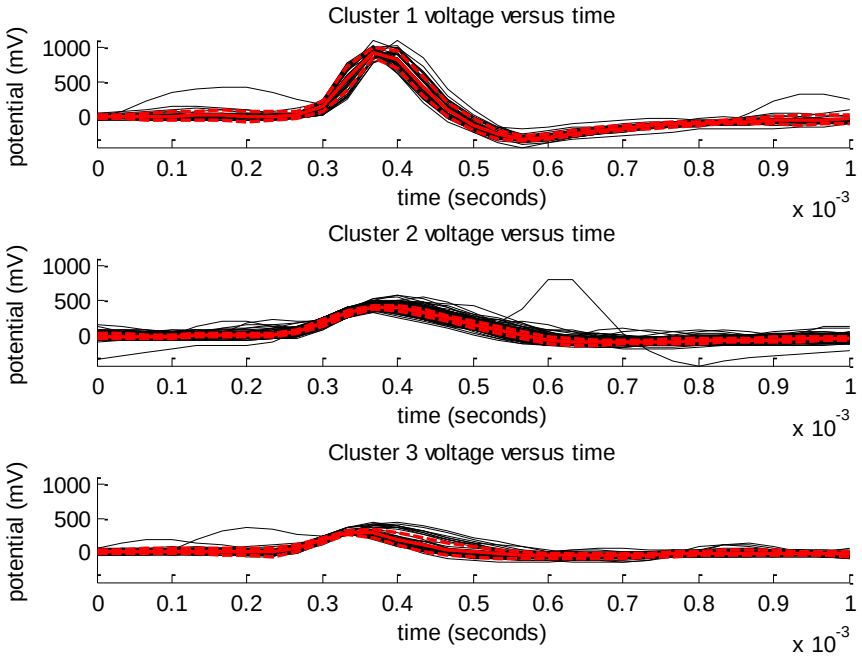
```
function [mu, Sigma, ppi, gam, LL]=func_GMM(InitParams,Spikes)
% [mu, Sigma, ppi]=func_GMM(InitParams,Spikes)
% EM algorithm for Gaussian Mixture Model estimation
%
% xDim: data dimensionality
% zDim: number of mixture components
% N: number of data points
%
% INPUTS:
% InitParams - a 1x1 structure containing two fields
% InitParams.mu - initialization of mean vectors of GMs (xDim x zDim)
% InitParams.Sigma - initialization of covariance matrices of GMs (xDim
x xDim x
% zDim)
% Spikes - input data (xDim x N)
%
% OUTPUTS:
% mu - estimated mean vectors of GMs (xDim x zDim)
% Sigma - estimated covariance matrices of GMs (xDim x xDim x zDim)
% ppi - estimated weights of GMs
% gam - estimated responsibilities of each cluster to each data point
(zDim x xDim)
% LL - estimated log-likelihood at each iteration
mu = InitParams.mu;
ppi = InitParams.pi;
K = size(mu, 2);
[D, N] = size(Spikes);
Sigma = InitParams.Sigma;
const = -0.5 * D * log(2*pi);
for i = 1:100
% === E-step ===
logMat = nan(K, N);
for k = 1:K
S = Sigma(:,:,k);
xdif = bsxfun(@minus, Spikes, mu(:,k));
term1 = -0.5 * sum((xdif' * inv(S)) .* xdif', 2); % N x 1
term2 = const - 0.5 * log(det(S)) + log(ppi(k)); % scalar
logMat(k,:) = term1' + term2;
end
% Evaluate log P({x})
astar = max(logMat, [], 1);
adif = bsxfun(@minus, logMat, astar);
nLL = log(sum(exp(adif), 1)) + astar; % 1 x N
LL(i) = sum(nLL);
gam = exp(bsxfun(@minus, logMat, nLL)); % K x N (responsibilities)
gam = bsxfun(@rdivide, gam, sum(gam, 1)); % for numerical stability
% === M-step ===
Neff = sum(gam, 2);
ppi = Neff' / N;
for k = 1:K
mu(:,k) = (Spikes * gam(k,:)') / Neff(k);
xdif = bsxfun(@minus, Spikes, mu(:,k));
S = bsxfun(@times, xdif, gam(k,:)) * xdif' / Neff(k);
Sigma(:,:,k) = (S + S') / 2; % for numerical stability
end
end
return;
```

a.)
The EM algorithm converges in 5 iterations.



b.)
The $\pi_k = [0.0761, 0.8333, 0.0906]^T$

c.)



3.)

This set of initialization parameters causes cluster 1 to have only 11 data points with nonzero responsibilities, $P(z_n = 1)$. This results in a rank deficient covariance matrix for cluster 1, and we cannot invert a low rank matrix.