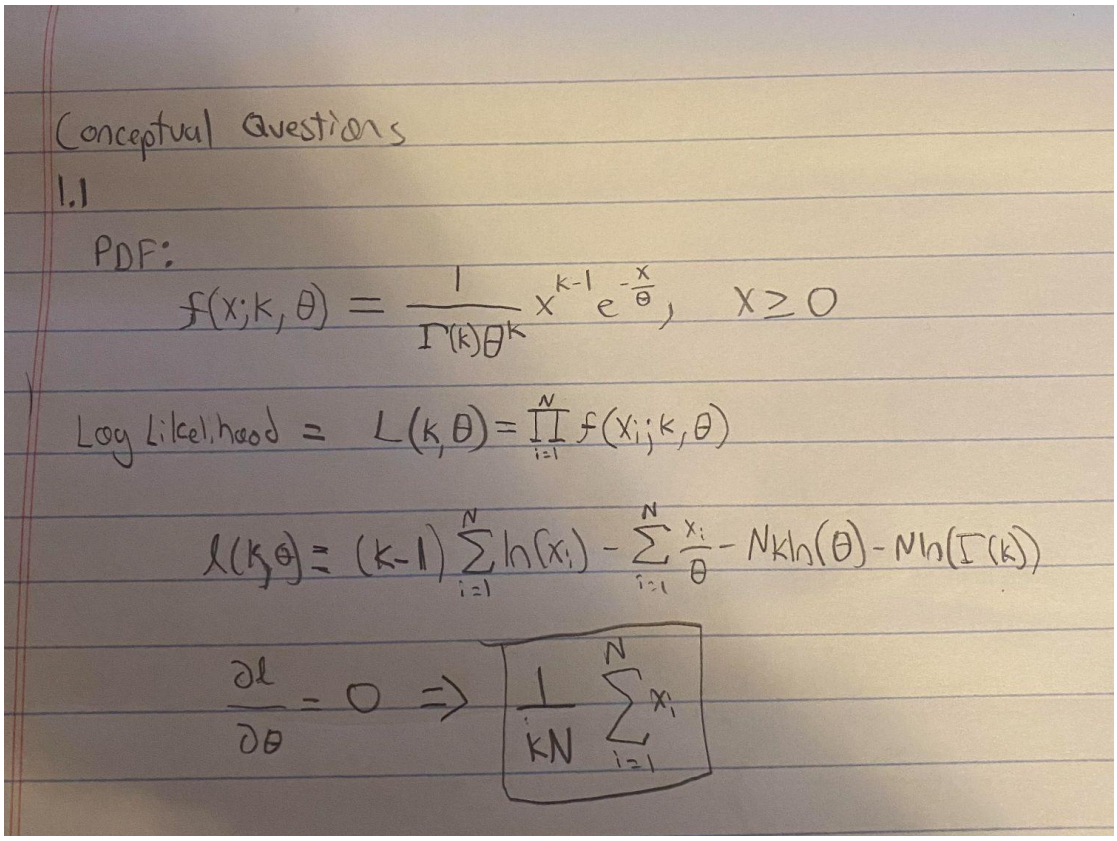
CDA Homework 3
Author: Joseph D Ciaravino

1.1 - see attachment



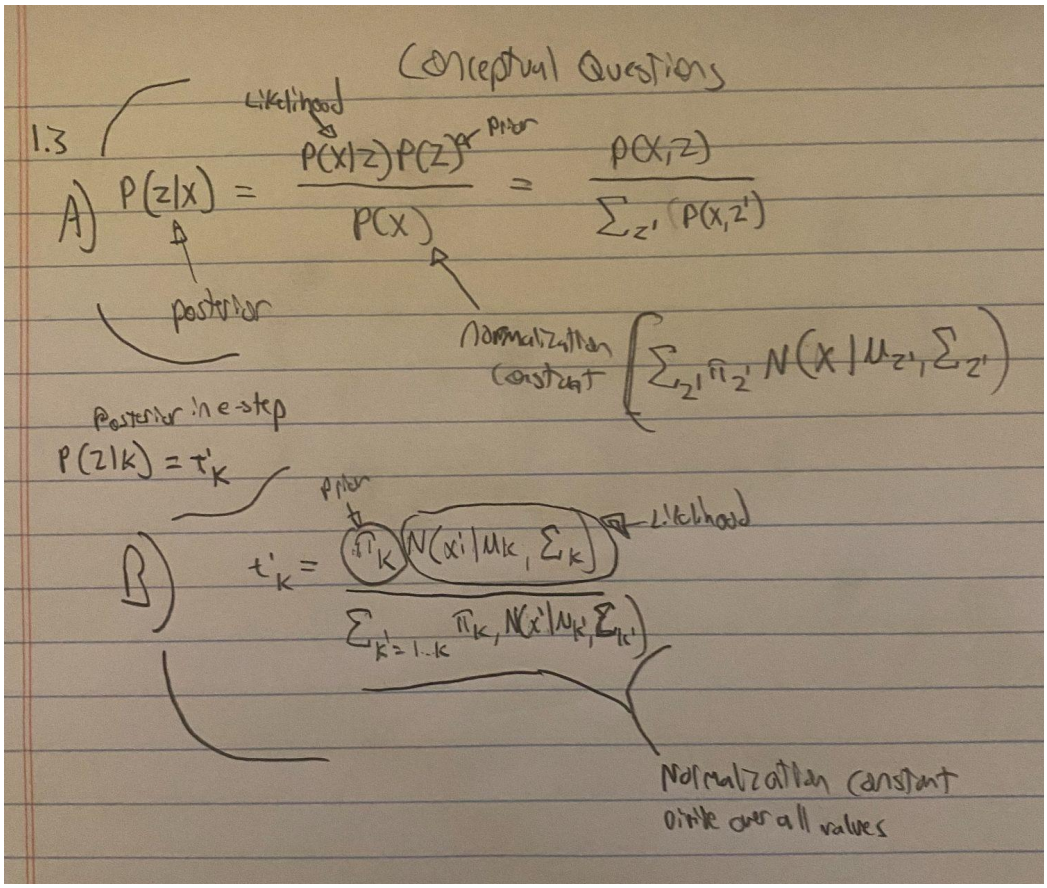
1.2 Please compare the pros and cons of KDE over histogram, and give atleast one advantage and disadvantage to each.

KDE
Pro - flexible (various smoothing kernel functions) - kernel bandwidth can change smoothness as well.
Cons - Poor in terms of memory requirement. All data will need to be in memory (m*n), expensive computationally.

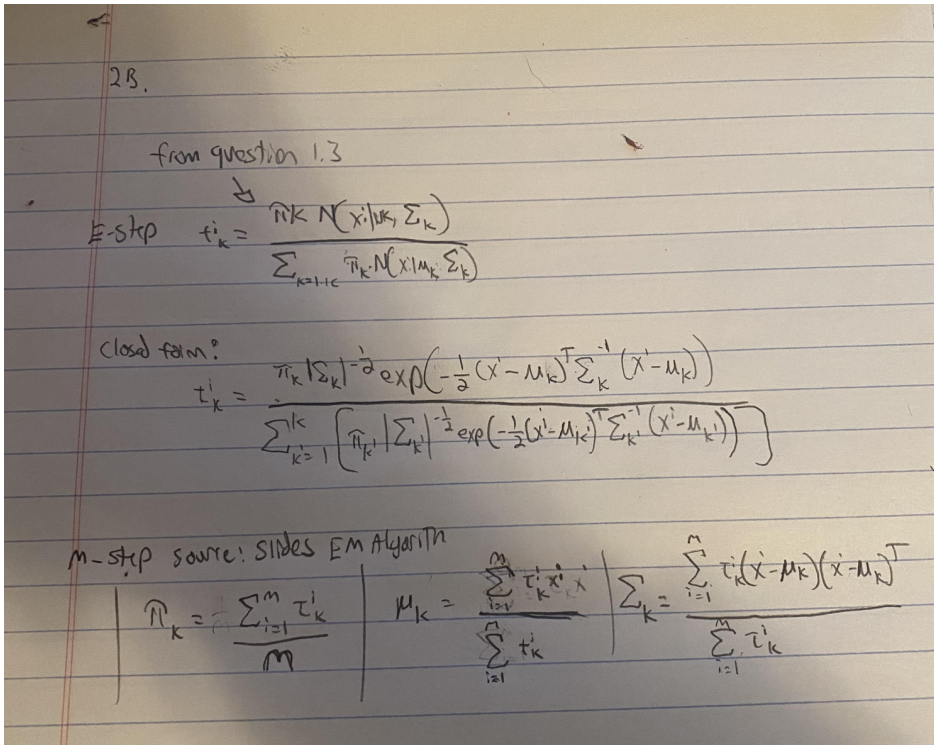
Histogram
Pros - Can easily expand for more data (placing in bins is computation). Parameters increase with 1/bin size which is an advantage over KDE in that it can be more memory efficient in certain circumstances, depending on bin size.
Cons - output depends on bins size. Not efficient for high-dimensional data

1.3

Shown below are how Bayes rule in part(A) is used to form the basis for constructing the E-step in EM for GMM. The components in Section (A) are linked to components in fully formed calculation in Section (B)



2.A



2B.

from question 1.3

E-step $t_k^i = \frac{\pi_k N(x^i|\mu_k, \Sigma_k)}{\sum_{k=1}^{k} \pi_k N(x^i|\mu_k, \Sigma_k)}$

closed form?

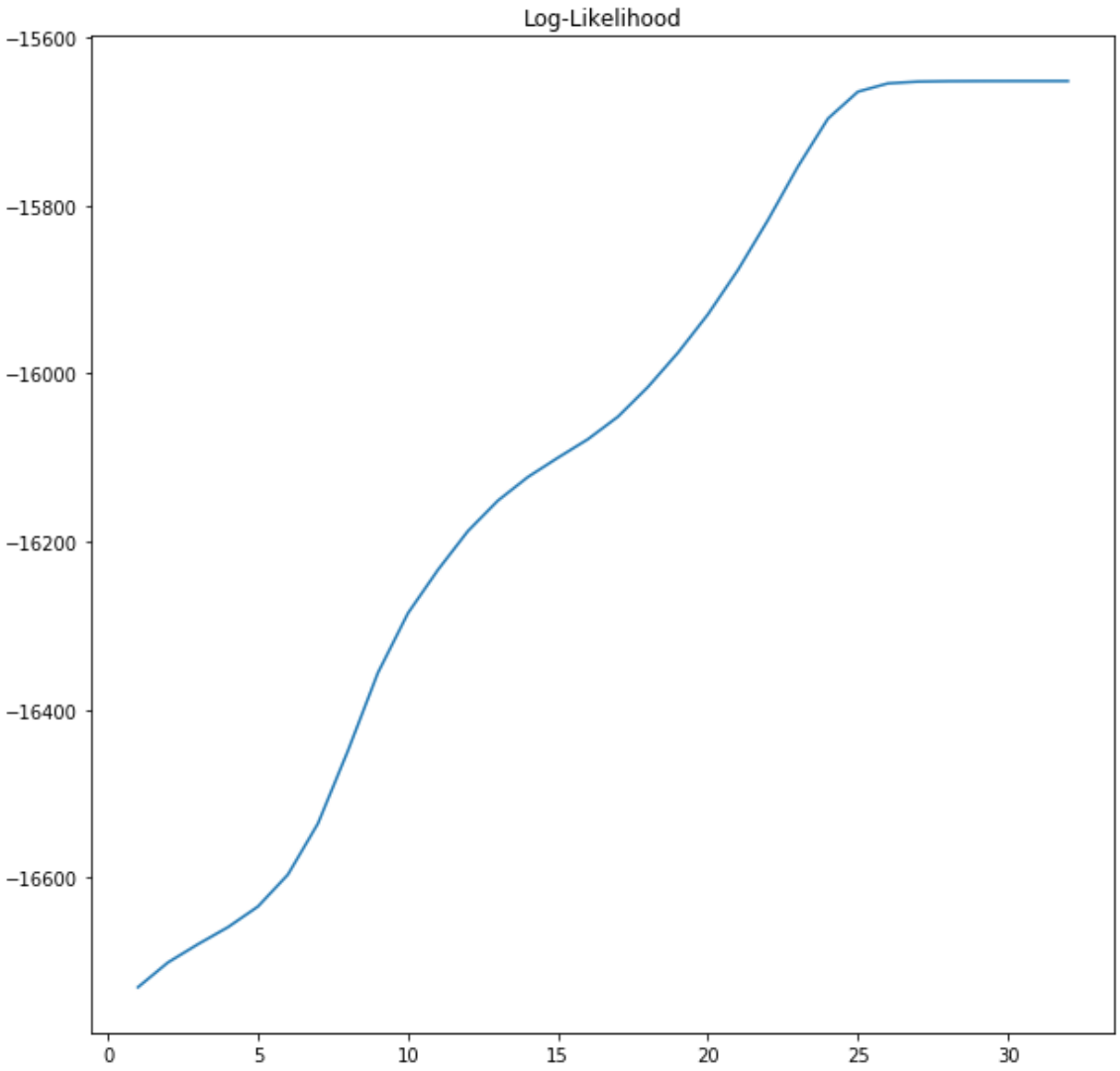
$$t_k^i = \frac{\pi_k |\Sigma_k|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x^i-\mu_k)^T \Sigma_k^{-1}(x^i-\mu_k)\right)}{\sum_{k'=1}^{k}\left[\pi_{k'}|\Sigma_{k'}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x^i-\mu_{k'})^T \Sigma_{k'}^{-1}(x^i-\mu_{k'})\right)\right]}$$

M-step source: slides EM Algorith

$\hat{\pi}_k = \frac{\sum_{i=1}^{m} \tau_k^i}{m}$ | $\mu_k = \frac{\sum_{i=1}^{m} \tau_k^i x^i x^i}{\sum_{i=1}^{m} t_k^i}$ | $\Sigma_k = \frac{\sum_{i=1}^{m} \tau_k^i (x^i-\mu_k)(x^i-\mu_k)^T}{\sum_{i=1}^{m} \tau_k^i}$

2.B

As shown below the Log-likelihood function converges after a reasonable number of steps and time (in this instance 32) decreasing the log-likelihood until the function is maximized.



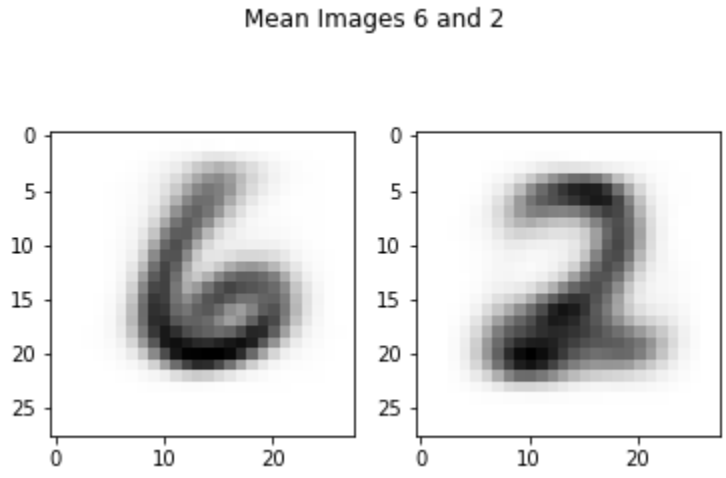
Convergence steps

step 0-15651.720172122845
step 1-35534.29268537524
step 2-16730.786860246895
step 3-16701.282077903703
step 4-16679.542064863785
step 5-16659.263831150532
step 6-16634.671790932607
step 7-16596.589689366436
step 8-16535.842263954175
step 9-16449.100461670067

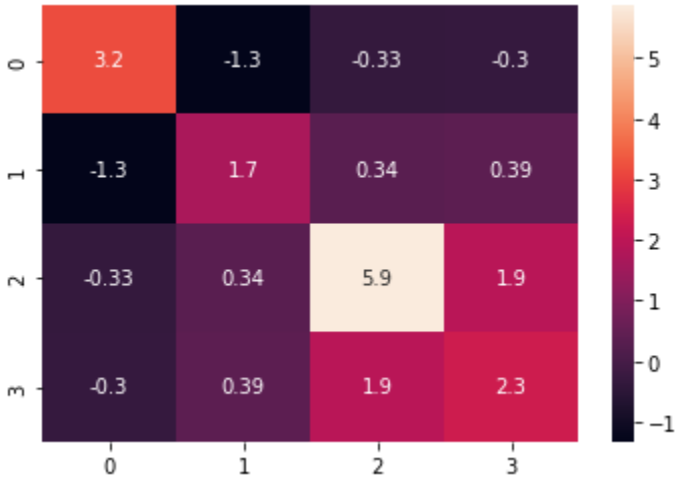
step 10-16356.310514457975
step 11-16285.398108684658
step 12-16233.76097593973
step 13-16187.542795014568
step 14-16151.274063243598
step 15-16123.161782855062
step 16-16100.215349168291
step 17-16078.156998488743
step 18-16051.362895303719
step 19-16015.97885868341
step 20-15975.251266938812
step 21-15929.44656565035
step 22-15876.666302886388
step 23-15817.224481862124
step 24-15753.475177275322
step 25-15696.398958787231
step 26-15664.364618231984
step 27-15654.515252597856
step 28-15652.379628648454
step 29-15651.89065830424
step 30-15651.765713163051
step 31-15651.732126330218
step 32-15651.722902182253
training coverged

2C.

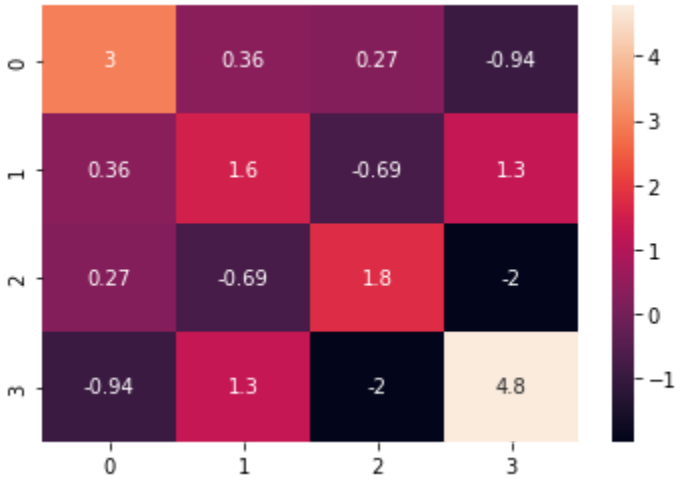
Mean Images



First Covariance Matrix



Second Covariance Matrix



Weights

Component 1: 0.5134041139402375
Component 2: 0.48659588605976245

2.D

Confusion Matrix - EM GMM

	precision	recall	f1-score	support
0	0.94	0.99	0.96	975
1	0.99	0.93	0.96	1015
accuracy			0.96	1990
macro avg	0.96	0.96	0.96	1990
weighted avg	0.96	0.96	0.96	1990

\

Confusion Matrix - Kmeans Algorithm

	precision	recall	f1-score	support
0	0.95	0.93	0.94	1044
1	0.93	0.94	0.93	946
accuracy			0.94	1990
macro avg	0.94	0.94	0.94	1990
weighted avg	0.94	0.94	0.94	1990

The precision for 2 labels is slightly higher for k-means than the GMM, however GMM outperforms for the 6 digit by a greater margin. Overall the GMM computationally doesn't take much longer than kmeans and seems to have greater accuracy.

Additionally, GMM is a "soft" classifier as opposed to kmeans hard classification using Euclidean distance, so by simply inferring labels using tau, we are missing some extra information on bayesian probability.