# Homework 2:Mixture Models and the EM Algorithm

Luis A. Valbuena Reyes

***Abstract*— In this document, we present the solution for homework 2. Trying to follow the guidelines for writing papers, we develop each the requirements of the assignment on Sections II, III, and IV.**

## I. INTRODUCTION

In this document, we present the results obtained after the implementation of two algorithms to conduct unsupervised learning. These algorithms are expectation maximization with Gaussian mixture models and k-means. Even though we arrive at expressions for the mean, covariance matrix and posterior probability that maximizes the log-likelihood of the probability distribution, we need to conduct an iterative process. We also need to assume a preliminary number of latent variables.

Although is not obvious how this methodology works, this algorithm drives the means to the maximum of the log-likelihood, which are arbitrarily close to the actual means of the clusters, but it can also lead the means to saddle point in the probability distribution.

One particular application of these two algorithm is the example presented in class where there is a communication channel in which a signal is corrupted by white noise and is independent of the data. The signal inside the communication channel is described by an equation of the form $x[n] = y[n] + ay[n-1]$ and given the combinations $(1,1)$, $(1,-1)$, $(-1,1)$, and $(-1,-1)$ for $y[n]$ and $y[n-1]$ respectively, we have the outputs $(1+a+\delta[n])$, $(-1+a+\delta[n])$, $(1-a+\delta[n])$, and $(-1-a+\delta[n])$.

Another application presented on [1] is image segmentation. We want to partition an image into regions with "homogeneous visual appearance or which corresponds to objects or parts of objects"

## II. THEORY

Based on the material found in [1]. We want to assign probability distributions to a set of samples to attain clustering data. Initially, we don't know how many clusters are present on the set of samples, therefore we assume an arbitrary number $N$. The $N$ latent variables are represented with $z$. We are also assuming that Gaussian distributions are very good approximations for the data set. Then the approach taken here is to maximize the likelihood of the probability distributions that we assigned with respect to the means $\mu_k$, the covariance matrices $\Sigma_k$, and the posterior probabilities (responsibilities) of each cluster we are initially assuming.

We start with the expression:

$$\begin{aligned}\mathbb{P}(x) &= \sum_{k=1}^{N} \mathbb{P}(z=k)\mathbb{P}(x|z=k),\\ &= \sum_{k=1}^{N} \pi_k \mathbb{P}(x|z=k),\end{aligned}$$

where $\pi_k = \mathbb{P}(z=k)$ is the posterior probability, and $\mathbb{P}(x|z=k)$ is the likelihood. As we are conducting a maximization of $\mathbb{P}(x|z=k)$ for all $k$, which is assumed to be Gaussian, then we take $\log \mathbb{P}(\mathbf{X}|\mu, \mathbf{\Sigma}, \pi)$ to avoid operations with the exponent and conduct the maximization with respect to $\mu_k$, $\Sigma_k$, and $\pi_k$. The expression of $\log \mathbb{P}(\mathbf{X}|\mu, \mathbf{\Sigma}, \pi)$ is

$$\log \mathbb{P}(\mathbf{X}|\mu, \mathbf{\Sigma}, \pi) = \sum_{n=1}^{M} \log \Big( \sum_{k=1}^{N} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \Big), \quad (1)$$

where $M$ is the number of samples After taking the derivative of $\log \mathbb{P}(x|z=k)$ with respect to $\mu_k$, $\Sigma_k$, and $\pi_k$ and making it equal to zero, we have:

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n, \quad (2a)$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T, \quad (2b)$$

$$\pi_k = \frac{N_k}{N}, \quad (2c)$$

where $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$, and function $\gamma(z_{nk})$ depends on whether the approach taken is Gaussian mixture models (see Section II-B) or k-means (see Section II-C).

### *A. Expectation Maximization*

The expectation maximization algorithm is composed of two main steps: the *E step* and the *M step*, and goes as follows:

1) Initialize $\mu_k$, $\Sigma_k$, and $\pi_k$ and evaluate Eq. 1 (or another performance criterion).
2) *E step*: evaluate $\gamma(z_{nk})$.
3) *M step*: compute 2a, 2b, and 2c.
4) evaluate 1 (or another performance criterion) again and determine to stop or continue.

The stopping criterion will be treated on Section III-B.

Luis Valbuena is with the Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, 87131-0001, {lavalbuenar@unm.edu}

### B. Gaussian Mixture Models

The Gaussian mixture model, which is the approach mos commonly used on expectation maximization, consists of : linear combinations of Gaussian distributions of the form:

$$f(x) = \sum_{k=1}^{N} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k), \quad (3$$

where $\sum_{k=1}^{N} \pi_k = 1$. Then, function $\gamma(z_{nk})$ is given by

$$\gamma(z_{nk}) = \frac{\mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{N} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$$

### C. K-means

In the case of the k-means approach, function $\gamma(z_{nk},$ becomes an indicator function, i.e.,

$$\gamma(z_{nk}) = \begin{cases} 1, & \text{if } k = \arg\min_k \{\sqrt{(x_n - \mu_k)^T \Sigma_k (x_n - \mu_k)}\} \\ 0, & \text{otherwise,} \end{cases}$$

where $\sqrt{(x_n - \mu_k)^T \Sigma_k (x_n - \mu_k)}$ is the Mahalanobis distance.

## III. Experiments

As stated in the homework, “The purpose of this set of experiments is to learn the basics of Gaussian Mixture models and how to code an Expectation Maximization algorithms as well as how to track its behavior. The results will be compared with those of the k-means algorithm”. For this theoretical example, the same generated artificial data is fed into both implementations. The artificial data is generated with a multi Gaussian distribution using the following mean:

$$\mu_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} -1 \\ -2 \end{bmatrix}, \quad \mu_3 = \begin{bmatrix} 3 \\ -3 \end{bmatrix}, \quad (4)$$

and covariance matrices:

$$\Sigma_1 = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \quad \Sigma_3 = \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix}, \quad (5)$$

We generated 100 samples for the first and second group, and 200 samples for the third group. The generated artificial data can be seen on Fig. 1.

### A. EM Algorithm

The execution of the expectation maximization algorithm with with Gaussian mixtures is presented in Fig. 2(a), Fig. 2(b), Fig. 2(c), and Fig. 2(d). The initialization values for the means were $\mu_1 = [-2 \quad 5]^T$, $\mu_2 = [0.1 \quad 4.5]^T$, and $\mu_3 = [4.2 \quad 0.2]^T$ and the covariance matrices $\Sigma_k$ were initialized as identity matrices. In order to avoid cluttering the images with too many level curves, we decided to plot level curves from $L = 0.001$ to $L = 0.019$ in increments of $0.001$. The trajectory of the means as the algorithm evolves in time is presented in Fig. 6(a).

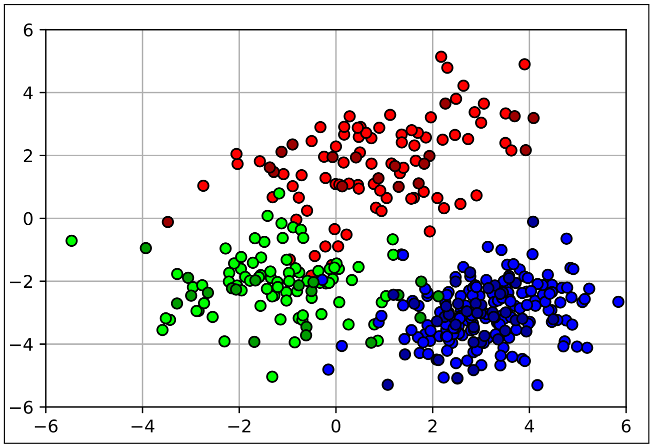


Fig. 1. Generated data with the parameter specification of Eq. 4, and Eq. 5. The blue group corresponds to $\mu_1$, $\Sigma_1$, the red group corresponds to $\mu_2$, $\Sigma_2$, and the green group corresponds to $\mu_3$, $\Sigma_3$.

Note that in the first iteration, Fig. 2(a), $\mu_2$ and $\mu_3$ are set relatively close that the maximum level curve sets the two groups as one; but as the computation is carried out,Fig. 2(b), the curve of the maximum level begins to split. At the $14^{th}$ iteration we can see three closed curves appear. At the end, in Fig. 2(d) we can see there is not a major change on the form of the closed curves. Also note that the maximum level curves are getting stretch as the algorithm progresses, giving account of the modification of the covariance matrices.

### B. Representation of the Data Likelihood

The log-likelihood is presented in Eq. 1 and the evolution of this measure is depicted on Fig.**??**. As the algorithm finds the means, covariance matrices and posterior probabilities based on the maximization of the log-likelihood, the best stopping criterion should be to halt the iterations when we perceive a decrement in the log-likelihood quantity.

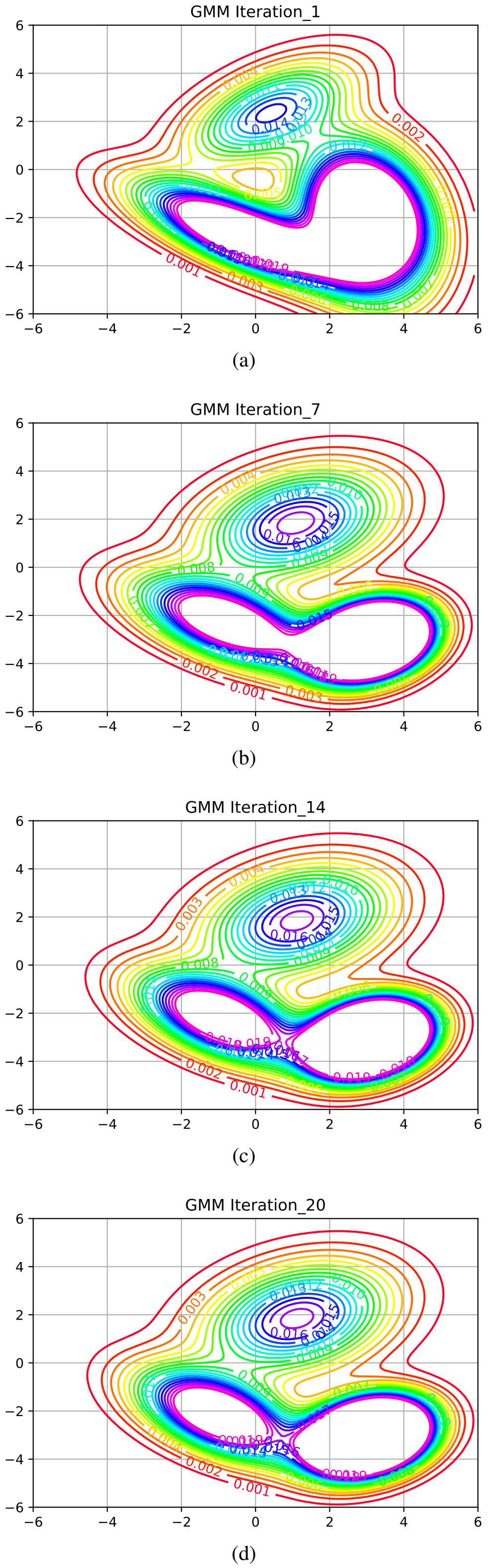
However, note that Eq. 1 requires the computation with all the samples $M$, over the number of latent variables $N$, which is $O(NM)$. Note that in Fig. 6(a) we can see that after the first iteration, the means have very small displacements, which hints that a more appropriate stopping criterion could be the Euclidean norm of the means’ displacements. The displacements of the means and the norm of the displacements as the expectation maximization algorithm with with Gaussian mixtures evolves is presented in Fig. 6(c). Note that the norm bounds the displacements of the means, guaranteeing that all the displacement are below or equal the stopping value assigned to the norm. In our case, we took a very conservative approach and required the norm of the displacements to be $0.01$

### C. Unsupervised Classification

A scatter plot of the data is depicted in Fig. 3, where we assigned the final iteration of Eq. 3 to the size of the ball of each sample.

A classification scheme based on the GMM model would be to take each Gaussian that makes part of Eq. 3 and assign a threshold $\epsilon_k$ on each function. Then, a sample $x_n$ belongs to a feature $k$ if

$$\pi_k \mathcal{N}(x|\mu_k, \Sigma_k) > \epsilon_k.$$



(a)

(b)

(c)

(d)

Fig. 2. Evolution of the Expectation Maximization algorithm using Gaussian mixture models at iterations (a) 1, (b) 7, (c) 14, and (d) 20.

A representation of the classification parameters explained before is depicted on Fig. 4, where the level curves with $L = \epsilon_k$ are calculated for each $\pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$.

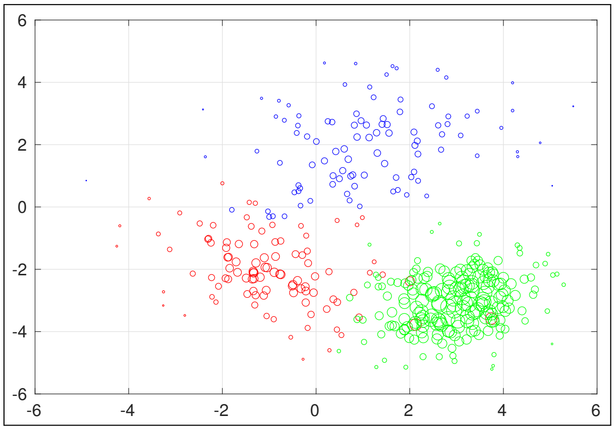


Fig. 3. Scatter plot. The bigger the balls are, the higher the probability value the samples have assigned to them.

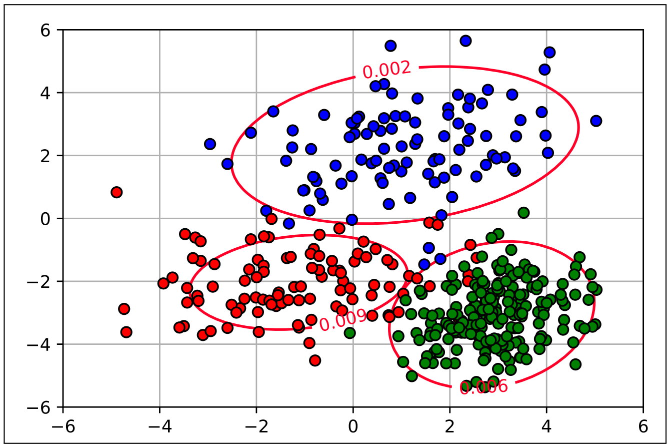


Fig. 4. Classification scheme. In this case, $\epsilon_{blue} = 0.004$, $\epsilon_{red} = 0.012$, and $\epsilon_{green} = 0.006$

.

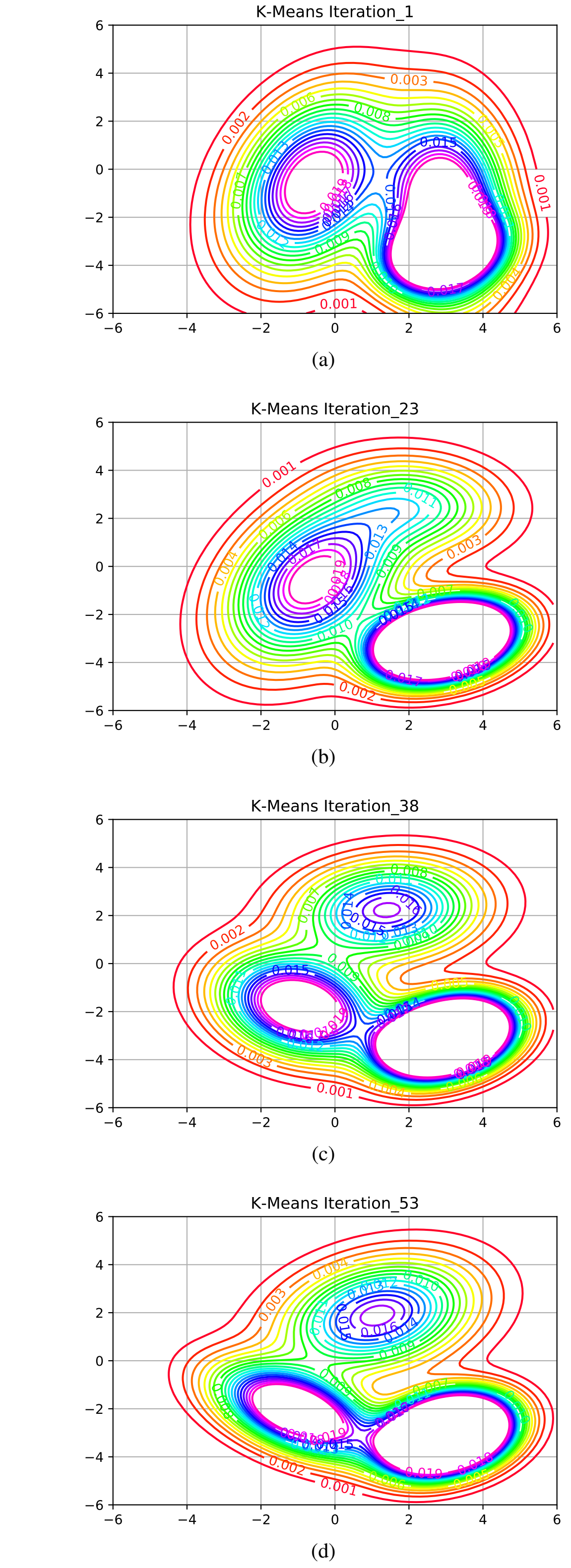
### D. Comparison to the K-means Algorithm

The other approach available for unsupervised classifications is k-means. We are taking the same artificial data depicted in Fig. 5 as input for our implementation of k-means.

*1) Mahalanobis distance:* A positive-definite matrix $A$ can generate a norm of the form $\sqrt{x^T Ax}$. Although the covariance matrices $\Sigma_k$ are only positive-semidefinite matrices, a norm that takes into account the correlation among the samples in a problem of classification is very helpful. Instead of using the euclidean norm on function $\gamma(z_{nk})$ for the k-means procedure, we use the Mahalanobis distance which is defined as

$$d_{\Sigma_k} = \sqrt{(x_n - \mu_k)^T \Sigma_k (x_n - \mu_k)},$$

*2) K-means procedure and Parameter update:* The initialization values for the means were $\mu_1 = [-0.76 \quad -2.16]^T$, $\mu_2 = [3.40 \quad -3.38]^T$, and $\mu_3 = [3.33 \quad -2.53]^T$ and the covariance matrices $\Sigma_k$ were initialized as identity matrices. The trajectory of the means as the algorithm evolves in time is presented in Fig. 6(b).



(a)

(b)

(c)

(d)

Fig. 5. Output of *svmtrain* with different values of $\sigma$. (a), $\sigma = 0.05$. (b), $\sigma = 1$. (c), $\sigma = 1.4$.

Because means $\mu_1$ and $\mu_3$ are very close to the mean of the group of green markers, the first iteration shown in Fig. 5(a) suggest there are going to be two groups. Note that in this case Fig. 6(b) shows that the displacements of the means is reduced compared to how the means moved on the Gaussian mixture model, therefore for this particular choice of initial means the algorithm is going to take longer to converge to our stopping criterion. As the algorithm progresses and Fig. 6(b) shows, $\mu_1$ starts moving to the top of the space starting to create a closed curve around it, as shown in Fig. 5(b). By the $38^{th}$ the three groups are clearly defined and in Fig. 5(d) we can see that the covariance matrices have influenced the closed curves by elongating the groups at the bottom of the graph. We confronted the Log-likelihood for Expectation Maximization with Gaussian mixture models in both the train set and the test set: while the Log-likelihood for the train data in monotonically increasing, the Log-likelihood for the test data has a maximum. That is the stopping criterion: the maximum of the Log-likelihood for the test data .

*3) Parameter initialization, growing-pruning and use in an 10 dimensional problem:* The parameter initialization goes as follows:

1) Start by choosing a sample randomly from the set of samples. As it is assumed that the distribution of the clusters inside the set of samples is Gaussian, there is about a 65 % change the sample is close to the actual mean of a cluster withing a standard deviation distance.
2) Compute the distance between your selected sample and the rest of the samples and sort them into increasing order.
3) Assign a probability to each distance according to

$$\mathbb{P}(x_i) = \frac{d_i}{\sum_j d_j}.$$

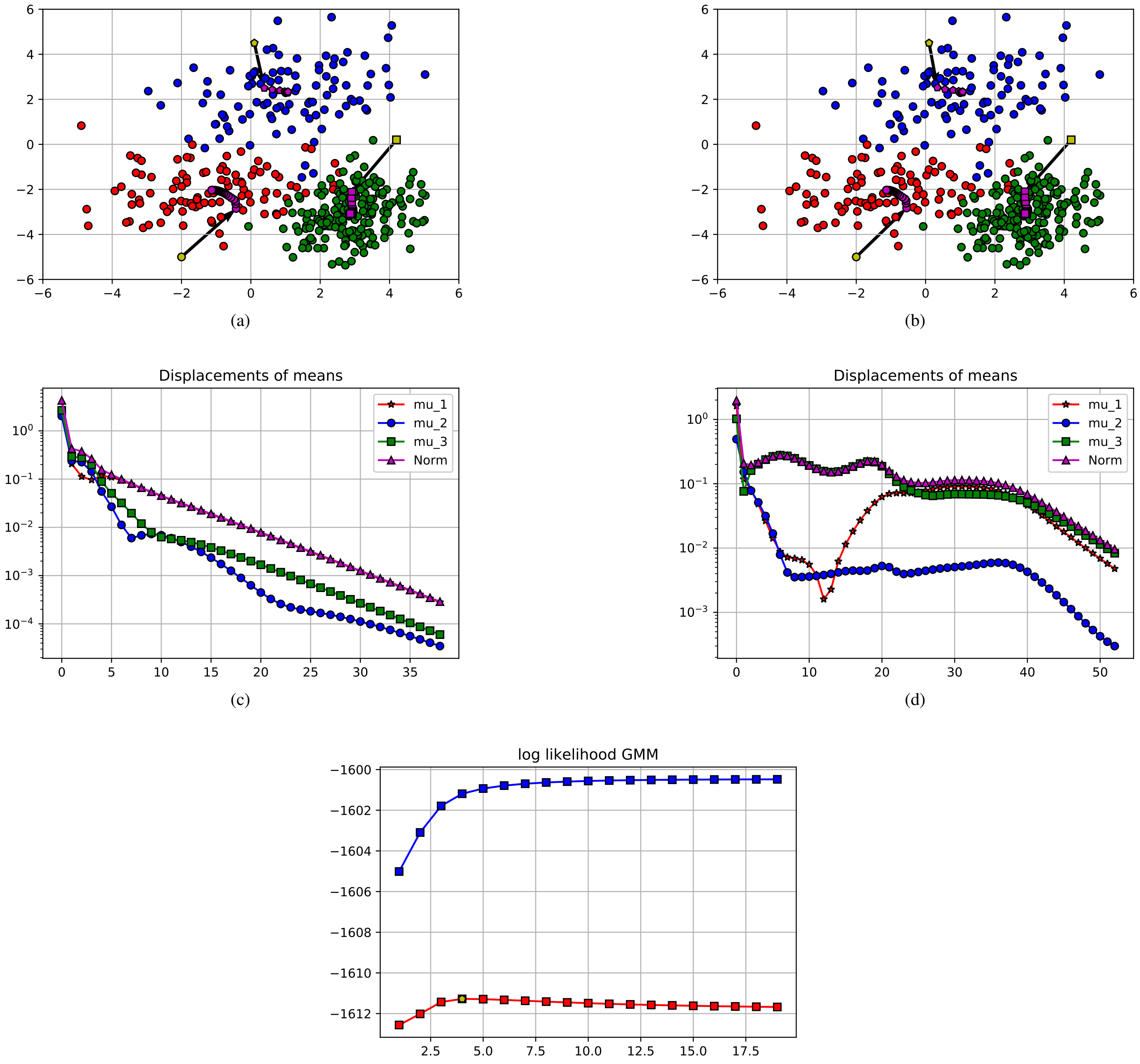
4) Select your initial mean in regions of the probability you just assigned where the slope is not pronounced.

Consider Fig. 7. The red dots indicate the samples where the slope of the probability function assigned to the distance changes dramatically. In the regions between them the slope is kept constant, suggesting a cluster. Therefore the selection of the initial means should be samples of these regions.

In contrast to this activity where we fabricated the sample data, we don't know how many clusters are present in the sample set. Therefore we can assign more clusters than necessary or less. If we assign more clusters than required, some means are going to converge to the same target mean while having similar covariance matrices, then it is necessary to do *pruning*, that is, compute a single center with the average of the converging means and their respective covariance matrices. On the other hand, if the ratio of the singular values of the covariance matrix is very high, we split the mean into the direction related to the highest singular value, this is known as *growing*.

## IV. Conclusion

The stopping criterion: because we are calculating the maximum of an increasing function such as the log-likelihood which argument is a Gaussian distribution, we are driven into the direction of stopping the algorithm when



(a)

(b)

(c)

(d)

(e)

Fig. 6. Comparison between Expectation Maximization with Gaussian mixture models and k-means. (a) Trajectory of the means, and (c) displacement of means for Expectation Maximization with Gaussian mixture models. (b) Trajectory of the means,, and (e) displacement of means for k-means. (e) Log-likelihood for Expectation Maximization with Gaussian mixture models

the log-likelihood stops increasing. However, a computation with all the samples over all the assigned latent variables makes it expensive, which lead us to use the Euclidean norm of the displacements. We selected a very conservative stopping value, which was reflected in the increased number of iterations.

Despite the fact the means $\mu_1$ and $\mu_3$ were very close to the mean of the generated green group on the k-means execution, the algorithm managed to steer $\mu_1$ to create a new cluster, $\mu_1$ did not get stuck on the group three.

Even though we are maximizing Eq. 3, we chose its individual components as the requested classifier of Section III-C. The idea behind that decision is that Eq. 3 is seen as a linear combination of Gaussian distributions, where the Gaussian distributions are thought of linearly independent functions.

## References

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006. [Online]. Available: www.cse.psu.edu/~rtc12/CSE586Spring2010/papers/prmlMixturesEM.pdf

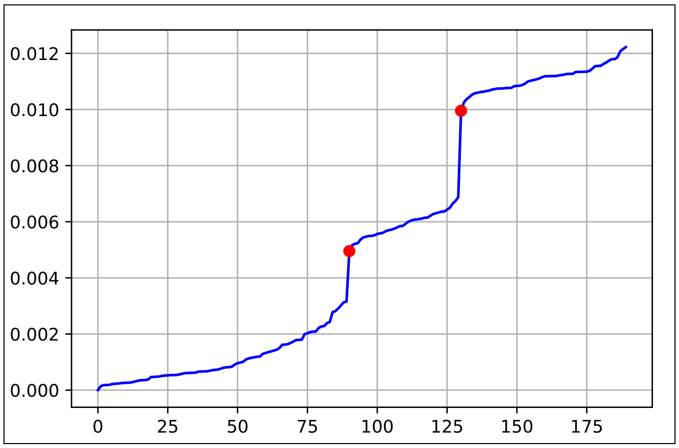


Fig. 7.  Selection of initial means.