# MLCV: Programming Assignement 3

Mohamed N'AITN'BARK

17/11/1990

## 1 Expectation Maximization for a Mixture of gaussians

**The data used in this part is generated from a 3 gaussians mixture distribution:** $p_1 N(\mu_1, \Sigma_1) + p_2 N(\mu_2, \Sigma_2) + p_3 N(\mu_3, \Sigma_3)$

$\mathbf{p}$ $=[0.25, 0.4, 0.35]$ ,$\mu_1 = [1, 2]$ , $\mu_2 = [-3, -5]$ , $\mu_3 = [6, -2]$ , $\Sigma_1 = \begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix}$ ,$\Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ ,$\Sigma_3 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$.

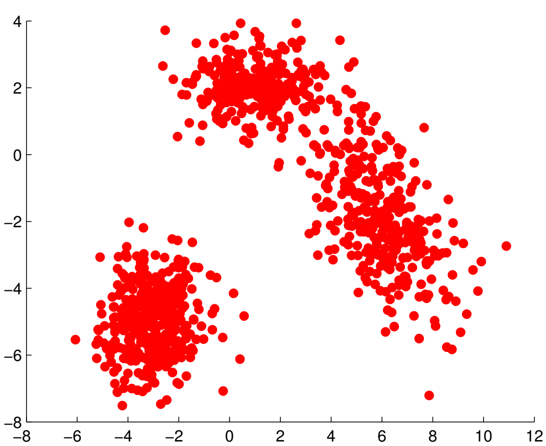


Figure 1: Our data

### 1.1 K-Means:

**The steps of the implemented K-means algorithm are:**

1. Initialize the centers by three randomly chosen points from the data.

2. Compute for every point the distance to each of the three centers.
3. Assign to each point the nearest center.
4. Update the value of each center by the value of the mean of the points assigned to it.
5. Compute the distortion.

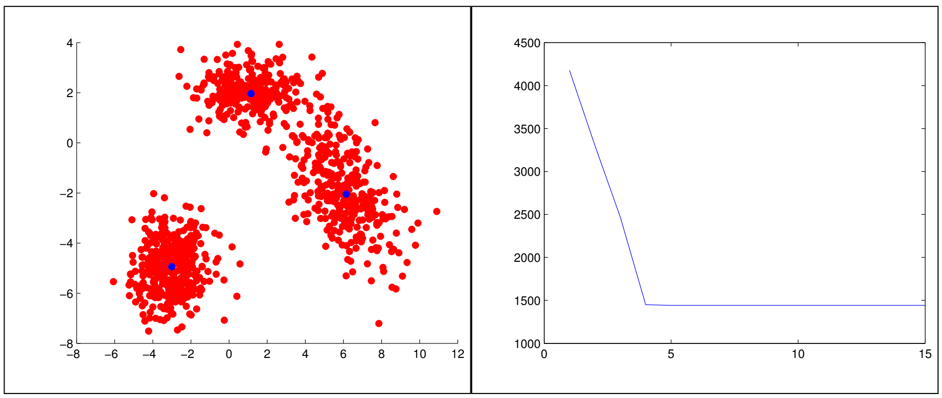


Figure 2: Right: The result of the KMeans algorithm. Left: the distortion as a function of iterations.

**We apply the same algorithm for 20 different initializations, then we choose the initialization that come with the final minimal distortion.**

## 1.2 Expectation Maximization:

**We apply the EM algorithm using the centers found by Kmeans as an initialization for the means, identity matrices for the covariance matrices and a uniform probability. We also keep track of the likelihood at each iteration.**

| Cluster | Probability | Center | Covariance matrix |
|---|---|---|---|
| 1 | 0.4 | $[-3, -4.94]$ | $\begin{bmatrix} 0.99 & 0.1 \\ 0.1 & 0.99 \end{bmatrix}$ |
| 2 | 0.35 | $[5.9, -1.0]$ | $\begin{bmatrix} 2.07 & -1.23 \\ -1.23 & 2.39 \end{bmatrix}$ |
| 3 | 0.25 | $[1, 2.05]$ | $\begin{bmatrix} 1.96 & 0.02 \\ 0.02 & 0.47 \end{bmatrix}$ |

**The EM algorithm gives the following clusters:**

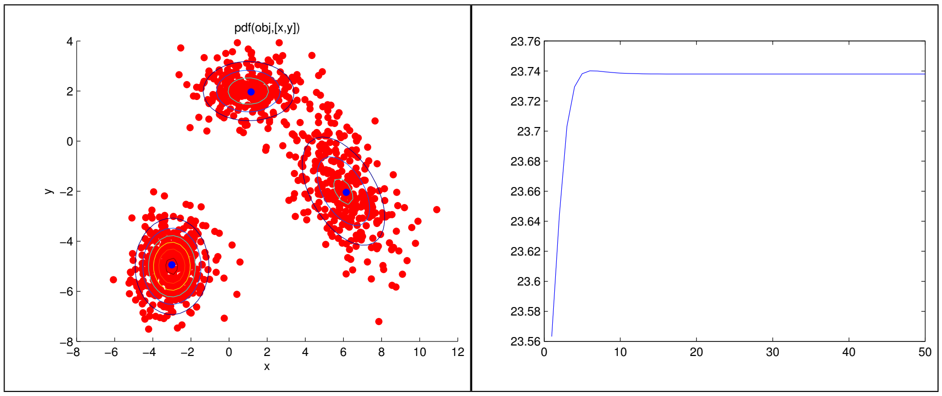


Figure 3: Right: The result of the EM algorithm. Left: the likelihood as a function of iterations.

**We can see that the algorithm recovered efficiently the parameters of our distribution.**

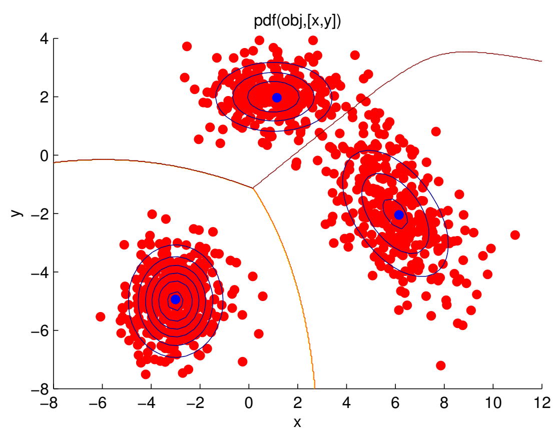


Figure 4: Decision boundaries

**Using the bayes rule, we assign to each point of regular grid on the domain of our data the cluster to which it belongs, then we use the** *contour* **function of matlab to plot the decision boundaries.**

## 2 Sum/Max-Product algorithms:

**Algorithm 1** Sum-Max Product algorithm

```
SumProduct(edge, graph):
    N←Neighbors of edge in graph
    result←[111...1]
    FOR neigbor IN N:
        result←result*Sum/MaxMessage(neighbor,edge,graph)
Sum/MaxMessage(edge2,edge1,graph):
    N←Neighbors of edge2 in graph except maybe edge1
    result←[111...1]
    IF N IS empty:
        result←∑_{edge2} Φ(edge1, edge2) or max_{edge2} Φ(edge1, edge2)
    ELSE:
        FOR neigbor IN N:
            result←result*SumMessage(neighbor,edge2)
        result←Φ(edge1, edge2).result  or  [max_{edge2} Φ(edge1, edge2) * result(edge2)]_{edge1}
```

**The implementation of the Sum-Product algorithm given here is based on a recursive approach:**

- Using the SumProduct algorithm for our graph we find that $Z = 1.915$ and $P(X_1) = [0.2914, 0.5311, 0.1775]$ and $P(X_2) = [0.4543, 0.5457]$.
- $P([1, 2, 4, 2, 1]) = 0.0056$, $P([3, 1, 2, 1, 2]) = 0.0033$ , $P([2, 2, 1, 2, 1]) = 0.0134$.