Representative Hard Clustering

\[ C_1, C_2, \ldots, C_k \]

\[ \mu_1, \mu_2, \ldots, \mu_k \]

\[ \text{SSE objective } = \text{ sum of squared errors} \]

Informally: minimize the squared distance of the point from the corresponding mean.

\[ \min J = \sum_{i=1}^{k} \sum_{j \in C_i} || x_j - \mu_i ||^2 \]

Search over all possible partitions.

\[ k = 1 \Rightarrow \text{Obvious solution: } \mu \text{ for the data} \]

\[ k = 2 \Rightarrow \text{NP-hard!} \]

**K-Means Algorithm**

- greedy
- randomized
- iterative algorithm
Cluster \((\bar{x}_j)\) = \(\arg \min_{i=1 \ldots k} \left\{ \| \bar{x}_j - \bar{\mu}_i \|_2^2 \right\}

**K-means \((D, k)\)**

- **How many clusters?**

1) **Initialization:**

   Randomly select any \(k\) points from \(D\)

   \(\bar{\mu}_i = \bar{x}_j\) for some random \(j \in \{1, 2, \ldots, n\}\)

   (try to select points that are far apart)

2) **Given** \(\bar{\mu}_1, \bar{\mu}_2, \ldots, \bar{\mu}_k\)

   for each \(\bar{x}_j \in D\)

   assign to closest mean \(\bar{\mu}_i\):

   \(\text{cluster } (\bar{x}_j) = \arg \min_{i=1 \ldots k} \left\{ \| \bar{x}_j - \bar{\mu}_i \|_2^2 \right\}\)

3) **Given the clusters:** \(C_1, C_2, \ldots, C_k\)
3) Given the clusters: $C_1, C_2, \ldots, C_k$

Update the means:
for $i = 1 \ldots k$:
$$\tilde{\mu}_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j$$

$O(n)$
$$P(C_i|x_j) = \frac{1}{1}$$

4) Convergence: repeat until:
$$\sum_{i=1}^{k} \left\| \tilde{\mu}_i^{(t+1)} - \tilde{\mu}_i^{(t)} \right\|^2 \leq \varepsilon$$
$\# \text{ of iterations?}$
$t : \text{ iteration}$
$O(n \cdot k \cdot d \cdot T)$

Meta: run multiple times & report the $B$ with smallest $SS$.

Works well for "convex" and well separated clusters.

Will not work here!
Soft-clustering $\sim$ probability $P(C_i|\mathbf{x}_j)$

\[ \Rightarrow \text{Expectation Maximization Algorithm} \]

\[ \Rightarrow \text{maximize the likelihood} \]

Clusters are from a mixture of Gaussians

Each cluster $C_i \sim N(x_j | \mu_i, \Sigma_i)$

$P(C_i) = \text{cluster prior prob}$
\[ f(x_j) = \sum_{i=1}^{k} \frac{\sum_{i}^k P(X_j|C_i) \cdot P(C_i)}{\bar{P}(x_j)} N(\bar{x}_j, \Sigma_i) \cdot P(C_i) \]

\[ P(x_j) = \sum_{i=1}^{k} P(X_j|C_i) \cdot P(C_i) \] 

**Mixture Model**

\[ L = \text{likelihood} = P(D|\theta) \]

\[ \theta = \text{parameters} \rightarrow \mu_i, \Sigma_i, P(C_i) : \forall i = 1 \ldots k \]

Unknown

\[ \max_{\theta} P(D|\theta) \equiv \max_{\theta} \ln P(D|\theta) \]

\[ = \max_{\theta} \left\{ \ln \left( \sum_{i=1}^{k} N(x_j|\mu_i, \Sigma_i) \cdot P(C_i) \right) \right\} \]

\[ \frac{\partial L}{\partial \mu_i}, \frac{\partial L}{\partial \Sigma_i}, \frac{\partial L}{\partial P(C_i)} \]

Direct solution is "recursive"

**EM: Expectation Maximization**

"guess" the labels \[ P(C_i|x_j) \]

\[ \text{Prob of cluster } i \text{ given point } x_j \]

Very easy to compute \[ \mu_i, \Sigma_i, P(C_i) \]
EM(D, K)

1) Initialization: guess $\hat{\mu}_i, \Sigma_i, P(C_i)$

\[ Z_i = 1 \]
\[ P(C_i) = \frac{1}{K} \]

2) Expectation step

Given $\hat{\mu}_i, \Sigma_i, P(C_i), \forall i = 1 \ldots K$

Compute the posterior prob $P(C_i|x_j) \forall x_j \forall C_i$

\[ P(C_i|x_j) = \frac{P(x_j|C_i) \cdot P(C_i)}{P(x_j)} \]

\[ P(C_i|x_j) = \frac{\prod_{i=1}^{K} P(x_j|C_i) \cdot P(C_i)}{\sum_{i=1}^{K} P(x_j|C_i) \cdot P(C_i)} \]

\[ \omega_j = P(C_i|x_j) = \frac{N(x_j; \hat{\mu}_i, \Sigma_i) \cdot P(C_i)}{\sum_{i=1}^{K} N(x_j; \hat{\mu}_i, \Sigma_i) \cdot P(C_i)} \]

Weight for cluster $C_i$
weight for cluster \( C_i \) from point \( x_j \)

\[
\sum_{i=1}^{\infty} \mathcal{N}(x_j | \mu_i, \Sigma_i) \cdot p(C_i) \]

\( \hat{x}_j \): weight of \( x_j \) for cluster \( i \)

3) **Maximization Step:** Given \( (\rho_{ij}) = p(C_i | x_j) \) \( \forall i, j \)

update \( \hat{\mu}_i, \Sigma_i, p(C_i) \)

\[
\hat{\mu}_i = \text{weighted mean} = \frac{\sum_{j=1}^{n} w_{ij} \cdot x_j}{\sum_{j=1}^{n} w_{ij}}
\]

\[
\Sigma_i = \text{weighted cov} = \frac{\sum_{j=1}^{n} w_{ij} (x_j - \hat{\mu}_i) (x_j - \hat{\mu}_i)^T}{\sum_{j=1}^{n} w_{ij}}
\]

\[
p(C_i) = \text{fraction of weight in } C_i
\]

\[
= \frac{\sum_{j=1}^{n} w_{ij}}{n}
\]

4) Repeat until convergence

\[
\sum_{i=1}^{k} \left\| \hat{\mu}_i^{(th)} - \mu_i^{(th)} \right\|^2 \leq \varepsilon
\]

5) Once we have \( \Theta = \{ \hat{\mu}_i, \Sigma_i, p(C_i) \} \) \( \forall i \)

we "can" assign each point to the nearest probable cluster
Cluster \( C(x_j) = \arg \max_{i=1 \cdots k} \frac{P(c_i|x_j)}{w_{ij}} \)

K-means is just a special case of EM
where \( P(c_i|x_j) = \frac{1}{\sum_{i=1}^{k} w_{ij}} \equiv w_{ij} \)

Further \( \sum_{i=1}^{k} w_{ij} = 1 \)

\( \Rightarrow \) one-hot prob assignment

Non-convex clusters?

density-based clustering

\( \varepsilon \)-radius

\( \varepsilon \)-ball or \( \varepsilon \)-neighborhood
\[ \text{minpts} \gets \text{minimum} \] 
\[ \# \text{points} \]
\[ \text{"density constraint"} \]
\[ |N_\epsilon(x_j)| \geq \text{minpts} \]

- \( \epsilon \)-ball or \( \epsilon \)-neighborhood

1) Compute "core" points \( \vec{x}_j \) such that 
\[ |N_\epsilon(x_j)| \geq \text{minpts} \]

2) \( \forall x_i, x_j \)
   \[ \text{if } x_i \in N_\epsilon(x_j) \]
   \[ \text{and } x_j \in N_\epsilon(x_i) \]
   add an edge between them

3) find connected component over the graph from step 2

4) for each component & each core point in them,
   include all border points in the cluster.

\[ \text{DBSCAN} = \text{density based clustering} \]