Problem with K-means
## Hard Assignment of Samples into Three Clusters

<table>
<thead>
<tr>
<th>Individual</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual 1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Individual 2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Individual 3</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Individual 4</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Individual 5</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Individual 6</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Individual 7</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Individual 8</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Individual 9</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Individual 10</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
## Probabilistic Soft-Clustering of Samples into Three Clusters

<table>
<thead>
<tr>
<th>Probability of</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual 1</td>
<td>0.1</td>
<td>0.4</td>
<td>0.5</td>
<td>1</td>
</tr>
<tr>
<td>Individual 2</td>
<td>0.8</td>
<td>0.1</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td>Individual 3</td>
<td>0.7</td>
<td>0.2</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td>Individual 4</td>
<td>0.10</td>
<td>0.05</td>
<td>0.85</td>
<td>1</td>
</tr>
<tr>
<td>Individual 5</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>Individual 6</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>Individual 7</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>Individual 8</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>Individual 9</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>Individual 10</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>1</td>
</tr>
</tbody>
</table>

- Each sample can be assigned to more than one clusters with a certain probability.
- For each sample, the probabilities for all clusters should sum to 1. (i.e., each row should sum to 1.)
- Each cluster is explained by a cluster center variable (i.e., cluster mean)
Probability Model for Data $P(X)$?
Mixture Model

- A density model $p(x)$ may be multi-modal.

Multi-modal: how do we model this?

Unimodal - Gaussian
Mixture Model

- We may be able to model it as a mixture of uni-modal distributions (e.g., Gaussians).
- Each mode may correspond to a different sub-population (e.g., male and female).
Learning Mixture Models from Data

- Given data generated from multi-modal distribution, can we find a representation of the multi-model distribution as a mixture of uni-modal distributions?
Gaussian Mixture Models (GMMs)

• Consider a mixture of $K$ Gaussian components:

$$p(x_n) = \sum_k p(x_n \mid z_n = k)p(z_n = k)$$

$$= \sum_k N(x_n \mid \mu_k, \Sigma_k)\pi_k$$
Gaussian Mixture Models (GMMs)

• Consider a mixture of \( K \) Gaussian components:

\[
p(x_n) = \sum_k p(x_n, z_n = k)p(z_n = k)
= \sum_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \pi_k
\]

• This probability model describes how each data point \( x_n \) can be generated
  - Step 1: Flip a \( K \)-sided die (with probability \( \pi_k \) for the \( k \)-th side) to select a cluster \( c \)
  - Step 2: Generate the values of the data point from \( \mathcal{N}(\mu_c, \Sigma_c) \)
Gaussian Mixture Models (GMMs)

- Consider a mixture of $K$ Gaussian components:

$$p(x_n) = \sum_k p(x_n, z_n = k)p(z_n = k)$$

$$= \sum_k N(x_n | \mu_k, \Sigma_k)\pi_k$$

- Parameters for $K$ clusters: $\theta = \{\mu_k, \Sigma_k, \pi_k, k = 1, \ldots, K\}$
Learning mixture models

• Latent variable model: data are only partially observed!
  – \( x_i \): observed sample data
  – \( z_i = \{z_i^1, ..., z_i^K\} \): Unobserved cluster labels (each element 0 or 1, only one of them is 1)

• MLE estimate
  – What if all data \((x_i, z_i)\) are observed?
    • Maximize the data log likelihood for \((x_i, z_i)\) based on \(p(x_i, z_i)\)
    • Easy to optimize!
  – In practice, only \(x_i\)'s are observed
    • Maximize the data log likelihood for \(x_i\) based on \(p(x_i)\)
    • Difficult to optimize!
  – Maximize the expected data log likelihood for \((x_i, z_i)\) based on \(p(x_i, z_i)\)
    • Expectation-Maximization (EM) algorithm
Learning mixture models: fully observed data

- In **fully observed iid settings**, assuming the **cluster labels** $z_i$’s were **observed**, the log likelihood decomposes into a sum of local terms.

$$ l_c(\theta; D) = \sum_n \log p(x_n, z_n | \theta) = \sum_n \log p(z_n | \theta) + \sum_n \log p(x_n | z_n, \theta) $$

- The optimization problems for $\mu_k, \Sigma_k$ and for $\pi_k$ are decoupled, and a closed-form solution for MLE exists.
MLE for GMM with fully observed data

- If we are doing MLE for **completely observed data**

- **Data log-likelihood**

\[
I(\theta; D) = \log \prod_n p(z_n, x_n) = \log \prod_n p(z_n | \pi) p(x_n | z_n, \mu, \sigma)
\]

\[
= \sum_n \log \prod_k \pi_k^{z_n} + \sum_n \log \prod_k N(x_n ; \mu_k, \sigma)^{z_n}
\]

\[
= \sum_n \sum_k z_n^k \log \pi_k - \sum_n \sum_k z_n^k \frac{1}{2\sigma^2} (x_n - \mu_k)^2 + C
\]

- **MLE**

\[
\hat{\pi}_{k,MLE} = \arg \max_{\pi} I(\theta; D),
\]

\[
\hat{\mu}_{k,MLE} = \arg \max_{\mu} I(\theta; D)
\]

\[
\hat{\sigma}_{k,MLE} = \arg \max_{\sigma} I(\theta; D)
\]

- **What if we do not know** \(z_n\)?
Learning mixture models

- In **fully observed iid settings**, assuming the cluster labels $z_i$’s were observed, the log likelihood decomposes into a sum of local terms.

$$l_c(\theta; D) = \sum_n \log p(x_n, z_n | \theta)$$

- With latent variables for cluster labels

$$l_c(\theta; D) = \sum_n \log p(x_n | \theta)$$

$$= \sum_n \log \sum_z p(x_n, z | \theta) = \sum_n \log \sum_z p(z | \theta) p(x_n | z, \theta)$$

  - all the parameters become coupled together via **marginalization**

- Are they equally difficult?

  Depends on $\pi_k$  
  Depends on $\mu_k, \Sigma_k$
**Theory underlying EM**

- Recall that according to MLE, we intend to learn the model parameter that would have maximized the likelihood of the data.

- But we do not observe $z$, so computing

$$l_c(\theta; D) = \sum_n \log \sum_z p(x_n, z | \theta) = \sum_n \log \sum_z p(z | \theta)p(x_n | z, \theta)$$

is difficult!

- Optimizing the log-likelihood for MLE is difficult!

- What shall we do?
Complete vs. Expected Complete Log Likelihoods

- The complete log likelihood:

\[
I(\theta; D) = \log \prod_n p(z_n, x_n) = \log \prod_n p(z_n | \pi) p(x_n | z_n, \mu, \sigma) \\
= \sum_n \log \prod_k \pi_k^{z_{nk}} + \sum_n \log \prod_k N(x_n; \mu_k, \sigma)^{z_{nk}} \\
= \sum_n \sum_k z_{nk} \log \pi_k - \sum_n \sum_k \frac{1}{2\sigma^2} (x_n - \mu_k)^2 + C
\]

- The expected complete log likelihood

\[
\langle I_c(\theta; x, z) \rangle = \sum_n \langle \log p(z_n | \pi) \rangle_{p(z|x)} + \sum_n \langle \log p(x_n | z_n, \mu, \Sigma) \rangle_{p(z|x)} \\
= \sum_n \sum_k \langle z_{nk} \rangle \log \pi_k - \frac{1}{2} \sum_n \sum_k \langle z_{nk} \rangle (x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k) + \log|\Sigma_k| + C
\]

Depends on \( \pi_k \)  
Depends on \( \mu_k, \Sigma_k \)
Complete vs. Expected Complete Log Likelihoods

• The complete log likelihood:

\[
I(\theta; D) = \log \prod_{n} p(z_n, x_n) = \log \prod_{n} p(z_n | \pi) p(x_n | z_n, \mu, \sigma) \\
= \sum_{n} \log \prod_{k} \pi_{z_n}^{k} + \sum_{n} \log \prod_{k} N(x_n; \mu_k, \sigma)^{z_n} \\
= \sum_{n} \sum_{k} z_n^{k} \log \pi_{k} - \sum_{n} \sum_{k} z_n^{k} \frac{1}{2\alpha^2} (x_n - \mu_k)^2 + C
\]

• The expected complete log likelihood

\[
\langle I_c(\theta; x, z) \rangle = \sum_{n} \langle \log p(z_n | \pi) \rangle_{p(z|x)} + \sum_{n} \langle \log p(x_n | z_n, \mu, \Sigma) \rangle_{p(z|x)} \\
= \sum_{n} \sum_{k} \langle z_n^{k} \rangle \log \pi_{k} - \frac{1}{2} \sum_{n} \sum_{k} \langle z_n^{k} \rangle (x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k) + \log |\Sigma_k| + C
\]

• EM optimizes the expected complete log likelihood
EM Algorithm

Maximization (M)-step:
- Find mixture parameters

Expectation (E)-step:
- Re-assign samples $x_i$’s to clusters
- Impute the unobserved values $z_i$

Iterate until convergence
K-Means Clustering Algorithm

Find the cluster means

\[ \bar{\mu}_k = \frac{1}{C_k} \sum_{i \in C_k} x_i \]

Re-assign samples \( x_i \)'s to clusters

\[ \arg\max_k \| x_i - \mu_k \|_2^2 \]

Iterate until convergence
The Expectation-Maximization (EM) Algorithm

• **Start:**
  - "Guess" the centroid $\mu_k$ and covariance $\Sigma_k$ of each of the K clusters

• **Loop**

(a) \hspace{2cm} (c) \hspace{2cm} (d) \hspace{2cm} (e)

(f) \hspace{2cm} (g) \hspace{2cm} (h) \hspace{2cm} (i)
The Expectation-Maximization (EM) Algorithm

- A “soft” k-means

**E step:**
\[
\tau_{n}^{k(t)} = \langle z_{n}^{k} \rangle_{q^{(t)}} = p(z_{n}^{k} = 1 | x, \mu^{(t)}, \Sigma^{(t)})
\]

**M step:**
\[
\pi_{k}^{(t+1)} = \frac{\sum_{n} \tau_{n}^{k(t)}}{N} = \frac{\langle n_{k} \rangle}{N}
\]
\[
\mu_{k}^{(t+1)} = \frac{\sum_{n} \tau_{n}^{k(t)} x_{n}}{\sum_{n} \tau_{n}^{k(t)}}
\]
\[
\Sigma_{k}^{(t+1)} = \frac{\sum_{n} \tau_{n}^{k(t)} (x_{n} - \mu_{k}^{(t+1)}) (x_{n} - \mu_{k}^{(t+1)})^{T}}{\sum_{n} \tau_{n}^{k(t)}}
\]
Compare: K-means

- The EM algorithm for mixtures of Gaussians is like a "soft version" of the K-means algorithm.
- In the K-means “E-step” we do hard assignment:
  \[ z_n^{(t)} = \arg \max_k (x_n - \mu_k^{(t)})^T \sum_k^{-1(t)} (x_n - \mu_k^{(t)}) \]
- In the K-means “M-step” we update the means as the weighted sum of the data, but now the weights are 0 or 1:
  \[ \mu_k^{(t+1)} = \frac{\sum_n \delta(z_n^{(t)}, k)x_n}{\sum_n \delta(z_n^{(t)}, k)} \]
Expected Complete Log Likelihood Lower-bounds
Complete Log Likelihood

• For any distribution \( q(z) \), define **expected complete log likelihood**:

\[
\left\langle I_c(\theta; x, z) \right\rangle_q = \sum_z q(z \mid x, \theta) \log p(x, z \mid \theta)
\]

– Does maximizing this surrogate yield a maximizer of the likelihood?

• Jensen’s inequality

\[
I(\theta; x) = \log p(x \mid \theta) = \log \sum_z p(x, z \mid \theta)
= \log \sum_z q(z \mid x) \frac{p(x, z \mid \theta)}{q(z \mid x)}
\geq \sum_z q(z \mid x) \log \frac{p(x, z \mid \theta)}{q(z \mid x)} \quad \Rightarrow \quad I(\theta; x) \geq \left\langle I_c(\theta; x, z) \right\rangle_q + H_q
\]
Closing notes

• Convergence

• Seed choice

• Quality of cluster

• How many clusters
Convergence

• Why should the K-means algorithm ever reach a fixed point?
  – -- A state in which clusters don’t change.

• K-means is a special case of a general procedure the Expectation Maximization (EM) algorithm.
  – Both are known to converge.
  – Number of iterations could be large.
Seed Choice

• Results can vary based on random seed selection.

• Some seeds can result in convergence to sub-optimal clusterings.
  – Select good seeds using a heuristic (e.g., doc least similar to any existing mean)
  – Try out multiple starting points (very important!!)
  – Initialize with the results of another method.
What Is A Good Clustering?

• Internal criterion: A good clustering will produce high quality clusters in which:
  – the intra-class (that is, intra-cluster) similarity is high
  – the inter-class similarity is low
  – The measured quality of a clustering depends on both the obj representation and the similarity measure used

• External criteria for clustering quality
  – Quality measured by its ability to discover some or all of the hidden patterns or latent classes in gold standard data
  – Assesses a clustering with respect to ground truth
How Many Clusters?

• Number of clusters K is given
  – Partition n docs into predetermined number of clusters

• Finding the “right” number of clusters is part of the problem
  – Given objs, partition into an “appropriate” number of subsets.
  – E.g., for query results - ideal value of K not known up front - though UI may impose limits.

• Tradeoff between having more clusters (better focus within each cluster) and having too many clusters

• Nonparametric Bayesian Inference
Cross validation

- We can also use cross validation to determine the correct number of classes.
- Recall that GMMs is a generative model. We can compute the likelihood of the held-out data to determine which model (number of clusters) is more accurate.

\[
p(x_1 \cdots x_n \mid \theta) = \prod_{j=1}^{n} \left( \sum_{i=1}^{k} p(x_j \mid C = i) w_i \right)
\]
Cross validation
Gaussian mixture clustering
## Clustering methods: Comparison

<table>
<thead>
<tr>
<th></th>
<th>Hierarchical</th>
<th>K-means</th>
<th>GMM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Running time</strong></td>
<td>naively, (O(N^3))</td>
<td>fastest (each iteration is linear)</td>
<td>fast (each iteration is linear)</td>
</tr>
<tr>
<td><strong>Assumptions</strong></td>
<td>requires a similarity / distance measure</td>
<td>strong assumptions</td>
<td>strongest assumptions</td>
</tr>
<tr>
<td><strong>Input parameters</strong></td>
<td>none</td>
<td>(K) (number of clusters)</td>
<td>(K) (number of clusters)</td>
</tr>
<tr>
<td><strong>Clusters</strong></td>
<td>subjective (only a tree is returned)</td>
<td>exactly (K) clusters</td>
<td>exactly (K) clusters</td>
</tr>
</tbody>
</table>
What you should know about Mixture Models

• Gaussian mixture models
  – Probabilistic extension of K-means for soft-clustering
  – EM algorithm for learning by assuming data are only partially observed
    • Cluster labels are treated as the unobserved part of data

• EM algorithm for learning from partly unobserved data
  – MLE of $\theta = \arg\max_{\theta} \log P(data|\theta)$
  – EM estimate: $\theta = \arg\max_{\theta} E_{Z|X,\theta}[\log P(X, Z|\theta)]$
    • Where $X$ is observed part of data, $Z$ is unobserved