Clustering  (Bishop ch 9)

Reference:

Data Mining by Margaret Dunham (a slide source)

On-line instructor evaluations – in class TA evals
Clustering

- Clustering is *unsupervised* learning, there are no class labels
- Want to find groups of “similar” instances
- Often use a distance measure (such as Euclidean distance) for dis-similarity
- Can use cluster membership/distances as additional (created) features
Clustering Examples

- **Segment** customer database based on similar buying patterns.
- Group houses in a town into neighborhoods based on features (location, sq ft, stories, lot size)
- Identify new plant species
- Identify similar Web usage patterns
Clustering Problem

- Given data \( D=\{x_1, x_2, \ldots, x_n\} \) of feature vectors and an integer value \( k \), the **Clustering Problem** is to define a mapping where each \( x_i \) is assigned to one cluster \( K_j \), \( 1 \leq j \leq k \).

- A **Cluster**, \( K_j \), contains precisely those vectors mapped to it.

- Unlike classification problem, clusters are not known a priori.
Impact of Outliers on Clustering

What are the “best” two clusters?
Types of Clustering

- **Hierarchical** – Creates Tree of clusterings
  - Agglomerative (bottom up – merge “closest”)
  - Divisive (top down - less common)
- **Partitional** – One set of clusters created, usually # of clusters supplied by user

- Clusters often non-overlapping (“hard”), but sometimes “soft”
Agglomerative: merge closest clusters

- Which clusters are closest?
- **Single Link**: smallest distance between points
- **Complete Link**: largest distance between points
- **Average Link**: average distance between points
- **Centroid**: distance between centers
Levels of Clustering

a) Six Clusters

b) Four Clusters

c) Three Clusters

d) Two Clusters

e) One Cluster
Dendrogram

• **Dendrogram**: a tree data structure which illustrates hierarchical clustering techniques.
• Each level shows clusters for that level.
  – Leaf – individual clusters
  – Root – one cluster
• A cluster at level $i$ is the union of its children clusters at level $i+1$. 

![Dendrogram Diagram](image)
Partitional Clustering

• Non-hierarchical - creates one level of clustering
• Since only one set of clusters is output, the user normally has to input the desired number of clusters, k.
• Often try different k and use “best” one
Partitional Algorithms

• K-Means
• Gaussian mixtures (EM)
• Many, many others
K-means clustering

1. Pick $K$ starting means, $\mu_1, \mu_2, \ldots, \mu_K$
   Can use: randomly picked examples, perturbations of sample mean, or equally spaced along principle component

2. Repeat until convergence:
   1. Split data into $K$ sets, $S_1, S_2, \ldots, S_K$
      s.t. $x_i \in S_k$ iff $\mu_k$ closest mean to $x_i$
   2. Update each $\mu_k$ to mean of $S_k$
k-means: Initial

After 1 iteration

After 2 iterations

After 3 iterations
K-Means Example

• Given: \{2,4,10,12,3,20,30,11,25\}, k=2
• Randomly assign means: m_1=3, m_2=4
• K_1={2,3}, K_2={4,10,12,20,30,11,25}, m_1=2.5, m_2=16
• K_1={2,3,4}, K_2={10,12,20,30,11,25}, m_1=3, m_2=18
• K_1={2,3,4,10}, K_2={12,20,30,11,25}, m_1=4.75, m_2=19.6
• K_1={2,3,4,10,11,12}, K_2={20,30,25}, m_1=7, m_2=25
• Stop as the clusters with these means stay the same.
Why \( k \)-means works

• For each example \( i \), let \( d_i \) be distance to mean of example’s cluster
• Consider sum of \( d_i \)
• Both steps of \( k \)-means reduces the sum
Tabular view of k-means

<table>
<thead>
<tr>
<th>Clusters</th>
<th>$\mu_1$</th>
<th>$\mu_2$</th>
<th>$\mu_3$</th>
<th>$\mu_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Soft k-means clustering

<table>
<thead>
<tr>
<th>EXAMPLES</th>
<th>Clusters</th>
<th>( \mu_1 )</th>
<th>( \mu_2 )</th>
<th>( \mu_3 )</th>
<th>( \mu_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>( x_1 )</td>
<td>.6</td>
<td>.2</td>
<td>.1</td>
<td>.1</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>( x_2 )</td>
<td>.2</td>
<td>.1</td>
<td>.5</td>
<td>.2</td>
</tr>
<tr>
<td>( x_3 )</td>
<td>( x_3 )</td>
<td>.4</td>
<td>.3</td>
<td>.1</td>
<td>.2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
From Soft clustering to EM

- Use weighted mean based on soft-clustering weights
- Soft cluster weights are probabilities: $P(\text{cluster} \mid \mathbf{x})$
- Uses Bayes’ rule: $P(\text{cluster} \mid \mathbf{x}) \propto P(\mathbf{x} \mid \text{cluster}) P(\text{cluster})$
- For each $\mathbf{x}$, the true cluster for $\mathbf{x}$ is a latent (unobserved) variable
C nominal: 1, 2, …, k unobserved

x Gaussian
Conditioned on C, Given in the data
Soft clustering to EM 2

• Assume parametric forms for
  – $P(\text{cluster})$ (multinomial)
  – $P(x|\text{cluster})$ (Gaussian)

• Iteratively:
  1. Smoothly estimate the cluster membership (latent variables) based on data and old parameters
  2. Update parameters to maximize the likelihood of data assuming new estimates are “truth”

  – This is the mixture of Gaussian EM algorithm
    see http://citeseer.ist.psu.edu/bilmes98gentle.html
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(c)$</td>
<td>.3</td>
<td>.2</td>
<td>.2</td>
<td>.3</td>
</tr>
<tr>
<td>$P(x</td>
<td>c)$ parameters</td>
<td>$\mu_1, \sigma_1$</td>
<td>$\mu_2, \sigma_2$</td>
<td>$\mu_3, \sigma_3$</td>
</tr>
<tr>
<td>$x_1$</td>
<td></td>
<td></td>
<td>.2</td>
<td>.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.1</td>
</tr>
<tr>
<td>$x_2$</td>
<td>.2</td>
<td>.1</td>
<td>.5</td>
<td>.2</td>
</tr>
<tr>
<td>$x_3$</td>
<td>.4</td>
<td>.3</td>
<td>.1</td>
<td>.2</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Problems with EM

- Local minima
  - Run several times, take best result
  - Use good initialization (perhaps k-means)
- Degenerate Gaussians - as $\sigma$ goes to zero, the likelihood goes to $\infty$
  - Fix a lower bound on $\sigma$
- Lots of parameters to learn
  - Use spherical Gaussians or shared co-variance matrices (or even fixed distributions)
EM summary

• Iterative method for maximizing likelihood
• General method - not just for Gaussian mixtures, but also HMMs, Bayes’ nets, etc.
• Generally works well, but can have local minima and degenerate situations
• Gets both clustering and distribution (mixture of Gaussians) - distributions can be used for Bayesian learning (e.g. learn $P(x|y)$ using a gaussian mixture model)
Mixture of Mixtures

• In classification, the input comes from a mixture of classes (supervised).
• If each class-conditional distribution is also a mixture, e.g., of Gaussians, we have a mixture of mixtures:

\[
p(x \mid C_i) = \sum_{j=1}^{k_i} p(x \mid G_{ij}) p(G_{ij})
\]

\[
p(x) = \sum_{i=1}^{K} p(x \mid C_i) p(C_i)
\]
EM summary

- Iterative method for maximizing likelihood
- General method - not just for Gaussian mixtures, but also HMMs, Bayes’ nets, etc. with latent (hidden) variables
- Generally works well, but can have local minima and degenerate situations
- Gets both clustering and distribution (mixture of Gaussians) - distributions can be used for Bayesian learning (e.g. learn $p(x|t)$ using a Gaussian mixture model)