Pattern Clustering

• Unsupervised Clustering
  – Find interesting patterns or groupings in a given set of data.
    • Example: perform the task of “segmenting” the images (i.e., partitioning pixels on an image into regions that correspond to different objects or different faces of objects in the images).
Pattern Clustering

• Conventional clustering algorithms find a “Hard partition” of a given dataset based on certain criteria that evaluate the goodness of a partition.
  – “Hard Partition” - each datum belongs to exactly one cluster of the partition.
  – In many real-world clustering problems, some data points partially belong to multiple clusters (e.g. A pixel in a MRI may correspond to a mixture of two different types of tissues.)
Soft Clustering Algorithms

- Find a “soft partition” of a given data set based on certain criteria.
  - A datum can partially belong to multiple clusters
  - In practice, most soft clustering algorithms do generate a soft partition that also forms a fuzzy partition.
Fuzzy C-Means Algorithm

• Best known fuzzy clustering algorithm
  – produces a constrained soft partition
  • I.e.
    \[ \sum_{c_l} \mu_{ci}(x_l) = 1 \]

• Both fuzzy c-means and probabilistic clustering produce a partition of similar properties, the clustering criteria underlying these algorithms are very different!
Fuzzy C-Means Algorithm

• Generalizes a hard clustering algorithm called the c-means algorithm
  – Hard C-Means algorithm (HCM) aims to identify compact well-separated clusters
    • compact cluster has a “ball like” shape
    • the center of the ball is called the center or the prototype of the cluster.
Example of two clusters that are not compact and well separated
Example of two clusters that are compact, but not well separated
Fuzzy C-Means Algorithm

- Like HCM, fuzzy c-means also tries to find a good partition by searching for prototypes \( v_i \) that minimize the objective function \( J_m \).

\[
J_m(P, V) = \sum_{i=1}^{k} \sum_{x_k \in X} (\mu_{C_i}(x_k))^m \| x_k - v_i \|^2
\]

- FCM also needs to search for membership function \( \mu_{C_i} \) that minimize \( J_m \).
Fuzzy C-Means Theorem

- A constrained fuzzy partition \{C_1, C_2, ..., C_k\} can be a local minimum of the objective function \( J_m \) only if the following conditions are satisfied:

\[
\mu_{C_i}(x) = \frac{1}{\sum_{j=1}^{k} \left( \frac{||x-v_j||^2}{||x-v_i||^2} \right)^{m-1}} \quad 1 \leq i \leq k, \, x \in X \quad \text{(A)}
\]

\[
v_i = \frac{\sum_{x \in X} (\mu_{C_i}(x))^m \times x}{\sum_{x \in X} (\mu_{C_i}(x))^m} \quad 1 \leq i \leq k \quad \text{(B)}
\]

- Based on this theorem, FCM updates the prototypes and the membership function iteratively using Equation 13.4 and 13.5 until a convergence criterion is reached.
Fuzzy C-Means Algorithm

FCM \((X, c, m, \varepsilon)\)

- \(X\): an unlabeled data set
- \(c\): the number of clusters to form
- \(m\): the parameter in the objective function
- \(\varepsilon\): a threshold for the convergence criteria

Initialize prototype \(V = \{v_1, v_2, \ldots, v_c\}\)

Repeat

\[
V^{\text{Previous}} \leftarrow V
\]

Compute membership functions using Equation \(B\)

Update the prototype, \(v_i\) in \(V\) using Equation \(A\)

Until \(\sum_{i=1}^{\infty} \| v_i^{\text{Previous}} - v_i \| \leq \varepsilon\)
Fuzzy C-Means - Example

- Use FCM to partition the data set into two clusters (c=2), suppose we set the parameter m in FCM at 2, and initial prototypes to \( v_1 = (5,5) \) \( v_2 = (10,10) \)
Fuzzy C-Means - Example

Fuzzy Logic: Intelligence, Control, and Information, J. Yen and R. Langari, Prentice Hall
Fuzzy C-Means - Example

- The initial membership functions of the two clusters are calculated using Equation A

\[
\mu_{C_1}(x_1) = \frac{1}{\sum_{j=1}^{2} \left( \frac{\|x_1 - v_1\|^2}{\|x_1 - v_j\|^2} \right)}
\]

\[
\|x_1 - v_1\|^2 = 32 + 7^2 = 9 + 49 = 58
\]

\[
\|x_1 - v_2\|^2 = 8^2 + 2^2 = 64 + 4 = 68
\]

\[
\mu_{C_1}(x_1) = \frac{1}{\frac{58}{58} + \frac{58}{68}} = \frac{1}{1 + 0.853} = 0.5397
\]
Example (Continued)

Similarly,

\[
\begin{align*}
\mu_{C_2}(x_1) &= \frac{1}{68 + 68} = 0.4603 \\
\mu_{C_1}(x_2) &= \frac{1}{17 + 17} = 0.6852 \\
\mu_{C_2}(x_2) &= \frac{1}{37 + 37} = 0.3148 \\
\mu_{C_1}(x_3) &= \frac{1}{68 + 68} = 0.2093 \\
\mu_{C_2}(x_3) &= \frac{1}{18 + 18} = 0.7907 \\
\mu_{C_1}(x_4) &= \frac{1}{36 + 36} = 0.4194 \\
\mu_{C_2}(x_4) &= \frac{1}{26 + 26} = 0.5806 \\
\mu_{C_1}(x_5) &= \frac{1}{53 + 53} = 0.197
\end{align*}
\]
Example (Continued)

- Similarly,

\[
\mu_{C_2}(x_5) = \frac{1}{\frac{13}{53} + \frac{13}{13}} = 0.803
\]

\[
\mu_{C_1}(x_6) = \frac{1}{\frac{82}{82} + \frac{82}{52}} = 0.3881
\]

\[
\mu_{C_2}(x_6) = \frac{1}{\frac{52}{82} + \frac{52}{52}} = 0.6119
\]
Example (Continued)

- FCM algorithm then updates the prototypes according to Equation B
  - For $V_1$

\[
V_1 = \frac{\sum_{k=1}^{6} (\mu_{C_1}(x_k))^2 \times x_k}{\sum_{k=1}^{6} (\mu_{C_1}(x_k))^2}
\]

\[
\begin{align*}
&= \frac{0.5397^2 \times (2, 12) + 0.6852^2 \times (4, 9) + 0.2093^2 \times (7, 13) + 0.4194^2 \times (11, 5) + 0.1972^2 \times (12, 7) + 0.3881^2 \times (14, 4)}{0.5397^2 + 0.6852^2 + 0.2093^2 + 0.4194^2 + 0.1972^2 + 0.3881^2} \\
&= (7.2761, 10.044) \\
&= (6.6273, 9.1484)
\end{align*}
\]
Example (Continued)

- FCM algorithm then updates the prototypes according to Equation B
  - For $V_2$

\[
\begin{align*}
V_2 &= \frac{\sum_{k=1}^{6} (\mu_{C_2}(x_k))^2 \times x_k}{\sum_{k=1}^{6} (\mu_{C_2}(x_k))^2} \\
&= \frac{0.4603^2 \times (2, 12) + 0.3148^2 \times (4, 9) + 0.7909^2 \times (7, 13) + 0.5806^2 \times (11, 5) + 0.803^2 \times (12, 7) + 0.6119^2 \times (14, 4)}{0.4603^2 + 0.3148^2 + 0.7909^2 + 0.5806^2 + 0.803^2 + 0.6119^2} \\
&= \begin{pmatrix} 22.326 & 19.4629 \\ 2.2928 & 2.2928 \end{pmatrix} \\
&= (9.7374, 8.4887)
\end{align*}
\]
Example (Continued)

- The updated prototype $v_1$ and $v_2$ are move closer to the center of the cluster formed by $x_1, x_2, x_3$ and $x_4, x_5, x_6$, respectively.

- Show Matlab example: run clustex.
Pattern Recognition Systems

Bezdex et al., 1986
Classification and Pattern Recognition

- Training data for classification
- Classification
- Pattern recognition
- Discrimination

Feedback

New data

Dong, 1987
Image Processing

\[ X = f(m, n) = \begin{bmatrix}
  x_{11} & x_{12} & \cdots & x_{1n} & \cdots & x_{1N} \\
  x_{21} & x_{22} & \cdots & x_{2n} & \cdots & x_{2N} \\
  x_{31} & x_{32} & \cdots & x_{3n} & \cdots & x_{3N} \\
  \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\
  x_{M1} & x_{M2} & \cdots & x_{Mn} & \cdots & x_{MN}
\end{bmatrix} \]
Fuzzy sets: Pixels

\[ X = \begin{bmatrix}
  \mu_{11}/x_{11} & \mu_{12}/x_{12} & \cdots & \mu_{1n}/x_{1n} & \cdots & \mu_{1N}/x_{1N} \\
  \mu_{21}/x_{21} & \mu_{22}/x_{22} & \cdots & \mu_{2n}/x_{2n} & \cdots & \mu_{2N}/x_{2N} \\
  \mu_{31}/x_{31} & \mu_{32}/x_{32} & \cdots & \mu_{3n}/x_{3n} & \cdots & \mu_{3N}/x_{3N} \\
  \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
  \mu_{M1}/x_{M1} & \mu_{M2}/x_{M2} & \cdots & \mu_{Mn}/x_{Mn} & \cdots & \mu_{MN}/x_{MN}
\end{bmatrix} \]