# Hierarchical Clustering
# *k*-Means Clustering
# Khasha Dehnad

# Clustering Task

- Clustering refers to grouping records, observations, or tasks into classes of similar objects
- Cluster is collection records similar to one another
- Records in one cluster dissimilar to records in other clusters
- Clustering is unsupervised data mining task
- Therefore, no target variable specified
- Clustering algorithms segment records and maximize homogeneity in subgroups
- Similarity to records outside cluster minimized

# Clustering Task *(cont'd)*

- For example, Claritas, Inc. provides demographic profiles of geographic areas, according to zip code
- PRIZM segmentation system clusters zip codes in terms of lifestyle types
- Recall clusters identified for 90210 Beverly Hills, CA
- Cluster 01: *Blue Blood Estates*

  "Established executives, professionals, and 'old money' heirs that live in America's wealthiest suburbs..."
- Cluster 10: *Bohemian Mix*
- Cluster 02: *Winner's Circle*
- Cluster 07: *Money and Brains*
- Cluster 08: *Young Literati*

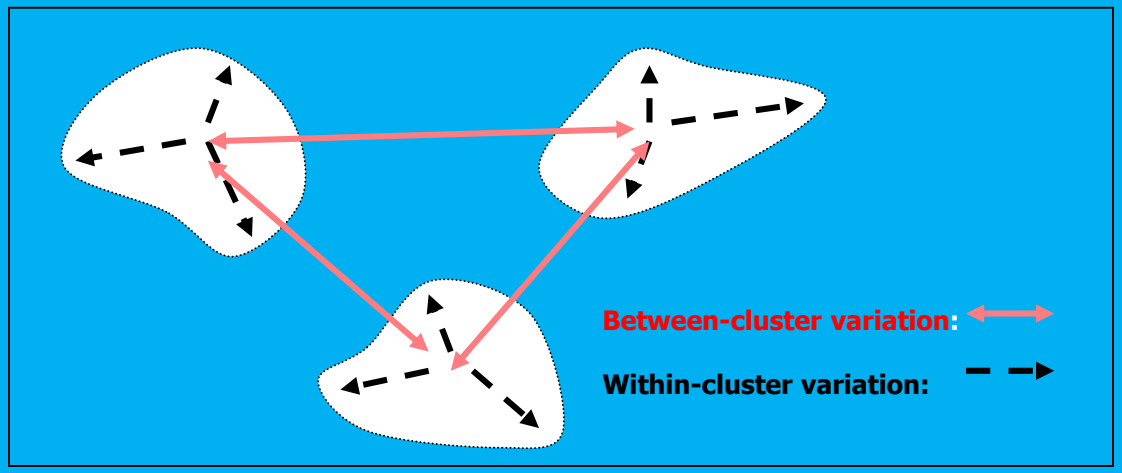
http://www.claritas.com/MyBestSegments/Default.jsp?ID=20#

# Clustering Task *(cont'd)*

- Clustering Tasks in Business and Research
  - Target marketing for niche product, without large marketing budget
  - Segment financial behavior into benign and suspicious categories
  - Gene expression clustering, where genes exhibit similar characteristics

  - Clustering often performed as preliminary step in data mining process
  - Clustering results used as input to other data mining techniques

# Clustering Task *(cont'd)*




- Clustering identifies groups of highly-similar records
- Algorithms construct clusters where between-cluster variation (BCV) large, as compared to within-cluster variation (WCV)
- Analogous to concept behind analysis of variance

# Clustering Task *(cont'd)*

- Applying cluster analysis to enormous databases helpful
- Reduces search space for downstream algorithms

- Cluster analysis addresses similar issues encountered in classification
  - Similarity measurement
  - Recoding categorical variables
  - Standardizing and normalizing variables
  - Number of clusters

# Distance Function

- How is similarity defined between an unclassified record and its neighbors?
- A distance metric is a real-valued function *d* used to measure the similarity between coordinates *x*, *y*, and *z* with properties:

1. $d(x, y) \geq 0, \text{and } d(x, y) = 0 \text{ if and only if } x = y$
2. $d(x, y) = d(y, x)$
3. $d(x, z) \leq d(x, y) + d(y, z)$

- Property 1: Distance is always non-negative
- Property 2: Commutative, distance from "A to B" is distance from "B to A"
- Property 3: Triangle inequality holds, distance from "A to C" must be less than or equal to distance from "A to B to C"

# Clustering Task *(cont'd)*

- Measuring Similarity
  - Euclidean Distance measures distance between records

$$d_{\text{Euclidean}}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}, \text{ where}$$

$\mathbf{x} = x_1, x_2, \ldots, x_m$ and $\mathbf{y} = y_1, y_2, \ldots, y_m$ represent $m$ attribute values of two records

  - Other distance measurements include City-Block Distance and Minkowski Distance

$$d_{\text{City-Block}}(\mathbf{x},\mathbf{y}) = \sum_i |x_i - y_i|$$

$$d_{\text{Minkowski}}(\mathbf{x},\mathbf{y}) = \sum_i |x_i - y_i|^q$$

# Clustering Task *(cont'd)*

- "Different From" function measures similarity between categorical attributes

$$\text{different}(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{otherwise} \end{cases}$$

- Substitute *different(x,y)* for each categorical attribute in Euclidean Distance function
- Normalizing data enhances performance of clustering algorithms
- Use Min-max Normalization or Z-Score Standardization

$$\text{Min - Max Normalization} = \frac{X - \min(X)}{\max(X) - \min(X)} \qquad \text{Z - Score Standardization} = \frac{X - \text{mean}(X)}{\text{standard deviation}(X)}$$

# Hierarchical Clustering Methods

- Clustering algorithms either Hierarchical or Non-Hierarchical
- Hierarchical
  - Tree like cluster structure (dendogram) created through recursive partitioning (Divisive Methods) or combining (Agglomerative Methods) existing clusters
  - Divisive Methods
  - All records initialized into single cluster
  - At each iteration, most dissimilar record split off into separate cluster
  - Continues until each record represents single cluster

# Hierarchical Clustering Methods *(cont'd)*

- Agglomerative Methods
- Each observation initialized to become its own cluster
- At each iteration two closest clusters aggregated together
- Number of clusters reduced by one, each step
- Eventually, all records combined into single cluster
- Agglomerative more popular hierarchical method
- Therefore, focus remains on this approach
- Measuring distance between records straightforward once recoding and normalization applied
- However, how is distance between clusters determined?

# Hierarchical Clustering Methods *(cont'd)*

- Distance Between Clusters
  - Several criteria examined to determine distance between clusters, A and B
  - Single Linkage
  - Known as Nearest-Neighbor Approach
  - Minimum distance between any record in cluster A, and any record in cluster B
  - Cluster similarity based on most similar records from each cluster
  - Tends to form long, slender clusters
  - Sometime heterogeneous records clustered together

# Hierarchical Clustering Methods *(cont'd)*

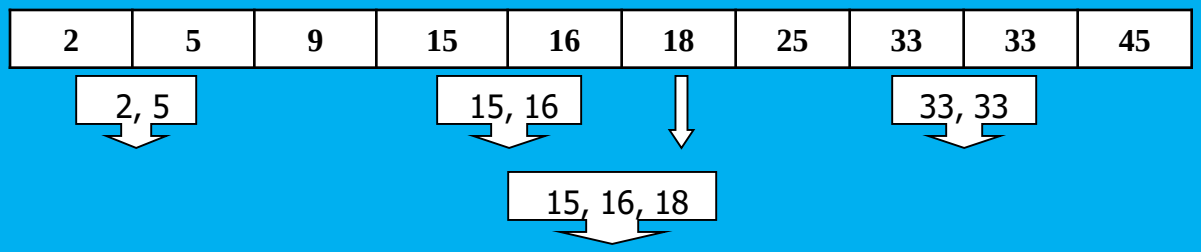
- Measure is average distance of records in cluster A, from records in cluster B
- Resulting clusters have approximately equal within-cluster variability

- Next, linkage methods examined using small data set

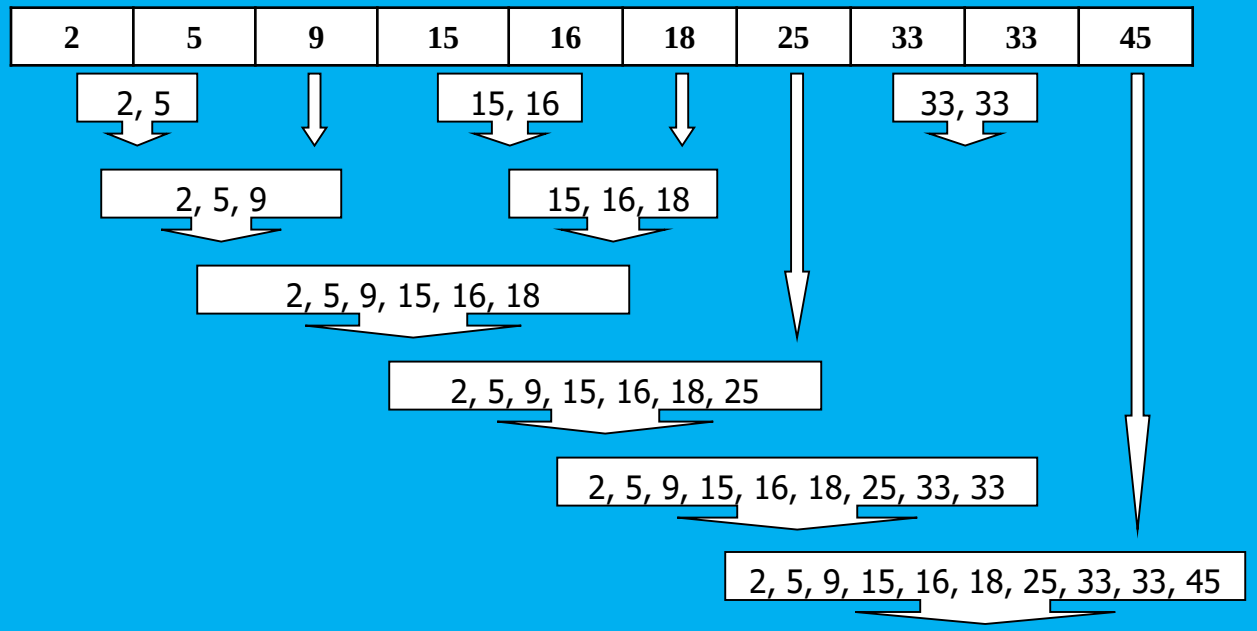
**2 5 9 15 16 18 25 33 33 45**

# Single-Linkage Clustering

- To begin, each record assigned to its own cluster
- Single-linkage seeks minimum distance between any two records, in separate clusters
- Step 1: Minimum cluster distance is between clusters {33} and {33}. Distance = 0, clusters combined
- Step 2: Clusters {15} and {16} combined, where distance = 1
- Step 3: Cluster {15, 16} combined with cluster {18}
- Step 4: Clusters {2} and {5} combined



# Single-Linkage Clustering *(cont'd)*




- Agglomeration continues similarly Steps 4 – 9
- Above, last cluster {2, 5, 9, 15, 16, 18, 25, 33, 33, 45} contains all records in data set

# *k*-Means Clustering

- *k*-Means effective at finding clusters in data

- *k*-Means Algorithm
  - Step 1: Analyst specifies *k* = number of clusters to partition data
  - Step 2: *k* records randomly assigned to initial clusters
  - Step 3: For each record, find the nearest cluster center,
    Each cluster center "owns" subset of records, we have a partition of data set into *k* clusters, $C_1$, $C_2$, ...., $C_k$
  - Step 4: For each of *k* clusters, find cluster centroid
    Update cluster center location to centroid
  - Step 5: Repeats Steps 3 – 5 until convergence or termination

# *k*-Means Clustering *(cont'd)*

- Nearest criterion in Step 3 typically Euclidean Distance

- Determining Cluster Centroid
  - Assume *n* data points ($a_1$, $b_1$, $c_1$), ($a_2$, $b_2$, $c_2$), ..., ($a_n$, $b_n$, $c_n$)
  - Centroid of points is center of gravity of points
  - Located at point ($\Sigma a_i/n$, $\Sigma b_i/n$, $\Sigma c_i/n$)
  - For example, points (1, 1, 1), (1, 2, 1), (1, 3, 1), and (2, 1, 1) have centroid

$$\left(\frac{1+1+1+2}{4}, \frac{1+2+3+1}{4}, \frac{1+1+1+1}{4}\right) = (1.25, 1.75, 1.00)$$

# *k*-Means Clustering *(cont'd)*

- *k*-Means algorithm terminates when centroids no longer change
- For *k* clusters, $C_1$, $C_2$, ...., $C_k$, all records "owned" by cluster remain in cluster

- Convergence criterion may also cause termination
- For example, no significant reduction in SSE

$$SSE = \sum_{i=1}^{k} \sum_{p \in C_i} d(p, m_i)^2, \text{ where}$$

$p \in C_i$ = each data point in cluster i

$m_i$ = represents centroid of cluster i

# Example of *k*-Means Clustering at Work

- Assume $k = 2$ to cluster following data points

| a | b | c | d | e | f | g | h |
|---|---|---|---|---|---|---|---|
| (1, 3) | (3, 3) | (4, 3) | (5, 3) | (1, 2) | (4, 2) | (1, 1) | (2, 1) |

- Step 1: $k = 2$ specifies number of clusters to partition
- Step 2: Randomly assign $k = 2$ cluster centers
  For example, $m_1 = (1, 1)$ and $m_2 = (2, 1)$

# Example of *k*-Means Clustering at Work

First Pass (Copied from book)

| | Centroid | |
|---|---|---|
| | d1 | d2 |
| m1 | 1 | 1 |
| m2 | 2 | 1 |

Clustering

| Point | d1 | d2 | Distance from m1 | Distance from m2 | Cluster Membership | SE |
|---|---|---|---|---|---|---|
| a | 1 | 3 | | | | |
| b | 3 | 3 | | | | |
| c | 4 | 3 | | | | |
| d | 5 | 3 | | | | |
| e | 1 | 2 | | | | |
| f | 4 | 2 | | | | |
| g | 1 | 1 | | | | |
| h | 2 | 1 | | | | |
| SSE | | | | | | |
| BCV | | | | | | |
| BCV/WCV | | | | | | |

# Example of *k*-Means Clustering at Work

First Pass (Copied from book)

| | Centroid d1 | d2 |
|---|---|---|
| m1 | 1 | 1 |
| m2 | 2 | 1 |

Clustering

| Point | d1 | d2 | Distance from m1 | Distance from m2 | Cluster Membership | SE |
|---|---|---|---|---|---|---|
| a | 1 | 3 | 2.00 | 2.24 | C1 | 4 |
| b | 3 | 3 | 2.83 | 2.24 | C2 | 5 |
| c | 4 | 3 | 3.61 | 2.83 | C2 | 8 |
| d | 5 | 3 | 4.47 | 3.61 | C2 | 13 |
| e | 1 | 2 | 1.00 | 1.41 | C1 | 1 |
| f | 4 | 2 | 3.16 | 2.24 | C2 | 5 |
| g | 1 | 1 | 0.00 | 1.00 | C1 | 0 |
| h | 2 | 1 | 1.00 | 0.00 | C2 | 0 |

| | |
|---|---|
| SSE | 36.00 |
| BCV | 1 |
| BCV/WCV | 0.028 |

Centroid (Newly calculated)

| | d1 | d2 |
|---|---|---|
| m1 | 1 | 2 |
| m2 | 3.6 | 2.4 |

# Example of *k*-Means Clustering at Work

Second Pass

| | d1 | d2 |
|---|---|---|
| m1 | 1 | 2 |
| m2 | 3.6 | 2.4 |

Clustering

| Point | d1 | d2 | Distance from m1 | Distance from m2 | Cluster Membership | SE |
|---|---|---|---|---|---|---|
| a | 1 | 3 | 1.00 | 2.67 | C1 | 1 |
| b | 3 | 3 | 2.24 | 0.85 | C2 | 0.72 |
| c | 4 | 3 | 3.16 | 0.72 | C2 | 0.52 |
| d | 5 | 3 | 4.12 | 1.52 | C2 | 2.32 |
| e | 1 | 2 | 0.00 | 2.63 | C1 | 0 |
| f | 4 | 2 | 3.00 | 0.57 | C2 | 0.32 |
| g | 1 | 1 | 1.00 | 2.95 | C1 | 1 |
| h | 2 | 1 | 1.41 | 2.13 | C1 | 2 |

| | |
|---|---|
| SSE | 7.88 |
| BCV | 2.631 |
| BCV/WCV | 0.334 |

Centroid (Newly calculated)

| | d1 | d2 |
|---|---|---|
| m1 | 1.25 | 1.75 |
| m2 | 4 | 2.75 |

# Example of *k*-Means Clustering at Work

Third Pass (Copied from book)

| | Centroid d1 | d2 |
|---|---|---|
| m1 | 1.25 | 1.75 |
| m2 | 4 | 2.75 |

Clustering

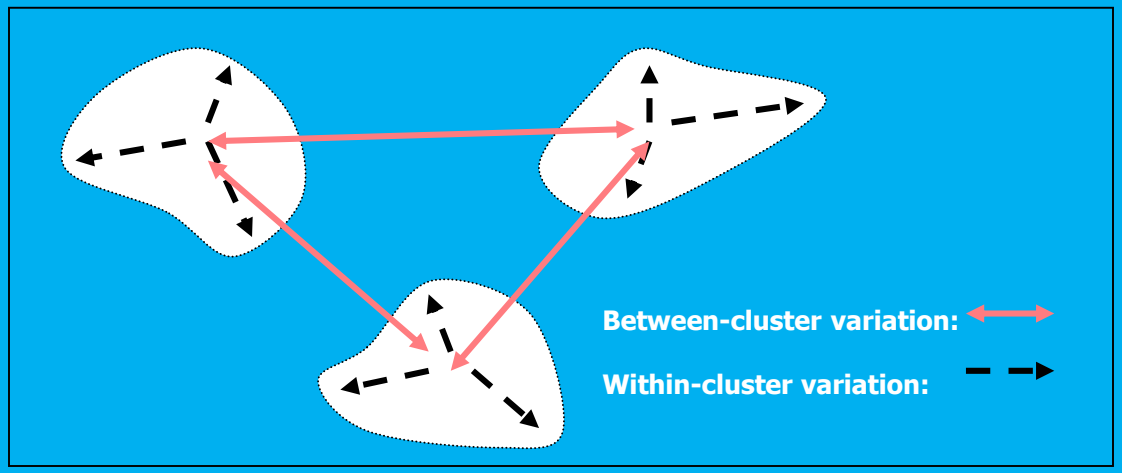
| Point | d1 | d2 | Distance from m1 | Distance from m2 | Cluster Membership | SE |
|---|---|---|---|---|---|---|
| a | 1 | 3 | 1.27 | 3.01 | C1 | 1.625 |
| b | 3 | 3 | 2.15 | 1.03 | C2 | 1.0625 |
| c | 4 | 3 | 3.02 | 0.25 | C2 | 0.0625 |
| d | 5 | 3 | 3.95 | 1.03 | C2 | 1.0625 |
| e | 1 | 2 | 0.35 | 3.09 | C1 | 0.125 |
| f | 4 | 2 | 2.76 | 0.75 | C2 | 0.5625 |
| g | 1 | 1 | 0.79 | 3.47 | C1 | 0.625 |
| h | 2 | 1 | 1.06 | 2.66 | C1 | 1.125 |

| | |
|---|---|
| SSE | 6.25 |
| BCV | 2.926 |
| BCV/WCV | 0.468 |

Centroid (Newly calculated)

| | d1 | d2 |
|---|---|---|
| m1 | 1.25 | 1.75 |
| m2 | 4 | 2.75 |

# Clustering Task *(Repeated slide)*




- Clustering identifies groups of highly-similar records
- Algorithms construct clusters where between-cluster variation (BCV) large, as compared to within-cluster variation (WCV)
- Analogous to concept behind analysis of variance

# Example of *k*-Means Clustering at Work

- Assume $k = 2$ to cluster following data points

| a | b | c | d | e | f | g | h |
|---|---|---|---|---|---|---|---|
| (1, 3) | (3, 3) | (4, 3) | (5, 3) | (1, 2) | (4, 2) | (1, 1) | (2, 1) |

- Step 1: $k = 2$ specifies number of clusters to partition
- Step 2: Randomly assign $k = 2$ cluster centers
  For example, $m_1 = (1, 1)$ and $m_2 = (2, 1)$

- First Iteration
  - Step 3: For each record, find nearest cluster center
    Euclidean distance from points to $m_1$ and $m_2$ shown

| Point | a | b | c | d | e | f | g | h |
|---|---|---|---|---|---|---|---|---|
| Distance from $m_1$ | 2.00 | 2.83 | 3.61 | 4.47 | 1.00 | 3.16 | 0.00 | 1.00 |
| Distance from $m_2$ | 2.24 | 2.24 | 2.83 | 3.61 | 1.41 | 2.24 | 1.00 | 0.00 |
| Cluster Membership | $C_1$ | $C_2$ | $C_2$ | $C_2$ | $C_1$ | $C_2$ | $C_1$ | $C_2$ |

# Example of *k*-Means Clustering at Work *(cont'd)*

- Cluster $m_1$ contains {a, e, g} and $m_2$ has {b, c, d, f, h}
- Cluster membership assigned, now SSE calculated

$$SSE = \sum_{i=1}^{k} \sum_{p \in C_i} d(p, m_i)^2$$
$$= 2^2 + 2.24^2 + 2.83^2 + 3.61^2 + 1^2 + 2.24^2 + 0^2 + 0^2 = 36$$

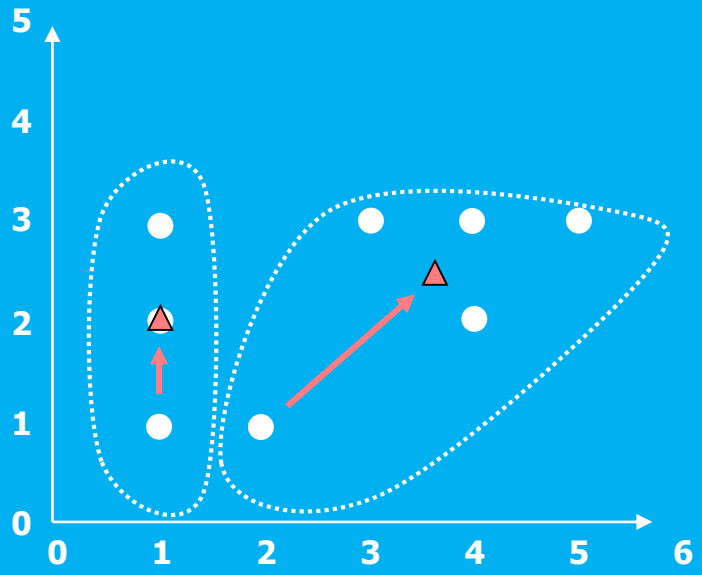
- Recall clusters constructed where between-cluster variation (BCV) large, as compared to within-cluster variation (WCV)

$$\frac{\text{BCV}}{\text{WCV}} = \frac{d(m_1, m_2)}{\text{SSE}} = \frac{1}{36} = 0.0278, \text{ where}$$
$$d(m_1, m_2) = \text{surrogate for BCV}$$
$$\text{SSE} = \text{surrogate for WCV}$$

- Ratio BCV/WCV expected to increase for successive iterations

# Example of *k*-Means Clustering at Work *(cont'd)*

- Step 4: For *k* clusters, find cluster centroid, update location
- Cluster 1 = [(1 + 1 + 1)/3, (3 + 2 + 1)/3] = (1, 2), Cluster 2 = [(3 + 4 + 5 + 4 + 2)/5, (3 + 3 + 3 + 2 + 1)/5] = (3.6, 2.4)
- Figure shows movement of clusters $m_1$ and $m_2$ (triangles) after first iteration of algorithm

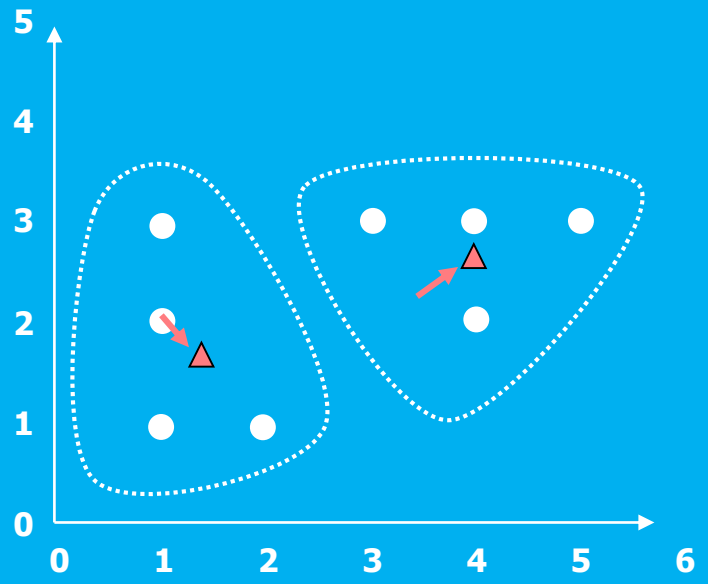


# Example of *k*-Means Clustering at Work *(cont'd)*

  - Step 5: Repeats Steps 3 – 4 until convergence or termination
- Second Iteration
  - Repeat procedure for Steps 3 – 4
  - Again, for each record find nearest cluster center $m_1$ = (1, 2) or $m_2$ = (3.6, 2.4)
  - Cluster $m_1$ contains {a, e, g, h} and $m_2$ has {b, c, d, f}
  - SSE = 7.86, and BCV/WCV = 0.3346
  - Note 0.3346 has increased compared to First Iteration value = 0.0278
  - Between-cluster variation increasing with respect to Within-cluster variation

# Example of *k*-Means Clustering at Work *(cont'd)*

- Cluster centroids updated to $m_1$ = (1.25, 1.75) or $m_2$ = (4, 2.75)
- After Second Iteration, cluster centroids shown to move slightly



# Example of *k*-Means Clustering at Work *(cont'd)*

- Third (Final) Iteration
  - Repeat procedure for Steps 3 – 4
  - Now, for each record find nearest cluster center $m_1$ = (1.25, 1.75) or $m_2$ = (4, 2.75)
  - SSE = 6.23, and BCV/WCV = 0.4703
  - Again, BCV/WCV has increased compared to previous = 0.3346
  - This time, no records shift cluster membership
  - Centroids remain unchanged, therefore algorithm terminates

# Example of *k*-Means Clustering at Work *(cont'd)*

- Summary
  - *k*-Means not guaranteed to find to find global minimum SSE
  - Instead, local minimum found
  - Invoking algorithm using variety of initial cluster centers improves probability of achieving global minimum
  - One approach places first cluster at random point, with remaining clusters placed far from previous centers (Moore)
  - What is appropriate value for *k*?
  - Potential problem for applying *k*-Means
  - Analyst may have *a priori* knowledge of *k*

# Example of *k*-Means Clustering at Work *(cont'd)*

- Outer loop to algorithm possible
- Cycles through different *k* values
- Results compared, selecting solution with smallest SSE

- What attributes to use as input?
- Some attributes likely more relevant than others
- Apply axis-stretching methods for quantifying attribute relevance discussed in Chapter 5