Clustering

M.M. Pedram
pedram@tmu.ac.ir
Tarbiat Moallem University of Tehran
(Fall 2011)

1

Clustering Outline

Goal : Provide an overview of the clustering problem and introduce some of the basic algorithms.

- Clustering Problem Overview
- Clustering Techniques
  - Hierarchical Algorithms
  - Partitional Algorithms
  - Genetic Algorithm
  - Clustering Large Databases

2

Clustering Examples

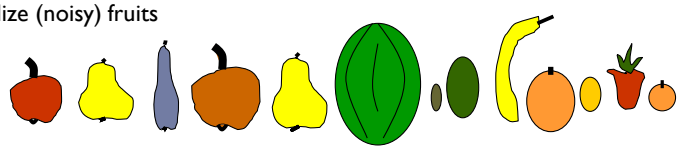
- ***Segment*** customer database based on similar buying patterns.
- Identify new plant species.
- Group houses in a town into neighborhoods based on similar features.
- Identify similar Web usage patterns

3

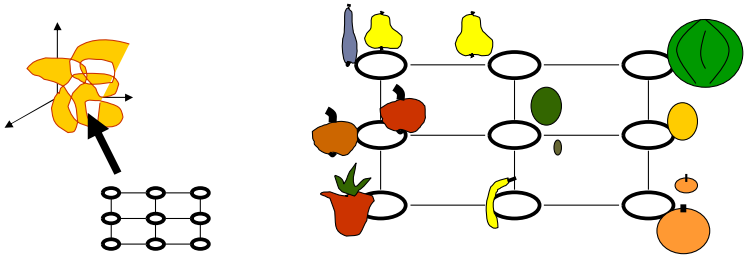
Identify new plant species

unsupervised methods:

Visualize (noisy) fruits



representation: $(\emptyset x, \emptyset y, \emptyset x/\emptyset y, \text{curvature}, \text{color}, \text{hardness}, \text{weight}, \cdots) \in \mathbb{R}^n$



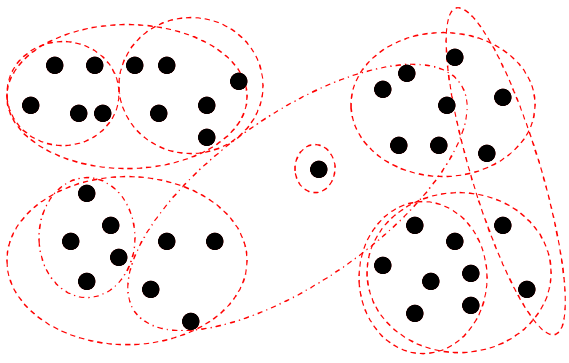
4

Clustering Example

Income	Age	Children	Marital Status	Education
$25,000	35	3	Single	High School
$15,000	25	1	Married	High School
$20,000	40	0	Single	High School
$30,000	20	0	Divorced	High School
$20,000	25	3	Divorced	College
$70,000	60	0	Married	College
$90,000	30	0	Married	Graduate School
$200,000	45	5	Married	Graduate School
$100,000	50	2	Divorced	College

5

Clustering Houses



Geographic Distance Based

6

Clustering vs. Classification

- No prior knowledge
  - Number of clusters
  - Meaning of clusters
- Unsupervised learning

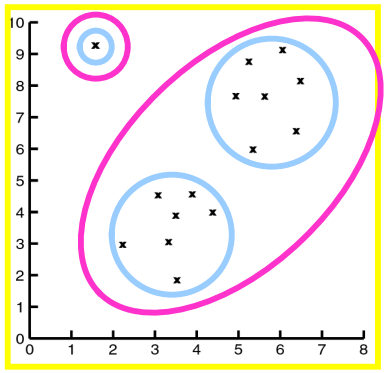
7

Clustering Issues

- Outlier handling
- Dynamic data
- Interpreting results
- Evaluating results
- Number of clusters
- Data to be used
- Scalability

8

Impact of Outliers on Clustering



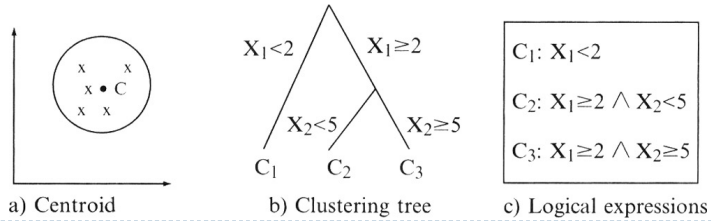
Clustering Problem

- Given a database $D=\{t_1,t_2,\ldots,t_n\}$ of tuples and an integer value k, the ***Clustering Problem*** is to define a mapping $f:D\rightarrow\{1,..,k\}$ where each t_i is assigned to one cluster K_j, $1 \leq j \leq k$.
- A ***Cluster***, K_j, contains precisely those tuples mapped to it.
- Unlike classification problem, clusters are not known *a priori*.

Description of discovered clusters

- Represent a cluster of points in an n-dimensional space (samples) by their centroid or by a set of distant (border) points in a cluster.

  Using the centroid to represent a cluster is the most popular schema. It works well when the clusters are compact or isotropic. When the clusters are elongated or non-isotropic, however, this schema fails to represent them properly.
- Represent a cluster graphically using nodes in a clustering tree.
- Represent clusters by using logical expression on sample attributes.



a) Centroid b) Clustering tree c) Logical expressions

Similarity Measures

- Some Notations:
  - X : space of samples
  - x or x_i : sample, feature vector, observation, i.e.,

    $x_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}$

    is a single data vector in a space of samples X.

  x_i can be:
  - physical object (a chair)
  - abstract object (a style of writing)

Similarity Measures

- Some Notations (Cont'd)

  x_{ij} can be:

  - Quantitative feature:
    1. continuous values (e.g., real numbers),
    2. discrete values (e.g., binary numbers {0, 1}, or integers),
    3. interval values (e.g., $\{x_{ij} \le 20, 20 < x_{ij} < 40, x_{ij} \ge 40\}$).
  - Qualitative feature:
    1. nominal or unordered (e.g., color is "blue" or "red"),
    2. ordinal (e.g., military rank with values "general", "colonel").

Similarity Measures

$s(x, x')$: Similarity measure

Properties:

- $s(x, x')$ is large when x and x' are two similar samples;
  the value of $s(x, x')$ is small when x and x' are not similar.
- similarity measure s is symmetric:

$$s(x, x') = s(x', x) \qquad \forall\, x, x' \in X$$

- similarity measure is (in most cases) normalized:

$$0 \le s(x, x') \le 1 \qquad \forall\, x, x' \in X$$

Similarity Measures

Very often a measure of dissimilarity is used instead of a similarity measure.

$d(x, x')$: dissimilarity measure

Dissimilarity is frequently called a distance.

Similarity Measures

Properties of $d(x, x')$:

- Nonnegative:

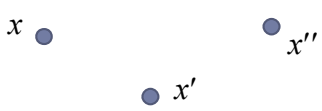
$$d(x, x') \geq 0$$

- distance measure is symmetric:

$$d(x, x') = d(x', x) \qquad \forall\, x, x' \in X$$

- distance measure requires triangular inequality (if it is accepted as a *metric distance measure*):

$$d(x, x') \leq d(x', x'') + d(x, x'') \qquad \forall\, x, x', x'' \in X$$



Similarity Measures

Common distances:

- L_1 metric or city block distance: $d_1(x_i, x_j) = \sum_{k=1}^{m} |x_{ik} - x_{jk}|$
- L_2 metric or Euclidean distance: $d_2(x_i, x_j) = \left(\sum_{k=1}^{m} (x_{ik} - x_{jk})^2 \right)^{1/2}$
- L_p metric or the Minkowski metric:

$$d_p(x_i, x_j) = \|x_i - x_j\|_p = \left(\sum_{k=1}^{m} |x_{ik} - x_{jk}|^p \right)^{1/p}$$

- Euclidian n-dimensional offers also cosine-correlation as a similarity measure:

$$S_{\cos}(x_i, x_j) = \frac{\langle x_i, x_j \rangle}{\|x_i\| \cdot \|x_j\|}$$

Distance Between Clusters

Distance Between Clusters

Measures for Distance between Clusters C_k and C_l:

a. ***Single Link*** : smallest distance between points

$$D_{\min}(C_k, C_l) = \min_{i,j} \|x_i - x_j\|_p \qquad x_i \in C_k \ , \ x_j \in C_l$$

b. ***Complete Link*** : largest distance between points

$$D_{\max}(C_k, C_l) = \max_{i,j} \|x_i - x_j\|_p \qquad x_i \in C_k \ , \ x_j \in C_l$$

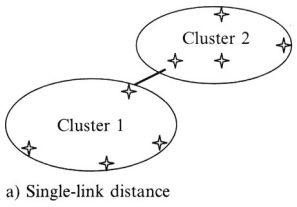
c. ***Centroid*** : distance between centroids

$$D_{mean}(C_k, C_l) = \|\mu_k - \mu_l\|_p \qquad x_i \in C_k \ , \ x_j \in C_l$$

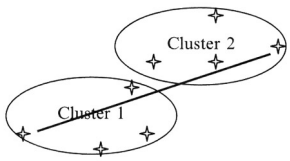
d. ***Average Link*** : average distance between points

$$D_{avg}(C_k, C_l) = \frac{1}{n_i n_j} \sum_i \sum_j \|x_i - x_j\|_p \qquad x_i \in C_k \ , \ x_j \in C_l$$

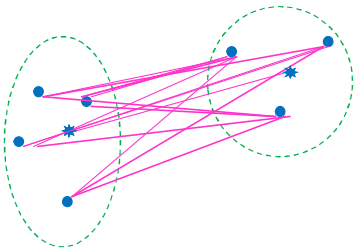
19



a) Single-link distance



b) Complete-link distance



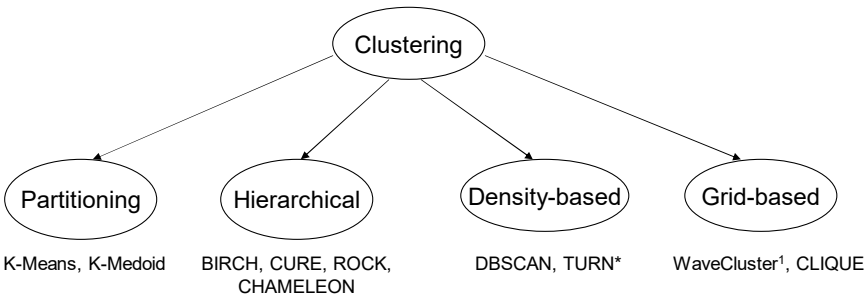
20

Types of Clustering

- ***Hierarchical*** : Nested set of clusters created.
- ***Partitional*** : One set of clusters created.
- ***Incremental*** : Each element handled one at a time.
- ***Simultaneous*** : All elements handled together.
- ***Overlapping/Non-overlapping***

21

clustering algorithms



[1]WaveCluster: A novel clustering approach based on wavelet transforms. Applies a multi-resolution grid structure on the data space. For more details, refer to "Wavecluster: a multi-resolution clustering approach for very large spatial databases", Proc. 24th Conf. on Very Large Databases.

22

Clustering Methods

1. Hierarchical clustering
   - Agglomerative (Bottom-up)
   - Conglomerative or Divisive (Top-Down)
2. Iterative square-error partitional clustering
3. Incremental

Hierarchical Clustering

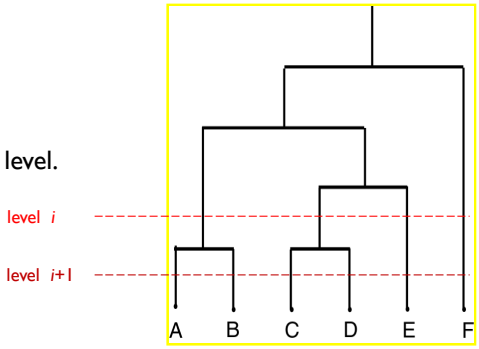
- Clusters are created in levels actually creating sets of clusters at each level.
- ***Agglomerative***
  - Initially each item in its own cluster
  - Iteratively clusters are merged together
  - Bottom Up
  - Most agglomerative algorithms are variants of the *single* or *complete-link* algorithms. They characterize the similarity between a pair of clusters.
  - More common than Divisive.
- **Conglomerative** or ***Divisive***
  - Initially all items in one cluster
  - Large clusters are successively divided
  - Top Down

Hierarchical Algorithms

- Single Link
- MST Single Link
- Complete Link
- Average Link

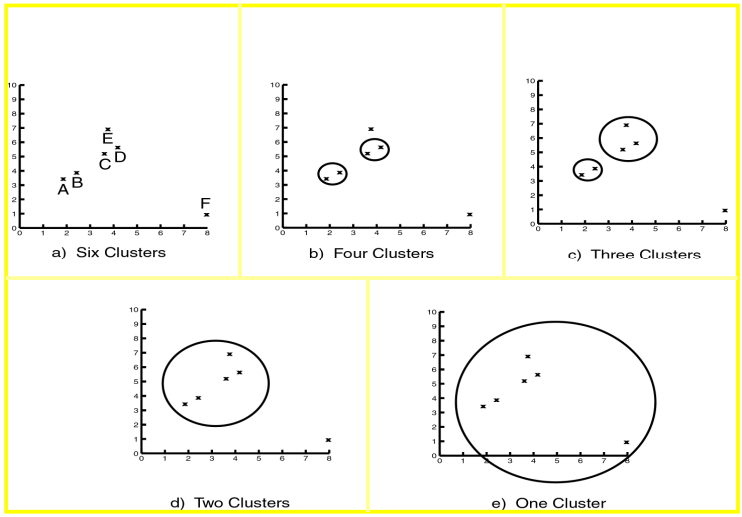
Dendrogram

- ***Dendrogram:*** a tree diagram (or 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.

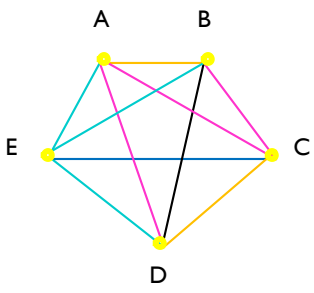
Levels of Clustering

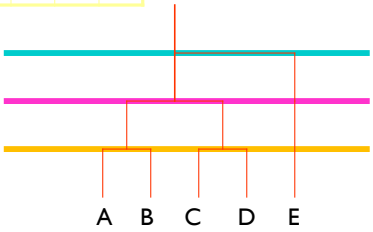


27

Agglomerative Example

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0



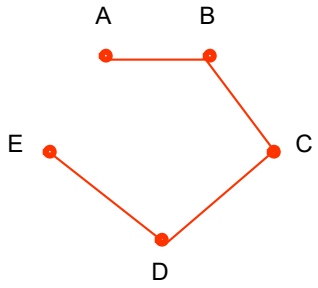


Threshold of

1 2 3 4 5

28

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0



29

Agglomerative Algorithm

Input:
$D = \{t_1, t_2, ..., t_n\}$ // Set of elements
A // Adjacency matrix showing distance between elements.
Output:
DE // Dendrogram represented as a set of ordered triples.
Agglomerative Algorithm:
$d = 0$;
$k = n$;
$K = \{\{t_1\}, ..., \{t_n\}\}$;
$DE = \{< d, k, K >\}$; // Initially dendrogram contains each element in its own cluster.
repeat
$oldk = k$;
$d = d + 1$;
A_d = Vertex adjacency matrix for graph with threshold distance of d;
$< k, K >= NewClusters(A_d, D)$;
if $oldk \neq k$ then
$DE = DE \cup < d, k, K >$; // New set of clusters added to dendrogram.
until $k = 1$

30

Agglomerative Algorithm

1. START: Clusters $C_1, C_2, ..., C_n$ each containing one object (singleton clusters)
2. Number_of_clusters = n
3. Compute the similarity matrix (or distance matrix)
4. **REPEAT**
   - Search the most similar pair of clusters C_i and C_j, merge C_i and C_j (C_{ij})
   - Compute the similarity of C_{ij} and the remaining clusters
   - Adjust the similarity matrix
   - Number_of_clusters --

UNTIL Number_of_clusters = 1

Note: Similarity measure could be any mentioned ones.

Agglomerative Example

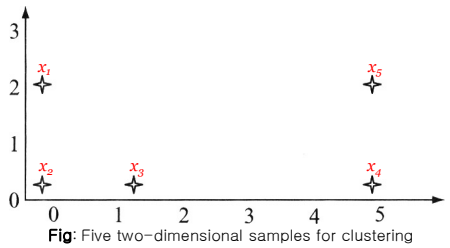


Fig: Five two-dimensional samples for clustering

x_1=(0,2)
x_2=(0,0)
x_3=(1.5,0)
x_4=(5,0)
x_5=(5,2)

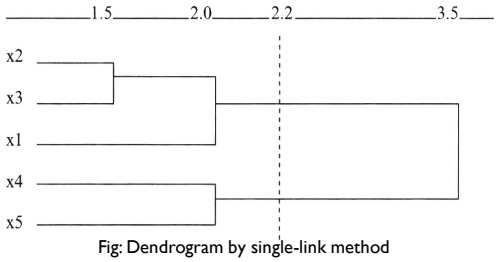
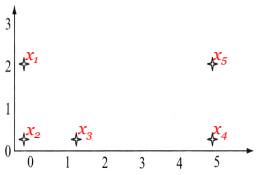


Fig: Dendrogram by single-link method

threshold measure of similarity s = 2.2
single-link algorithm creates only two clusters: $\{x_1, x_2, x_3\}$ and $\{x_4, x_5\}$.



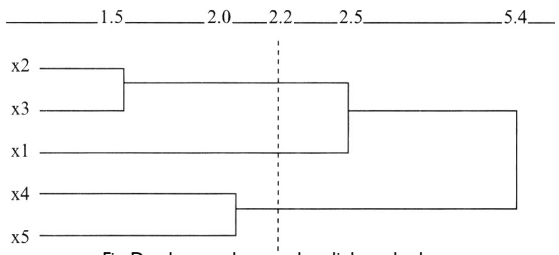


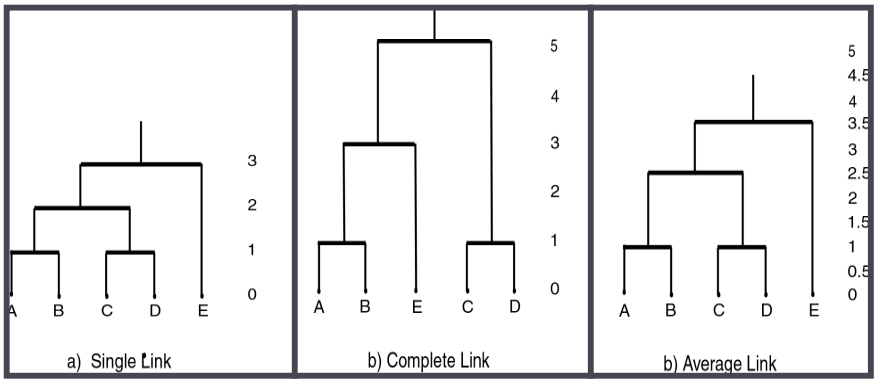
Fig: Dendrogram by complete-link method

threshold measure of similarity s = 2.2:
complete-link algorithm creates three clusters: $\{x_1\}$, $\{x_2, x_3\}$, and $\{x_4, x_5\}$.

Single Link

- View all items with links (distances) between them.
- Finds maximal connected components in this graph.
- Two clusters are merged if there is at least one edge which connects them.
- Uses threshold distances at each level.
- Could be agglomerative or divisive.

Single, Complete & Average Link Clustering



35

Hierarchical Clustering: Time and Space requirements

For a dataset consisting of N points:

- $O(N^2)$ space, since it requires storing the distance matrix.
  - N is the number of points.
- $O(N^3)$ time, in most of the cases.
- There are N steps and at each step the size, N^2, distance matrix must be updated and searched
- Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches, by using appropriate data structures

36

Squared Error

- The most commonly used partitional-clustering
- Minimizes squared error
- N samples has been partitioned into K clusters $\{C_1, C_2, \ldots, C_k\}$. Each C_k has n_k samples, so that $\sum n_k = N$, where $k = 1, \ldots, K$.
  - *centroid*: mean vector M_k of cluster C_k

$$M_k = \frac{1}{n_k}\sum_{i=1}^{n_k} x_{ki}$$

  - square-err: x_{ik} is the i^{th} sample belonging to cluster C_k

$$e_k^2 = \frac{1}{n_k}\sum_{i=1}^{n_k} \| x_{ki} - M_k \|_2^2$$

  - square-error for the entire clustering space containing K clusters

$$E_k^2 = \sum_{k=1}^{K} e_k^2$$

Squared Error Algorithm

Input:
  $D = \{t_1, t_2, ..., t_n\}$ // Set of elements
  k // Number of desired clusters.
Output:
  K // Set of clusters.
Squared Error Algorithm:
  assign each item t_i to a cluster;
  calculate center for each cluster;
  repeat
    assign each item t_i to the cluster which has the closest center ;
    calculate new center for each cluster;
    calculate squared error;
  until the difference between successive squared errors is below a threshold;

K-Means

- The simplest and most commonly used algorithm employing a square-error criterion.
- Initial set of clusters randomly chosen.
- Iteratively, items are moved among sets of clusters until the desired set is reached.
- High degree of similarity among elements in a cluster is obtained.
- Given a cluster $C_i=\{x_{i1}, x_{i2},\ldots, x_{im}\}$, the ***cluster mean*** is:

$$m_i = (1/m)(x_{i1} + \ldots + x_{im})$$

39

K-Means Algorithm

Input:
$D = \{t_1, t_2, ..., t_n\}$ // Set of elements
A // Adjacency matrix showing distance between elements.
k // Number of desired clusters.
Output:
K // Set of clusters.
K-Means Algorithm:
assign initial values for means $m_1, m_2, ..., m_k$;
repeat
assign each item t_i to the cluster which has the closest mean ;
calculate new mean for each cluster;
until convergence criteria is met;

40

K-Means Algorithm

1. Define *k*,
2. Take *k* objects as singleton clusters from a set of *n* objects randomly,
3. Assign each of the remaining *n* - *k* objects to the cluster with the nearest centroid,
4. Recompute the centroid of the gaining cluster after each assignment,
5. **REPEAT**
   - Assign each object to the cluster with the nearest centroid,
   - If object changes clusters, Compute the centroid of the new and the old cluster,

   UNTIL no more changes are recorded in sequence
6. Stop.

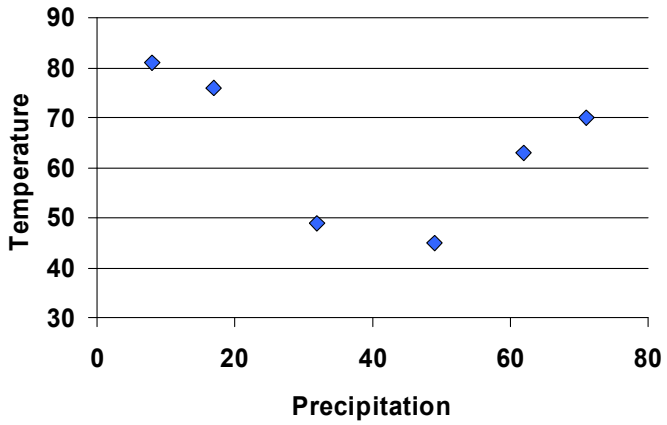
41

Clustering – k-Means

Precipitation	Temperature
8	81
71	70
62	63
49	45
17	76
32	49

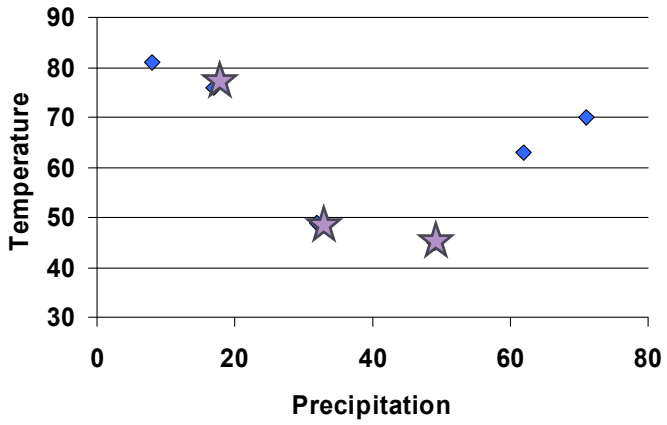
42

Clustering – k-Means



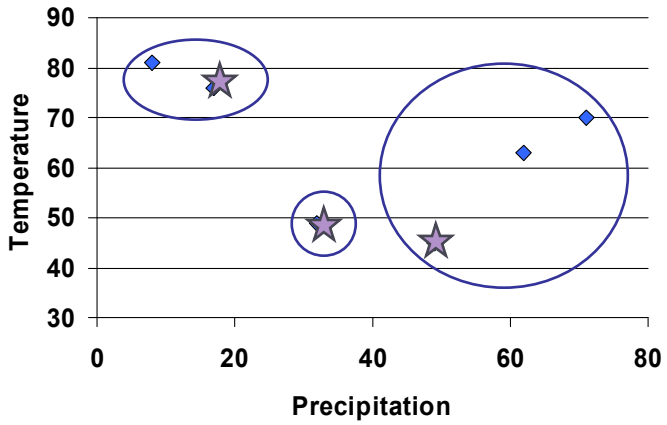
43

Clustering – k-Means



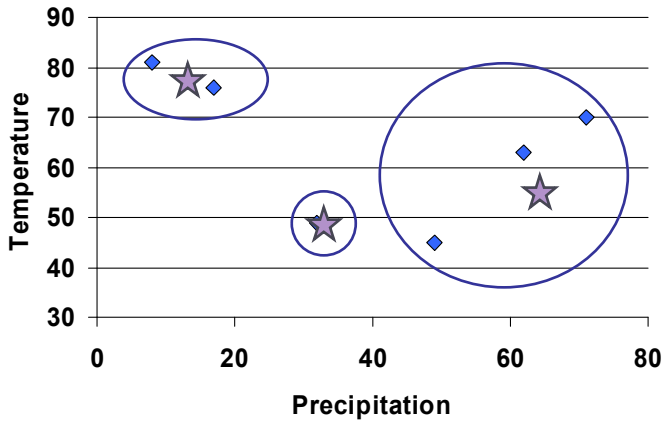
44

Clustering – k-Means



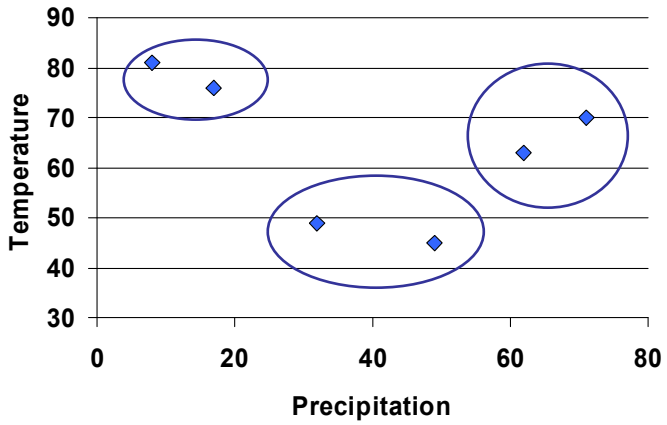
45

Clustering – k-Means



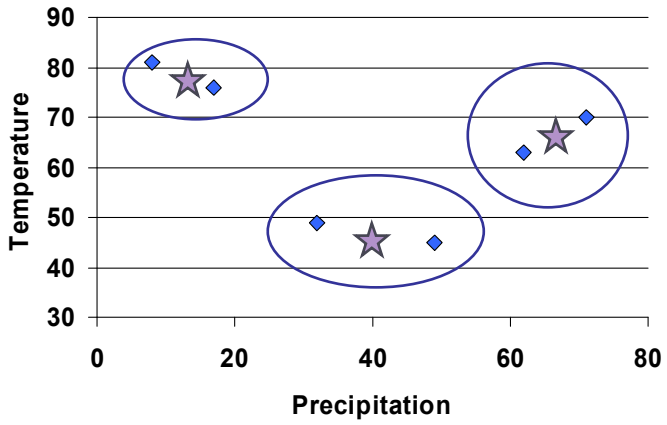
46

Clustering – k-Means



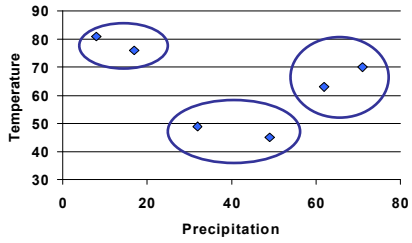
47

Clustering – k-Means



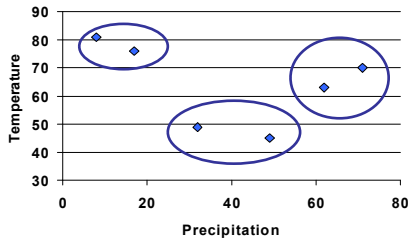
48

Clustering – k-Means



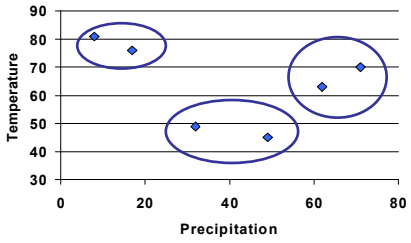
Cluster	Temperature	Precipitation
C1	70 – 85	0 – 25
C2	35 – 60	25 – 55
C3	50 – 80	50 – 80

Clustering – k-Means



Cluster	Temperature	Precipitation
C1	70 – 85	0 – 25
C2	35 – 60	25 – 55
C3	50 – 80	50 – 80

Clustering – k-Means



Cluster	Temperature	Precipitation	Ecosystem
C1	70 - 85	0-25	Desert
C2	35 - 60	25 - 55	Prairie
C3	50 – 80	50 – 80	Forest

51