

CSE 5243 INTRO. TO DATA MINING

Classification & Clustering

Huan Sun, CSE@The Ohio State University

09/21/2017

Classification: Advanced Methods

- Lazy Learners and K-Nearest Neighbors
- Neural Networks 

- Support Vector Machines

In Chapter 9,
recommended reading

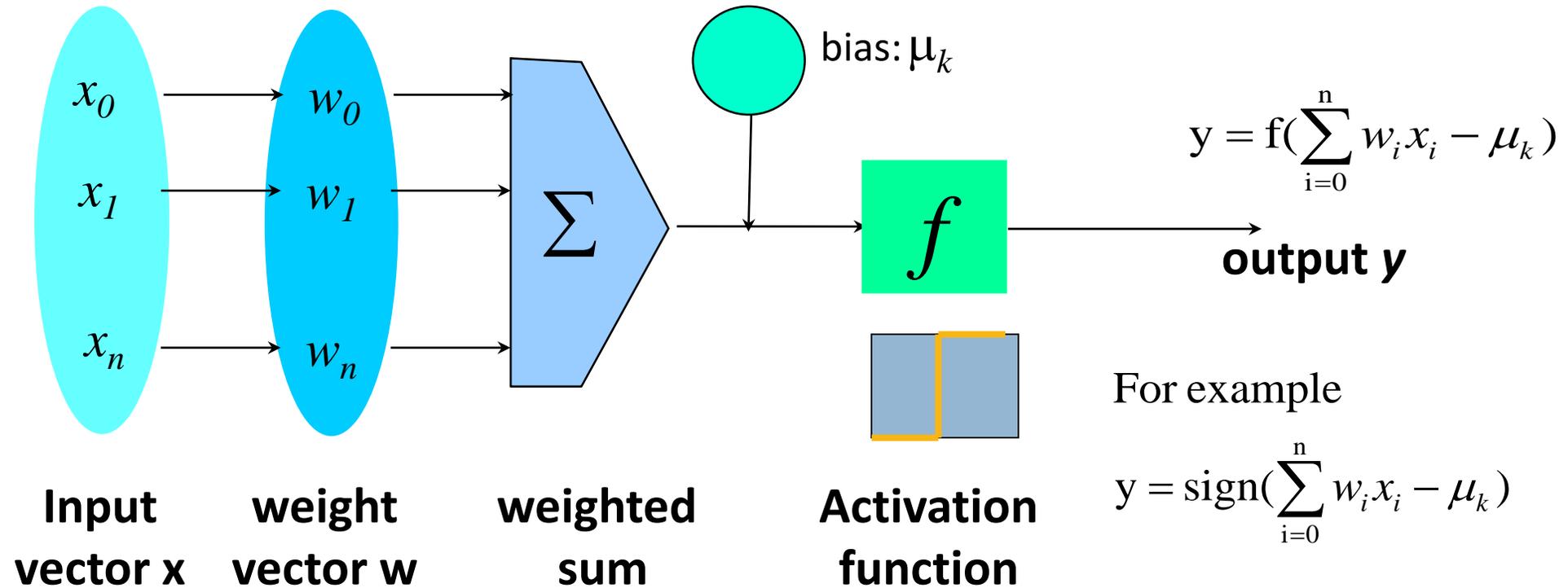
- Additional Topics: Semi-Supervised Methods, Active Learning, etc.

- Summary

Neural Network for Classification

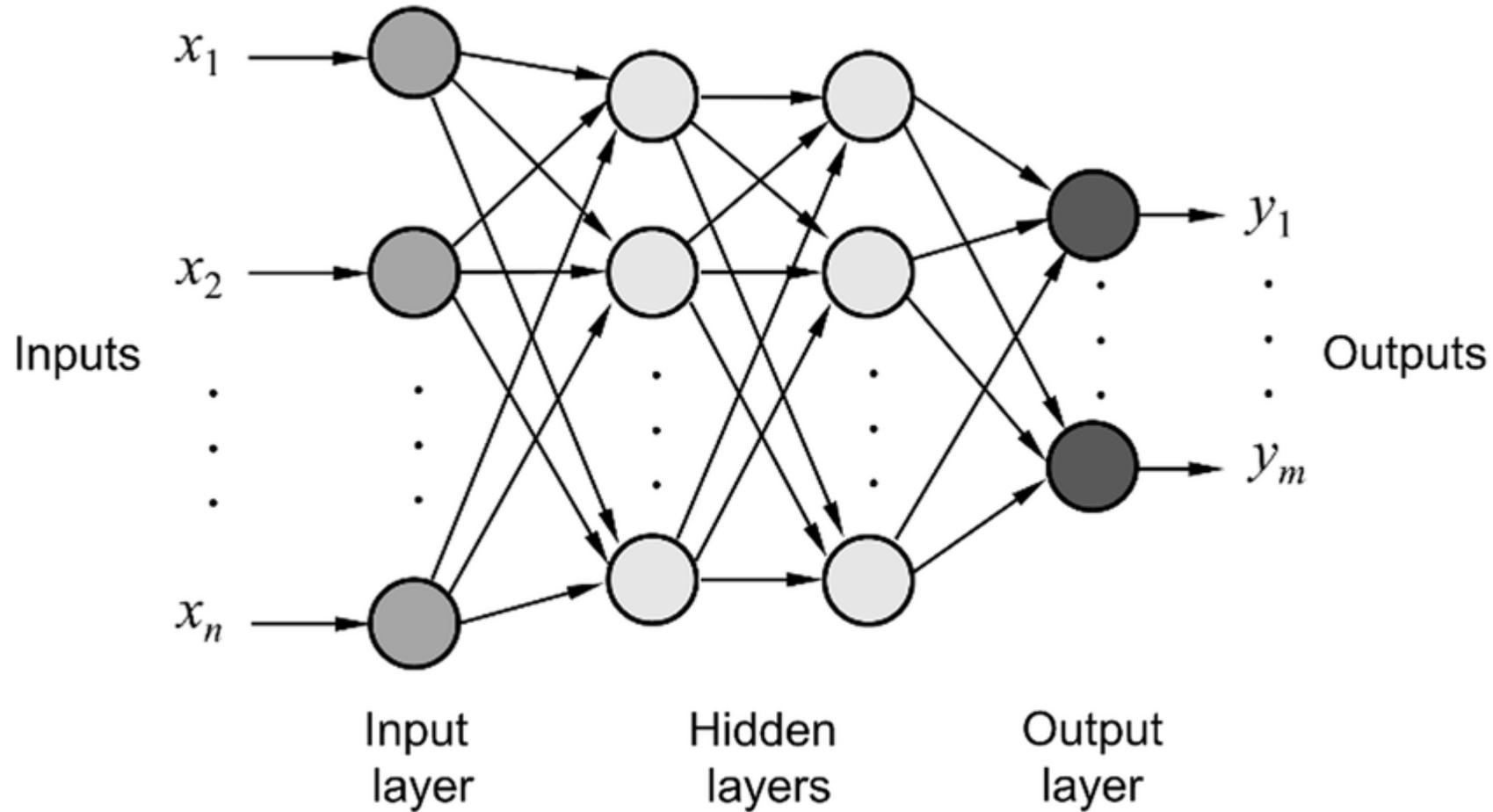
- Started by psychologists and neurobiologists to develop and test computational analogues of neurons
- A neural network: A set of connected input/output units where each connection has a **weight** associated with it
 - ▣ During the learning phase, the **network learns by adjusting the weights** so as to be able to predict the correct class label of the input tuples
- Also referred to as **connectionist learning** due to the connections between units
- Backpropagation: A **neural network** learning algorithm

Neuron: A Hidden/Output Layer Unit



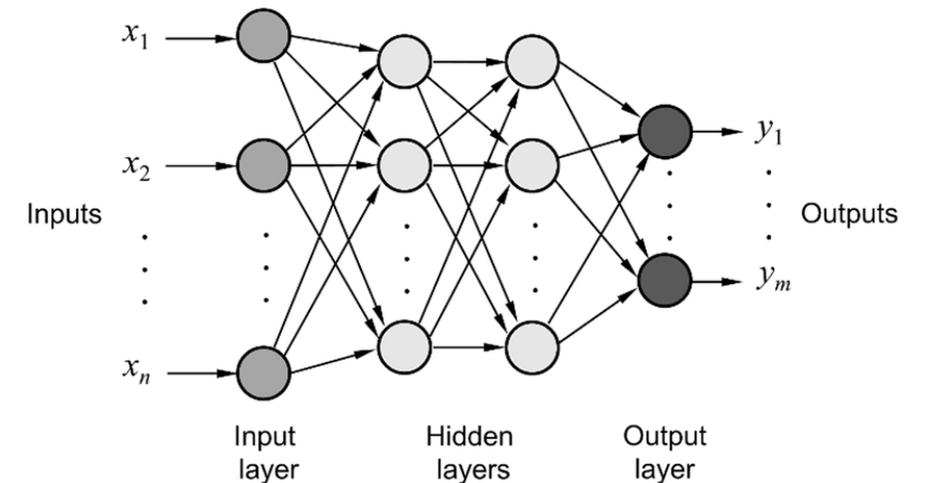
- An n -dimensional input vector \mathbf{x} is mapped into variable y by means of the scalar product and a nonlinear function mapping
- The inputs to unit are outputs from the previous layer. They are multiplied by their corresponding weights to form a weighted sum, which is added to the bias associated with unit. Then a nonlinear activation function is applied to it.

A Multi-Layer Feed-Forward Neural Network



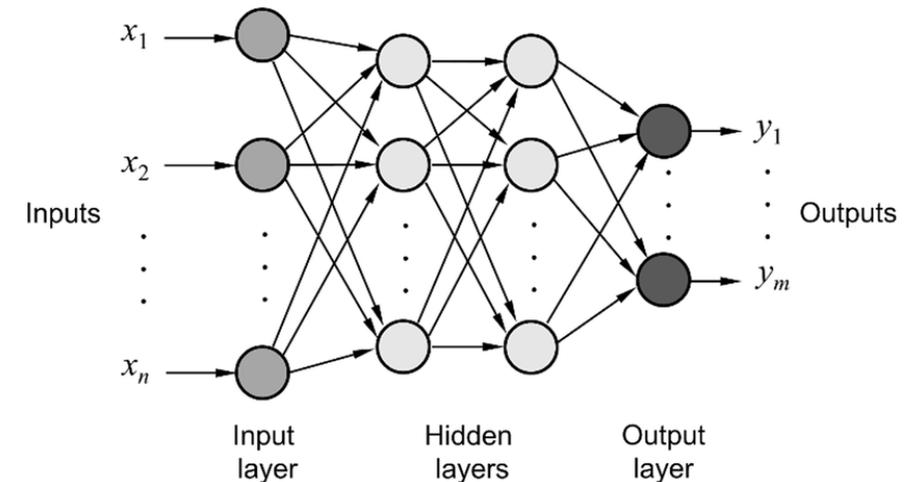
How a Multi-Layer Neural Network Works

- The **inputs** to the network correspond to the attributes measured for each training tuple
- Inputs are fed simultaneously into the units making up the **input layer**
- They are then weighted and fed simultaneously to a **hidden layer**



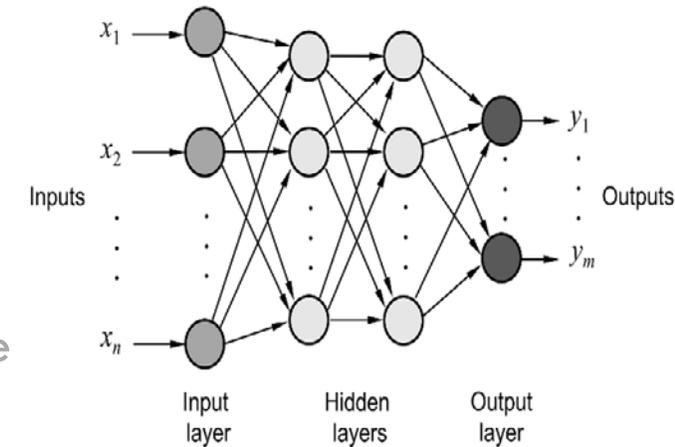
How a Multi-Layer Neural Network Works

- The **inputs** to the network correspond to the attributes measured for each training tuple
- Inputs are fed simultaneously into the units making up the **input layer**
- They are then weighted and fed simultaneously to a **hidden layer**
- The number of hidden layers is arbitrary
- The weighted outputs of the last hidden layer are input to units making up the **output layer**, which emits the network's prediction



How a Multi-Layer Neural Network Works

- The **inputs** to the network correspond to the attributes measured for each training tuple
- Inputs are fed simultaneously into the units making up the **input layer**
- They are then weighted and fed simultaneously to a **hidden layer**
- The number of hidden layers is arbitrary
- The weighted outputs of the last hidden layer are input to units making up the **output layer**, which emits the network's prediction
- The network is **feed-forward**: None of the weights cycles back to an input unit or to an output unit of a previous layer
- From a statistical point of view, networks perform **nonlinear regression**
 - ▣ Given enough hidden units and enough training samples, they can closely approximate any function

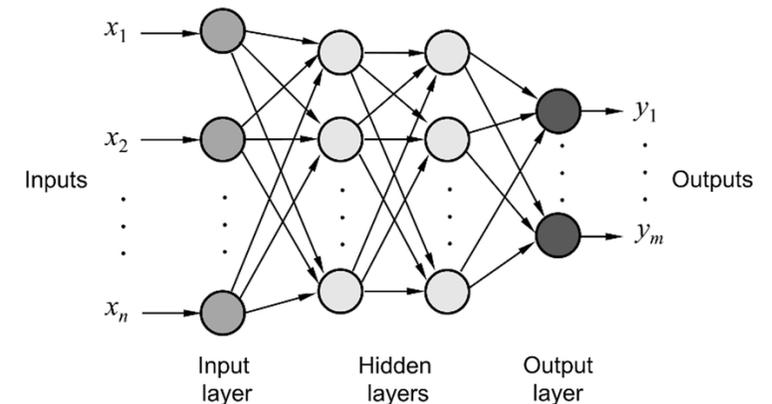


Defining a Network Topology

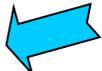
- Decide the **network topology**
 - Specify # of units in the *input layer*, # of *hidden layers* (if > 1), # of units in each *hidden layer*, and # of units in the *output layer*
- Normalize the input values for each attribute measured in the training
- **Output**, if for classification and more than two classes, one output unit per class is used
- Once a network has been trained and its accuracy is **unacceptable**, repeat the training process with a *different network topology* or a *different set of initial weights*

Back Propagation

- **Back propagation:** Reset weights on the "front" neural units and this is sometimes done in combination with training where the correct result is known
- Iteratively process a set of training tuples & compare the network's prediction with the actual known target value
- For each training tuple, the weights are modified to **minimize the mean squared error between the network's prediction and the actual target value**
- Modifications are made in the **"backwards"** direction: from the output layer, through each hidden layer down to the first hidden layer, hence **"backpropagation"**
- **Steps**
 - Initialize weights to small random numbers, associated with biases
 - Propagate the inputs forward (by applying activation function)
 - Backpropagate the error (by updating weights and biases)
 - Terminating condition (when error is very small, etc.)

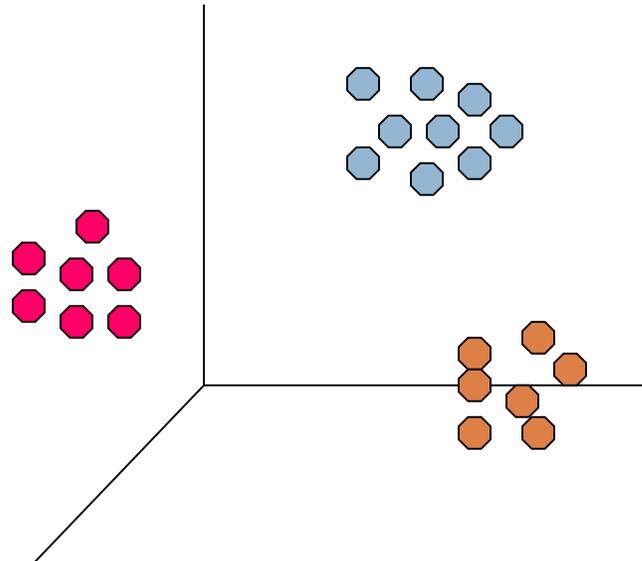


Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: An Introduction 
- Partitioning Methods
- Hierarchical Methods
- Density- and Grid-Based Methods
- Evaluation of Clustering
- Summary

What Is Cluster Analysis?

- **What is a cluster?**
 - A cluster is a collection of data objects which are
 - Similar (or related) to one another within the same group (i.e., cluster)
 - Dissimilar (or unrelated) to the objects in other groups (i.e., clusters)

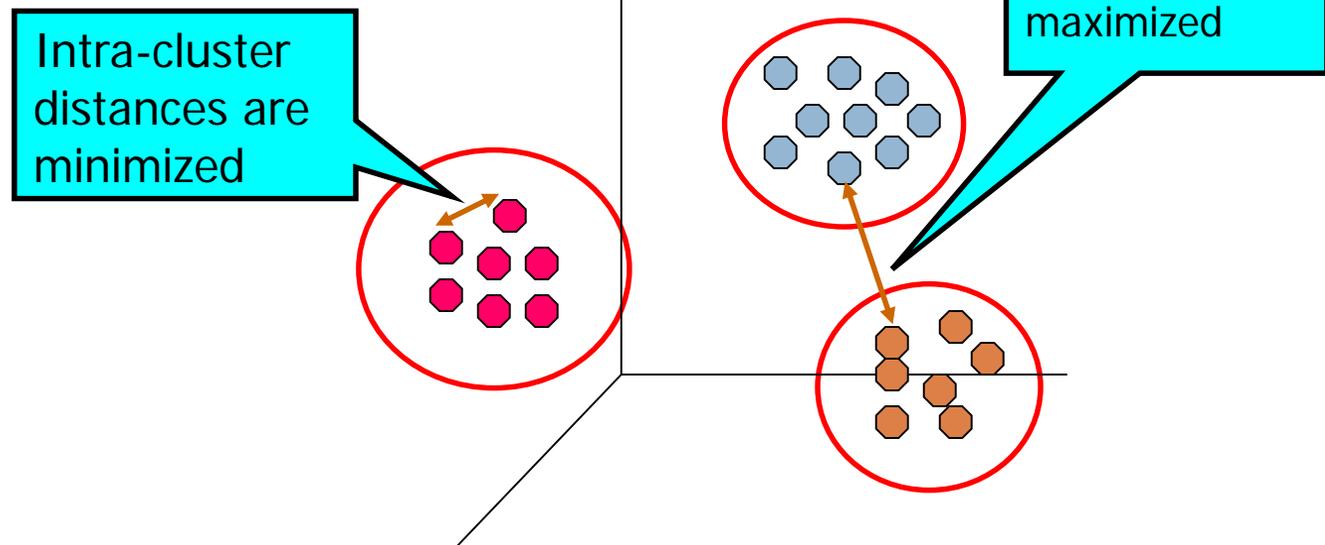


What Is Cluster Analysis?

- **What is a cluster?**
 - ▣ A cluster is a collection of data objects which are
 - Similar (or related) to one another within the same group (i.e., cluster)
 - Dissimilar (or unrelated) to the objects in other groups (i.e., clusters)
- **Cluster analysis (or *clustering, data segmentation, ...*)**
 - ▣ Given a set of data points, partition them into a set of groups (i.e., clusters), **such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups**

What Is Cluster Analysis?

- **What is a cluster?**
 - ▣ A cluster is a collection of data objects which are
 - Similar (or related) to one another within the same group (i.e., cluster)
 - Dissimilar (or unrelated) to the objects in other groups (i.e., clusters)
- **Cluster analysis (or *clustering, data segmentation, ...*)**
 - ▣ Given a set of data points, partition them into a set of groups (i.e., clusters), **such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups**



What Is Cluster Analysis?

- **What is a cluster?**
 - A cluster is a collection of data objects which are
 - Similar (or related) to one another within the same group (i.e., cluster)
 - Dissimilar (or unrelated) to the objects in other groups (i.e., clusters)
- **Cluster analysis (or *clustering*, *data segmentation*, ...)**
 - Given a set of data points, partition them into a set of groups (i.e., clusters), **such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups**
- Cluster analysis is **unsupervised learning** (i.e., no predefined classes)
 - This contrasts with *classification* (i.e., *supervised learning*)
- Typical ways to use/apply cluster analysis
 - As a stand-alone tool to get insight into data distribution, or
 - As a preprocessing (or intermediate) step for other algorithms

What Is Good Clustering?

- A good clustering method will produce high quality clusters, which should have
 - ▣ **High intra-class similarity:** *Cohesive* within clusters
 - ▣ **Low inter-class similarity:** *Distinctive* between clusters

What Is Good Clustering?

- A good clustering method will produce high quality clusters, which should have
 - ▣ **High intra-class similarity:** **Cohesive** within clusters
 - ▣ **Low inter-class similarity:** **Distinctive** between clusters
- **Quality function**
 - ▣ There is usually a separate “quality” function that measures the “goodness” of a cluster
 - ▣ It is hard to define “similar enough” or “good enough”
 - The answer is typically highly subjective
- There exist many similarity measures and/or functions for different applications
- **Similarity measure is critical for cluster analysis**

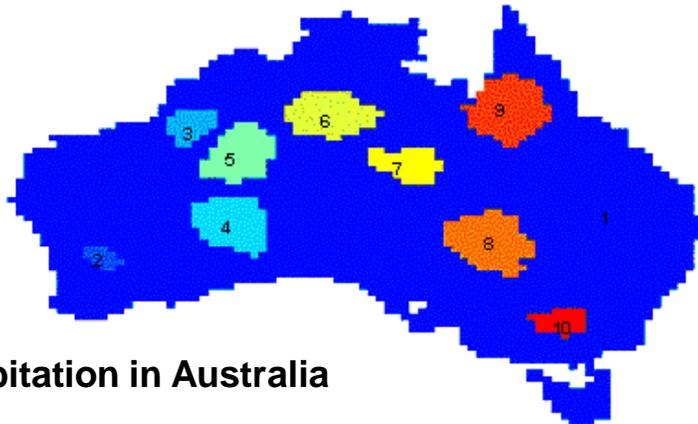
Cluster Analysis: Applications

□ Understanding

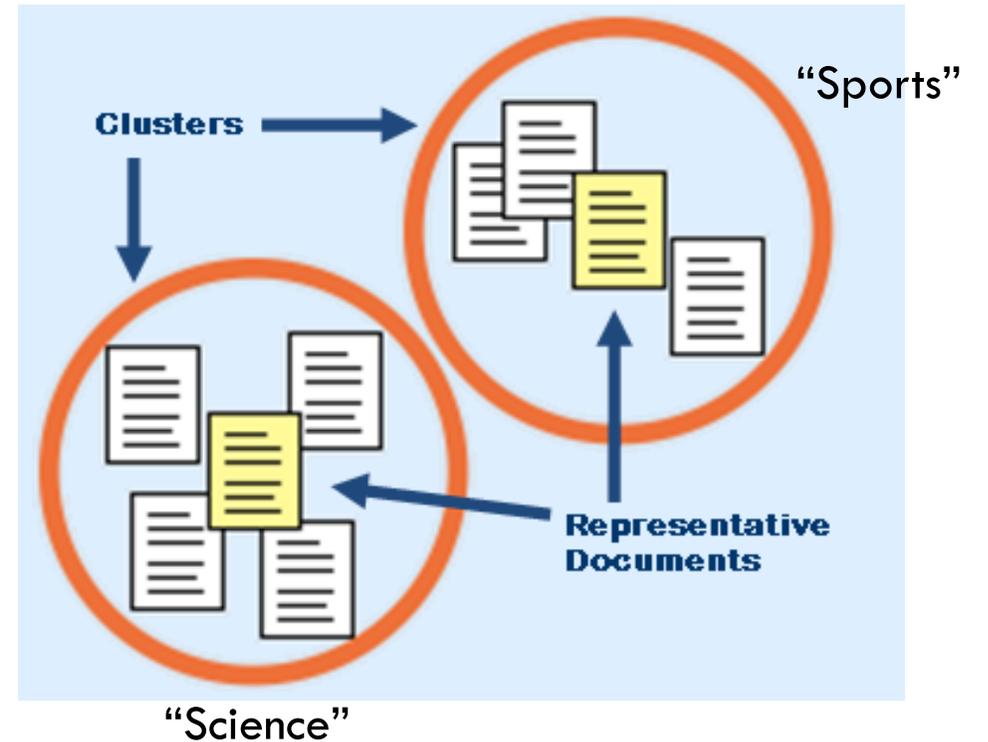
- Group related documents for browsing, group genes and proteins that have similar functionality, or group stocks with similar price fluctuations

□ Summarization

- Reduce the size of large data sets



Clustering precipitation in Australia



What is not Cluster Analysis?

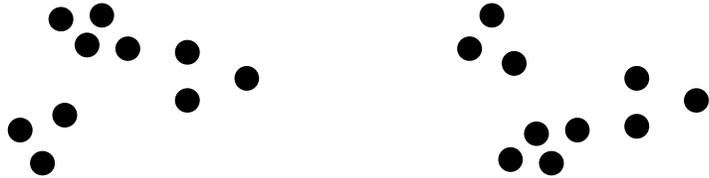
- Supervised classification
 - ▣ Have class label information
- Simple segmentation
 - ▣ Dividing students into different registration groups alphabetically, by last name
- Results of a query
 - ▣ Groupings are a result of an external specification
- Graph partitioning
 - ▣ Some mutual relevance and synergy, but areas are not identical

Notion of a Cluster can be Ambiguous

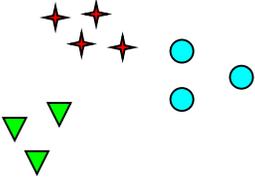


How many clusters?

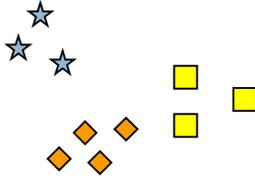
Notion of a Cluster can be Ambiguous



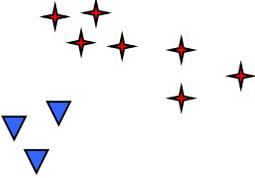
How many clusters?



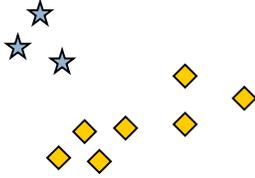
Six Clusters



Two Clusters



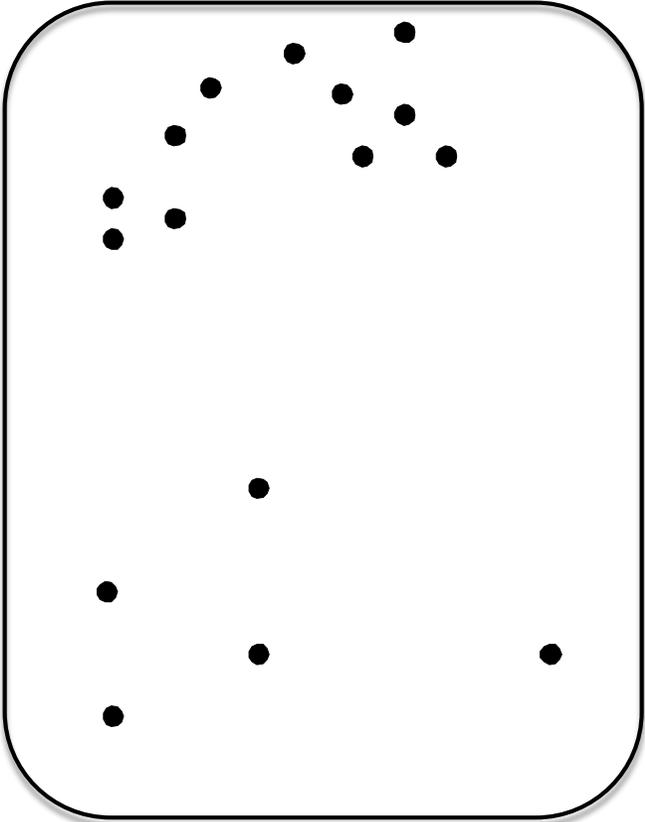
Four Clusters



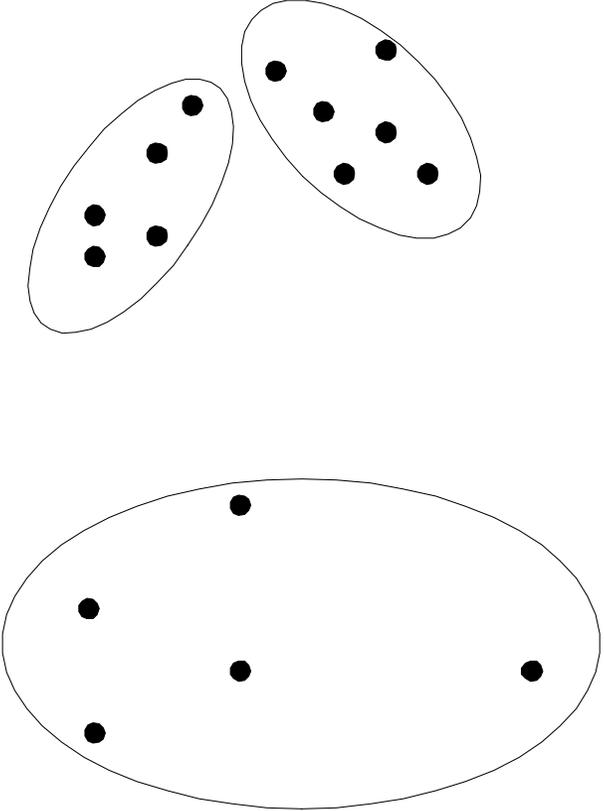
Types of Clusterings

- A **clustering** is a set of clusters
- Important distinction between **partitional** and **hierarchical** sets of clusters
- Partitional Clustering
 - A division of data objects into **non-overlapping** subsets (clusters) such that each data object is in exactly one subset

Partitional Clustering



Original Points

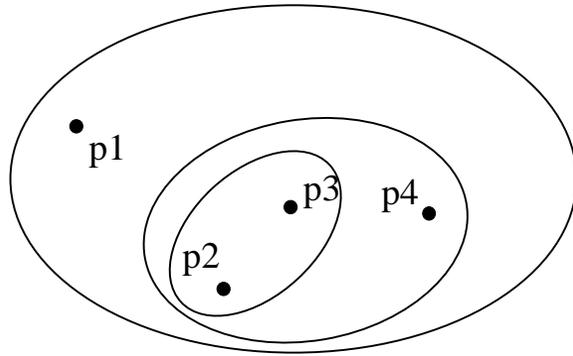


A Partitional Clustering

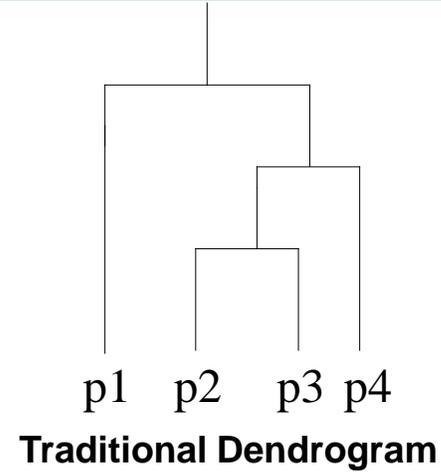
Types of Clusterings

- A **clustering** is a set of clusters
- Important distinction between **hierarchical** and **partitional** sets of clusters
- **Partitional Clustering**
 - ▣ A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- **Hierarchical clustering**
 - ▣ A set of nested clusters organized as a hierarchical tree

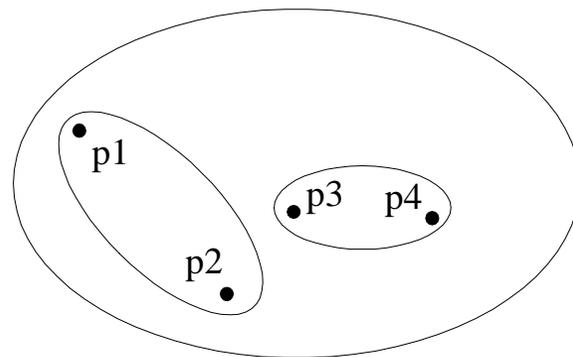
Hierarchical Clustering



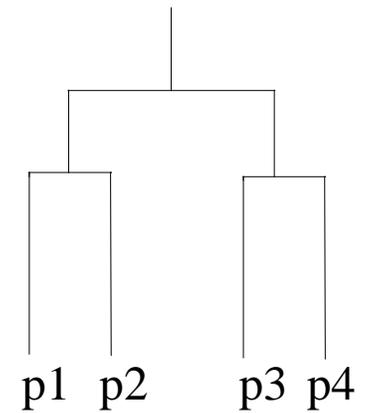
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering

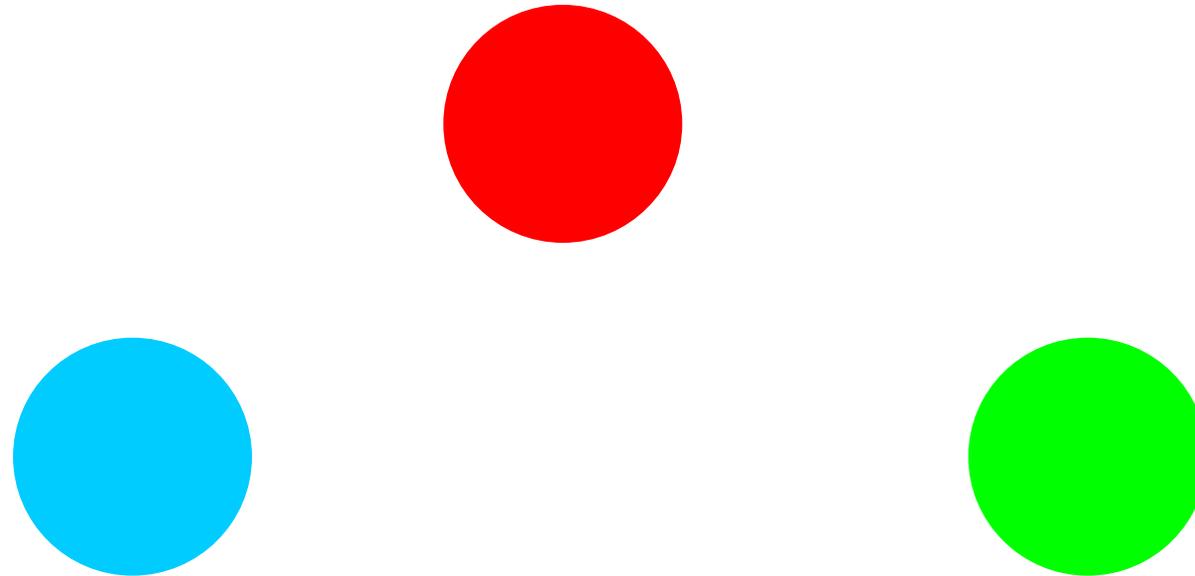


Non-traditional Dendrogram

Types of Clusters: Well-Separated

□ Well-Separated Clusters:

- A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



3 well-separated clusters

Types of Clusters: Center-Based

□ Center-based

- A cluster is a set of objects such that an object in a cluster is closer (more similar) to the “center” of a cluster, than to the center of any other cluster
- The center of a cluster is often a **centroid**, the average of all the points in the cluster, or a **medoid**, the most “representative” point of a cluster

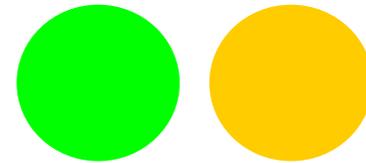
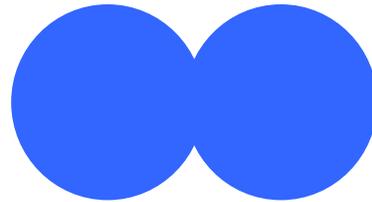
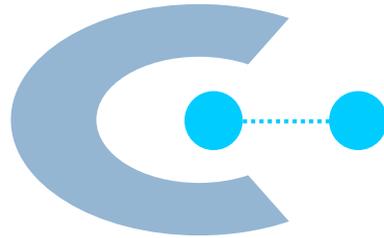
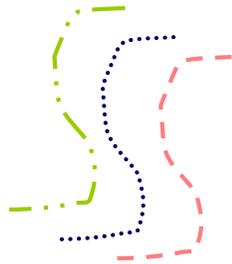


4 center-based clusters

Types of Clusters: Contiguity-Based

□ Contiguous Cluster (Nearest neighbor or Transitive)

- A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.

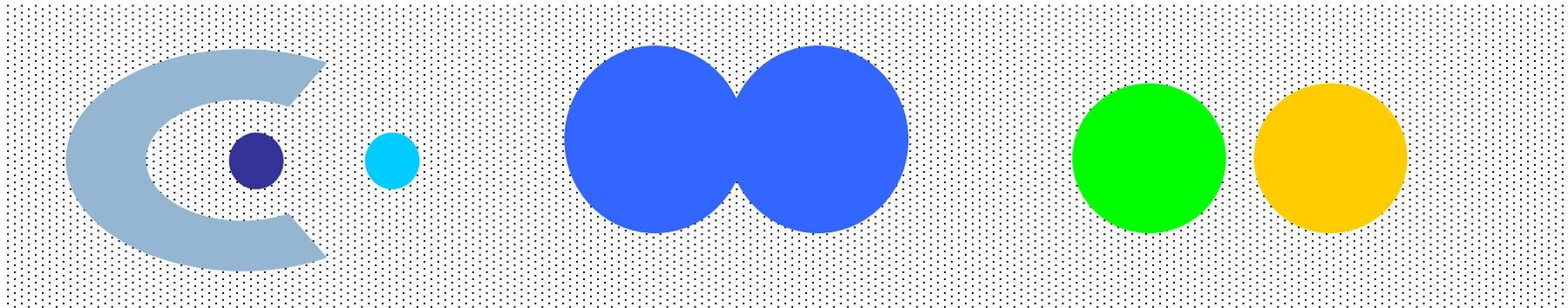


8 contiguous clusters

Types of Clusters: Density-Based

□ Density-based

- A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
- Used when the clusters are irregular or intertwined, and when noise and outliers are present.



6 density-based clusters

Characteristics of the Input Data Are Important

- Type of proximity or density measure
 - ▣ This is a derived measure, but central to clustering
- Sparseness
 - ▣ Dictates type of similarity
 - ▣ Adds to efficiency
- Attribute type
 - ▣ Dictates type of similarity
- Type of Data
 - ▣ Dictates type of similarity
 - ▣ Other characteristics, e.g., autocorrelation
- Dimensionality
- Noise and Outliers
- Type of Distribution

Clustering Algorithms

- K-means and its variants
- Hierarchical clustering
- Density-based clustering

K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a **centroid** (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K , must be specified
- The basic algorithm is very simple

-
- 1: Select K points as the initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning all points to the closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** The centroids don't change
-

K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a **centroid** (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K , must be specified
- The basic algorithm is very simple

Often chosen
randomly

-
- 1: Select K points as the initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning all points to the closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** The centroids don't change
-

K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a **centroid** (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K , must be specified
- The basic algorithm is very simple

1: Select K points as the initial centroids.

2: **repeat**

3: Form K clusters by assigning all points to the closest centroid.

4: Recompute the centroid of each cluster.

5: **until** The centroids don't change

Often chosen
randomly

Measured by Euclidean
distance, cosine similarity,
etc.

K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a **centroid** (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K , must be specified
- The basic algorithm is very simple

Often chosen randomly

1: Select K points as the initial centroids.

2: **repeat**

3: Form K clusters by assigning all points to the closest centroid.

4: Recompute the centroid of each cluster.

Typically the mean of the points in the cluster

5: **until** The centroids don't change

K-means Clustering – Details

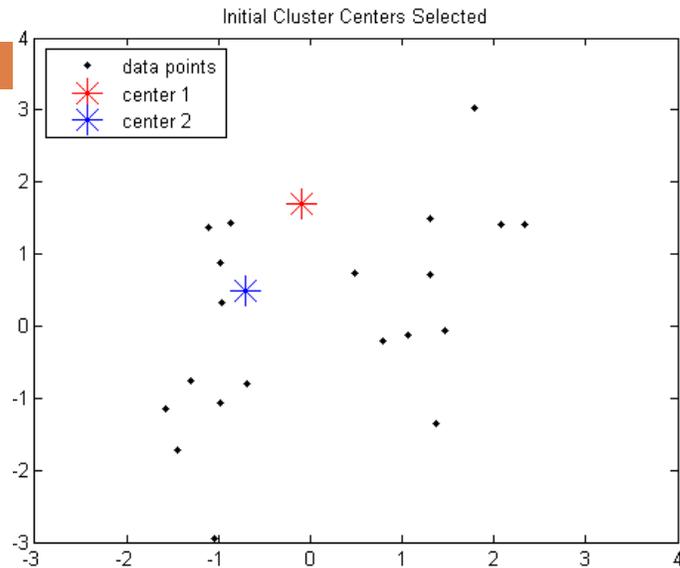
- Initial centroids are often chosen randomly.
 - ▣ Clusters produced vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- ‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc.

- K-means will converge for common similarity measures mentioned above.

- Most of the convergence happens in the first few iterations.
 - ▣ Often the stopping condition is changed to ‘Until relatively few points change clusters’

- Complexity is $O(n * K * I * d)$
 - ▣ n = number of points, K = number of clusters,
 I = number of iterations, d = number of attributes

Example: *K*-Means Clustering



The original data points &
randomly select $K = 2$ centroids

Execution of the *K*-Means Clustering Algorithm

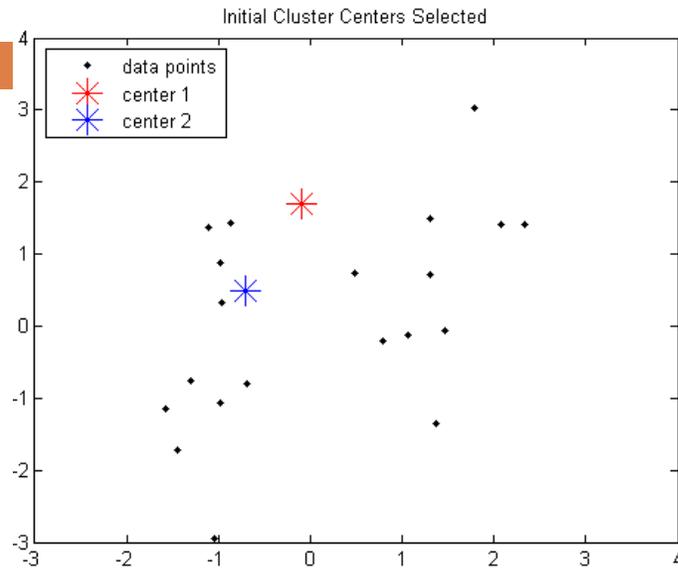
Select K points as initial centroids

Repeat

- Form K clusters by assigning each point to its closest centroid
- Re-compute the centroids (i.e., *mean point*) of each cluster

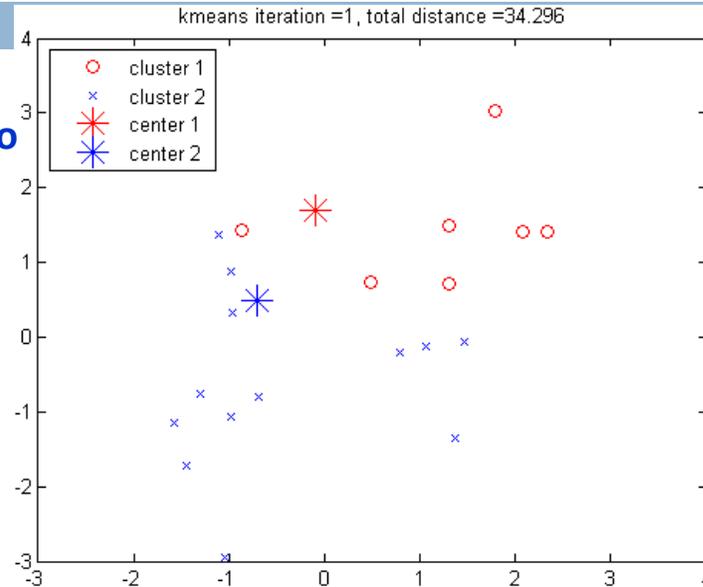
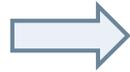
Until convergence criterion is satisfied

Example: *K*-Means Clustering



The original data points & randomly select $K = 2$ centroids

Assign
points to
clusters



Execution of the *K*-Means Clustering Algorithm

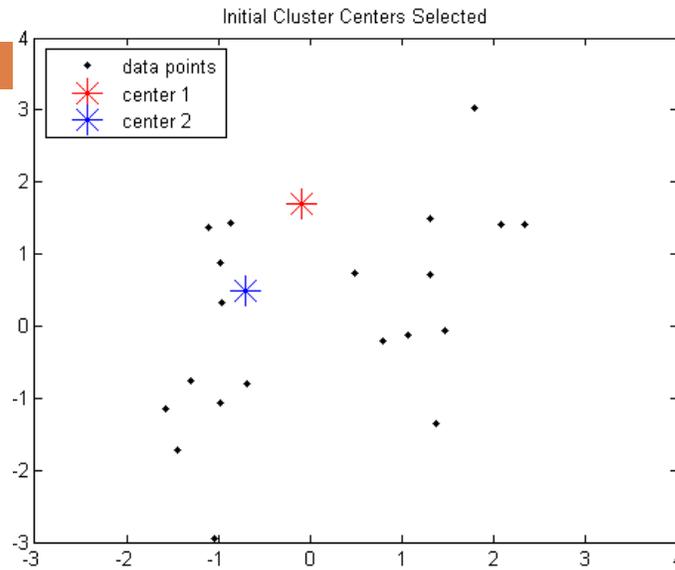
Select K points as initial centroids

Repeat

- Form K clusters by assigning each point to its closest centroid
- Re-compute the centroids (i.e., *mean point*) of each cluster

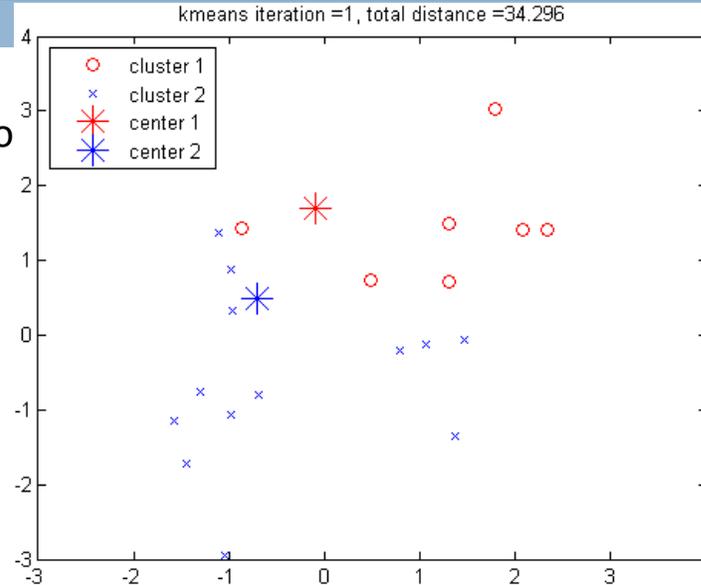
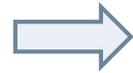
Until convergence criterion is satisfied

Example: *K*-Means Clustering

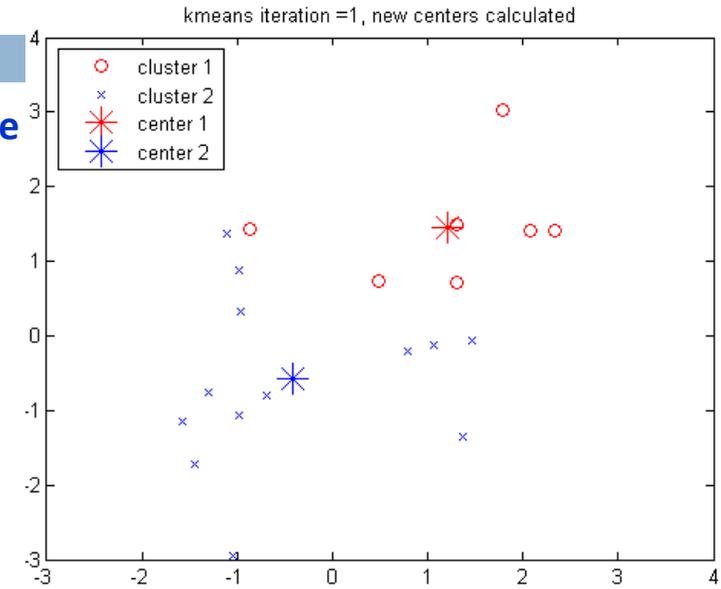


The original data points & randomly select $K = 2$ centroids

Assign points to clusters



Recompute cluster centers



Execution of the *K*-Means Clustering Algorithm

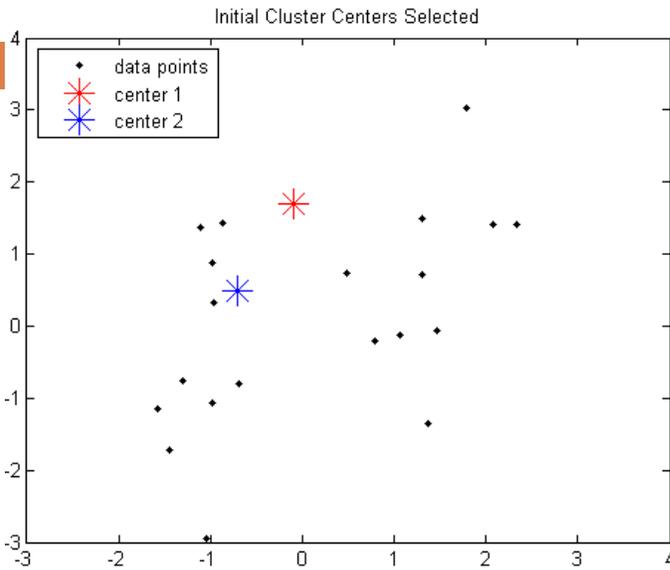
Select K points as initial centroids

Repeat

- Form K clusters by assigning each point to its closest centroid
- Re-compute the centroids (i.e., *mean point*) of each cluster

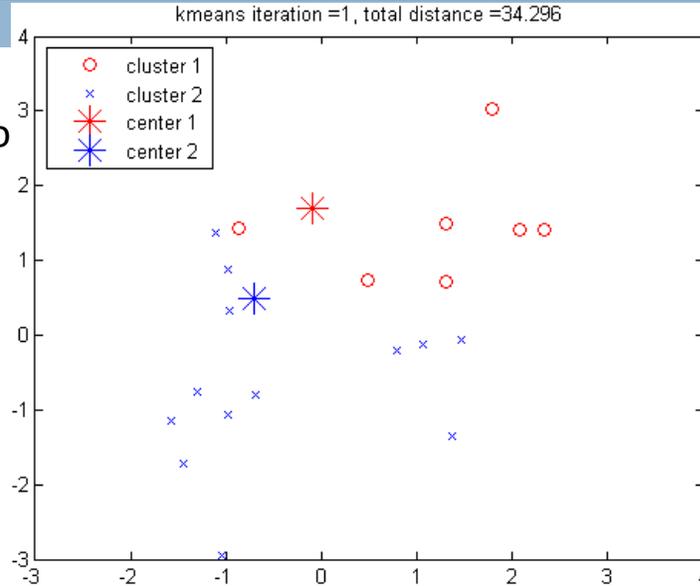
Until convergence criterion is satisfied

Example: *K*-Means Clustering

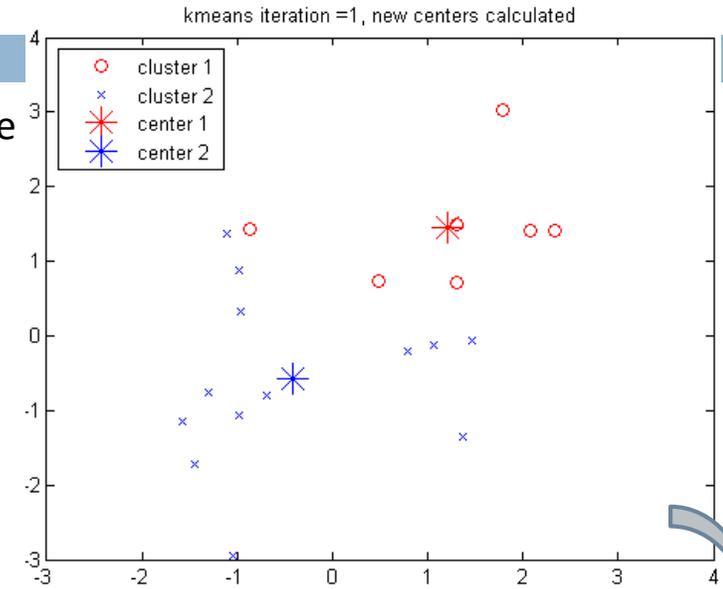


The original data points & randomly select $K = 2$ centroids

Assign points to clusters



Recompute cluster centers



Redo point assignment



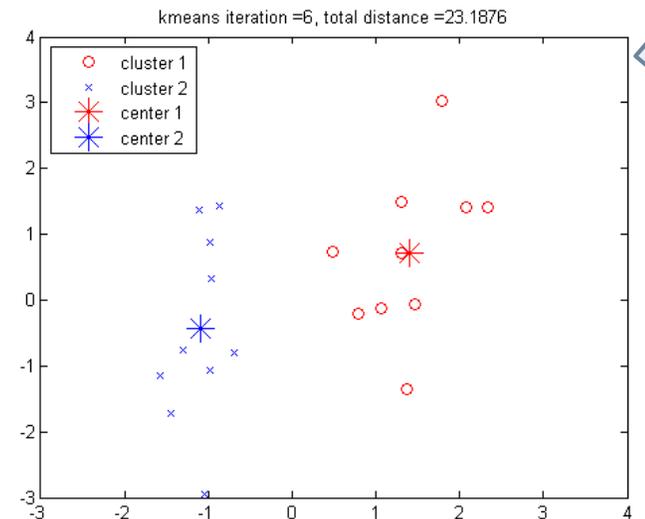
Execution of the *K*-Means Clustering Algorithm

Select K points as initial centroids

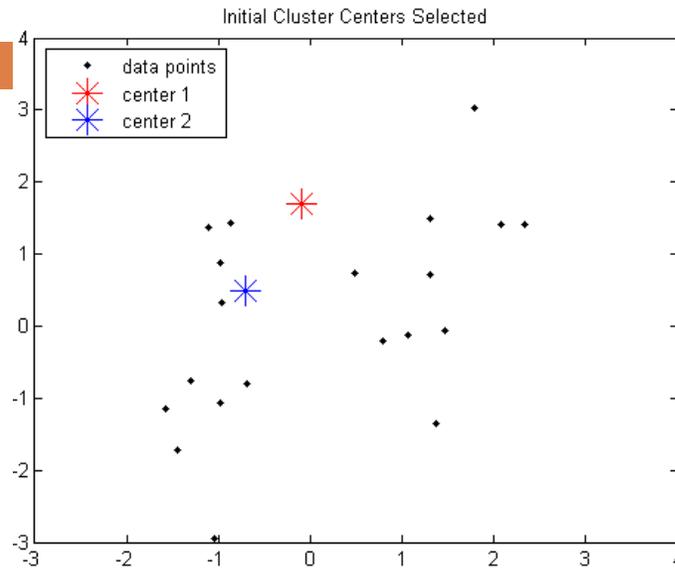
Repeat

- Form K clusters by assigning each point to its closest centroid
- Re-compute the centroids (i.e., *mean point*) of each cluster

Until convergence criterion is satisfied

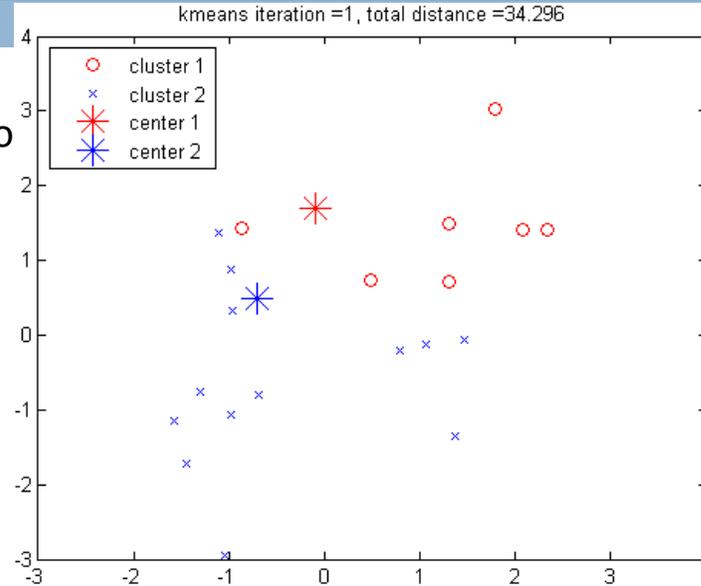


Example: *K*-Means Clustering



The original data points & randomly select $K = 2$ centroids

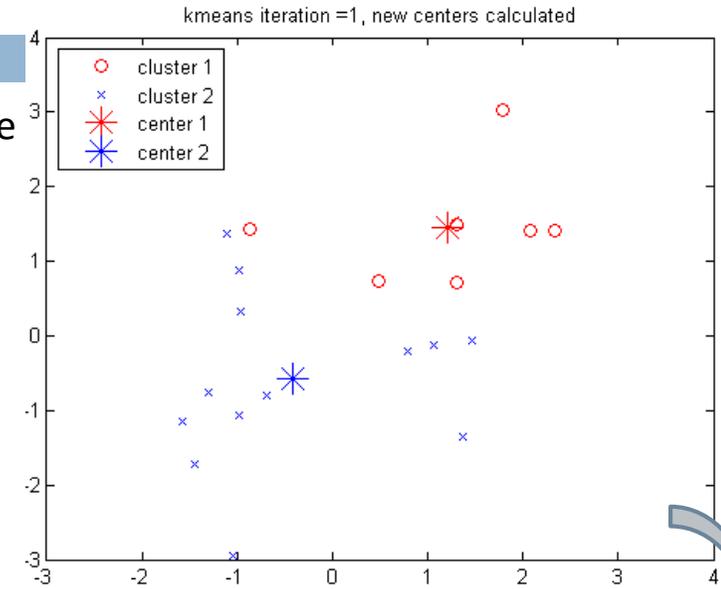
Assign points to clusters



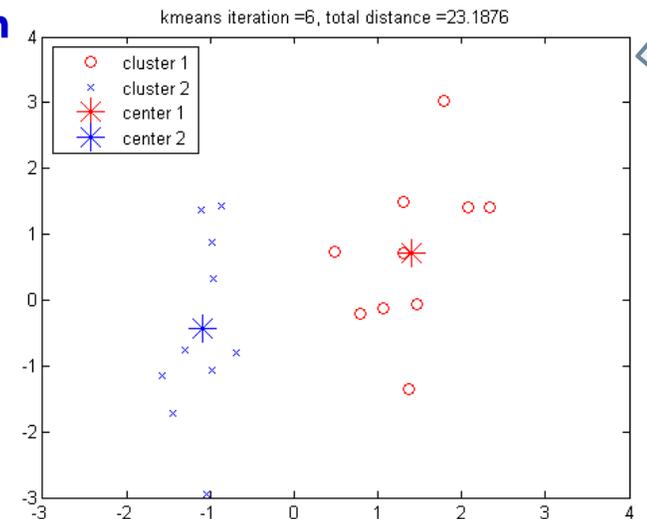
Recompute cluster centers



Next iteration



Redo point assignment



Execution of the *K*-Means Clustering Algorithm

Select K points as initial centroids

Repeat

- Form K clusters by assigning each point to its closest centroid
- Re-compute the centroids (i.e., *mean point*) of each cluster

Until convergence criterion is satisfied