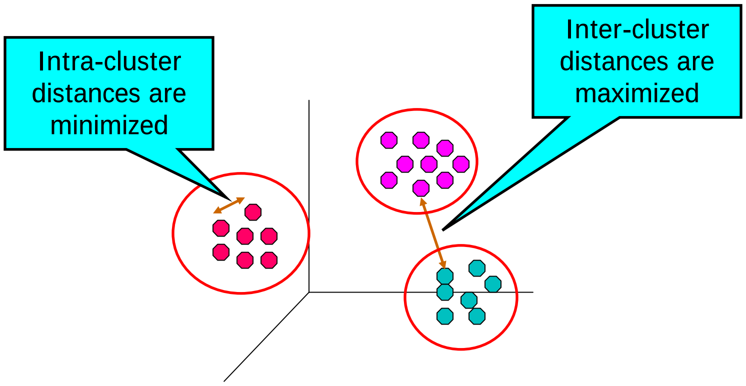
Cluster Analysis:
Basic Concepts and Algorithms

- **What does it mean clustering?**
  - Applications
- **Types of clustering**
- **K-means**
  - Intuition
  - Algorithm
  - Choosing initial centroids
  - Bisecting K-means
  - Post-processing
- **Strengths and weaknesses**
- **What’s next?**

Sections 2.4, 8.1, 8.2 of course book

What is Cluster Analysis?

- Finding groups of objects 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



Intra-cluster distances are minimized

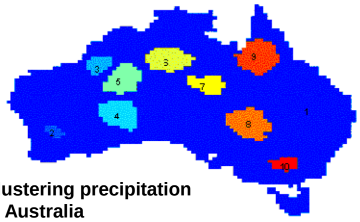
Inter-cluster distances are maximized

Applications of Cluster Analysis

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

	Discovered Clusters	*Industry Group*
1	Applied-Matl-DOWN,Bay-Network-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down, Natl-Semiconduct-DOWN,Oracl-DOWN,SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP, Schlumberger-UP	Oil-UP

- **Summarization**
  - Reduce the size of large data sets

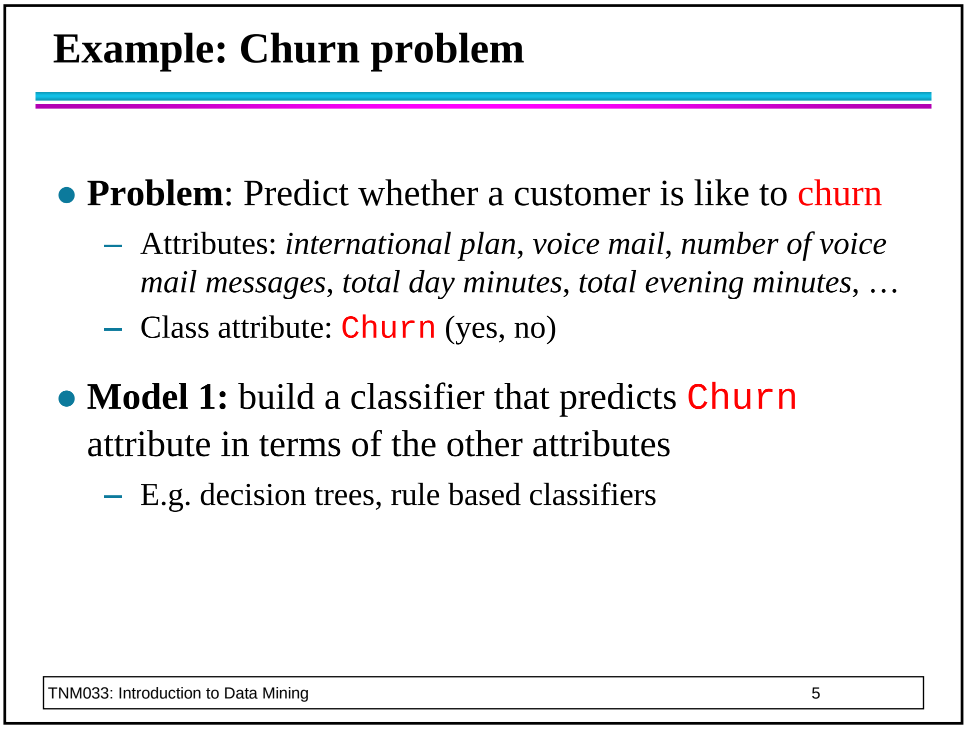


Clustering precipitation in Australia

What is not Cluster Analysis?

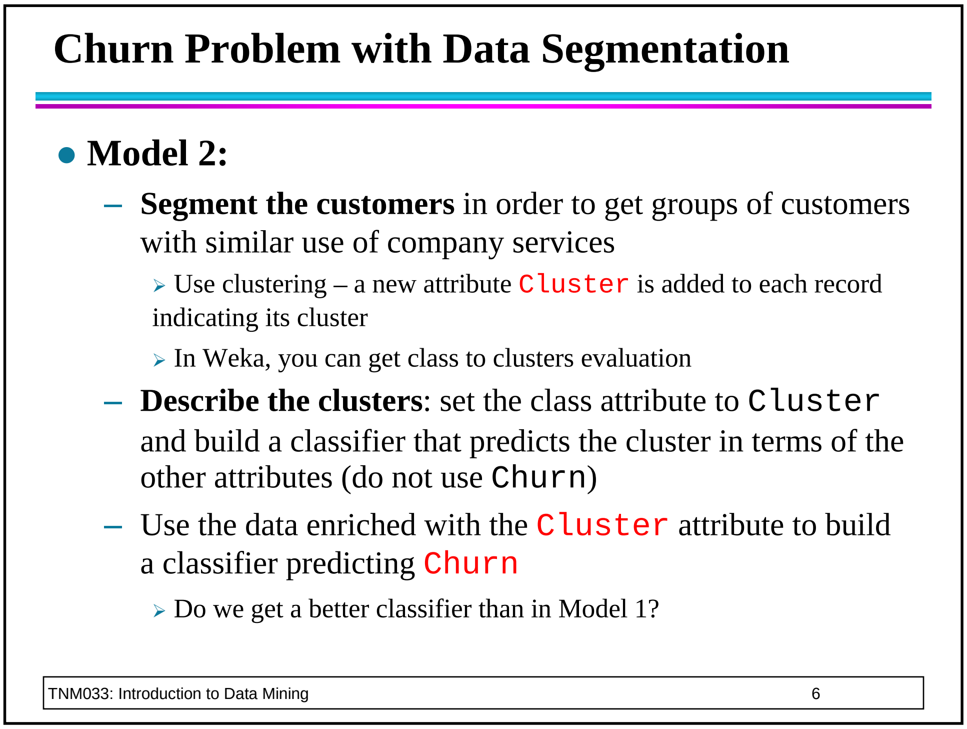
- Supervised classification
  - Have class label information
- Simple segmentation
  - Dividing students into different registration groups alphabetically, by last name
- Results of a query
  - Groupings are a result of an external specification

```
SELECT dept, division, AVG(salary)
FROM Table
GROUP BY dept, division
```



Example: Churn problem

- **Problem**: Predict whether a customer is like to churn
  - Attributes: *international plan, voice mail, number of voice mail messages, total day minutes, total evening minutes, …*
  - Class attribute: `Churn` (yes, no)
- **Model 1:** build a classifier that predicts `Churn` attribute in terms of the other attributes
  - E.g. decision trees, rule based classifiers



Churn Problem with Data Segmentation

- **Model 2:**
  - **Segment the customers** in order to get groups of customers with similar use of company services
    - Use clustering – a new attribute `Cluster` is added to each record indicating its cluster
    - In Weka, you can get class to clusters evaluation
  - **Describe the clusters**: set the class attribute to `Cluster` and build a classifier that predicts the cluster in terms of the other attributes (do not use `Churn`)
  - Use the data enriched with the `Cluster` attribute to build a classifier predicting `Churn`
    - Do we get a better classifier than in Model 1?

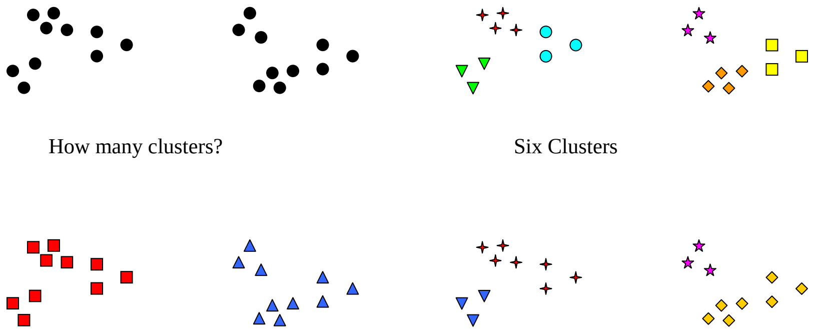
Weka: Class to clusters evaluation

```
Attribute              Full Data        C = 0         C = 1          C = 2
                       (3333)           (1221)        (1190)         (922)
==================================================================
Inter Plan               no             no            no             no
  no                   3010 ( 90%)      1117 ( 91%)   1063 ( 89%)    830 ( 90%)
  yes                   323 (  9%)      104 (  8%)    127 ( 10%)     92 (  9%)

VoiceMail Plan          no               no            no            yes
  yes                   922 ( 27%)      0 (  0%)      0 (  0%)       922 (100%)
  no                   2411 ( 72%)      1221 (100%)   1190 (100%)    0 (  0%)

No of Vmail Mesgs      0.1588            0            0              0.5741
                      +/-0.2684         +/-0          +/-0           +/-0.1482
```

Median

Standard deviation

Notion of a Cluster can be Ambiguous



How many clusters?

Six Clusters

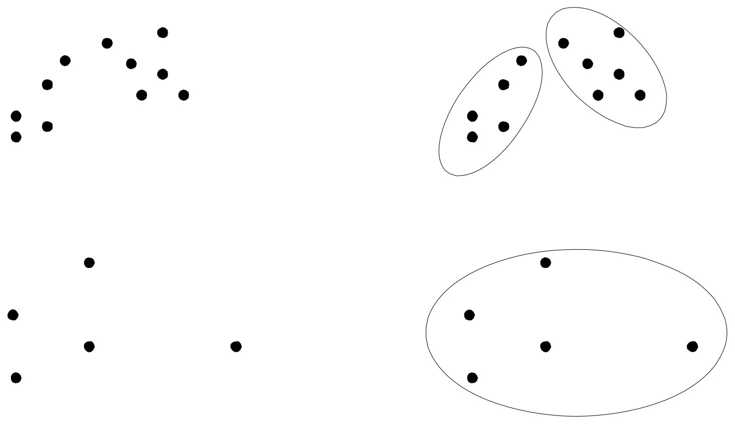
Two Clusters

Four Clusters

Types of Clusterings

- A clustering is a set of clusters
- Important distinction between **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

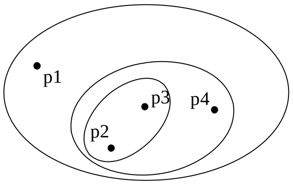
Partitional Clustering



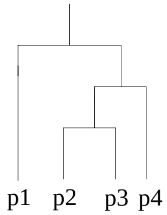
Original Points

A Partitional Clustering

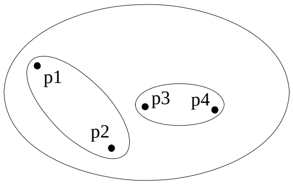
Hierarchical Clustering



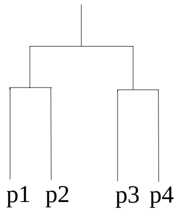
Hierarchical Clustering



Dendrogram



Hierarchical Clustering



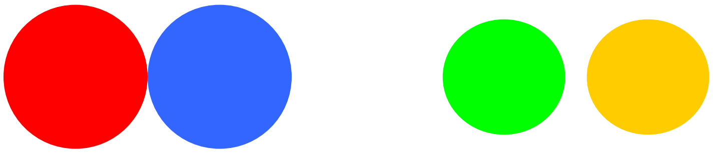
Dendrogram

How to Define a Cluster

- But, what is a cluster?
  - There are different ways to answer to this question

Types of Clusters: Center-Based

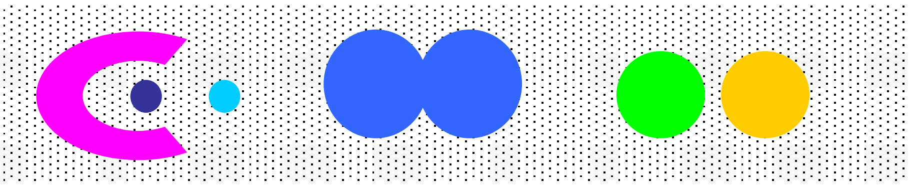
- Center-based
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of its cluster, than to the center of any other cluster
  - The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the most "representative"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



4 center-based clusters

Types of Clusters: 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.

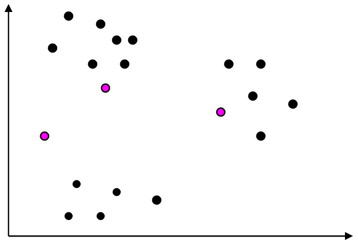


6 density-based clusters

Clustering Algorithms

- **K-means** and its variants
  - K-means is available in Weka
    - **Parameters**: Distance function (e.g. Euclidian, Manhattan) and number of clusters
- Hierarchical clustering
- Density-based clustering (**DBSCAN**)
  - Available in Weka

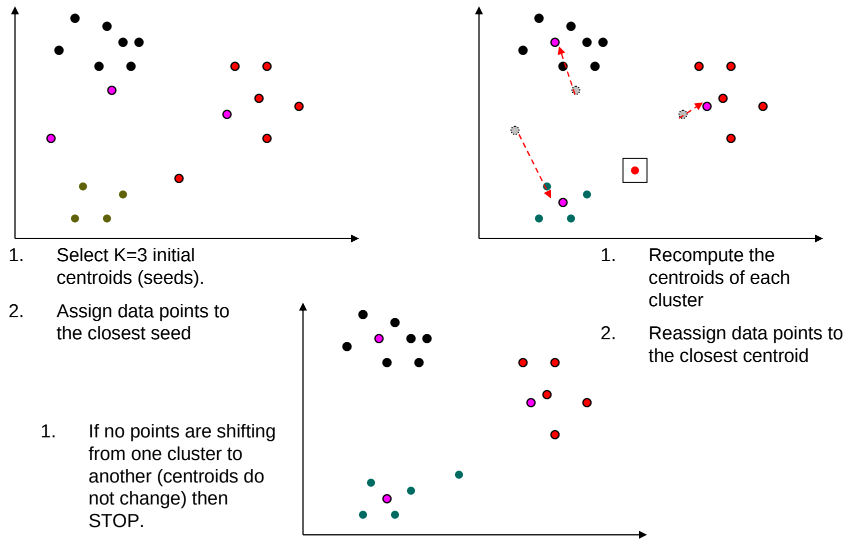
K-means: intuition



1. Select K=3 initial centroids (seeds).

- K = number of clusters
- K is a user-specified parameter

K-means: intuition



1. Select K=3 initial centroids (seeds).
2. Assign data points to the closest seed

1. Recompute the centroids of each cluster
2. Reassign data points to the closest centroid

1. If no points are shifting from one cluster to another (centroids do not change) then STOP.

Clustering and Objective Functions

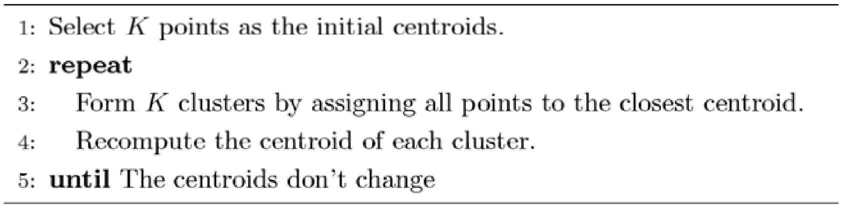
- Clusters defined by an objective function
  - Finds clusters that minimize (or maximize) an objective function.
  - Most common measure is Sum of Squared Error (SSE)

$$\text{SSE} = \sum_{i=1}^{K} \sum_{x \in C_i} \text{dist}^2(m_i, x)$$

  - For each point, the error is the distance to the nearest centroid
  - To get SSE, we square these errors and sum them.
  - x is a data point in cluster C_i and m_i is the centroid for cluster C_i
    - can show that m_i corresponds to the center (mean) of the cluster
- How to compute?
  - Enumerate all possible ways of dividing the points into clusters and evaluate the `goodness' of each potential set of clusters by using the given objective function.
    - Not feasible : problem is NP Hard!!!

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



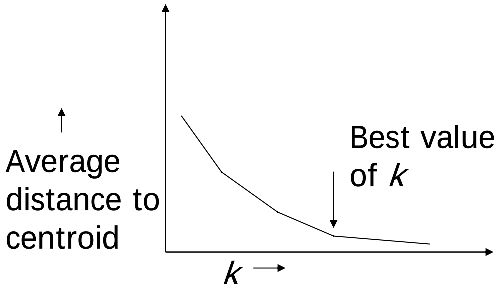
```
Select $K$ points as the initial centroids.
repeat
    Form $K$ clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change
```

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 * d * I)
  - n = number of points, K = number of clusters,
    I = number of iterations, d = number of attributes

Getting *K* Right

- Try different *k*, looking at the change in the average distance to centroid, as *k* increases.
- Average falls rapidly until right *k*, then changes little.

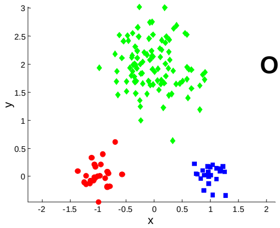


↑ Average distance to centroid

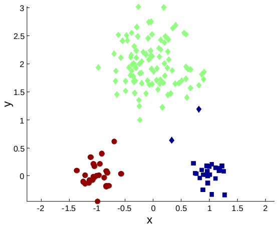
Best value of *k*

k →

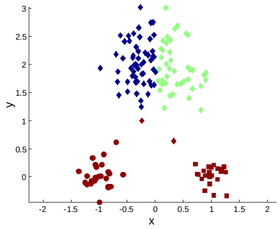
Two different K-means Clusterings



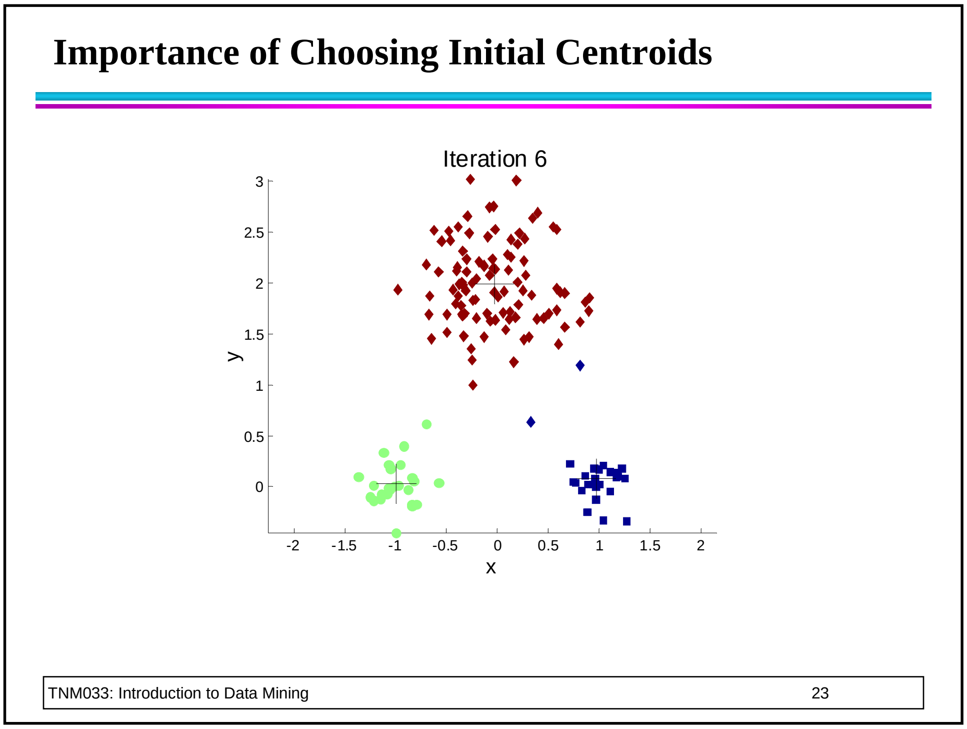
Original Points



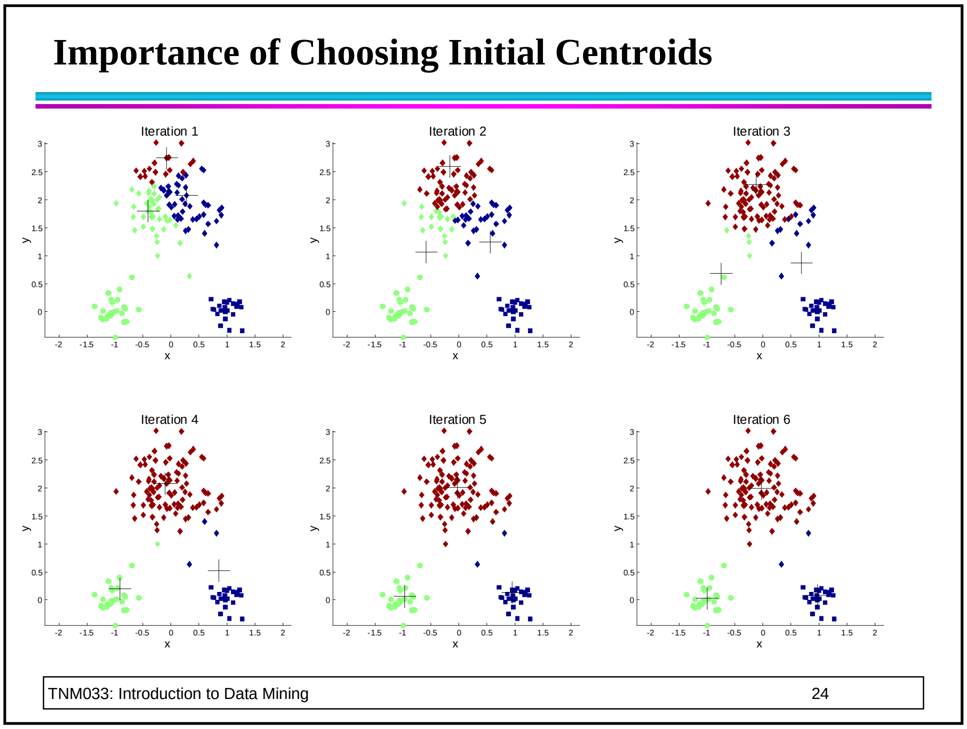
Optimal Clustering



Sub-optimal Clustering

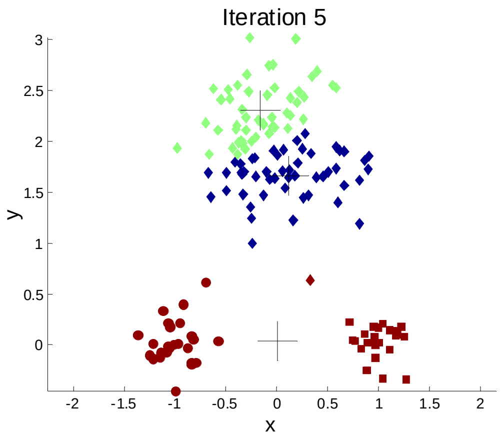


Importance of Choosing Initial Centroids

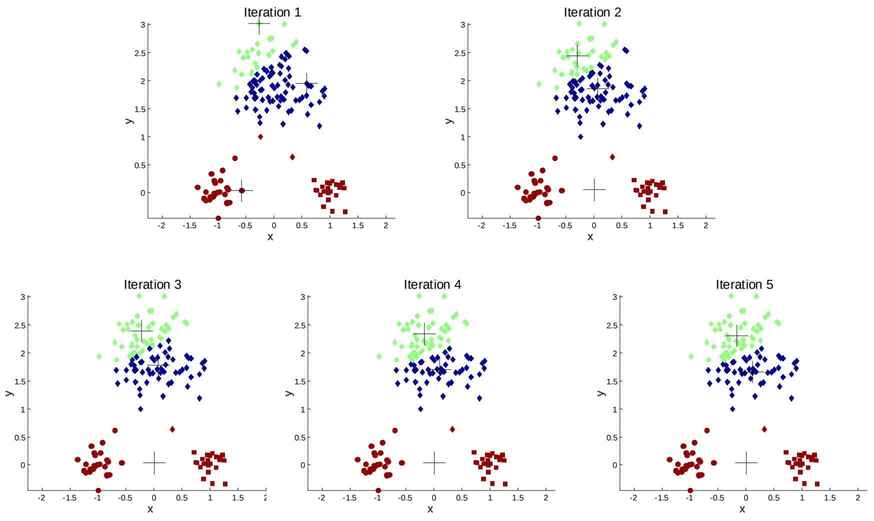


Importance of Choosing Initial Centroids

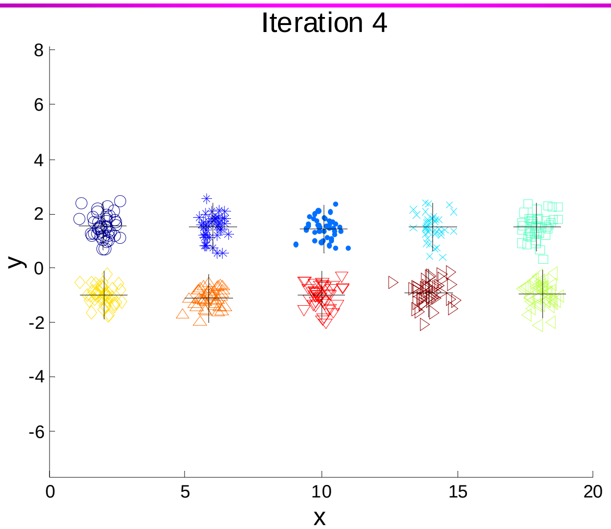
Importance of Choosing Initial Centroids …



Importance of Choosing Initial Centroids …

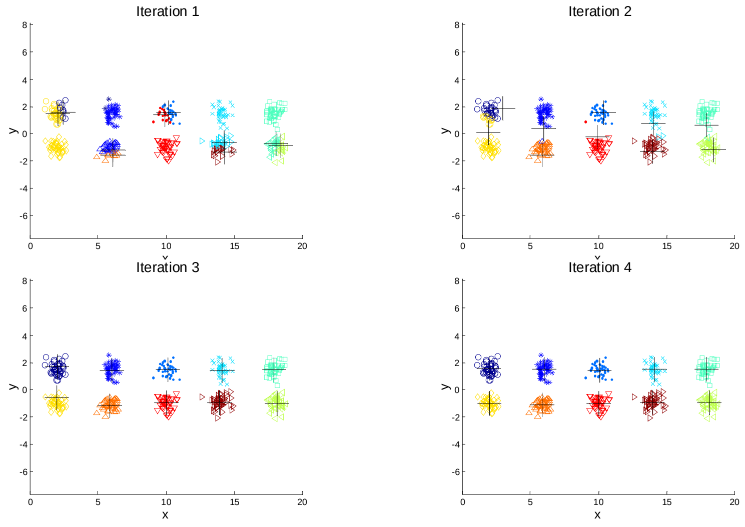


10 Clusters Example



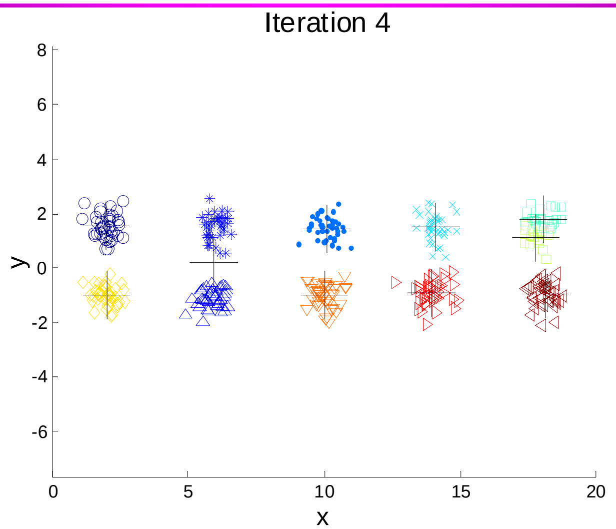
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



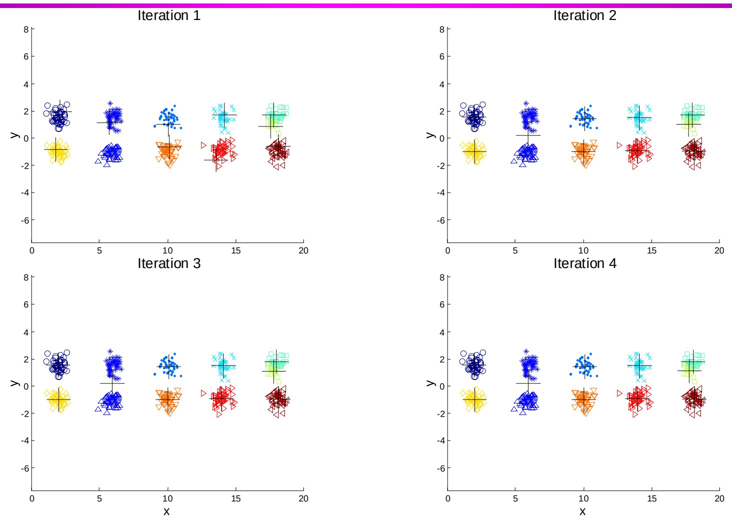
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



Starting with some pairs of clusters having three initial centroids, while other have only one.

10 Clusters Example



Starting with some pairs of clusters having three initial centroids, while other have only one.

Solutions to Initial Centroids Problem

- Multiple runs
  - Select the set of clusters with least SSE
  - May not help!
- Select a first point as the centroid of all points. Then, select (K-1) most widely separated points
  - **Problem**: can select outliers
  - **Solution**: Use a sample of points
- Post-processing
- Bisecting K-means
  - Not as susceptible to initialization issues

Pre-processing and Post-processing

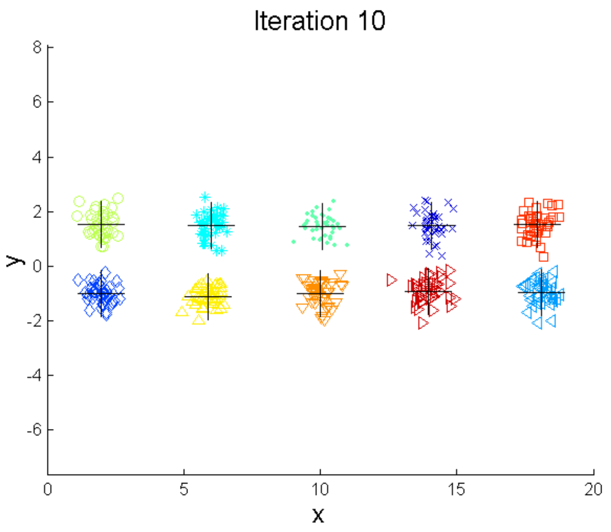
- Pre-processing
  - Normalize the data
    - Attribute values fall roughly into the same range
  - Eliminate outliers
    - Centroids may not be good representatives
    - SSE will be also higher
- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters, i.e., clusters with relatively high SSE or high standard deviation for an attribute
  - Merge clusters that are ‘close’ and that have relatively low SSE

Bisecting K-means

- **Bisecting K-means** algorithm
  - Variant of K-means that can produce a partitional or a hierarchical clustering

```
Initialize the list of clusters to contain the cluster containing all points.
repeat
  Select a cluster from the list of clusters
  for $i = 1$ to $number\_of\_iterations$ do
    Bisect the selected cluster using basic K-means
  end for
  Add the two clusters from the bisection with the lowest SSE to the list of clusters.
until Until the list of clusters contains $K$ clusters
```

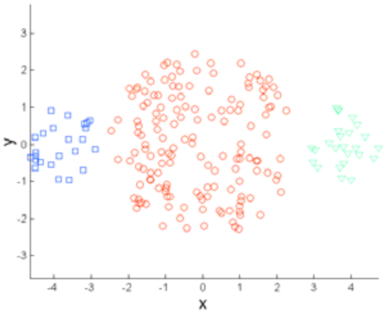
Bisecting K-means Example



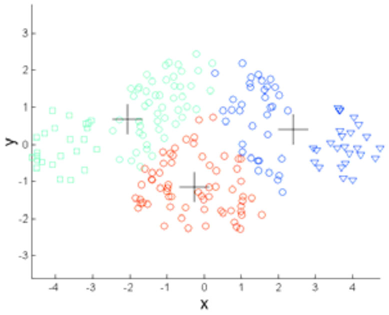
Strengths and Weaknesses of K-means

- **Strengths**
  - Efficient for medium size data
    - BIRCH and CURE for very large data sets
  - Bisecting K-means not so susceptible to initialization problems
- **Weaknesses**
  - Not suitable for all data types
    - Clusters are of differing sizes
    - Densities
    - Non-globular shapes
  - Outliers are a problem
  - Choice of seeds (initial centroids)

Limitations of K-means: Differing Sizes

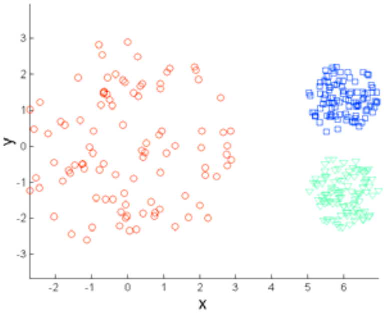


Original Points

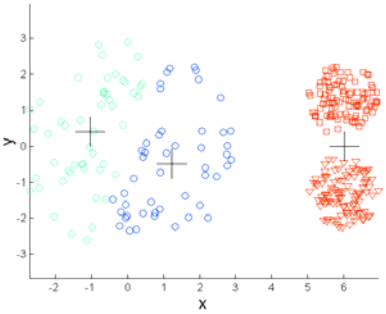


K-means (3 Clusters)

Limitations of K-means: Differing Density

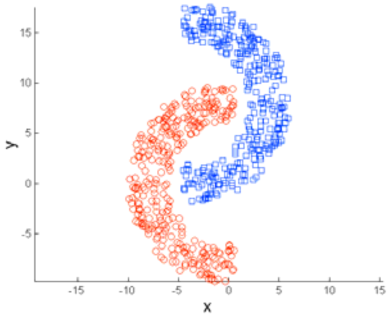


Original Points

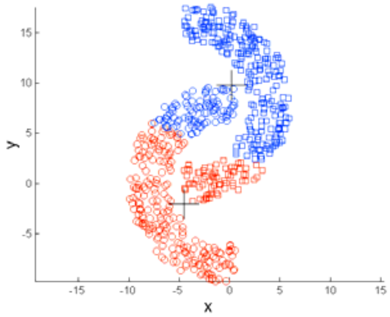


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes



Original Points



K-means (2 Clusters)

Next …

- Similarity measures
  - Is Euclidian distance appropriate for all types of problems?
- Hierarchical clustering
- DBSCAN algorithm
- Cluster validation