Analytics and Visualization of Big Data

Fadel M. Megahed

Lecture 14: Clustering

AUBURN UNIVERSITY
SAMUEL GINN
COLLEGE OF ENGINEERING

Department of Industrial and Systems Engineering

Spring 13

Refresher: Big Data Analytics Based on Types of Data

High dim. data	Graph data	Infinite data	Machine learning	Apps
Locality sensitive hashing	PageRank, SimRank	Filtering data streams	SVM	Recommender systems
Clustering	Community Detection	Web advertising	Decision Trees	Association Rules
Dimensionality reduction	Spam Detection	Queries on streams	Perceptron, kNN	Duplicate document detection

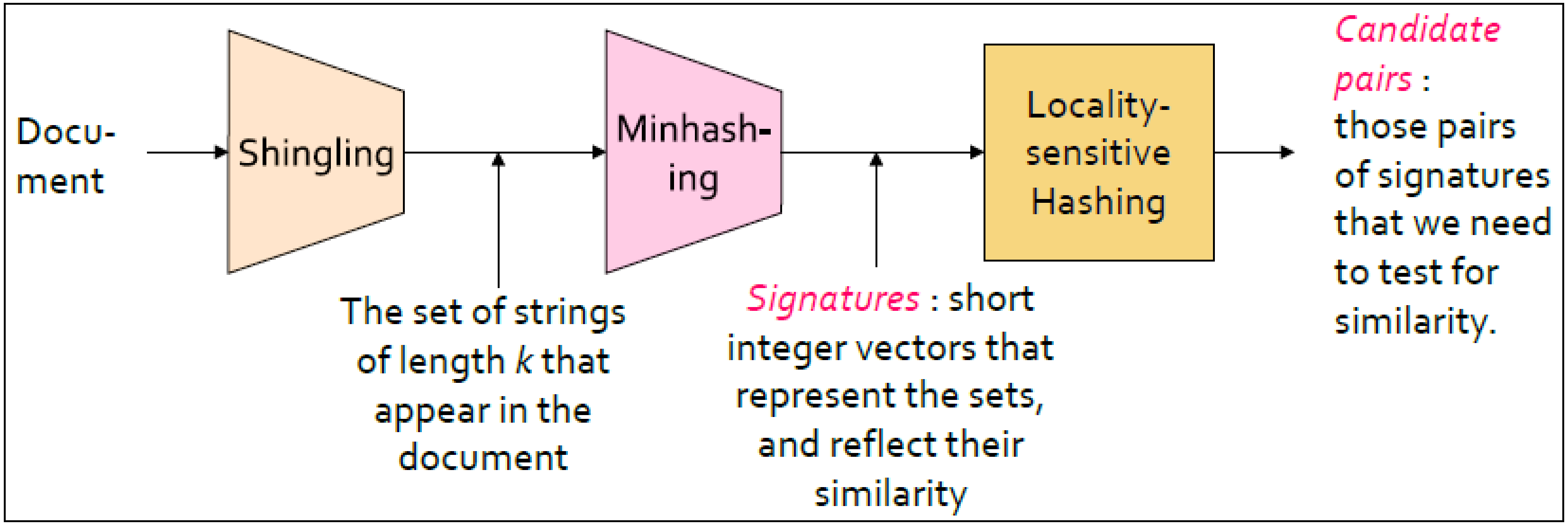
Source: Jure Leskovic, Stanford CS246, Lecture Notes, see http://cs246.stanford.edu

Refresher: High Dimensional Data

- Given a cloud of data points, we want to understand their underlying structure (what do we mean by that?)

Source: http://www.cs.toronto.edu/~laurens/drtoronto/Dimensionality_Reduction_%40_Toronto.html

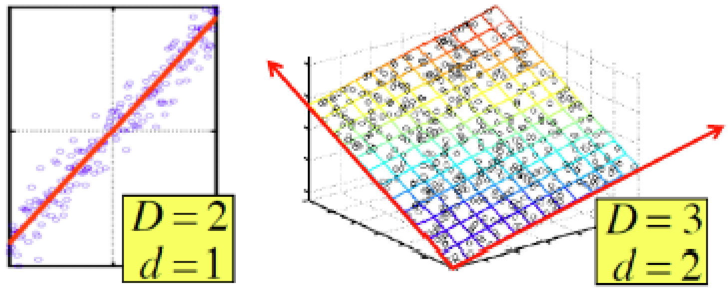
Refresher: Locality Sensitive Hashing



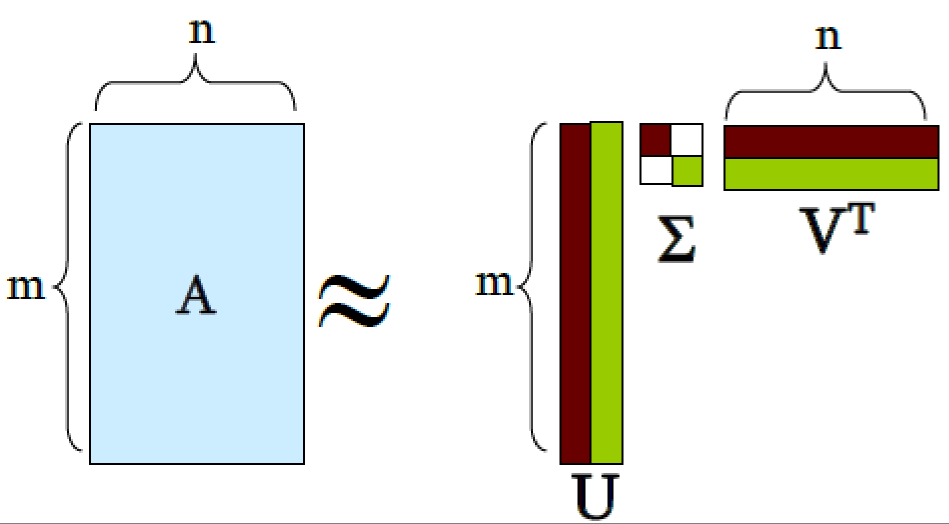
Steps for Locality Sensitive Hashing:

1. **Shingling:** convert docs to sets
2. **Minhashing:** convert large sets to short signatures, while preserving similarity
3. **Locality-sensitive hashing:** focus on pairs of signatures likely to be similar

Refresher: Dimensionality Reduction (PCA and SVD)



$$\mathbf{A} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i$$



Source: Jure Leskovic, Stanford CS246, Lecture Notes, see http://cs246.stanford.edu

Overview of Topics Covered in Chapter 7

Hierarchical Clustering

- In Euclidean Space
- Efficiency
- In Non-Euclidean Spaces

K-Means

- Basics
- Initialization
- Picking the Right Value of K
- BFR Algorithm

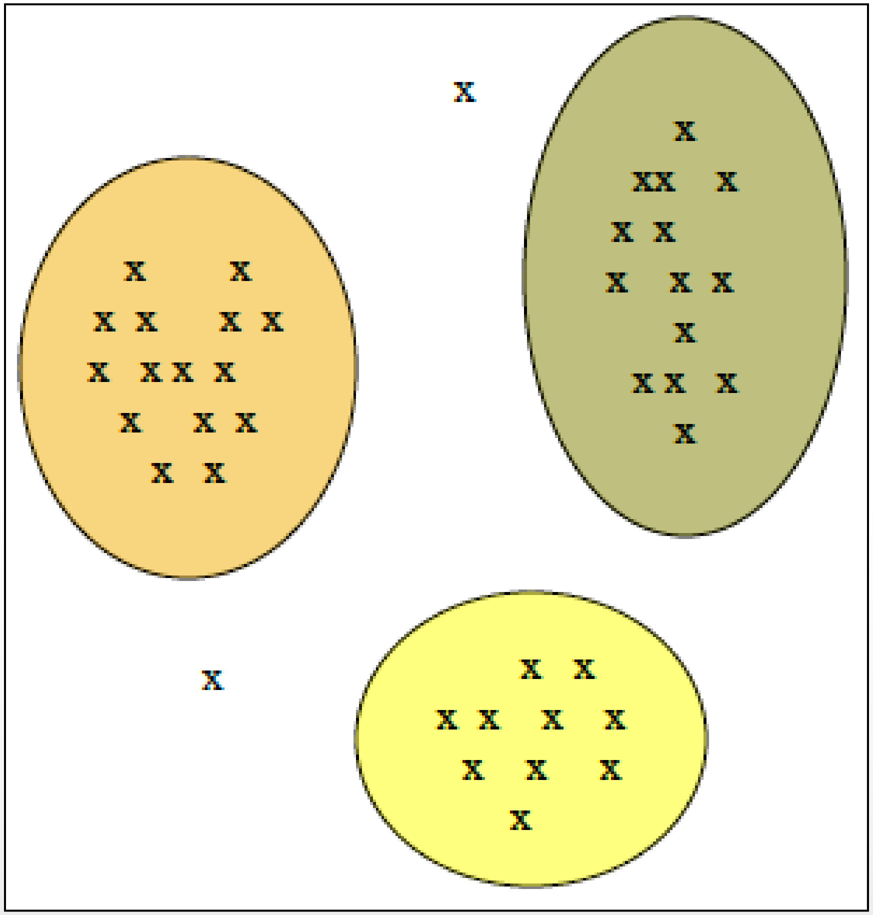
The Cure Algorithm

- Initialization
- Completion of the CURE Algorithm

An Overview of Today's Topic → Clustering

- Given a **set of points**, **group the points** into some **# clusters**, so that:
  - Members of a cluster are close/similar to each other
  - Members of different clusters are dissimilar

- Usually:
  - Points are in a high-dimensional space
  - Similarity is defined using a distance measure
    - Euclidean, Jaccard, …



Clustering has Many Applications in IE

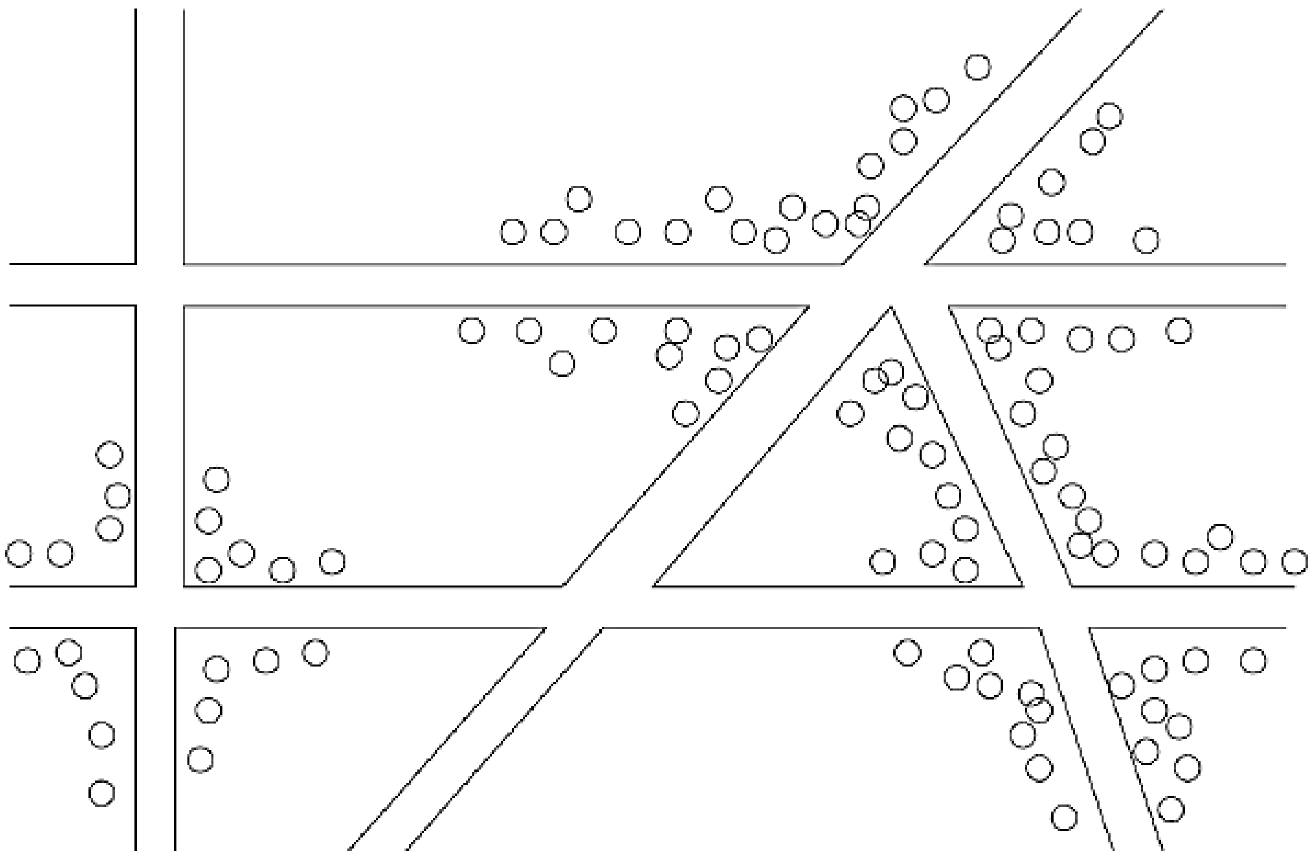
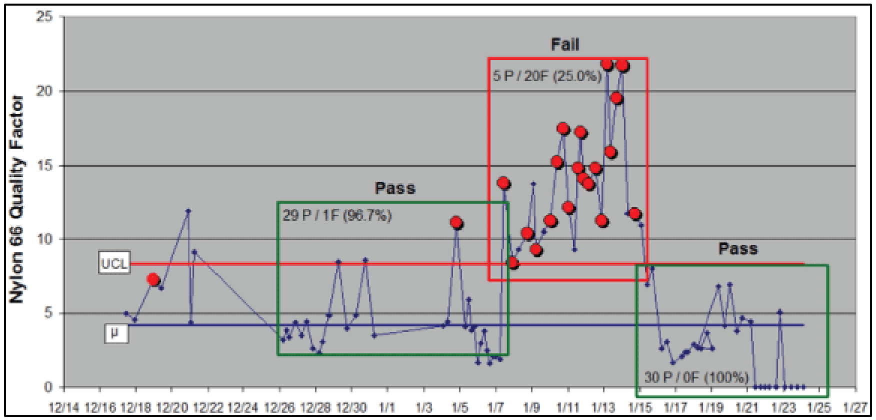
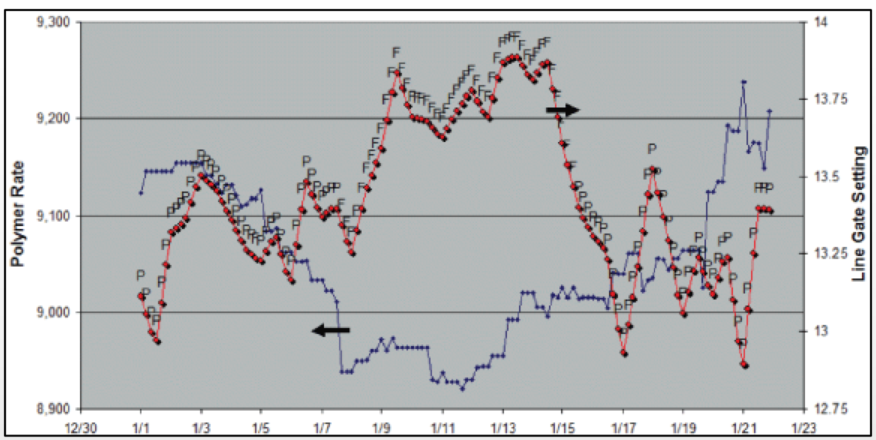


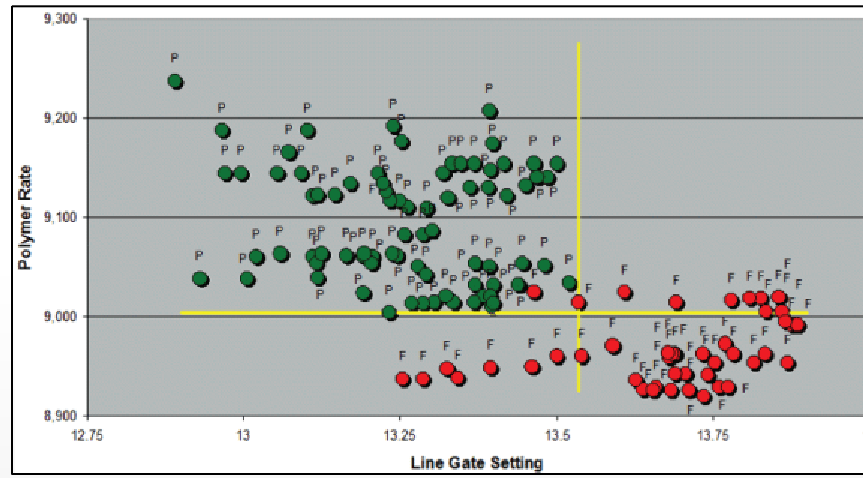
Figure 1.1: Plotting cholera cases on a map of London

Source: A. Rajaraman, J. Leskovec, J.D. Ullman. (2012). “Mining of Massive Datasets”. http://i.stanford.edu/~ullman/mmds.html

Clustering has Many Applications in IE

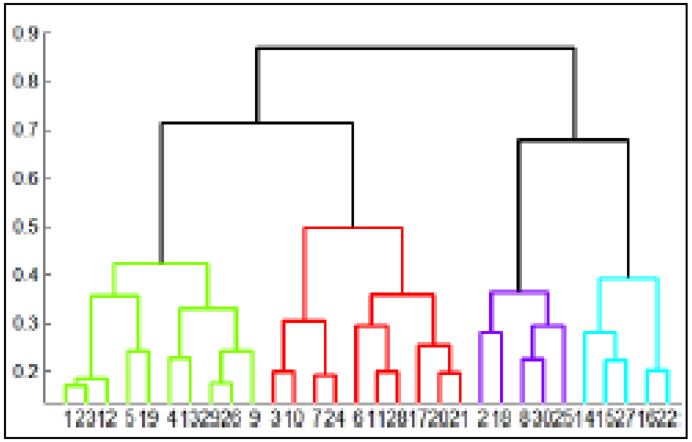




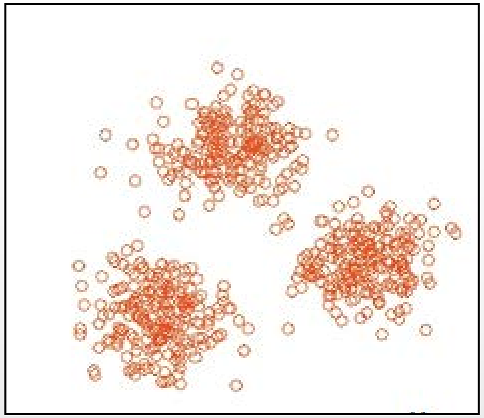


Source: http://www.isixsigma.com/tools-templates/sampling-data/process-data-mining-partitioning-variance/

Overview of Clustering Methods



- **Hierarchical Clustering:**
  - **Agglomerative (bottom up):**
    - Initially, each point is a cluster
    - Repeatedly combine the two "nearest" clusters into one
  - **Divisive (top down):**
    - Start with one cluster and recursively split it

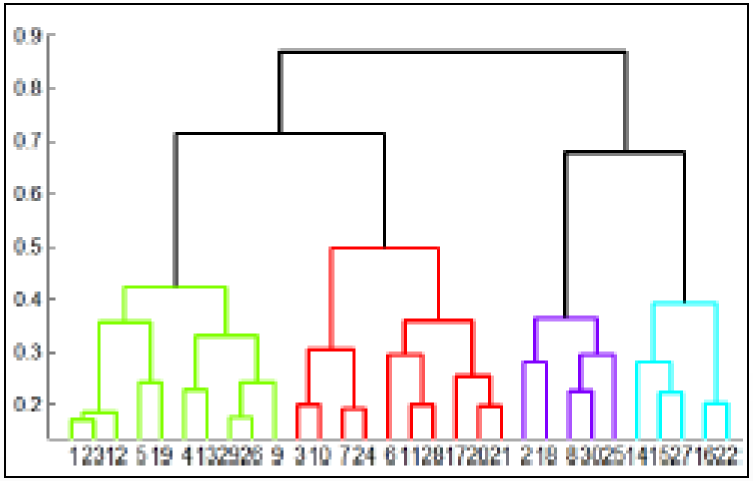


- **Point assignment:**
  - Maintain a set of clusters
  - Points belong to "nearest" cluster

Hierarchical Clustering

- **Key Operation:**
  - Repeatedly combine two nearest clusters

- **Important questions:**
  1. How will clusters be represented?
  2. How will we choose which two clusters to merge?
  3. When will we stop combining clusters?



Slide Adapted from: Jure Leskovic, Stanford CS246, Lecture Notes, see http://cs246.stanford.edu

Hierarchical Clustering – Some Details

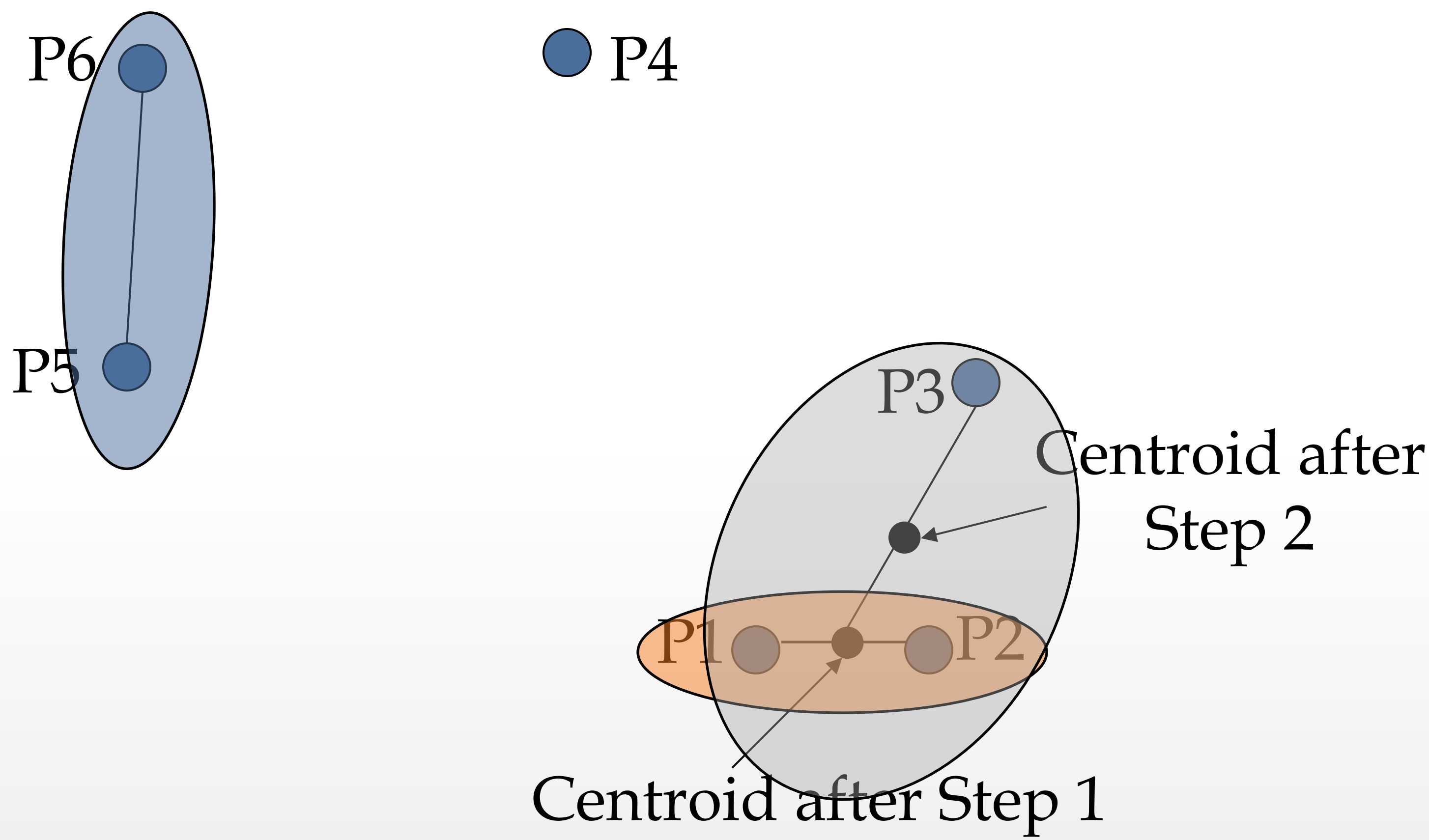
- **Operation: Repeatedly combine two nearest clusters**

1. **How will clusters be represented?**
   - **Key problem:** As you build clusters, how do you represent the location of each cluster, to tell which pair of clusters is closest?
     - **Euclidean case:** each cluster has a centroid = average of its (data)points
     - **Non-Euclidean case:** Very similar *(but use non-Euclidean distances)*

2. **How will we choose which two clusters to merge?**
   - Measure cluster distances by distances of centroids

3. **When will we stop combining clusters?**
   - When combining → inadequate cluster (e.g. avg distance between points in clusters increases) → Stop by producing a tree of clusters

Hierarchical Clustering – An Example



Hierarchical Clustering – Non-Euclidean Distances

- **What about the Non-Euclidean case?**
  - The only "locations" we can talk about are the points themselves
    - i.e., there is no "average" of two points

- **One Approach:**

1. **How will clusters be represented?**

   clustroid = (data)point *"closest"* to other points

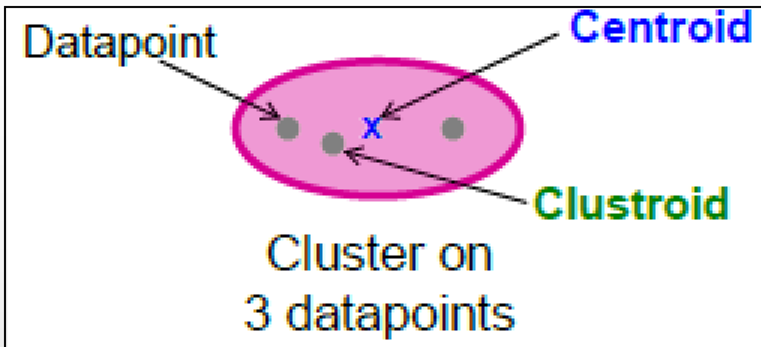
2. **How will we choose which two clusters to merge?**

   Treat clustroid as if it were centroid, when computing intercluster distances

1. **How will clusters be represented?**

   clustroid = point "closest" to other points

- **Possible meanings of "closest":**
  - Smallest maximum distance to other points
  - Smallest average distance to other points
  - Smallest sum of squares of distances to other points
    - For distance metric ***d*** clustroid ***c*** of cluster ***C*** is: $\min_{c} \sum_{x \in C} d(x,c)^2$



Centroid is the avg. of all (data)points in the cluster. This means centroid is an "artificial" point.
Clustroid is an existing (data)point that is "closest" to all other points in the cluster.

Slide Adapted from: Jure Leskovic, Stanford CS246, Lecture Notes, see http://cs246.stanford.edu

Implementation of Hierarchical Clustering Approaches

- **Naïve implementation of hierarchical clustering:**
  - At each step, compute pairwise distances between all pairs of clusters, then merge
  - O(*N*3)

- VERY COMPUTATIONALLY EXPENSIVE

Overview of Topics Covered in Chapter 7

Hierarchical Clustering

- In Euclidean Space
- Efficiency
- In Non-Euclidean Spaces

K-Means

- **Basics**
- **Initialization**
- **Picking the Right Value of K**
- BFR Algorithm

The Cure Algorithm

- Initialization
- Completion of the CURE Algorithm

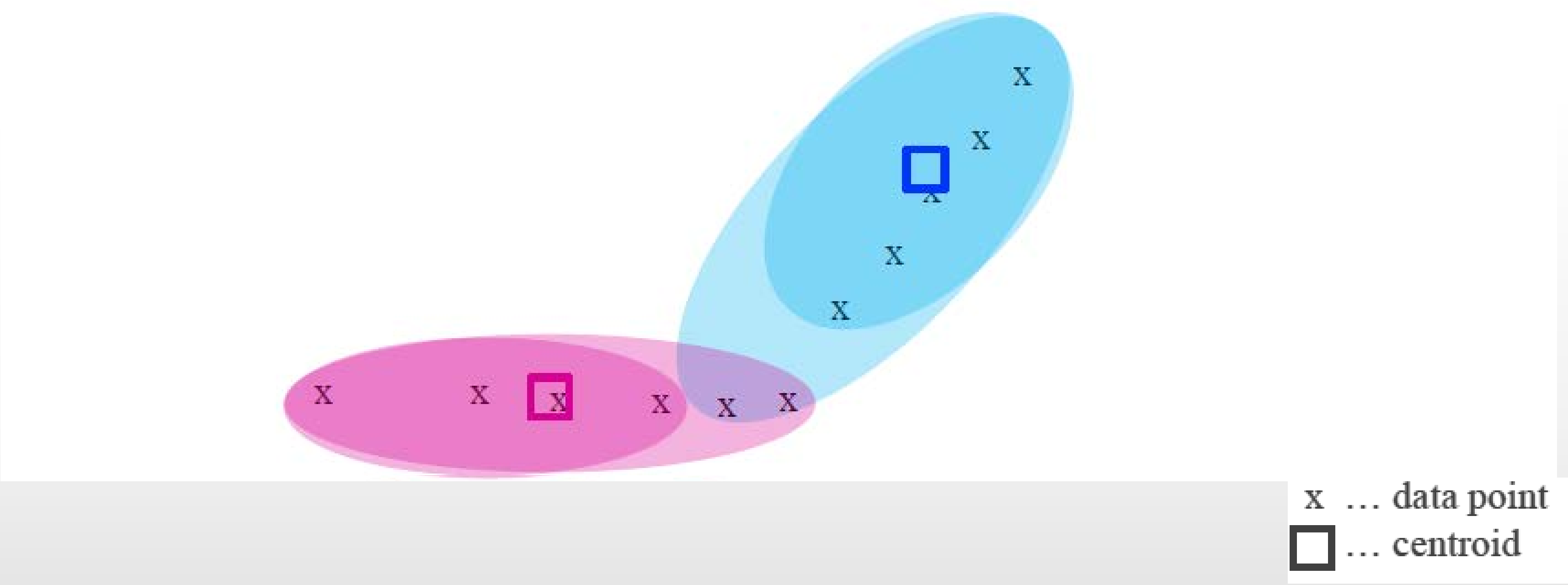
K-Means Clustering

- Most widely used clustering algorithm. It follows a very simple procedure whose main characteristics are:
  - Assumes Euclidean space/distance
  - Start by picking *k*, the number of clusters
  - Initialize clusters by picking one point per cluster
    - Example: Pick one point at random, then **k-1** other points, each as far away as possible from the previous points

Algorithm Basic K-means Algorithm.

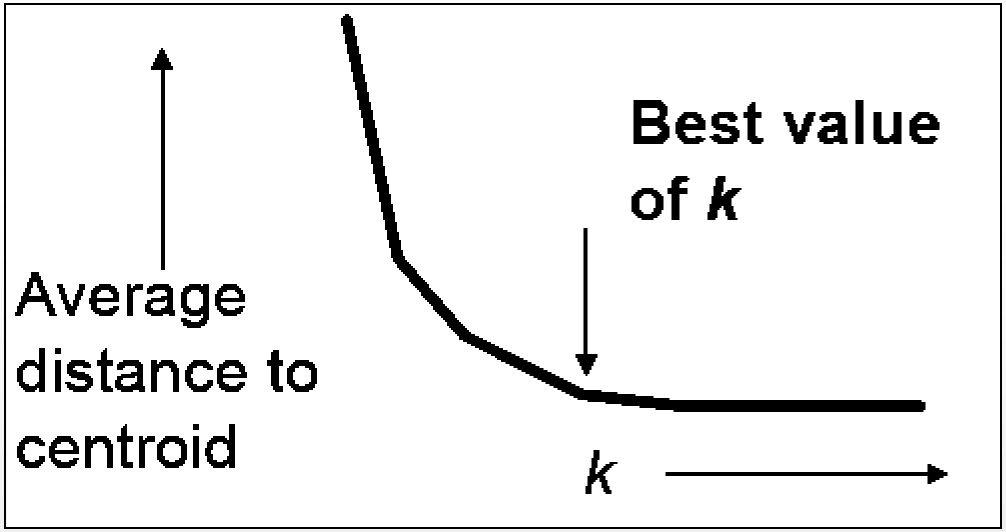
Select K points as the initial centroids.
repeat
    Form K clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change

Problem Definition: Assume that we have these 11 points, and we have initialized the k-means method by picking the highlighted points our two centroids (k=2, given)



Getting the *K* Right

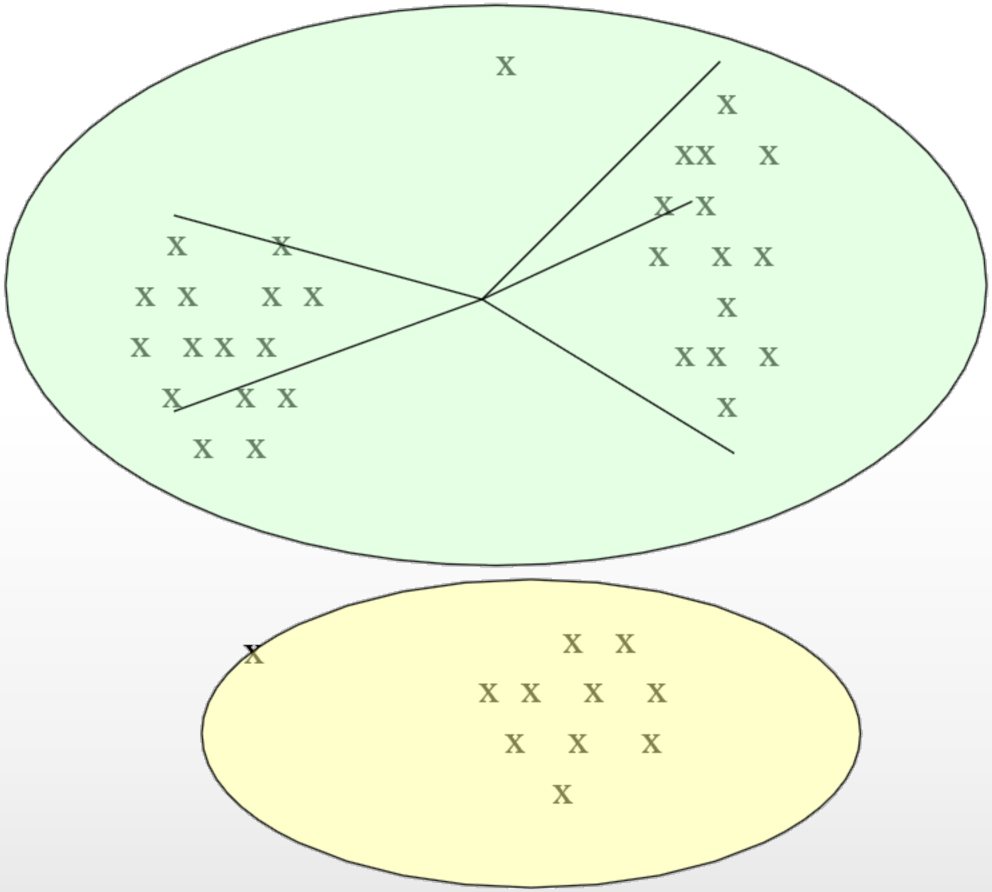
- Try different k, looking at the change in the average distance to centroid, as *k* increases.
- Average falls rapidly until right *k*, then changes little



Slide Adapted from: Jure Leskovic, Stanford CS246, Lecture Notes, see http://cs246.stanford.edu

Getting the *K* Right: An Example

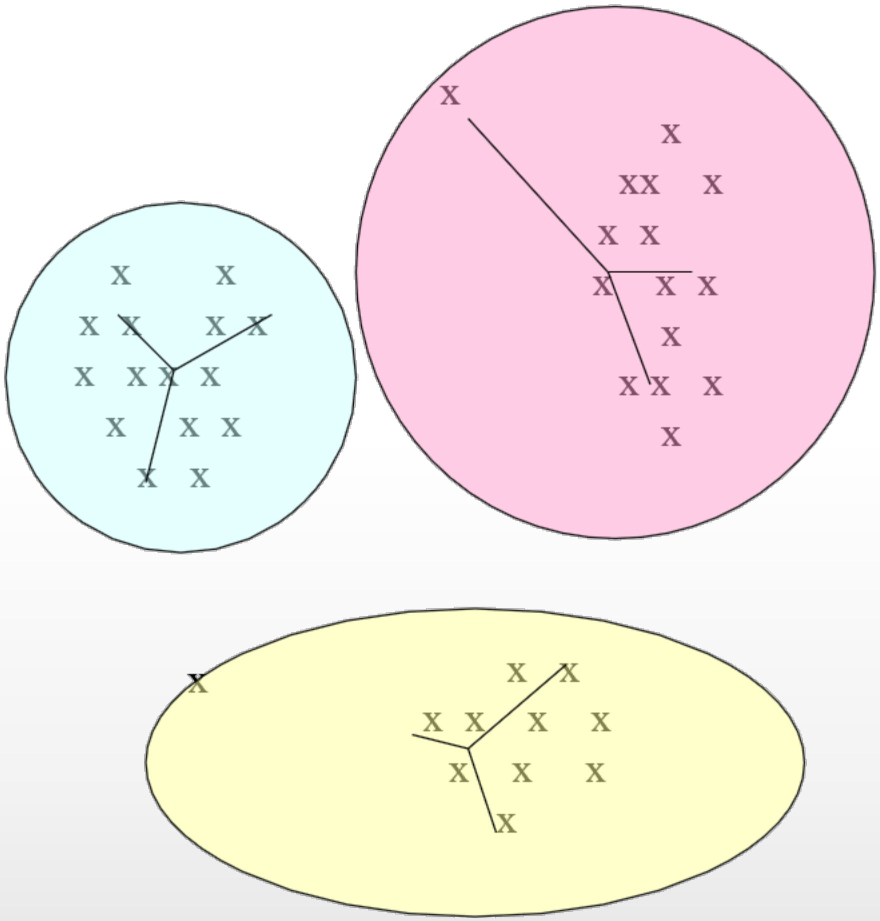
Too few;
many long
distances
to centroid.



Slide Adapted from: Jure Leskovic, Stanford CS246, Lecture Notes, see http://cs246.stanford.edu

Getting the *K* Right: An Example

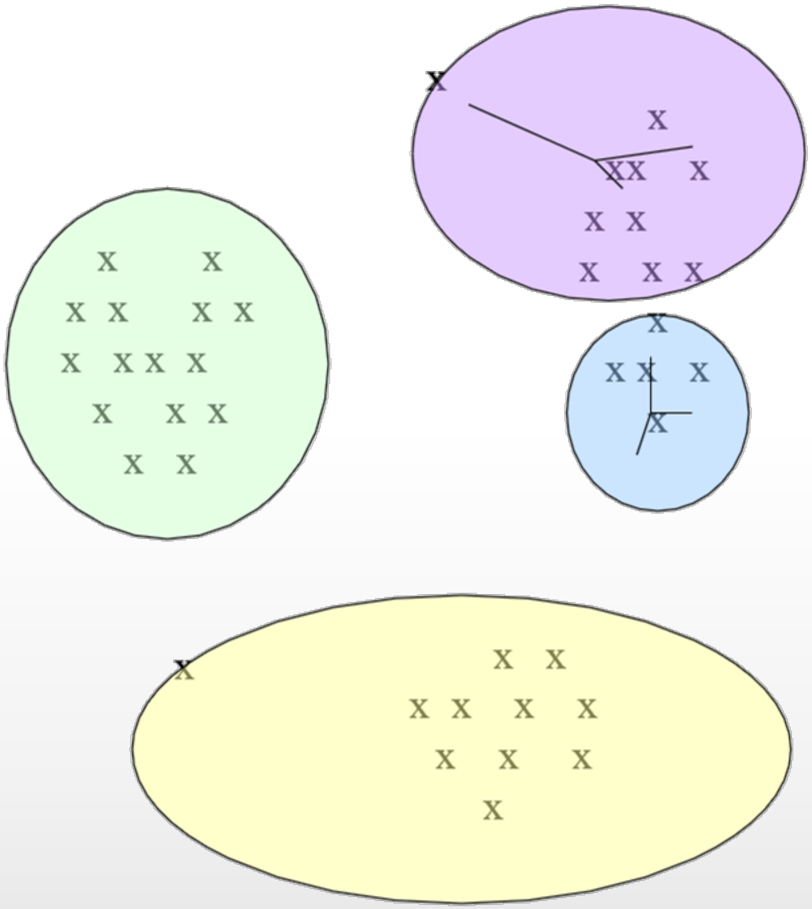
Just right;
distances
rather short.



Slide Adapted from: Jure Leskovic, Stanford CS246, Lecture Notes, see http://cs246.stanford.edu

Getting the *K* Right: An Example

Too many;
little improvement
in average
distance.



Slide Adapted from: Jure Leskovic, Stanford CS246, Lecture Notes, see http://cs246.stanford.edu