Hierarchical Clustering under Euclidean Case

华中科技大学 计算机科学与技术学院 School of Computer Science &Technology, HUST

(1) How to represent a cluster of many points?

- **Key problem:** As you merge 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*** (簇质心) = average of its (data) points
- 备注: 或者说**簇中平均点**, 也就是将簇内所有点进行算术平均得到的点.

(2) How to determine "nearness" of clusters?

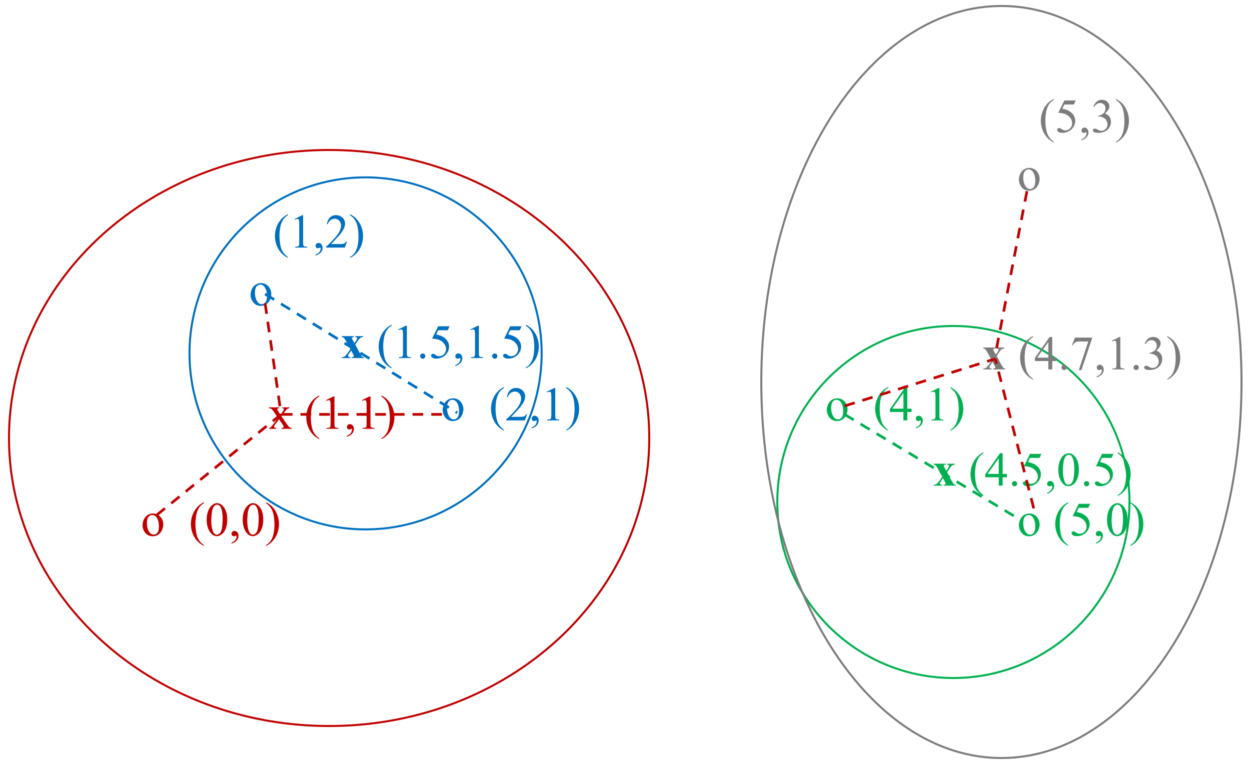
- Measure cluster distances by distances of centroids

Hierarchical Clustering under Euclidean Case

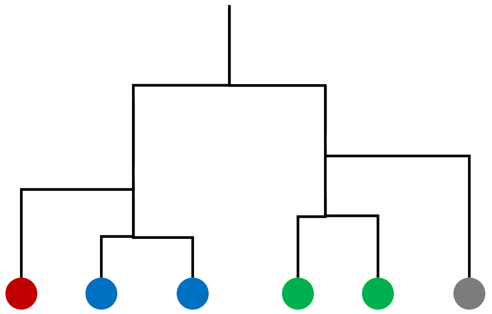
(3) When to stop combining clusters?

- **M clusters:** We could be told, or have a belief, about how many clusters there are in the data.
- At some point the best combination of **existing clusters** produces a cluster that is **inadequate:** E.g., we could insist that any cluster have an average distance between the centroid and its points no greater than some limit.
- Clustering until there is only **one cluster**: Meaningless to return a single cluster. Rather, we return the tree representing the way in which all the points were combined.

Example: Hierarchical clustering



Data:
o … data point
x … centroid



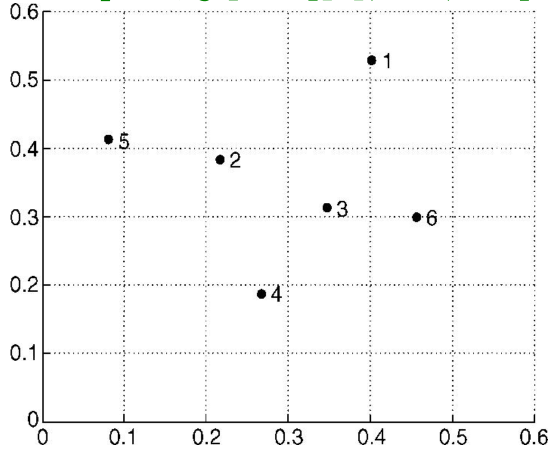
Dendrogram

Hierarchical clustering

- 在前面提及的问题(2)中, **How to determine “nearness” of clusters?**
  - Measure **cluster distances** by distances of centroids (簇质心之间的距离)
  - Then, repeatedly combine two nearest clusters
- **There are alternative rules for controlling hierarchical clustering**
  - We will explain next

Proximity Matrix

❑计算任意两个数据之间的距离得到一个**相似度矩阵(proximity matrix, 或称临近矩阵)**

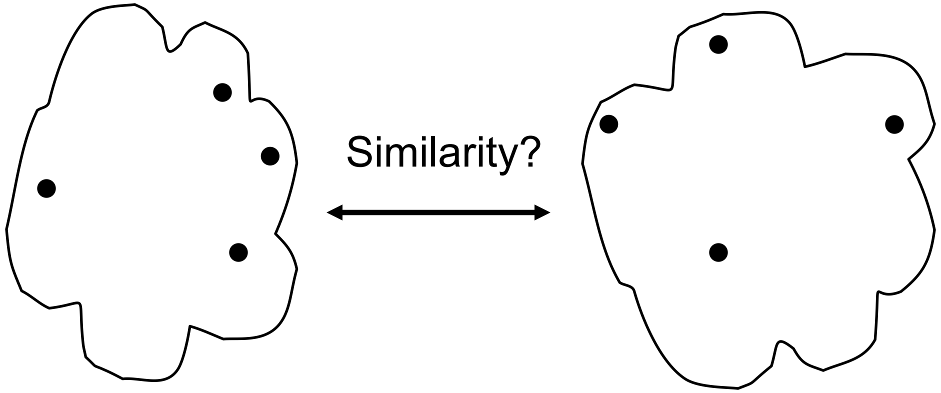


	x	*y*
P1	0.4005	0.5306
P2	0.2148	0.3854
P3	0.3457	0.3156
P4	0.2652	0.1875
P5	0.0789	0.4139
p6	0.4548	0.3022

Proximity Matrix:

	P1	P2	P3	P4	P5	P6
P1	0.0000	0.2357	0.2218	0.3688	0.3421	0.2347
P2	0.2357	0.0000	0.1483	0.2042	0.1388	0.2540
P3	0.2218	0.1483	0.0000	0.1513	0.2843	0.1100
P4	0.3688	0.2042	0.1513	0.0000	0.2932	0.2216
P5	0.3421	0.1388	0.2843	0.2932	0.0000	0.3921
p6	0.2347	0.2540	0.1100	0.2216	0.3921	0.0000

Inter-Cluster Similarity

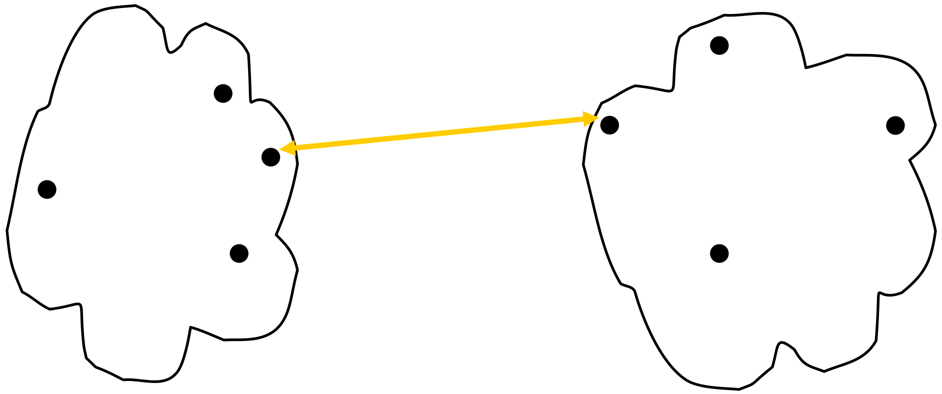


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix
(相似度矩阵)

- ❑MIN
- ❑MAX
- ❑Group Average
- ❑Distance Between Centroids
- ❑Other methods driven by an objective function
  - ➢Ward’s Method uses squared error(平方误差)

Inter-Cluster Similarity

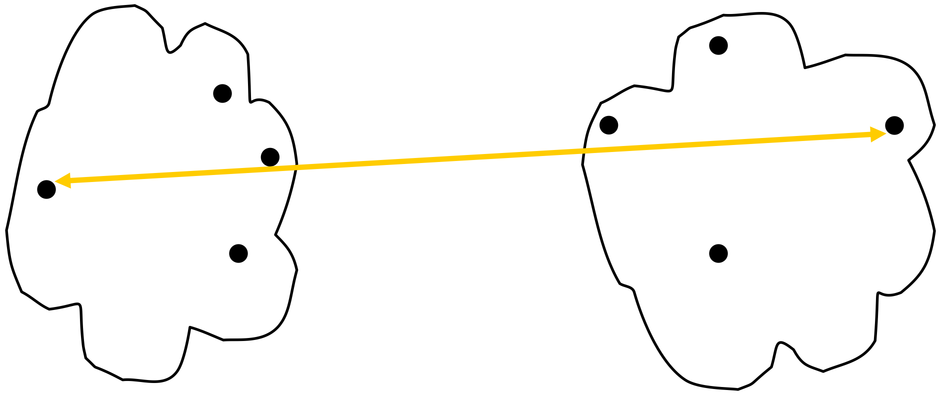


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’ s Method uses squared error

Inter-Cluster Similarity

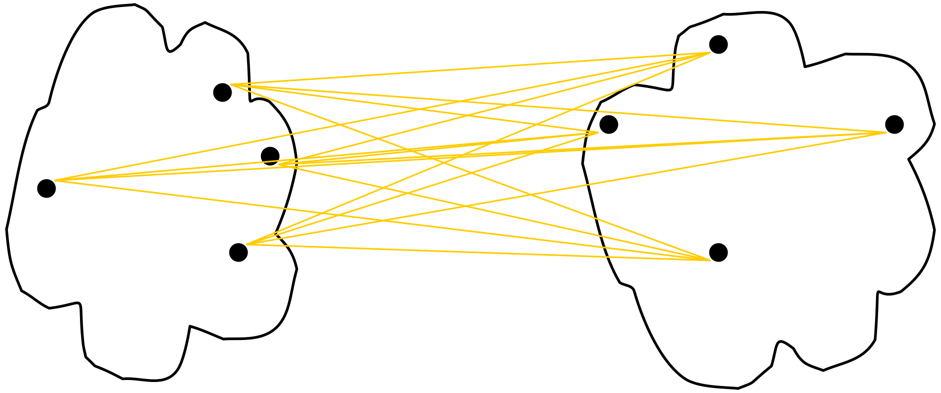


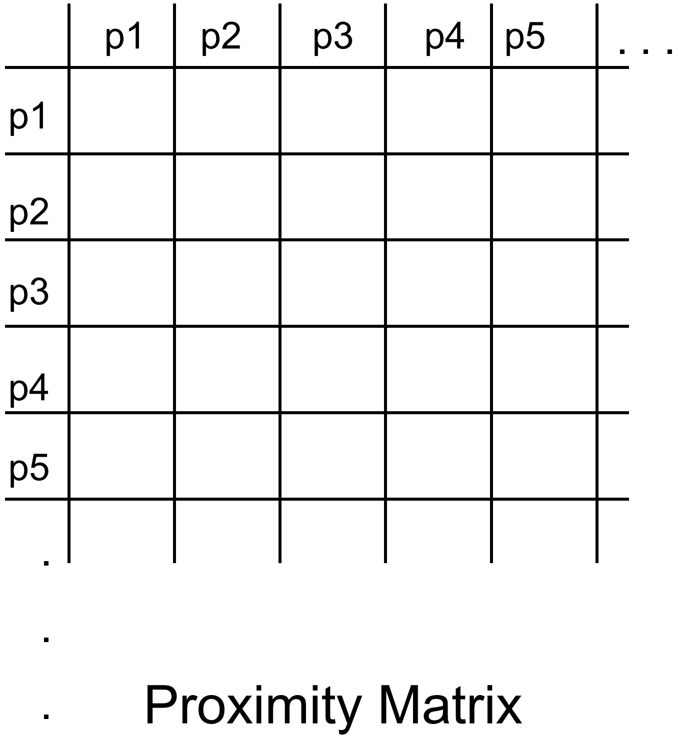
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- ❑MIN
- ❑MAX
- ❑Group Average
- ❑Distance Between Centroids
- ❑Other methods driven by an objective function
  - ➢Ward’ s Method uses squared error

Inter-Cluster Similarity

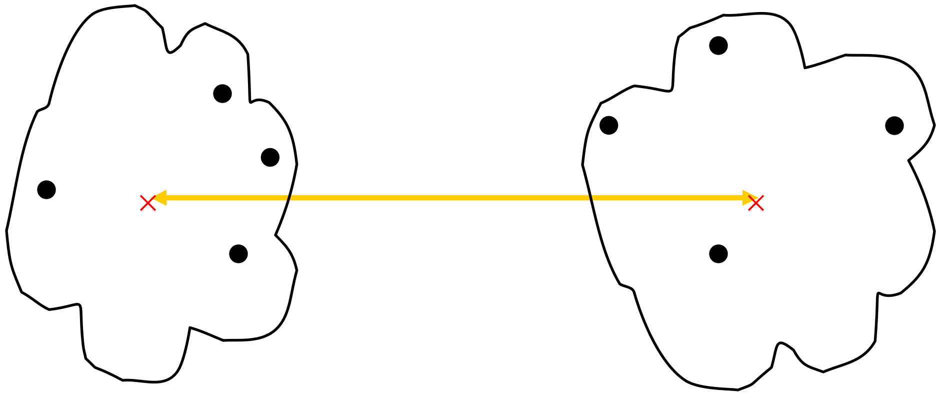




Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’ s Method uses squared error

Inter-Cluster Similarity



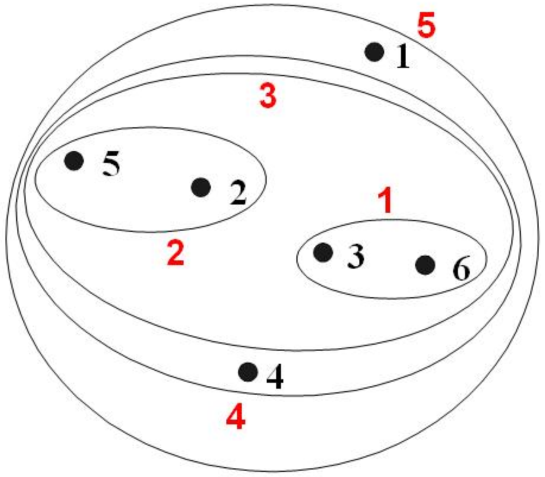
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

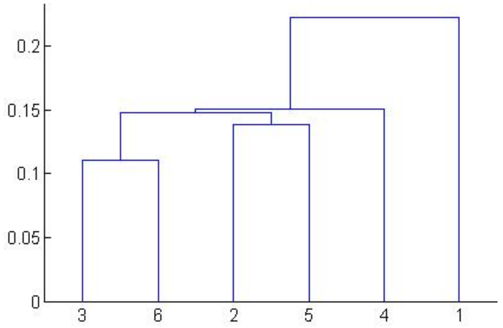
- MIN
- MAX
- Group Average
- Distance Between Centroids (也就是前面提及的簇质心之间的距离)
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Cluster Similarity: MIN

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

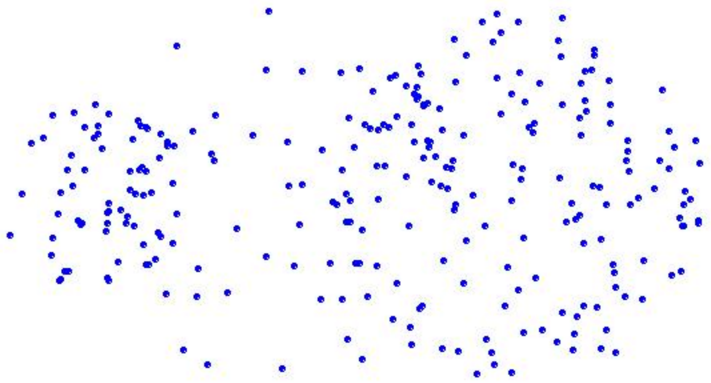


Nested Clusters

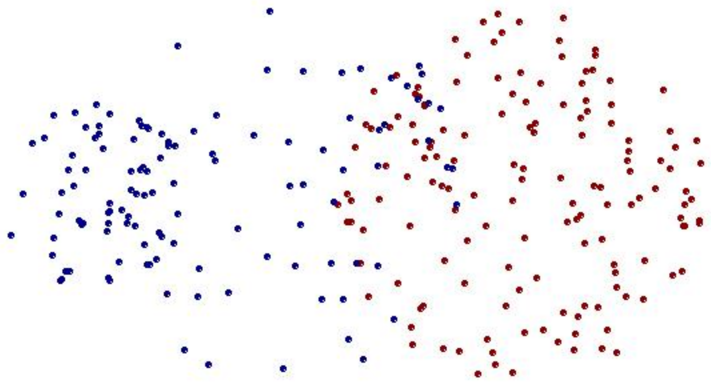


Dendrogram

Limitations of MIN



Original Points

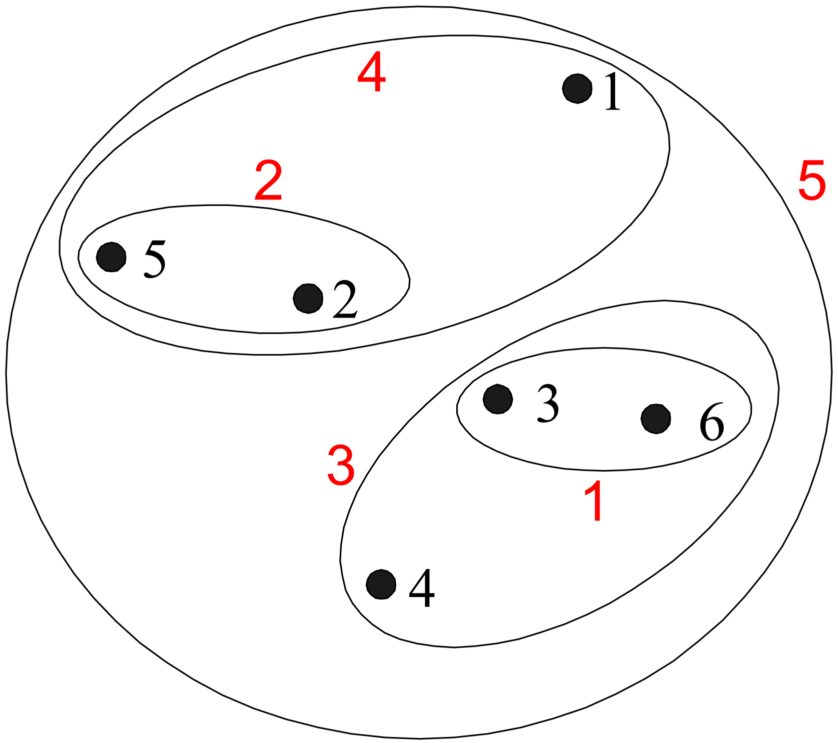


Two Clusters

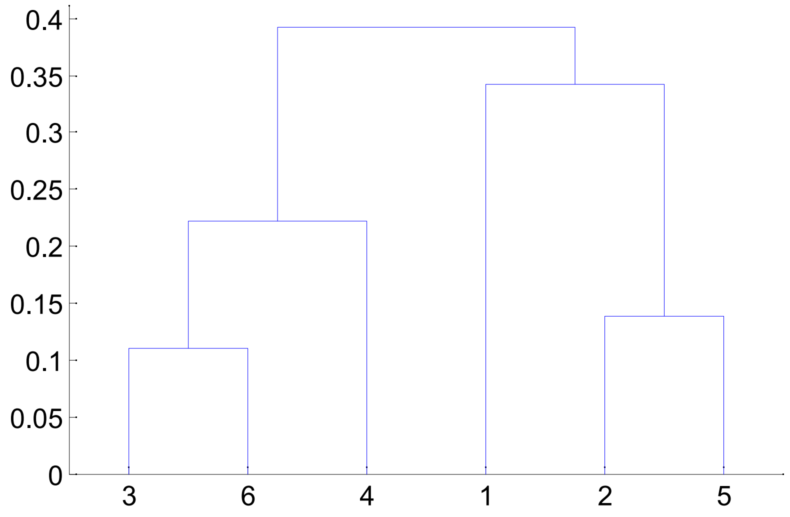
- Sensitive to noise and outliers

Cluster Similarity: MAX

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters
  - Determined by all pairs of points in the two clusters

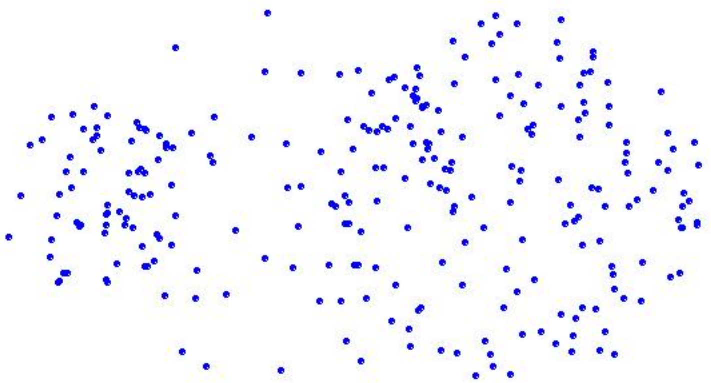


Nested Clusters

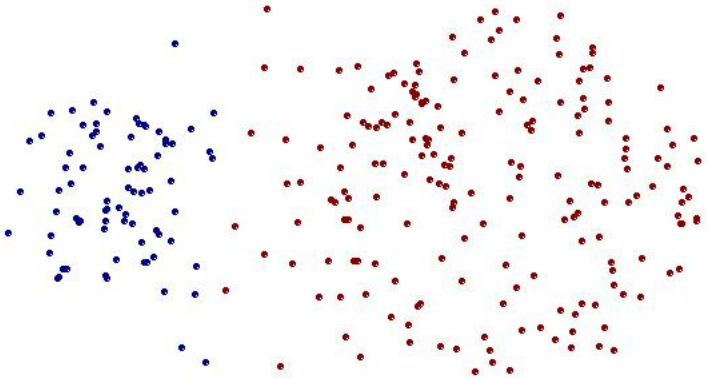


Dendrogram

Strength of MAX



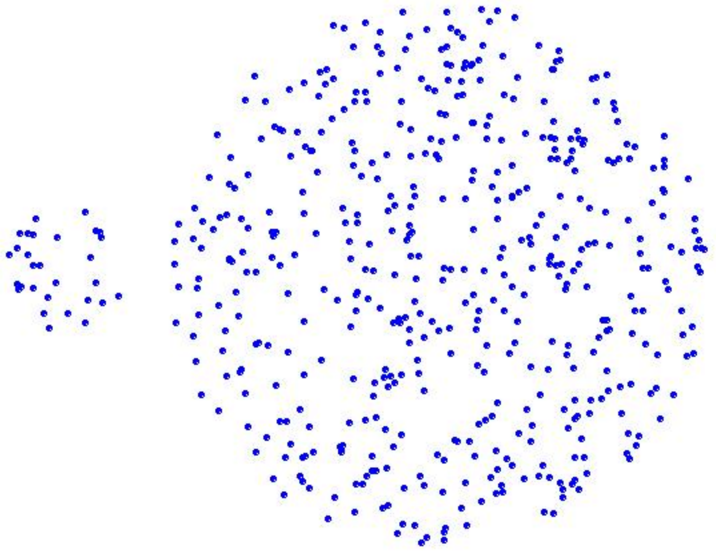
Original Points



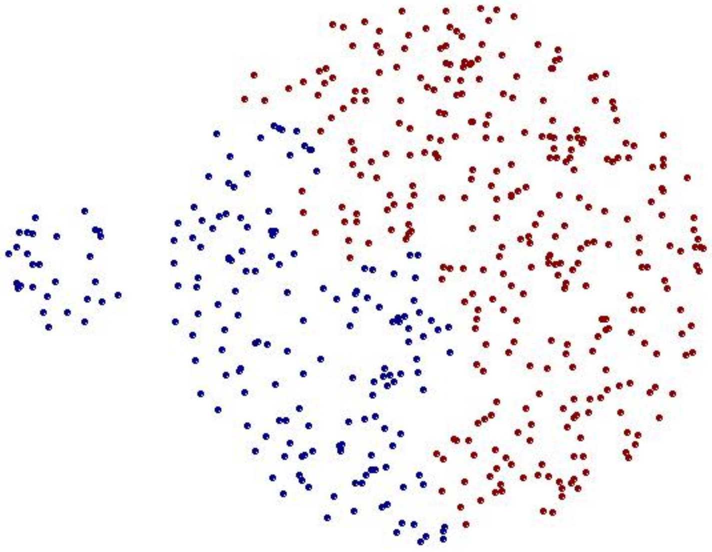
Two Clusters

- Less susceptible to noise and outliers

Limitations of MAX



Original Points



Two Clusters

- ❑Tends to break large clusters
- ❑Biased towards globular clusters

Cluster Similarity: Group Average

- Proximity of two clusters is the **average of pairwise proximity between points in the two clusters**:

$$proximity(Cluster_i, Cluster_j) = \frac{\sum_{p_i \in Cluster_i, p_j \in Cluster_j} proximity(p_i, p_j)}{|Cluster_i| * |Cluster_j|}$$

- Limitations: Scalability is a problem

Hierarchical Clustering in Non-Euclidean Case?

- **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

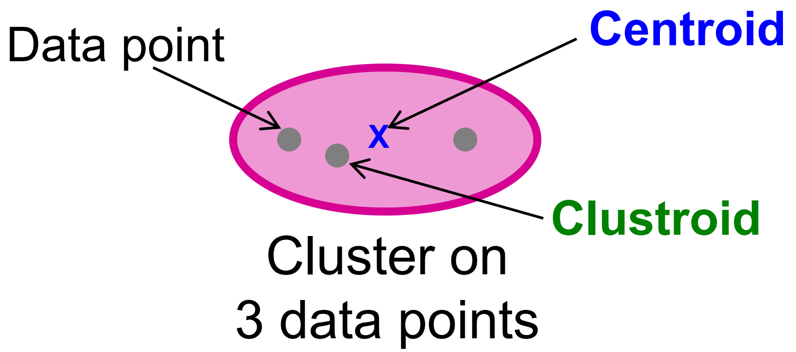
- **Approach 1:**
  - **(1) How to represent a cluster of many points?** ***Clustroid*** **(簇中心点)** = (data) point "***closest***" to other points
  - **(2) How do you determine the "nearness" of clusters?** Treat clustroid as if it were centroid, when computing inter-cluster distances
  - **(3) When to stop combining clusters?** Similar to previous

"Closest" Point in Non-Euclidean Case

- **(1) How to represent a cluster of many points?**

  Clustroid* (簇中心点)** = point "closest***" to other points

  Possible meanings of "closest":
  - **Approach 1:** Smallest maximum distance to other points (最大值)
  - **Approach 2:** Smallest average distance to other points (求和)
  - **Approach 3:** Smallest sum of squares of distances to other points (平方和)
    - 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.

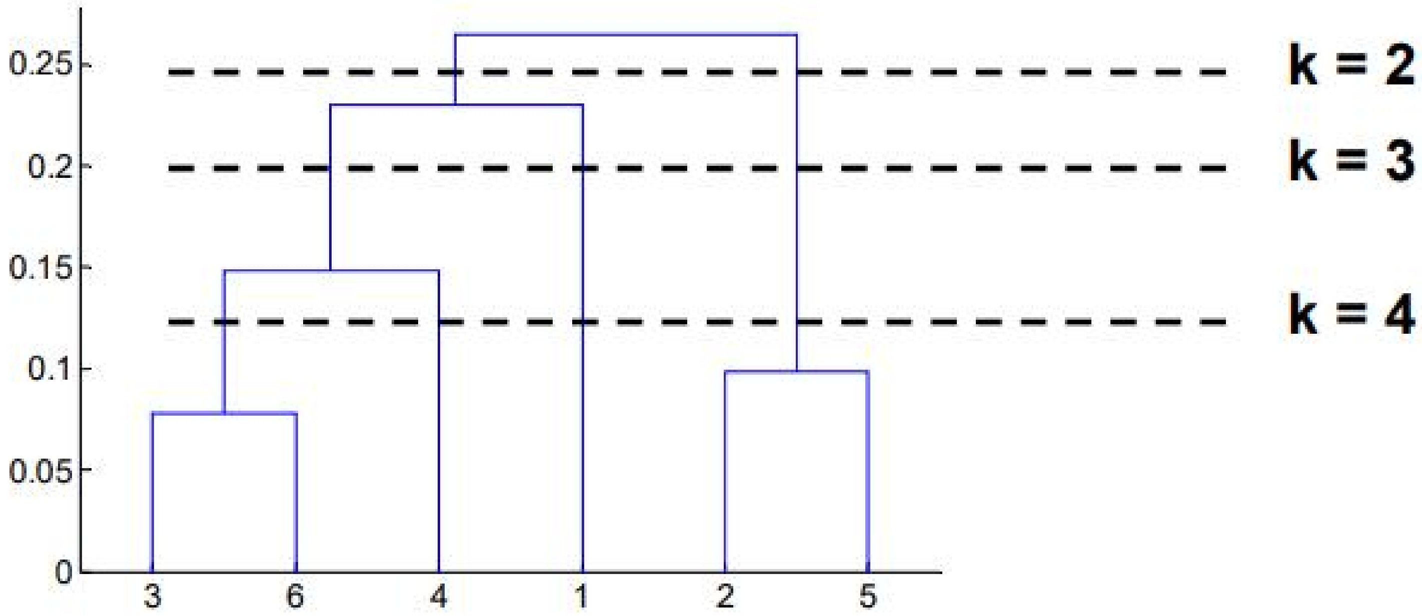
Defining Nearness in Non-Euclidean Case

华中科技大学 计算机科学与技术学院 School of Computer Science &Technology, HUST

- **(2) How do you determine the "nearness" of clusters?**
  - **Approach 1:** Treat clustroid as if it were centroid, when computing inter-cluster distances
  - **Approach 2: Intercluster distance (两个簇的距离)** = minimum of the distances between any two points, one from each cluster (两个簇中所有点之间的最短距离)
  - **Approach 3:** Pick a notion of "**cohesion**" **(内聚力)** of clusters, *e.g.*, maximum distance from the clustroid. Merge clusters whose *union* is most cohesive
    - **Approach 3.1**: Use the diameter (簇的直径) of the merged cluster = maximum distance between points in the cluster
    - **Approach 3.2**: Use the average distance （簇的平均距离） between points in the cluster
    - **Approach 3.3**: Use a density-based （簇的密度） approach. Take the diameter or avg. distance, e.g., and divide by the number of points in the cluster

From Hierarchical to Partitional Clustering

- Cut tree at some height to get the desired number of partitions *k*



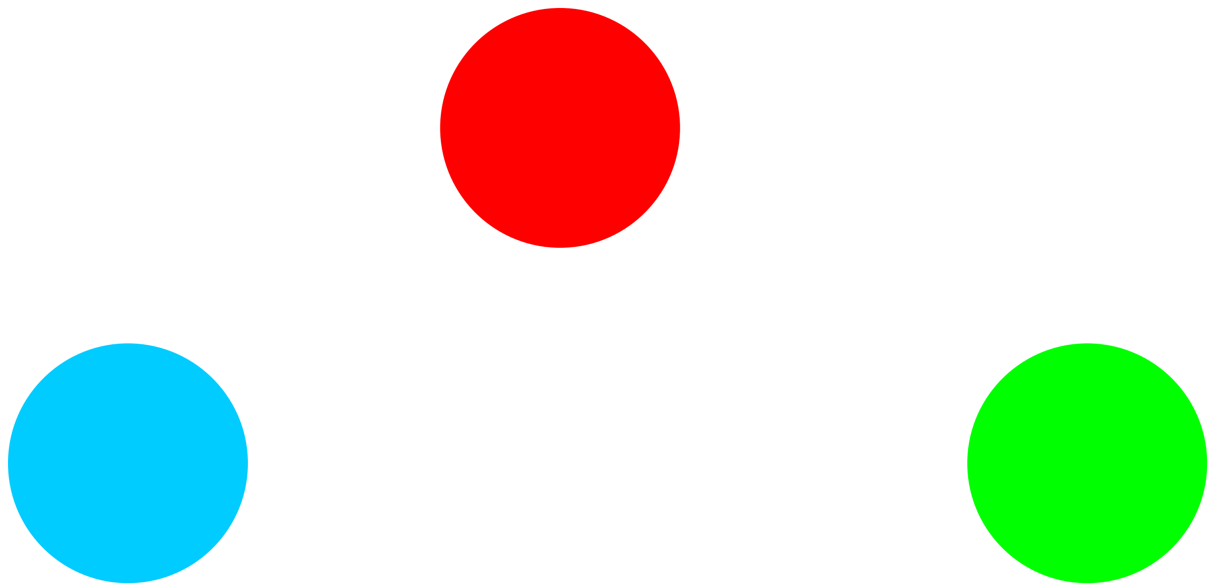
Hierarchical Clustering Overheads

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

- Careful implementation using priority queue can reduce time to O($N^2 \log N$)
  - **Still too expensive for really big datasets that do not fit in memory**

Types of Clusters: Well-Separated

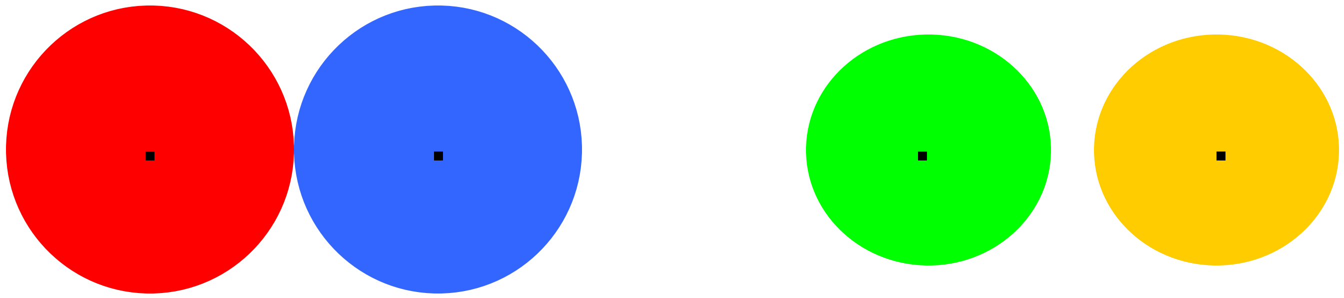
- Well-Separated Clusters (明显分离的簇):
  - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



3 well-separated clusters

Types of Clusters: Center-Based

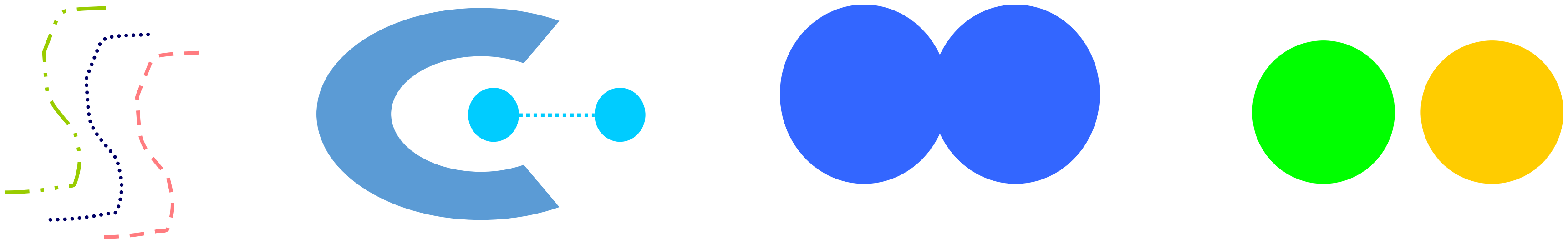
- Center-based Clusters (基于中心的簇):
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster
  - The center of a cluster is often a centroid (簇质心), the average of all the points in the cluster, the most "representative" point of a cluster



4 center-based clusters

Types of Clusters: Contiguity-Based

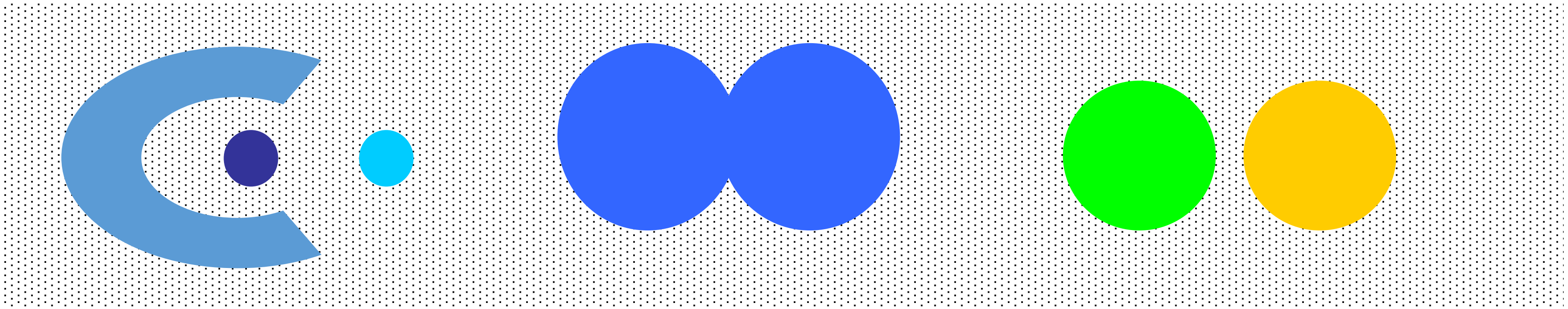
- Contiguous Clusters (基于邻近的簇, 或称基于图的簇, Nearest neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.



8 contiguous clusters

Types of Clusters: Density-Based

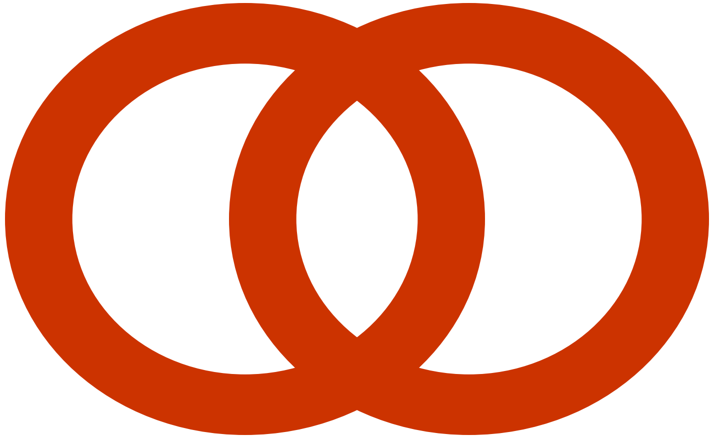
- Density-based Clusters (基于密度的簇):
  - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
  - Used when the clusters are irregular or intertwined, and when noise and outliers are present.



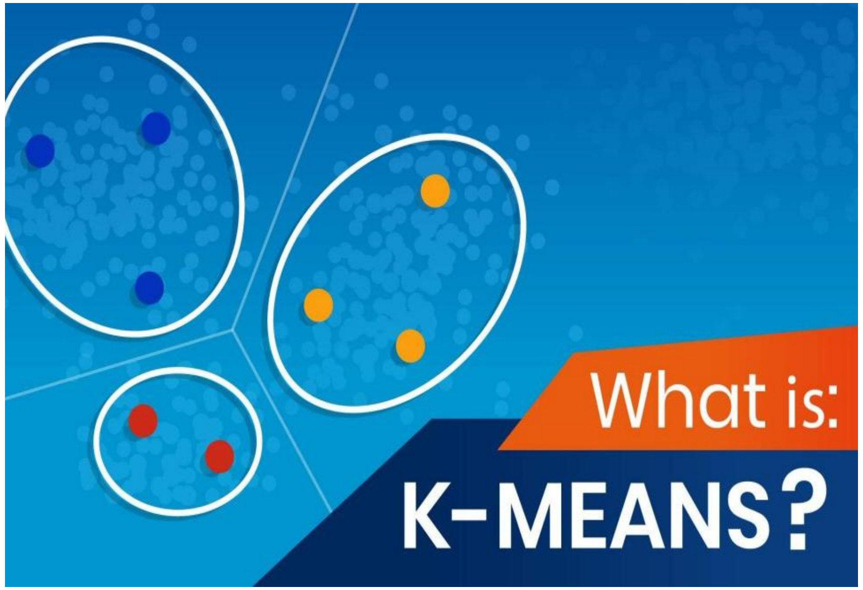
6 density-based clusters

Types of Clusters: Conceptual Clusters

- Shared Property or Conceptual Clusters (概念簇):
  - Finds clusters that share some common property or represent a particular concept.



2 Overlapping Circles



Section 5.3: k-means clustering

k–means Algorithm

- **k–means Algorithm** (K均值算法) assumes **Euclidean space/distance**

- Start by picking k, the number of clusters

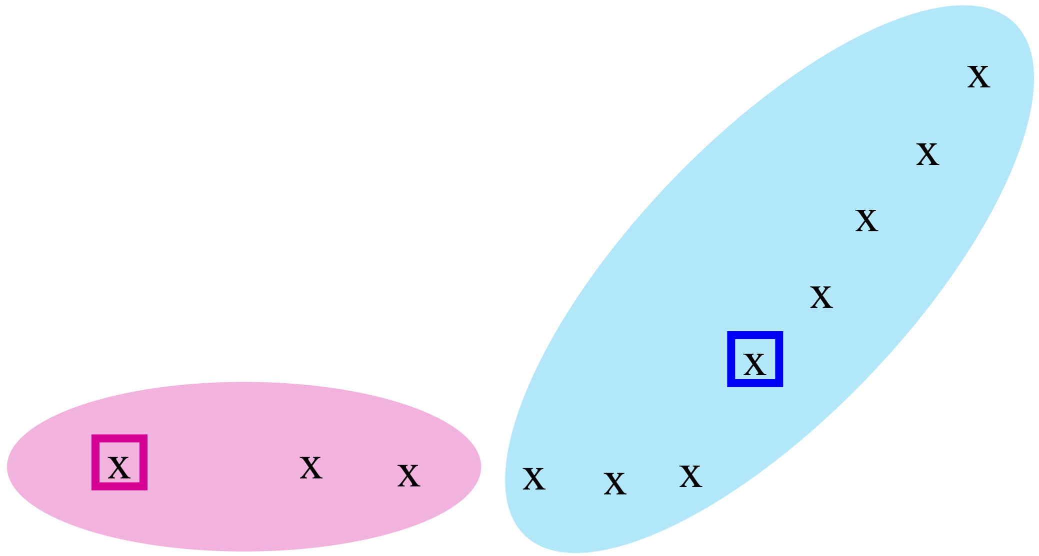
- Initialize clusters by picking one point per cluster
  - **Example:** Pick one point at random, then k**-1** other points, each as far away as possible from the previous points

Populating Clusters

- **1)** For each point, place it in the cluster whose current centroid (簇质心) it is nearest
- **2)** After all points are assigned, update the locations of centroids of the ***k*** clusters
- **3)** Reassign all points to their closest centroid
  - Sometimes moves points between clusters
- **Repeat 2 and 3 until convergence**
  - **Convergence:** Points don’t move between clusters and centroids stabilize

Example: Assigning Clusters

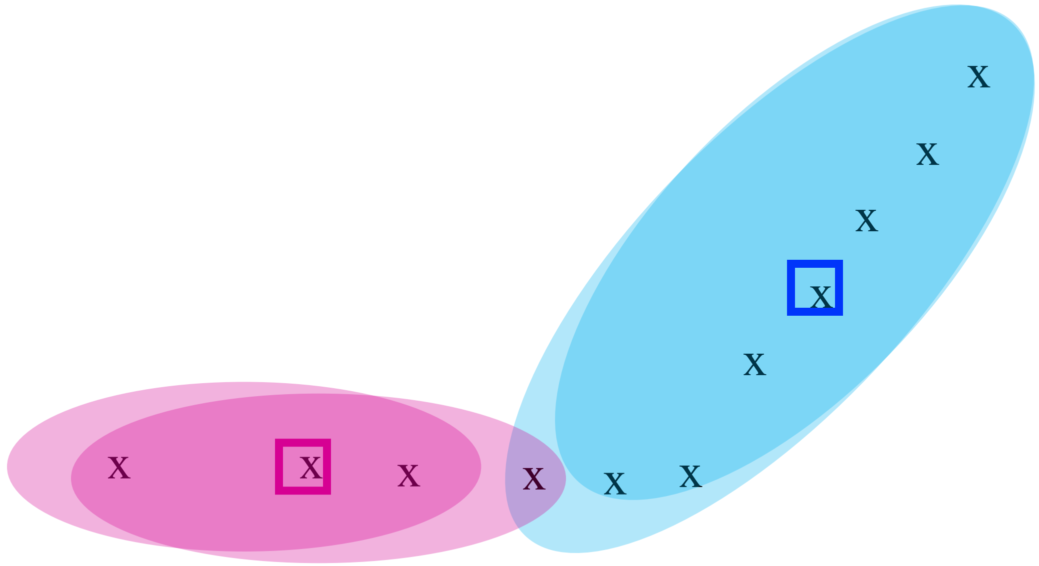
華中科技大學
计算机科学与技术学院
School of Computer Science &Technology，HUST



x … data point
… centroid

Clusters after round 1

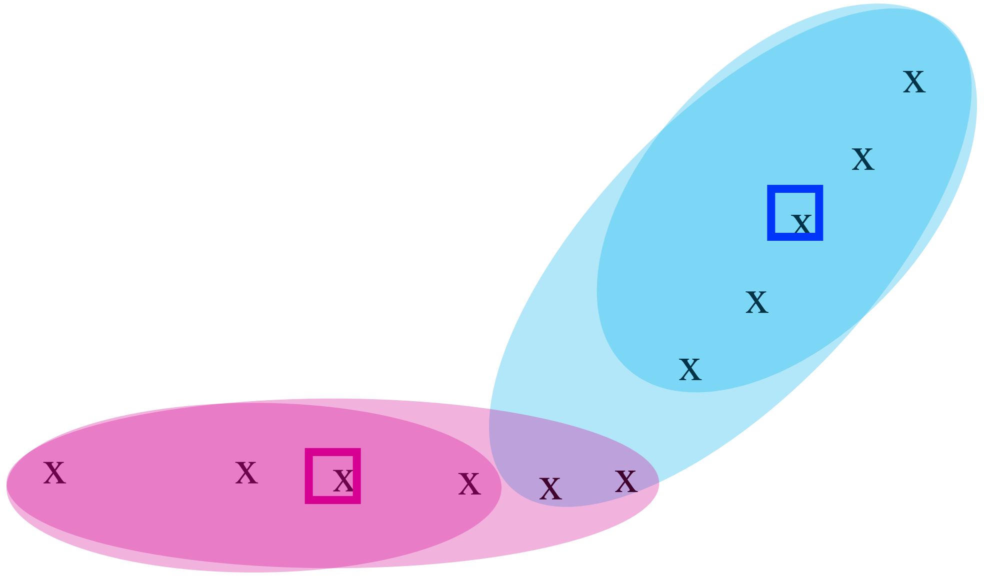
Example: Assigning Clusters



x … data point
□ … centroid

Clusters after round 2

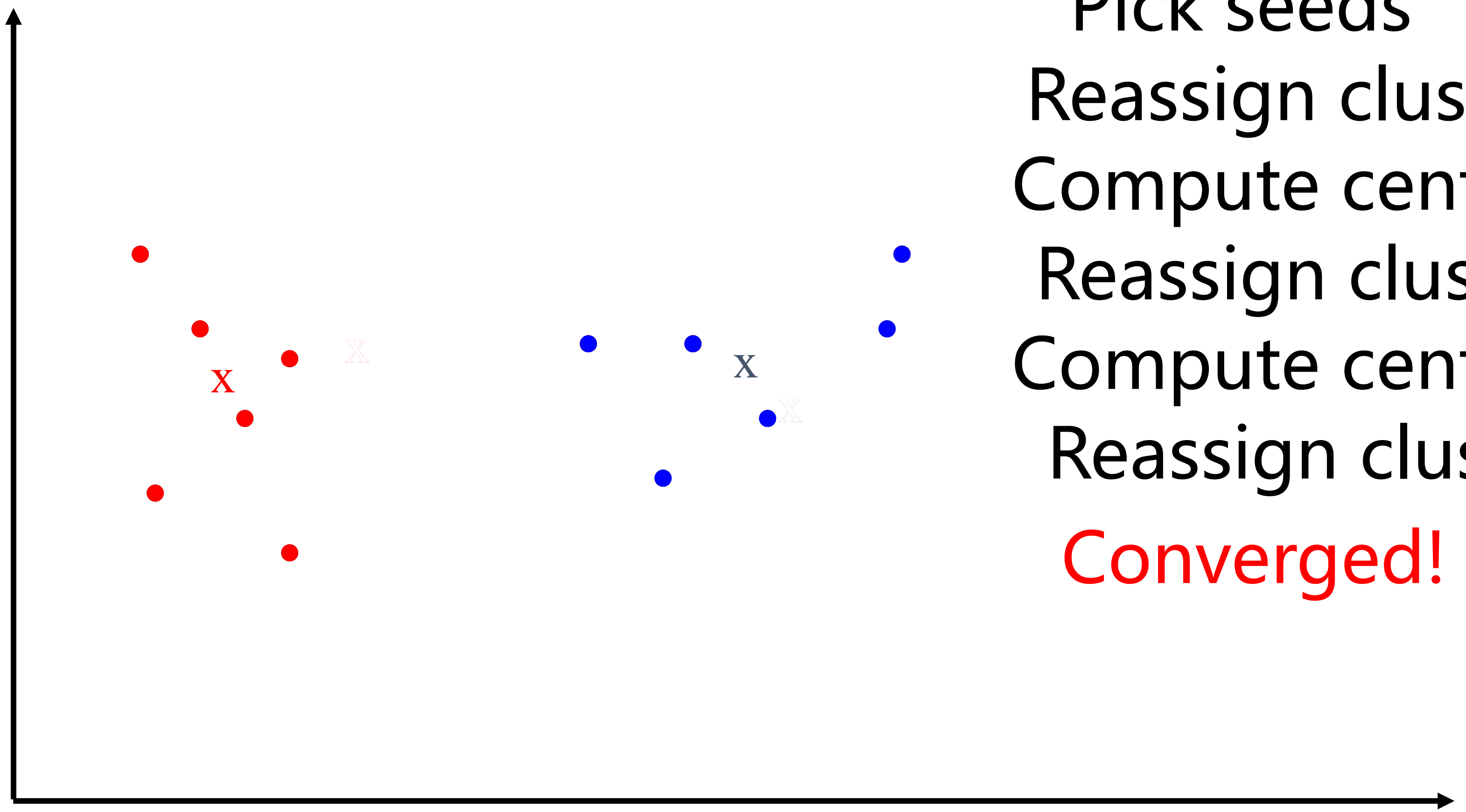
Example: Assigning Clusters



x … data point
□ … centroid

Clusters at the end

k-means Example (k=2)



Pick seeds
Reassign clusters
Compute centroids
Reassign clusters
Compute centroids
Reassign clusters

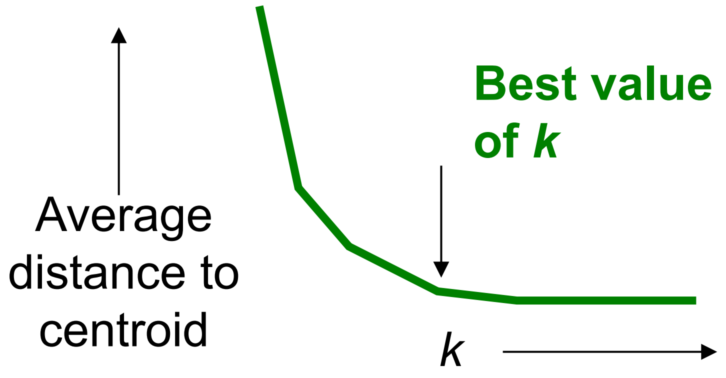
Converged!

Termination conditions

- Several possibilities termination conditions in k-means, e.g.,
  - A fixed number of iterations.
  - Point assignment unchanged.
  - Centroid positions don’t change.

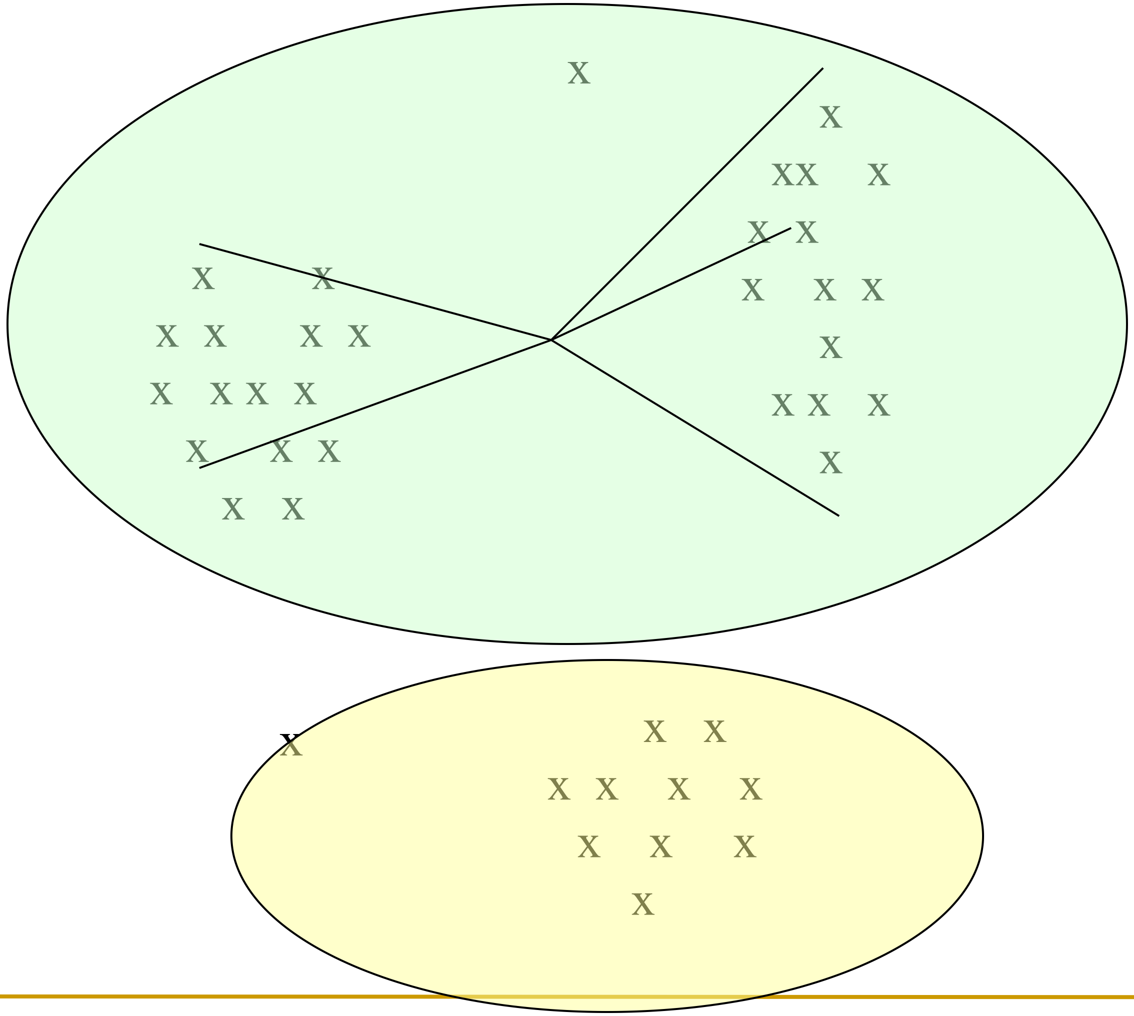
Getting the *k* right

- **How to select �?**
- 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



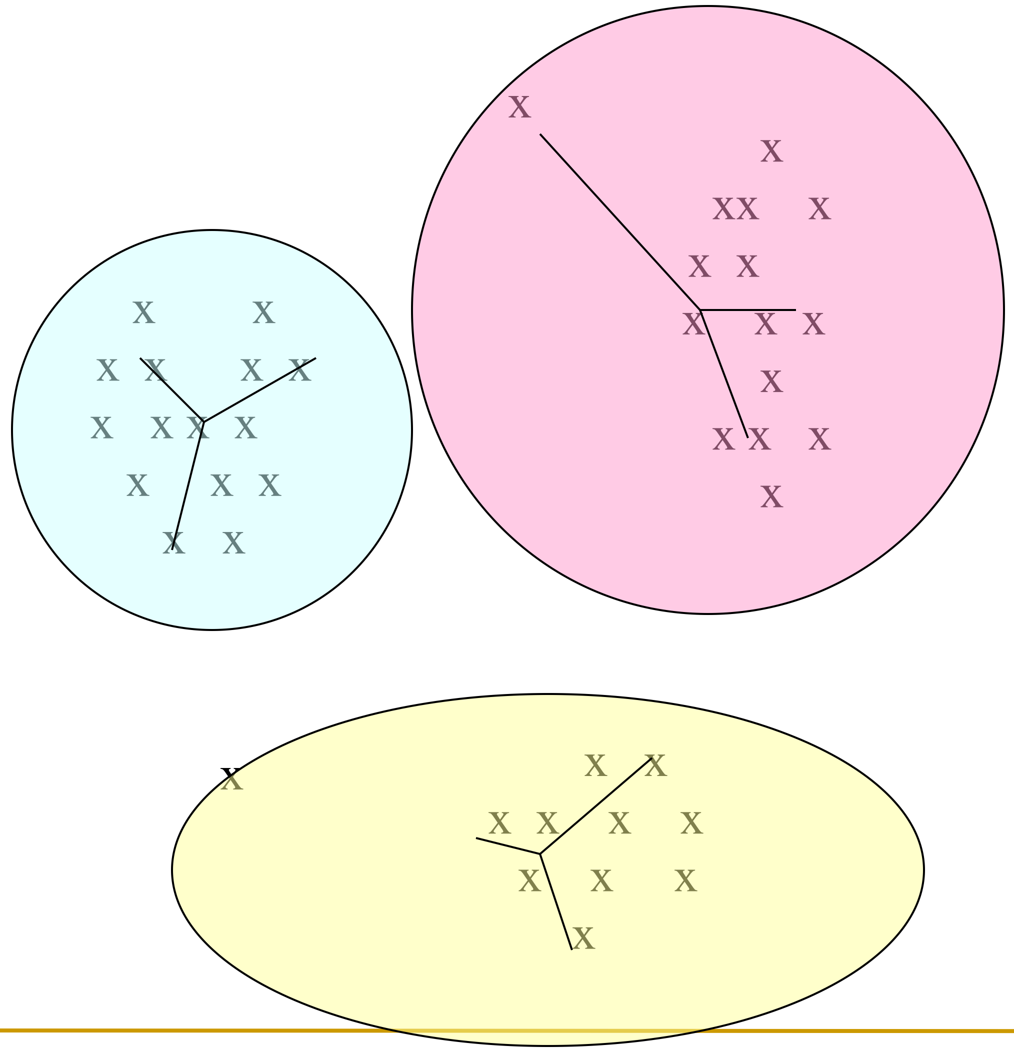
Example: Picking *k*

Too few;
many long
distances
to centroid.



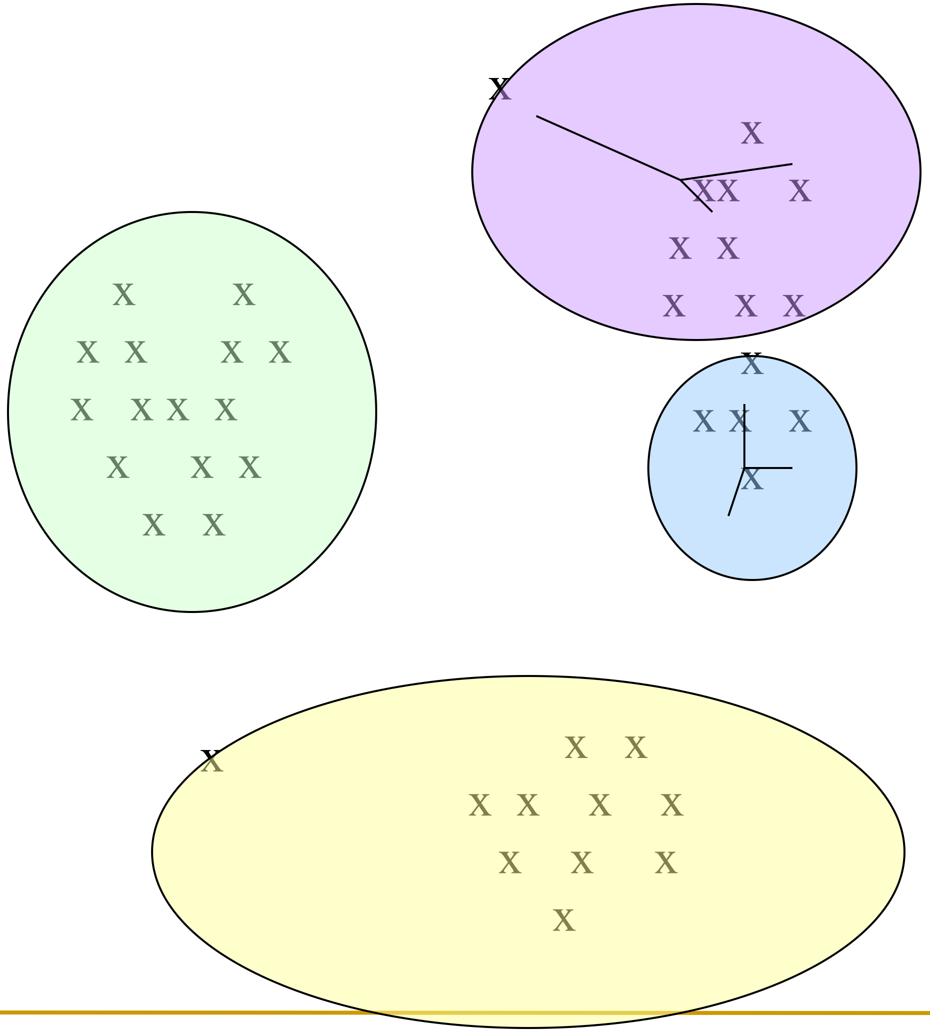
Example: Picking *k*

Just right;
distances
rather short.



Example: Picking *k*

Too many;
little improvement
in average
distance.



Picking the initial k point

Approach 1: pick random points

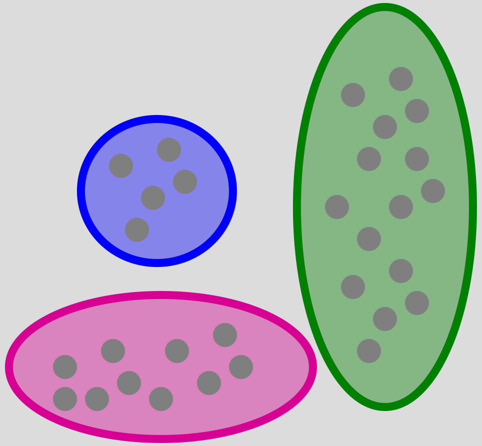
- Pick k random points

Approach 2: sampling

- Cluster a sample of the data using hierarchical clustering, to obtain k clusters
- Pick a point from each cluster (e.g., point closest to centroid)
- Sample fits in main memory

Approach 3: pick "dispersed" set of points

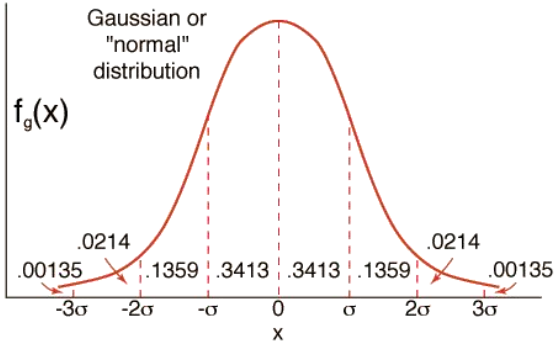
- Pick first point at random
- Pick the next point to be the one whose minimum distance from the selected points is as large as possible
- Repeat until we have k points



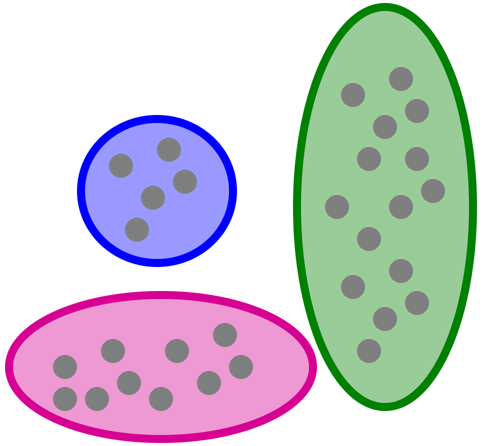
Section 5.4: BFR Algorithm

Extension of k-means to large data

BFR Algorithm



- **BFR** [Bradley-Fayyad-Reina] **(BFR算法)** is a variant of *k*-means designed to handle **very large** (disk-resident) data sets
- **Assumes** that clusters are normally distributed around a centroid in a Euclidean space
  - Standard deviations in different dimensions may vary
    - Clusters are axis-aligned ellipses



- **Efficient way to summarize clusters** (want memory required O(clusters) and not O(data))

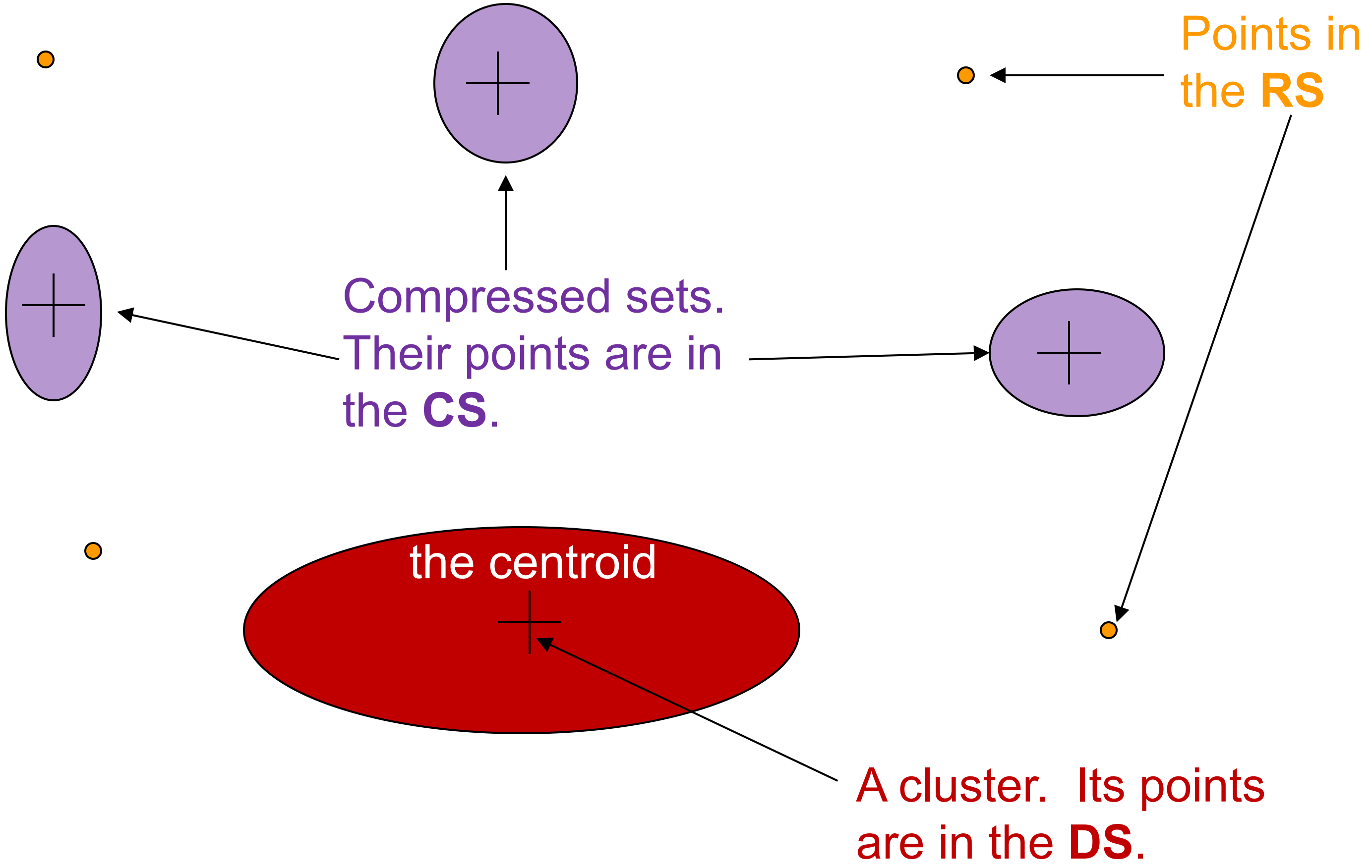
BFR Algorithm

- Points are read from disk one main-memory-full at a time
- **Most points from previous memory loads are summarized by simple statistics**
- To begin, from the initial load we select the initial k centroids by some sensible approaches, e.g.,:
  - Take k random points
  - Take a small random sample and cluster optimally
  - Take a sample; pick a random point, and then k–1 more points, each as far from the previously selected points as possible

Three Classes of Points

- **In BFR, 3 sets of points which we keep track of:**
- **Discard set (DS, 废弃集):**
  - Points close enough to a centroid to be summarized
- **Compression set (CS, 压缩集):**
  - Groups of points that are close together but not close to any existing centroid
  - These points are summarized, but not assigned to a cluster
- **Retained set (RS, 留存集):**
  - Isolated points waiting to be assigned to a compression set

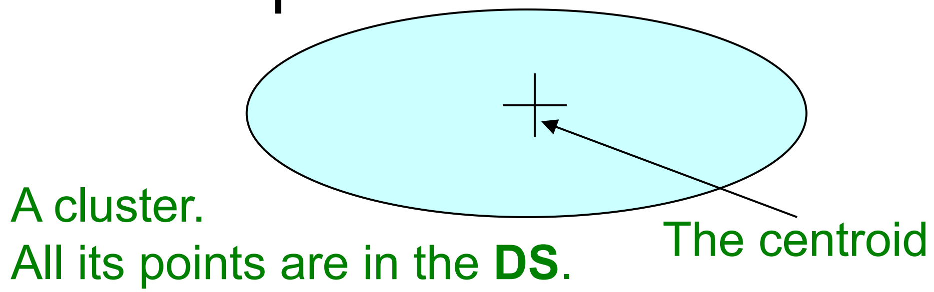
Example



Discard set (DS, 废弃集): Close enough to a centroid to be summarized
Compression set (CS, 压缩集): Summarized, but not assigned to a cluster
Retained set (RS, 留存集): Isolated points

Summarizing Sets of Points

- **For each cluster, the discard set (DS, 废弃集) is summarized by:**
  - The number of points, ***N***
  - The vector ***SUM***, whose i^{th} component is the sum of the coordinates of the points in the i^{th} dimension (i维度上的分量之和)
  - The vector ***SUMSQ***, i^{th} component = sum of squares of coordinates in i^{th} dimension (i维度上的分量平方和)
  - 简称***N-SUM-SUMSQ***方法



- **Similar for the compression set (CS, 压缩集)**

Example

❑Example for N-SUM-SUMSQ:

x (5,1)

x (7,0)

x (6,-2)

- *N*=3
- *SUM*=[18,-1]
- *SUMSQ*=[110,5]

Summarizing Points: Comments

- **2*d* + 1** values represent any size cluster
  - ***d* = number of dimensions**
- Average in **each dimension** (**the centroid, 簇质心**很容易被计算出来) can be calculated as $\mathbf{SUM}_i / N$
  - $\mathbf{SUM}_i$ = i^{th} component of SUM
- Variance (方差) of a cluster in dimension i is: $(\mathbf{SUMSQ}_i / N) - (\mathbf{SUM}_i / N)^2$
  - And **standard deviation (标准差)** is the square root of that
- 综上, 通过***N-SUM-SUMSQ***方法很容易得到簇内数目, 簇质心, 每个维度的标准差

Example

华中科技大学 计算机科学与技术学院 School of Computer Science &Technology, HUST

- **已知N-SUM-SUMSQ, 求该簇的数目, 簇质心和每个维度的标准差分别是多少。**

  N=3

  SUM=[18,-1]

  SUMSQ=[110,5]

  x (5,1)

  x (7,0)

  x (6,-2)

  - 解: 簇的点个数为3, 簇质心以及每个维度的标准差:
  - Centroid(簇质心): SUM/N=[6,-1/3]
  - Variance(方差) for i=1, ($SUMSQ_i/N$) – (SUM_i/N)^2=110/3-(18/3)^2=0.667, standard deviation (标准差)=$\sqrt{0.667} = 0.816$
  - Variance for i=2, ($SUMSQ_i/N$) – (SUM_i/N)^2=5/3-(-1/3)^2=1.56, standard deviation (标准差)=$\sqrt{1.56} = 1.25$

- **Next step: Actual clustering**