Clustering

Rafael Ferreira da Silva
rafsilva@isi.edu
http://rafaelsilva.com

https://xkcd.com/1781/

Clustering: Examples

- Examples

```
docker run -p 8888:8888 jupyter/scipy-notebook:2c80cf3537ca
```

- Use files:

```
Hierarchical-clustering-example.ipynb
shopping_data.csv
```

```
K-Means-clustering-example.ipynb
```

```
Clustering-comparison.ipynb
clusterable_data.npy
```

Roadmap

- Problem, types, and distance functions

- Hierarchical clustering

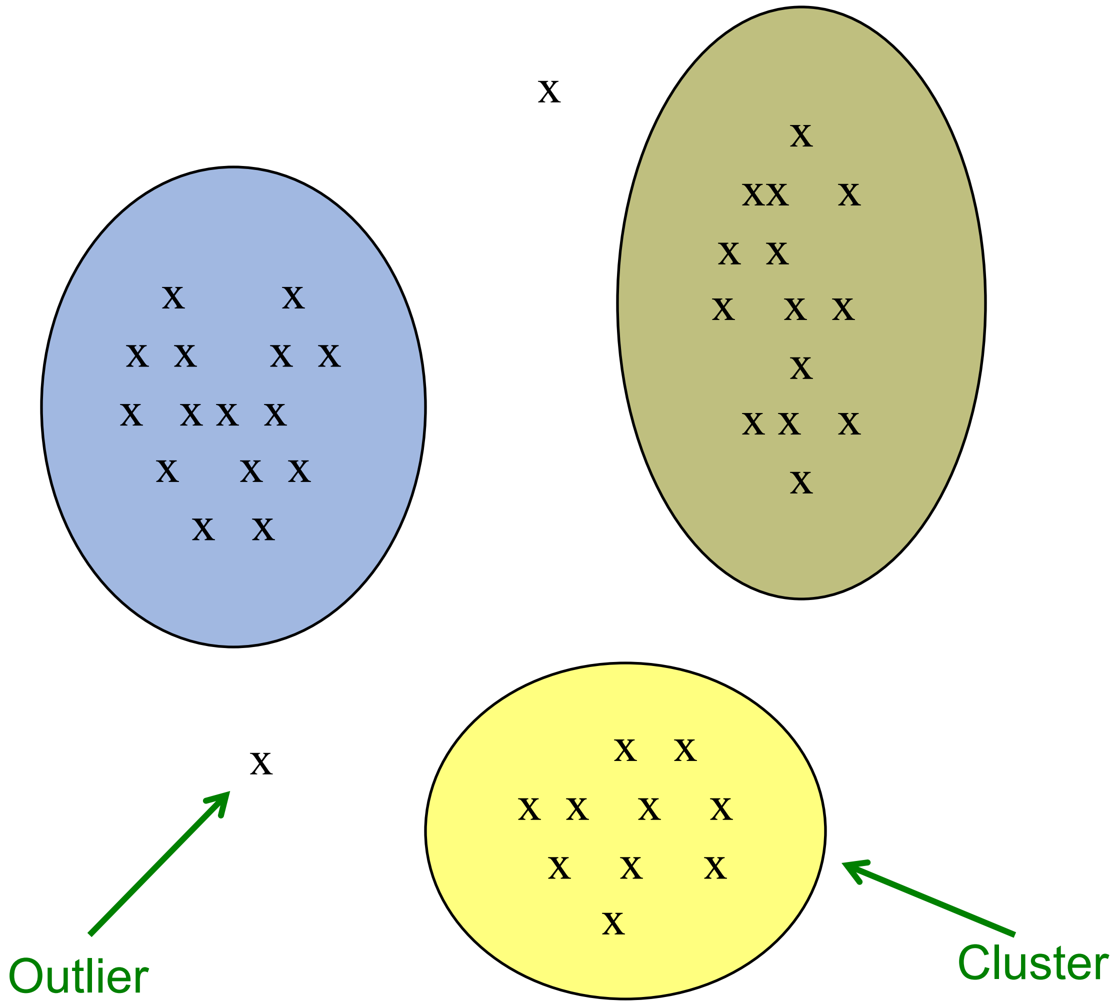
- Point assignment
  - K-means
  - BFR: extend k-means to handle large data set
  - CURE

- Curse of dimensionality

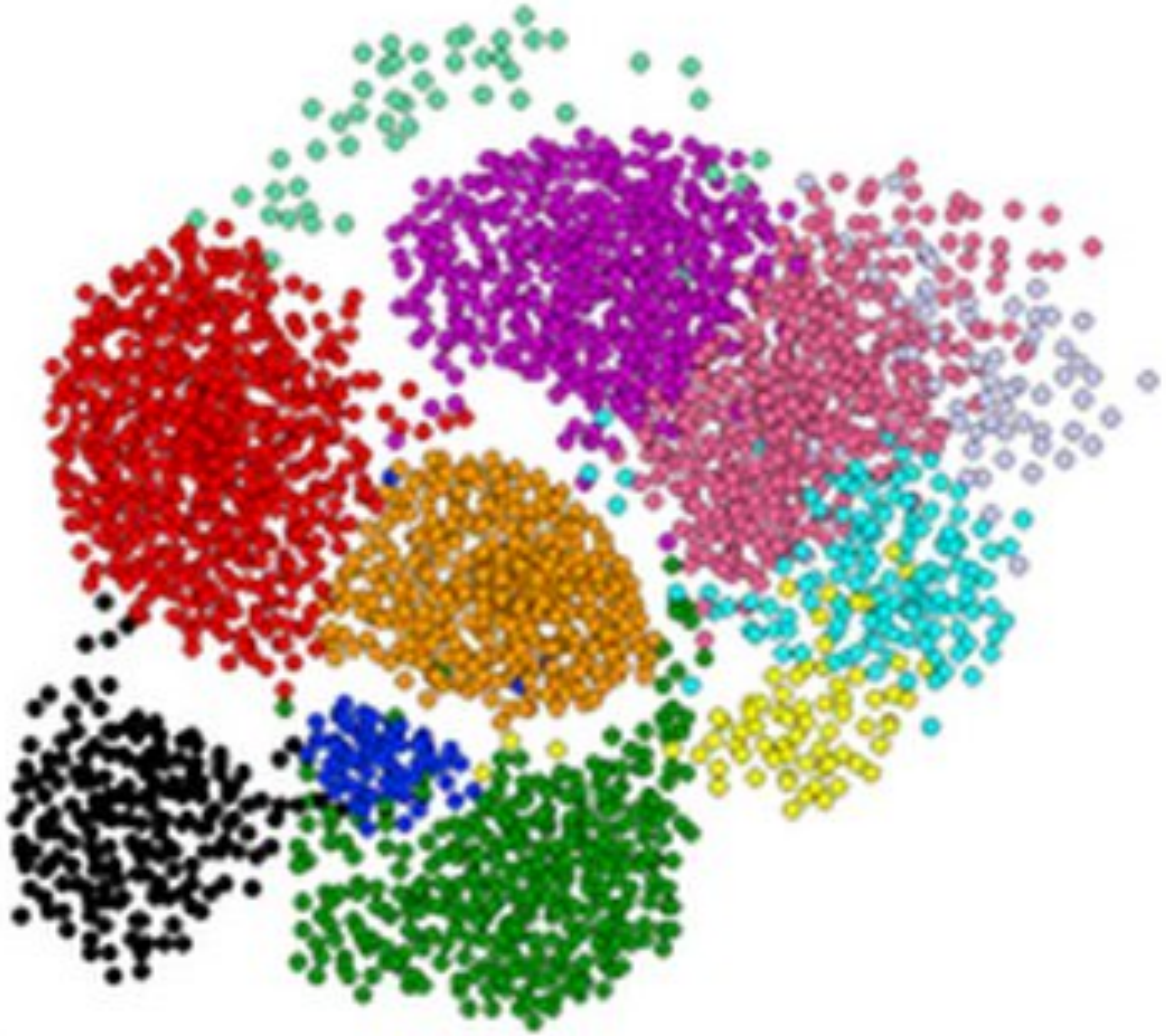
Clustering Problem

- Given a set of objects and a distance function
- Find groups/clusters of objects
- Desired properties:
  - Objects in the same group are close to each other
  - Objects in different groups are far away from each other

Example



Clustering can be hard



Clustering Stars

- Each represented by 7-dimensional point
  - Dimension = frequency band
  - Point = radiation signature

USC Viterbi School of Engineering Information Sciences Institute

Clustering Music CDs

- CD represented by a set of its buyers
  - Similar CD’s have similar buyers
- Use LSH in clustering large sets
  - Use LSH to efficiently find similar sets
  - Compute pairwise similarities of sets
  - Use the similarities in clustering (e.g., hierarchical)
- Advantage:
  - avoid computing similarity of dissimilar sets

Clustering Documents

- Document D represented as a word vector
  - $(w_1, w_2, \ldots, w_k)$, where $w_i = 1$ if it appears in D
- Measure similarity of document D_1 and D_2
  - Cosine(D_1, D_2)
- Similar documents likely on same topic

Types of Algorithms

- Hierarchical vs. point assignment
  - Hierarchical: Bottom-up iterative merging of clusters to form a multi-level clustering
  - Point assignment or partitional: one-level

- Euclidean or non-Euclidean
  - Cluster center/centroid makes sense only in Euclidean

- In-memory or not
  - In-memory: entire data can fit in main memory

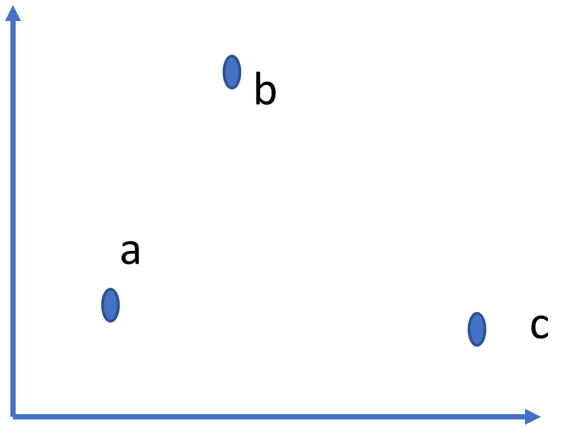
Varied Distance Functions

Distance function	Type of objects
Euclidean	Points in Euclidean space
Cosine	Vectors
Jaccard	Sets
Edit distance	Strings
Hamming distance	Bit vectors

Euclidean Distance

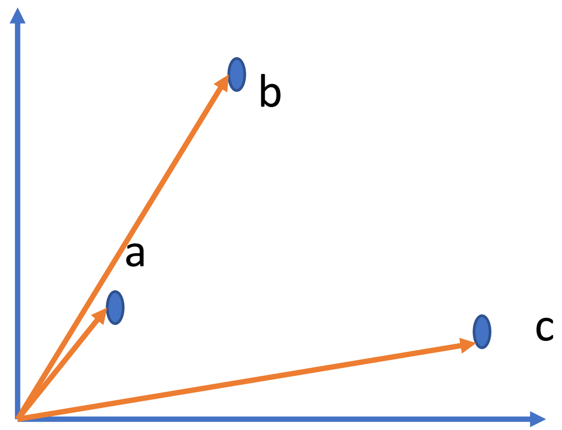
- Measures distance of two points in Euclidean space

$$d([x_1, x_2, \ldots, x_n],\ [y_1, y_2, \ldots, y_n]) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$



Cosine Distance

- Similarity = Cosine of angle btw vectors: A & B
- distance = 1- Cosine(A, B)



$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

USC Viterbi School of Engineering
Information Sciences Institute

Edit Distance

- For string data, distance btw x and y =
  - # of insertions or deletions of characters that turn x into y

- x = abcde, y = acfdeg
  - Edit(x, y) = 3
  - Delete b
  - Insert f after c
  - Insert g after e

USC Viterbi School of Engineering Information Sciences Institute

Hamming Distance

- For two bit vectors, distance btw x and y =
    - # of corresponding bits that differ

- x = 10101, y = 11110
    - Hamming(x, y) = 3

Roadmap

- Problem, types and distance functions
- Hierarchical clustering
- Point assignment
  - K-means
  - BFR
  - CURE
- Curse of dimensionality

Hierarchical Clustering

- Initially, a point is in a cluster by itself

How to pick and combine efficiently?

When to stop?

```
WHILE it is not time to stop DO
    pick the best two clusters to merge;
    combine those two clusters into one cluster;
END;
```

How to measure cluster distance?

USC Viterbi School of Engineering Information Sciences Institute

Centroid-Based Distance

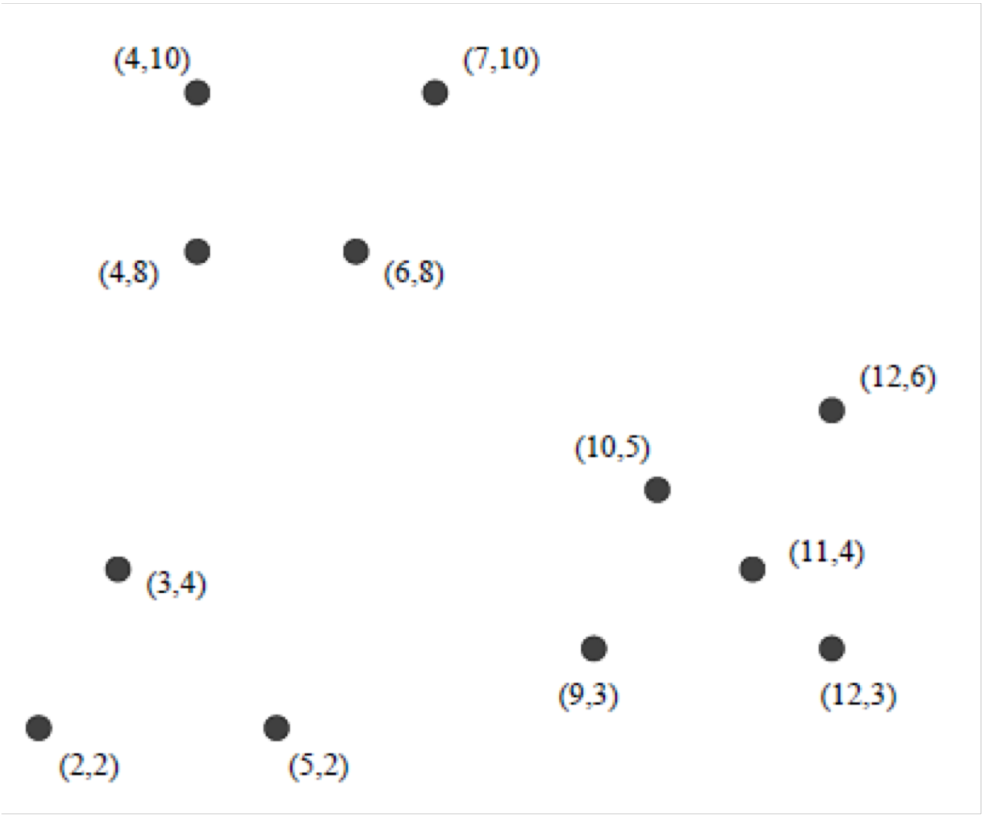
- Assume Euclidean space

- dist(C1, C2) = distance of their centroids
  - Coordinate of centroid = avg of that of all points in the cluster

- C1: {(1, 2), (2, 2)}
  - Centroid = (1.5, 2)

Example

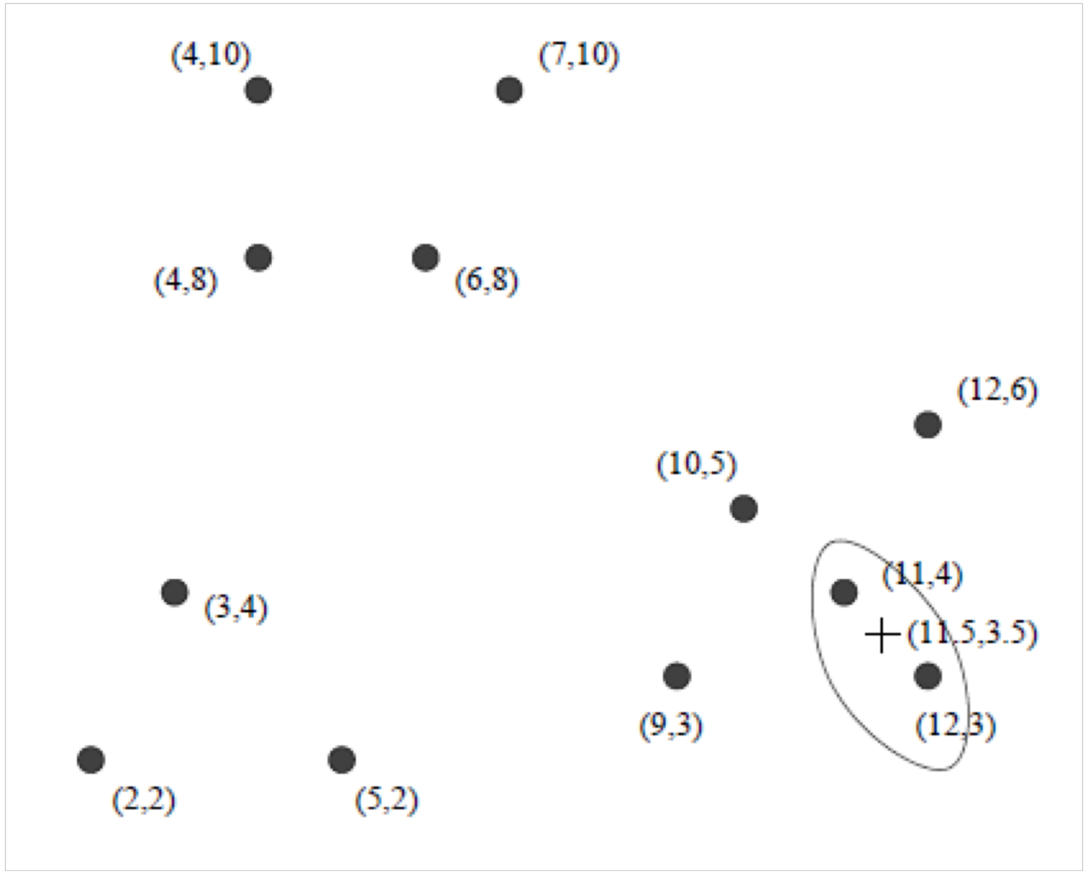
- Which two points are closest?
  - What is their distance?



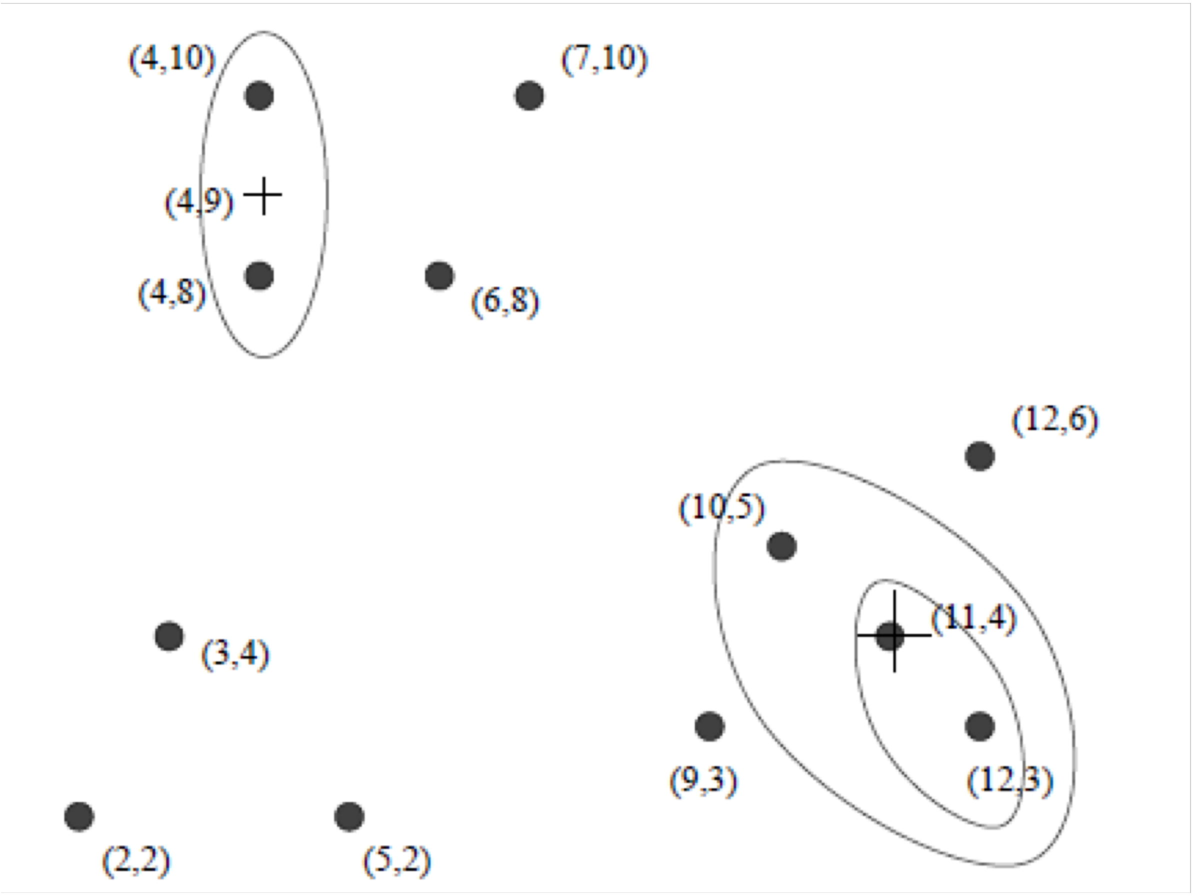
USC Viterbi
School of Engineering
Information Sciences Institute

First merge, compute centroid

- Which two clusters to be merged next?

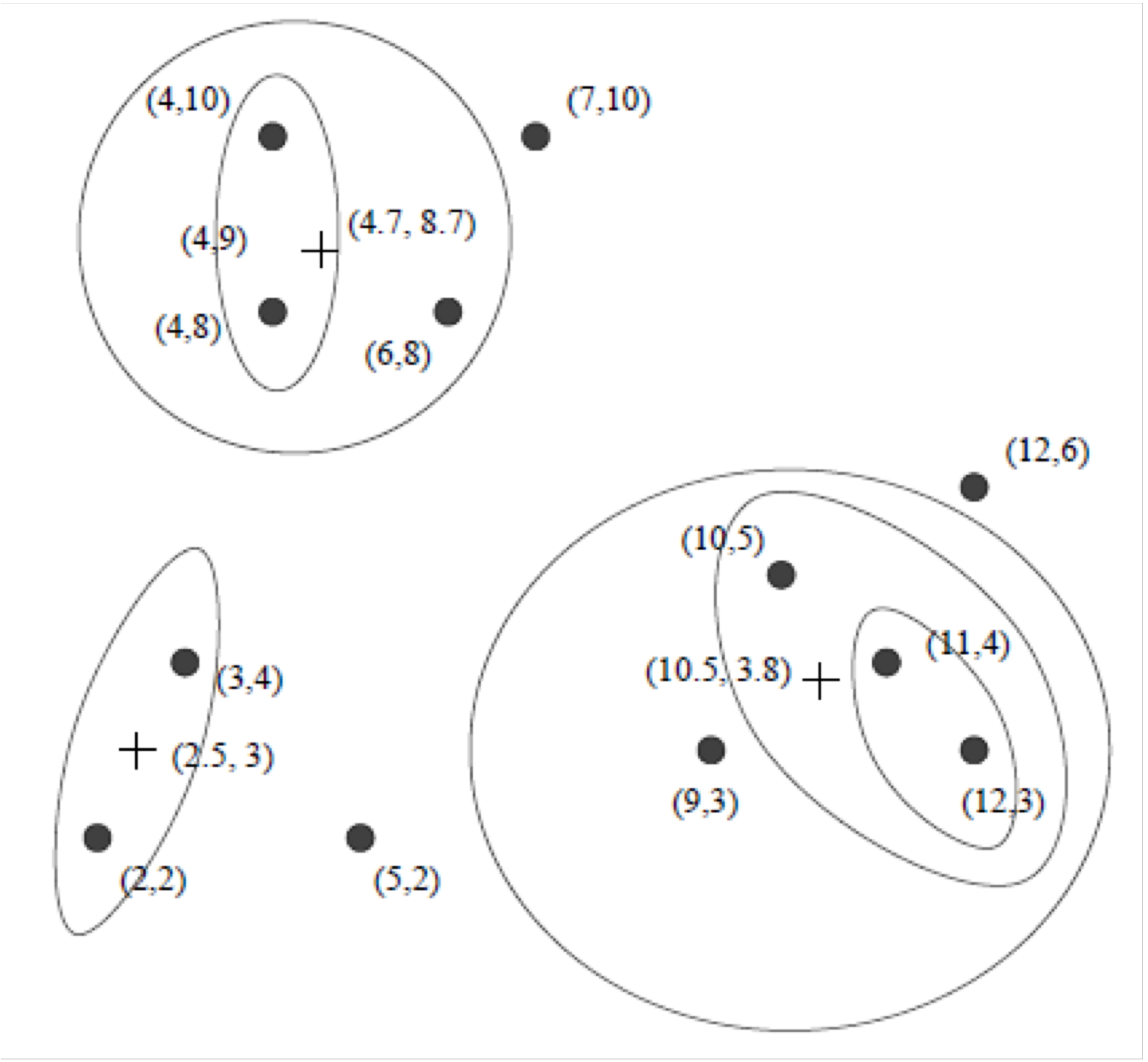


After two more merges



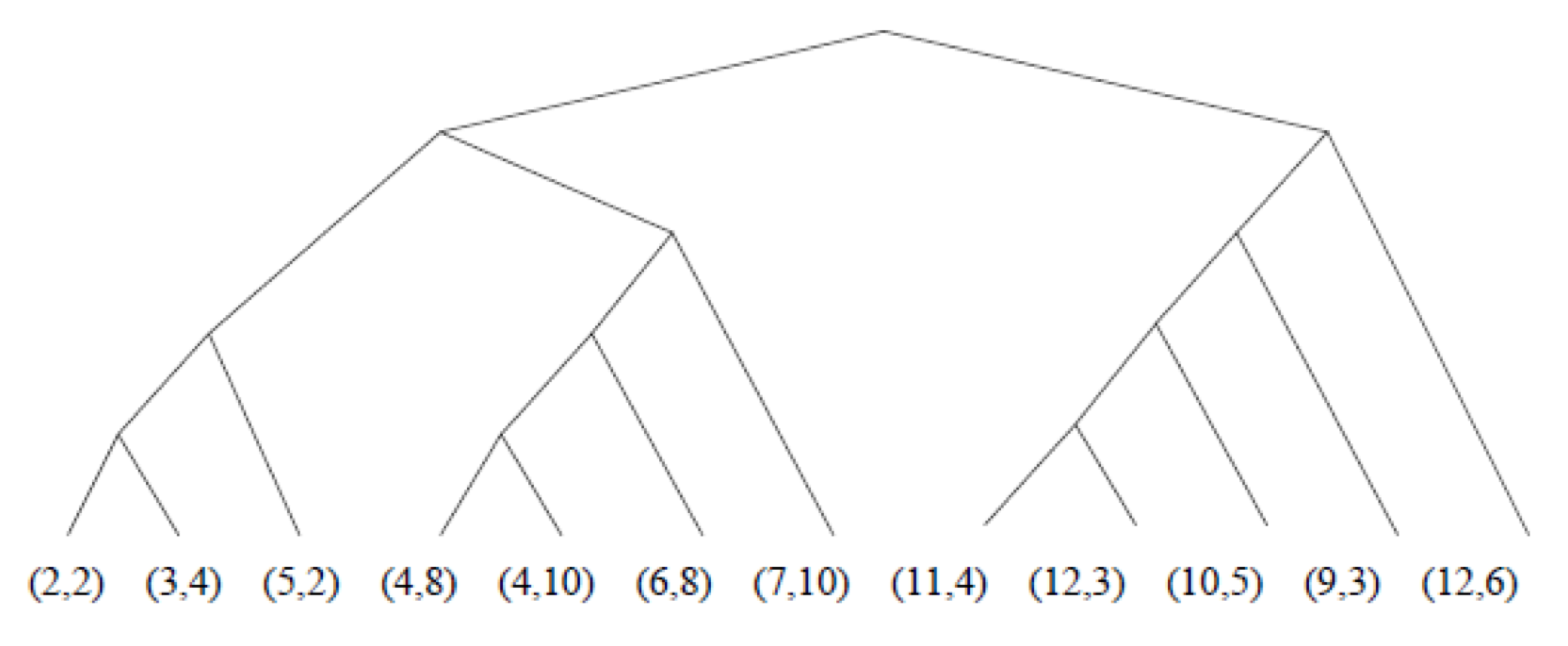
USC Viterbi
School of Engineering
Information Sciences Institute

After 3 more merges



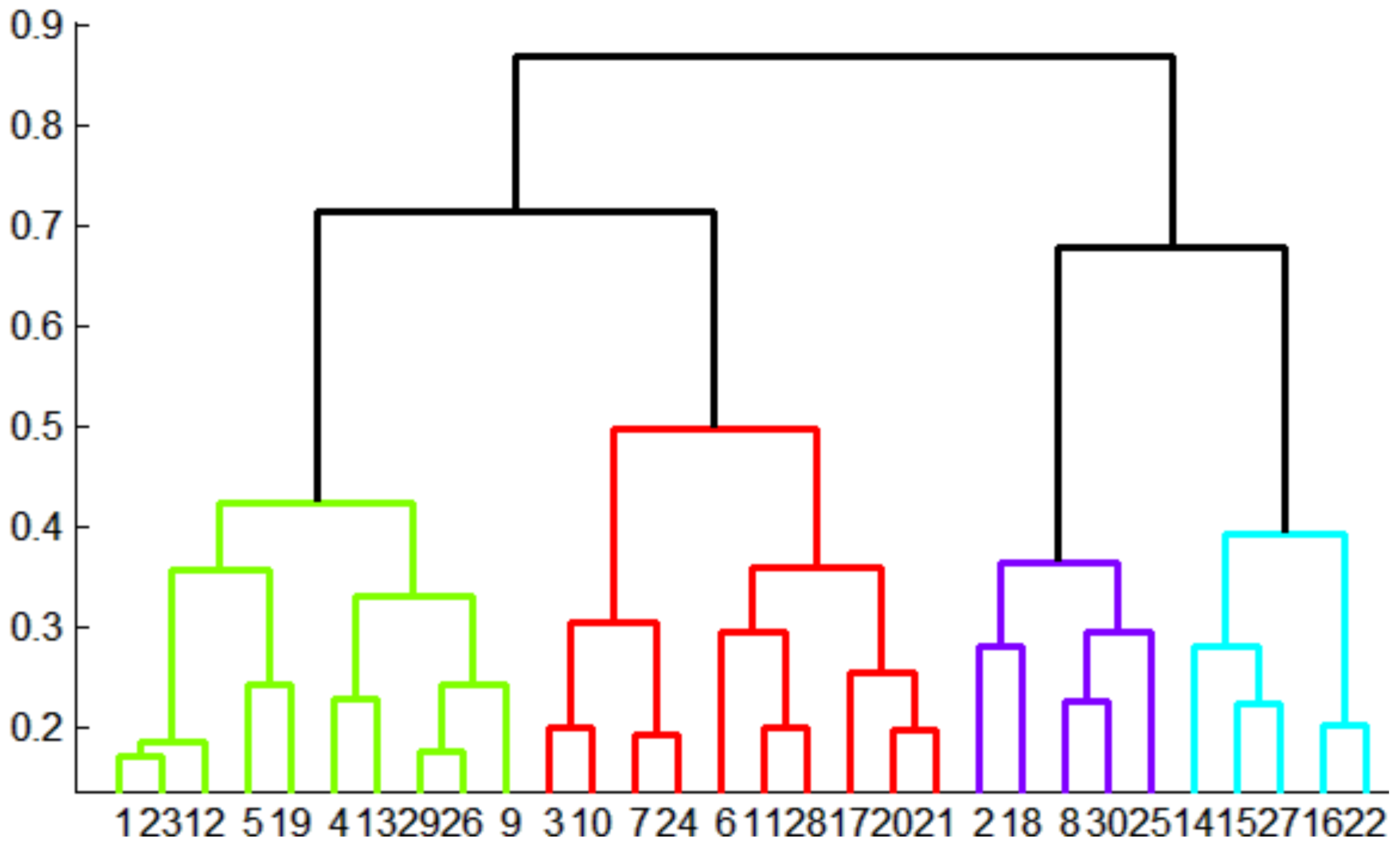
Final Result

- Represented as a tree



Dendrogram

- Can obtain k clusters from result for desired k
  - k can be any value between 1 and n



Complexity of Hierarchical Clustering

- n data points
- At most n – 1 step of merging

- Naive implementation, e.g., storing pairwise cluster distances in a matrix

	C1	**C2**	**C3**	**C4**
C1	0	2	3	2
C2		0	4	5
C3			0	3
C4				0

Complexity of Naive Implementation

- Initially, $O(n^2)$ for creating matrix and finding pair with minimum distance

- Subsequent merge, assuming matrix: k x k
  - Delete columns for old clusters: $O(k)$
  - Add new column for new cluster C': $O(k)$
  - Compute dist. of C' with other clusters: $O(k)$
  - Find new pair of clusters with min. dist: $O(k^2)$

=> Overall complexity: $O(n^3)$

Improved Version

- Use priority queue (e.g., heap-based) instead of matrix

1. Compute pairwise dist. of all points: $O(n^2)$
2. Build priority queue (time linear to size of queue), so: $O(n^2)$
3. Each merge:
   a) Remove entries for old clusters: $2n * O(\log(n))$
   b) Add entries for new cluster: $n * O(\log(n))$

=> Overall complexity: $O(n^2 \log(n))$

Other Measures of Cluster Distance

- Min/max of distances of any two points, one from each cluster
- Avg distance of all pairs of points, one from each cluster
- Merge two clusters if resulting cluster has
  - lowest radius (max dist btw point and centroid)
  - lowest diameter (max dist btw. two points)

Stopping Rules

- Stop if diameter/radius of next cluster > threshold

- Or stop if density of next cluster < threshold
  - Density: how many points per unit volume
  - Volume: estimated as some power, e.g., 1 or 2, of diameter/radius

Non-Euclidean Space

- Distance of two points can not be measured by their locations (i.e., no concept of coordinates)

- May use Jaccard, Cosine, Edit, Hamming to compute distances as appropriate

- E.g., Cosine can be used on two vectors
  - Even when measuring distance of their end points is not meaningful or desired
  - E.g., two similar documents, but one is twice as long as the other

Clusteroid

- Centroid is not meaningful in non-Euclidean

- Clusteroid = a point in the cluster that minimizes:
  - Sum of (squared) distances to other points, or
  - Maximum distance to another point

Example

- Consider a cluster of 4 points:
  - abcd, aecdb, abecb, ecdab

- Their edit distances:

	aecdb	abecb	ecdab
abcd	3	3	5
aecdb		2	2
abecb			4

Determine Clusteroid

- aecdb will be chosen as clusteroid
  - Located in “center” judged by all 3 measures

	aecdb	abecb	ecdab
abcd	3	3	5
aecdb		2	2
abecb			4

Point	Sum	Sum-sq	Max
abcd	11	43	5
aecdb	**7**	**17**	**3**
abecb	9	29	4
ecdab	11	45	5

Roadmap

- Problem, types, and distance functions
- Hierarchical clustering
- Point assignment
  - K-means
  - BFR
  - CURE
- Curse of dimensionality

K-means Algorithm

1. Pick k points as centroids of k clusters

2. Repeat until centroids stabilize
    a) For each point p,
        Find the centroid to which p is closest
        Add p to the cluster of that centroid
    b) Re-compute centroids of clusters

Point assignment

Complexity

Assume n data points

1. Pick k points as centroids of k clusters
   - O(k)

I: # of iterations

2. Repeat until centroids stabilize O(I*n*k)
   a) For each point p, O(n*k)
      Find the centroid to which p is closest
      Add p to the cluster of that centroid
   b) Re-compute centroids of clusters: O(k*n/k) or O(n)

Important Issues in k-means

- Picking points for initial centroids
  - May produce different clusters with diff. choices

- Picking right values for k: # of clusters

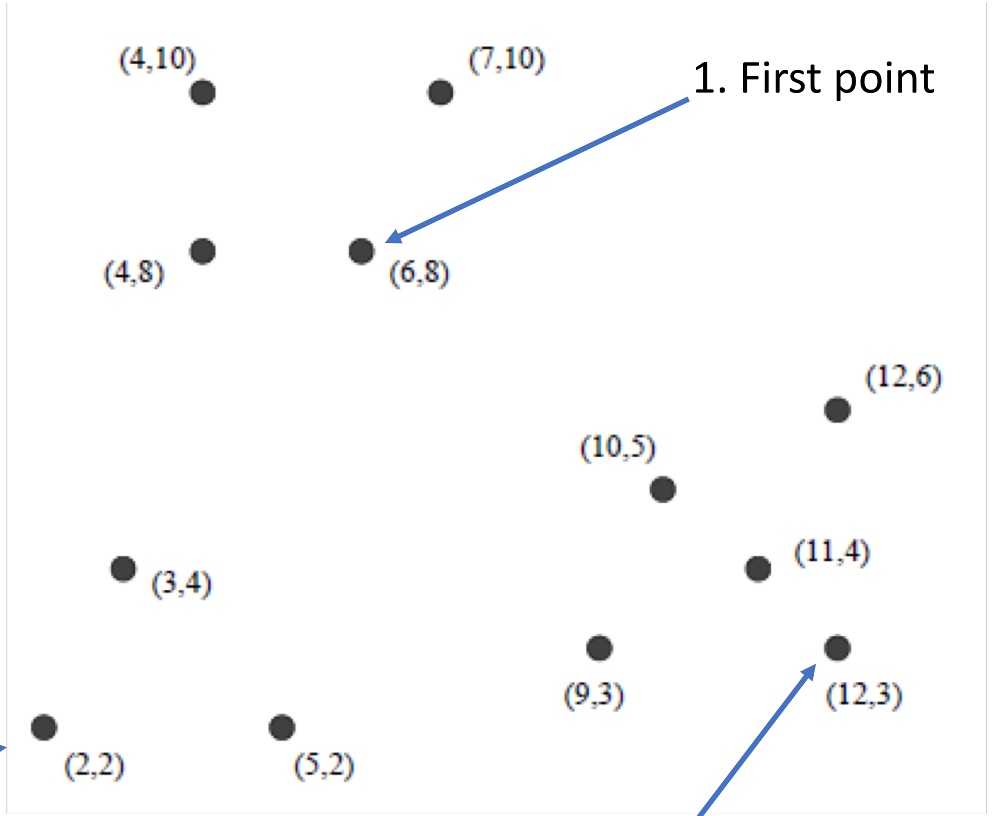
Picking Points for Initial Centroids

- Method 1: Pick points as far away as possible
  - First pick one randomly
  - Next, repeatedly pick one x far away from existing ones: distance between x and existing points = their minimum distance

- Method 2: produce k clusters by hierarchical methods, and select one point from each cluster
  - May cluster on a sample of points instead

Method 1 example

- First point: (6, 8)
- 2nd point: (12, 3)
- 3rd point: (2, 2)
- …



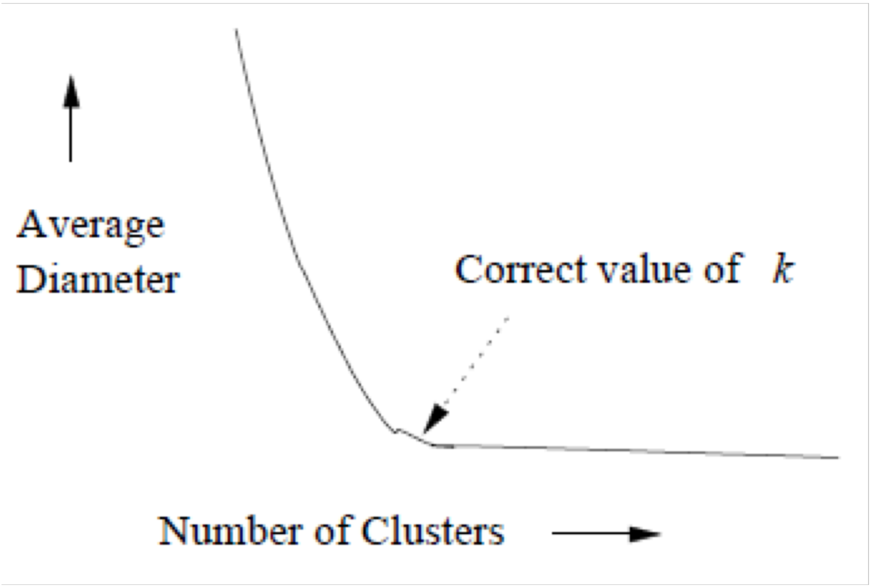
3. (2, 2) has maximum of minimum distance to (6,8) and (12,3)

2. (12,3) is farthest away from (6,8)

USC Viterbi School of Engineering Information Sciences Institute

Picking Right Value for k

- Cohesion of clusters increases dramatically
  - Before # of clusters is smaller than some k
- More cohesive if avg. diameter of clusters is smaller



Finding Right k

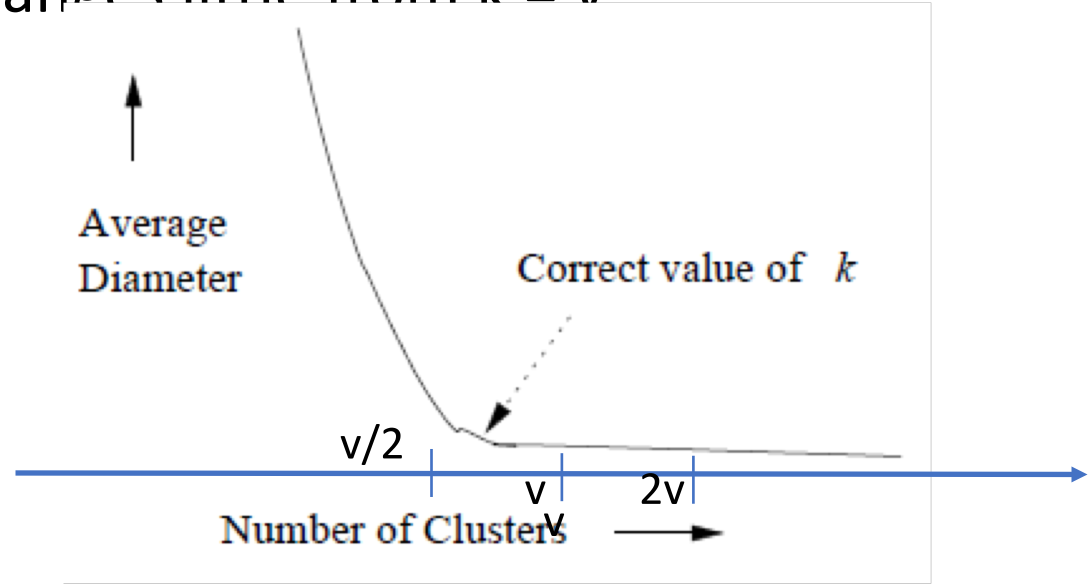
First, find the elbow of the curve:

- Run k-means for $k = 1, 2, 4, 8, \ldots 2^{m-1}$
  - i.e., double # of clusters at each clustering/run
- Stop at $k = 2v$
  - where cohesion changes little from $k = v$
  - Elbow: $[v/2, v]$

Since $k = 2^{m-1} = 2v$

$\Rightarrow m = 2 + \log_2 v$

(m: # of clusterings)



Finding Right k

- Next, binary search on [v/2, v]
  - Suppose current range [x, y]
  - Midpoint z = (x + y) /2
  - If not much change between [z, y]

    true k in [x, z]

    else

    true k in [z, y]
  - Continue search in [x, z] or [z, y]

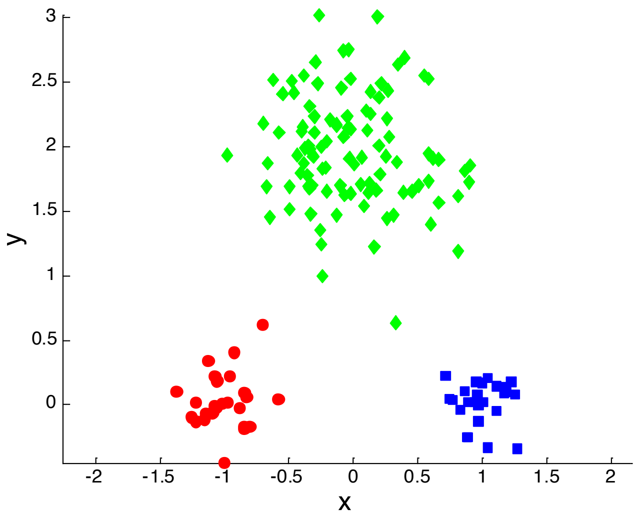
$\Rightarrow$Every search/clustering divides range by half

$\Rightarrow$# of clustering = $\log_2 (v/2) = \log_2 v - 1$

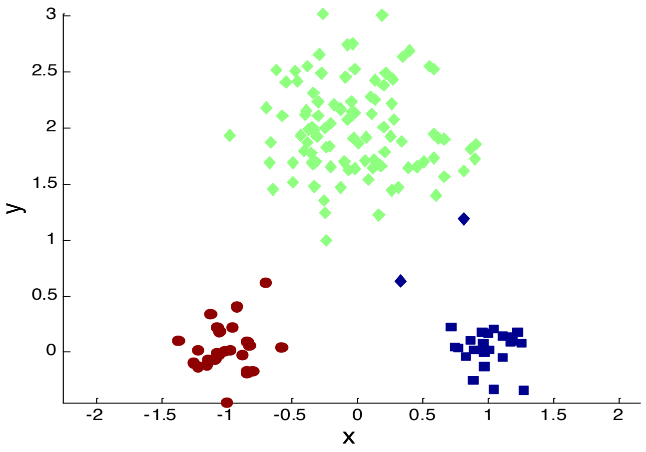
Overall, about $2\log_2 v$ clusterings

USC Viterbi School of Engineering Information Sciences Institute

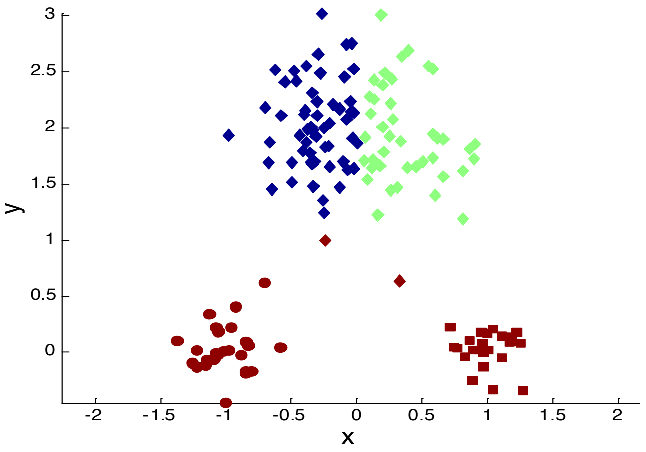
Two different K-means Clusterings



Original Points



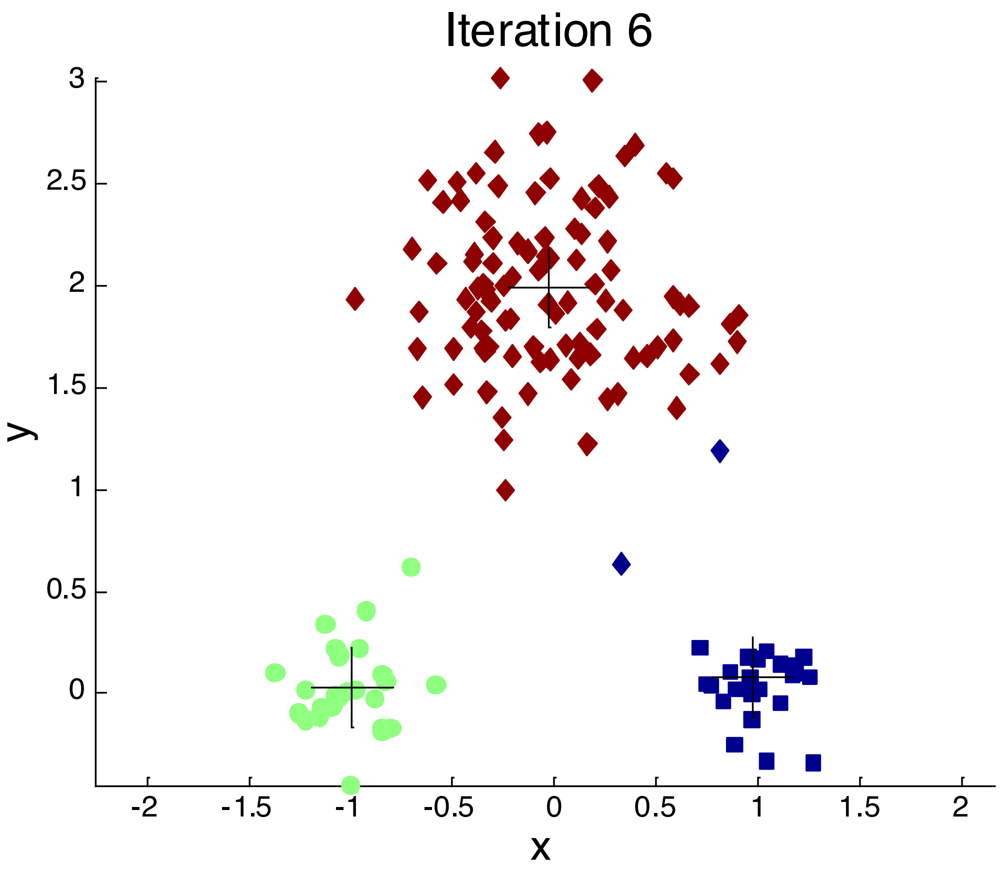
Optimal Clustering



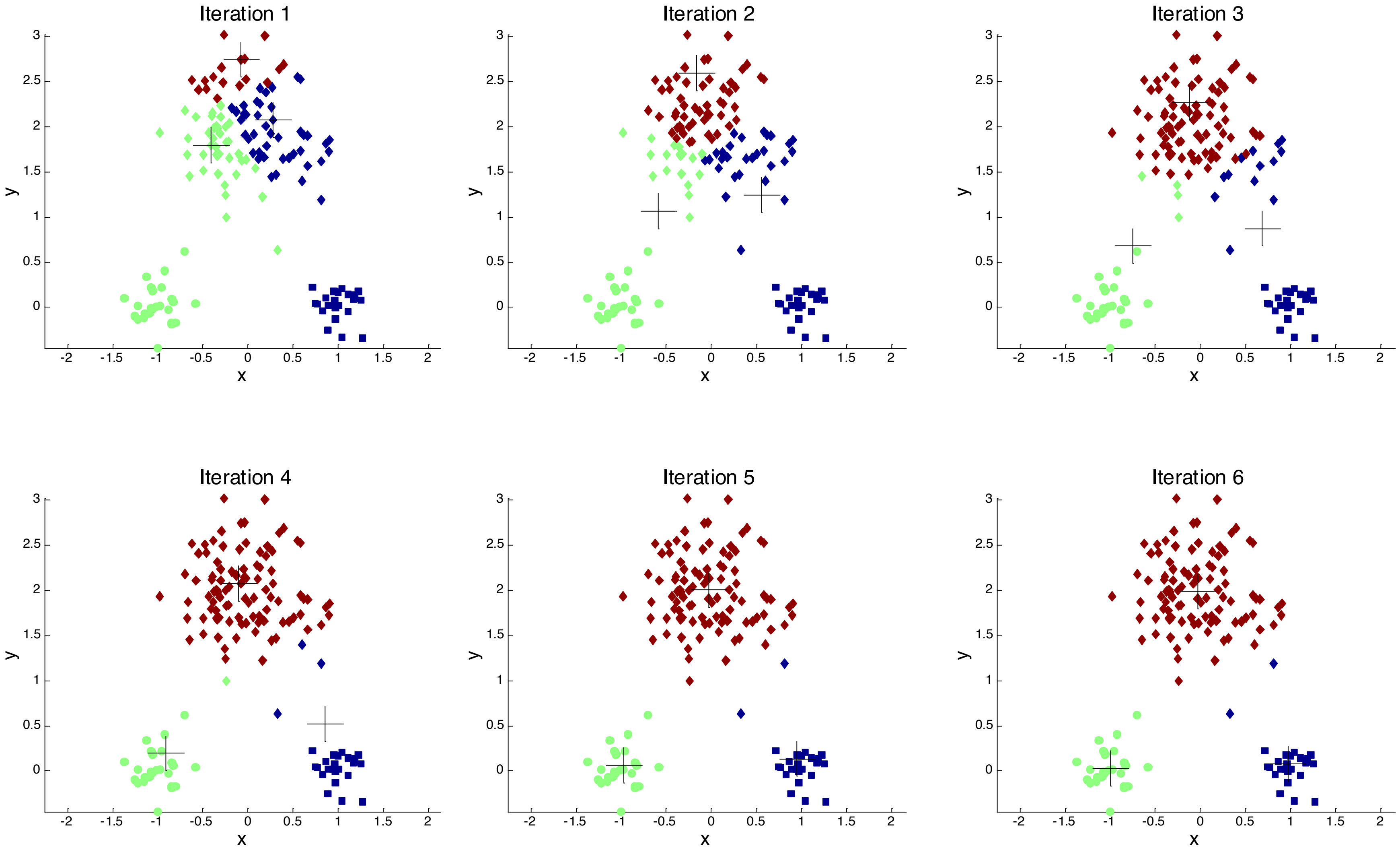
Sub-optimal Clustering

USC Viterbi
School of Engineering
Information Sciences Institute

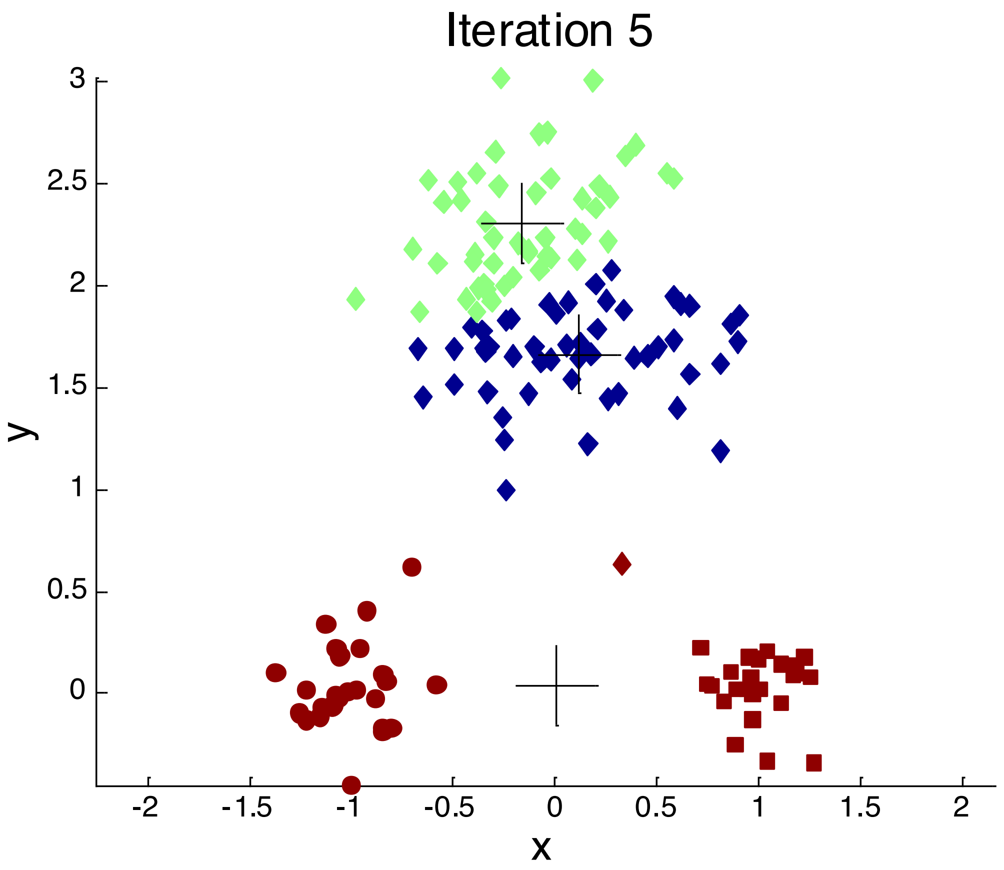
Importance of Choosing Initial Centroids



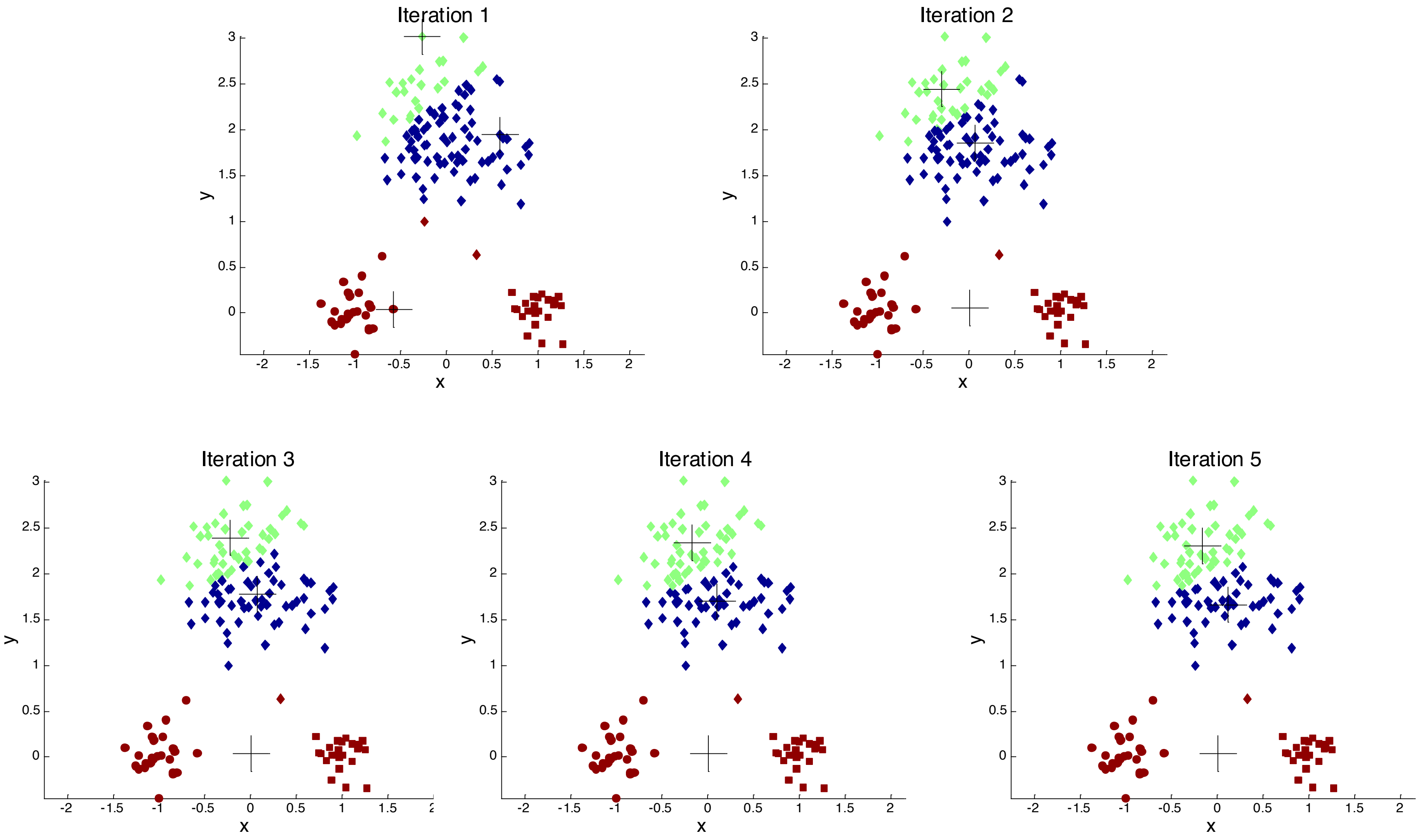
Importance of Choosing Initial Centroids



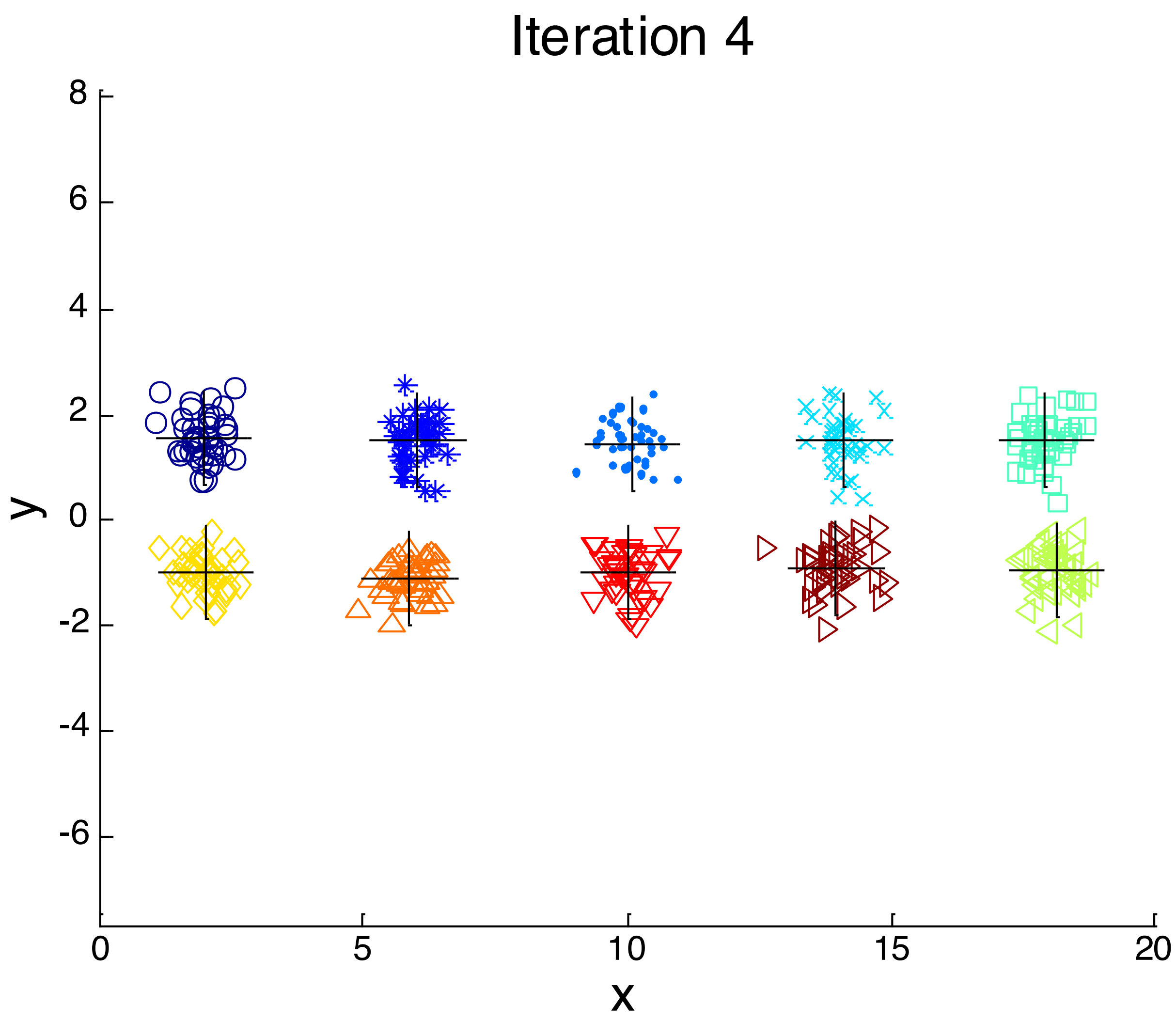
Importance of Choosing Initial Centroids …



Importance of Choosing Initial Centroids …

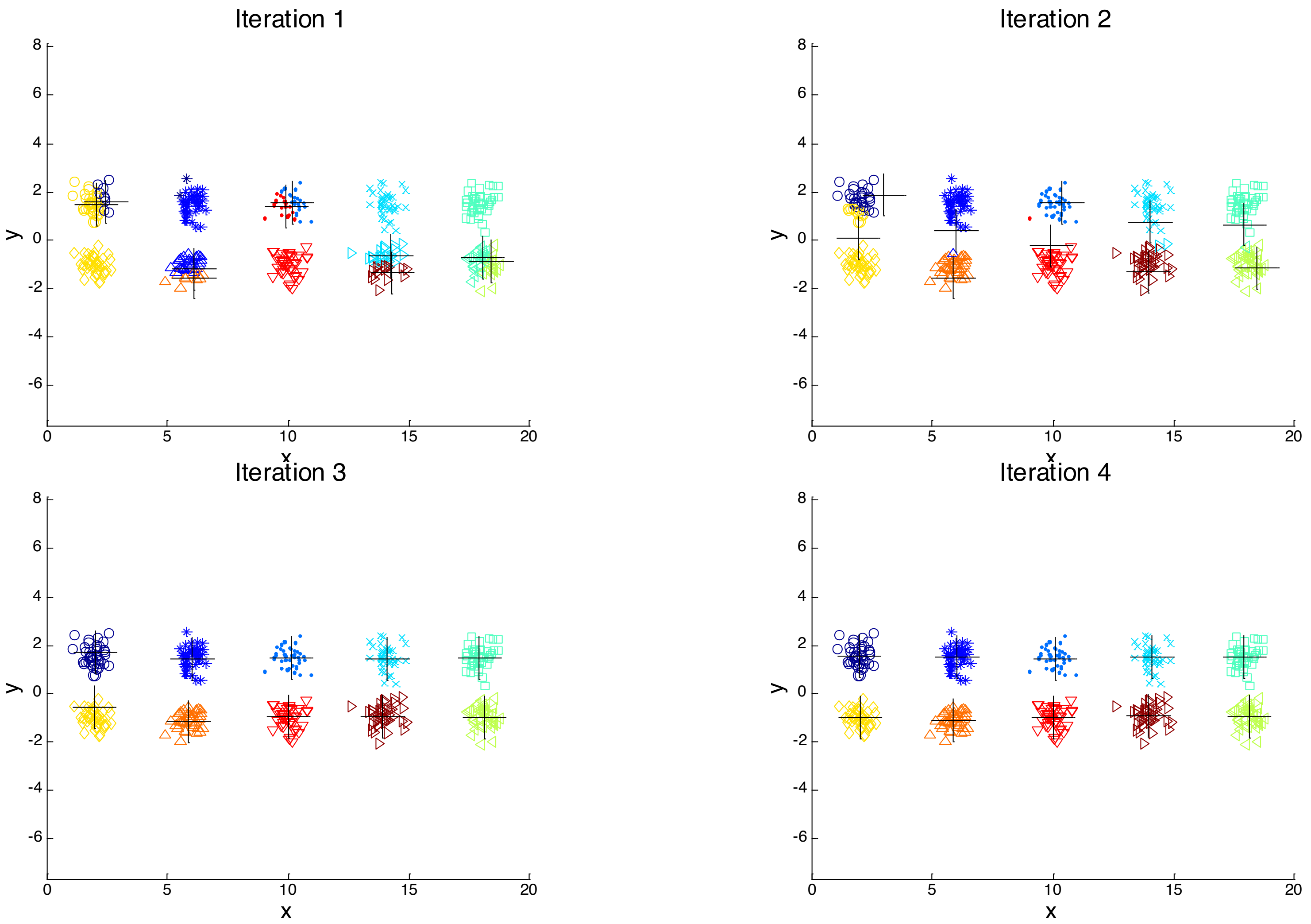


10 Clusters Example



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

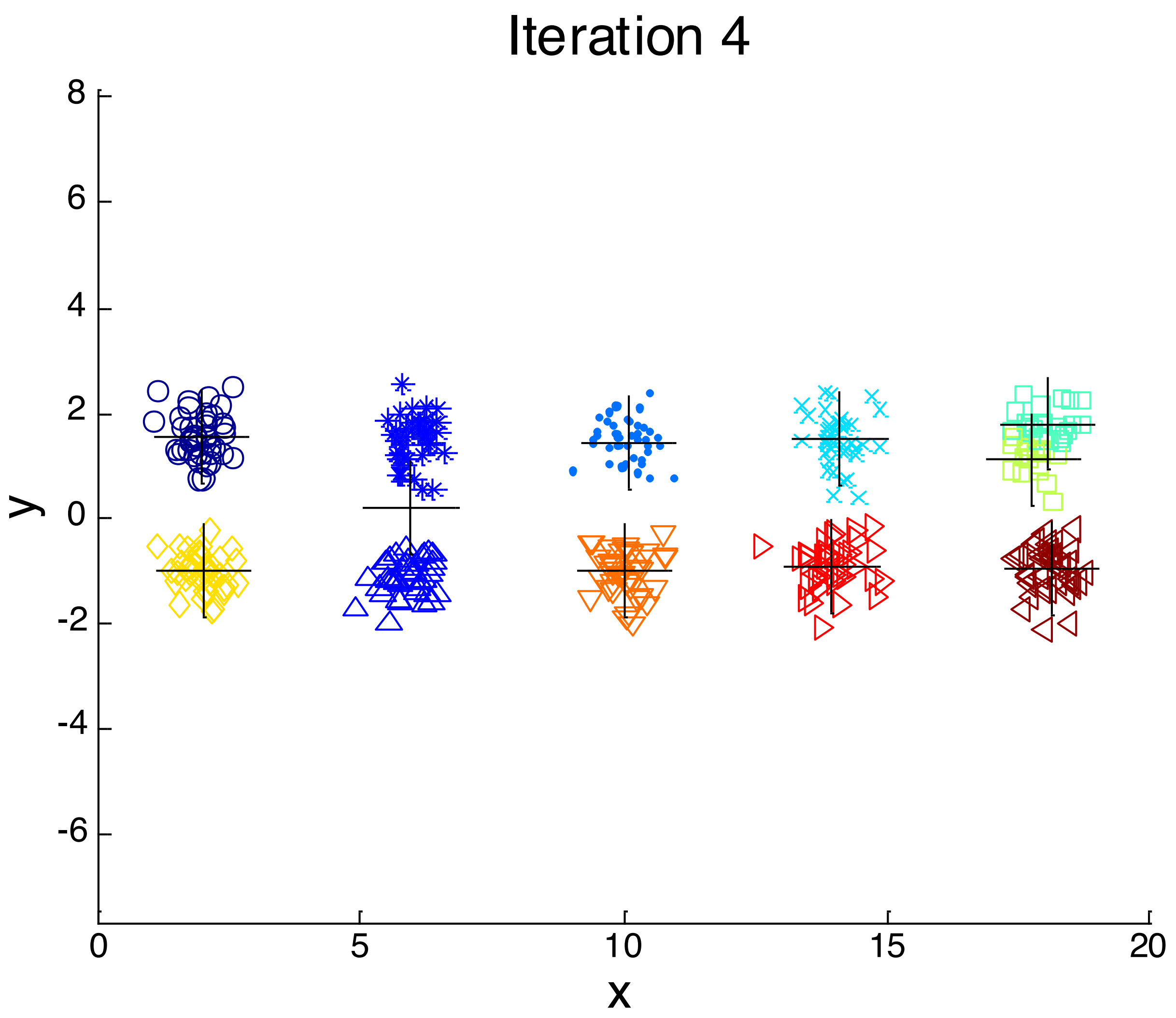
10 Clusters Example



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

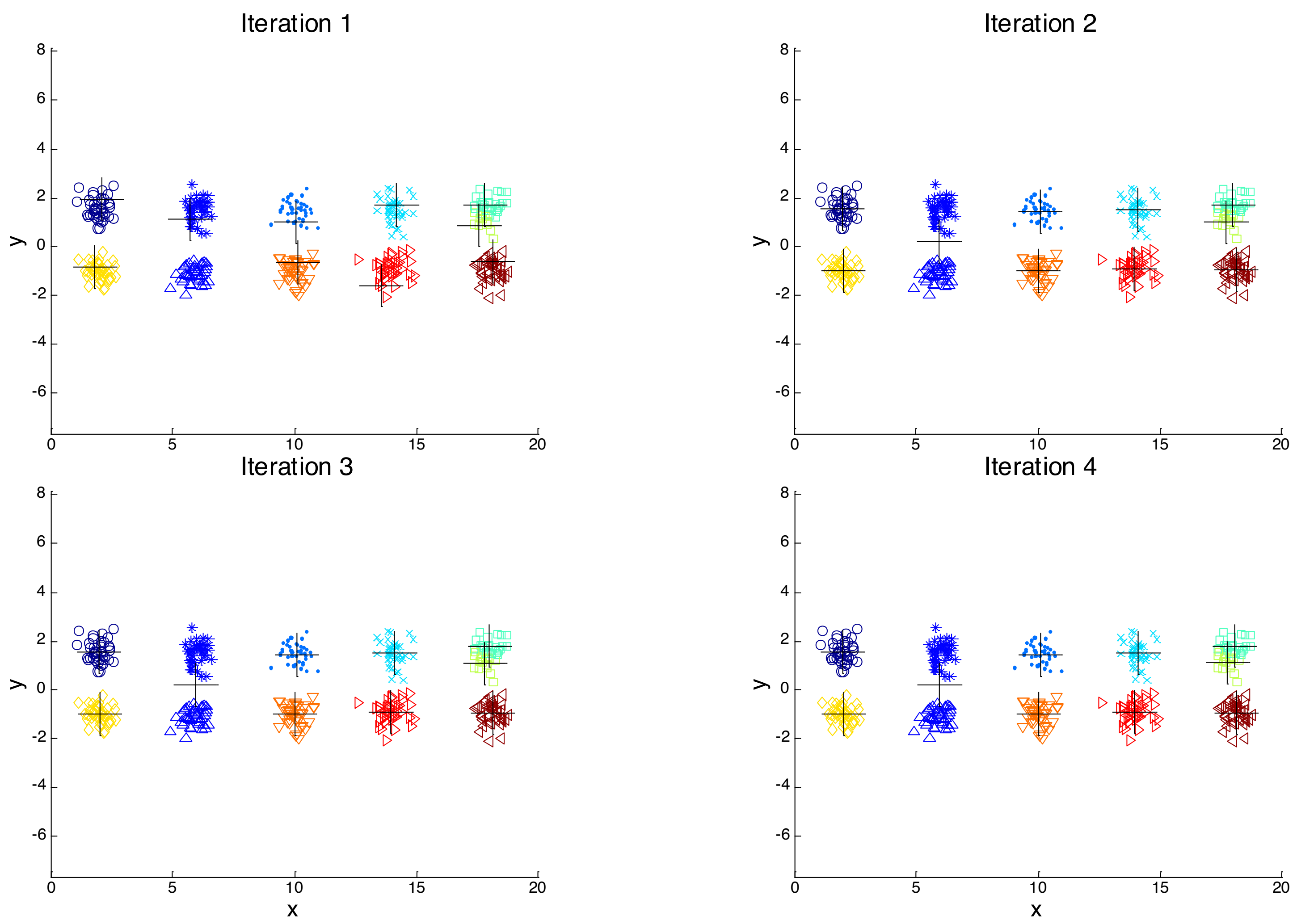
USC Viterbi School of Engineering *Information Sciences Institute*

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.

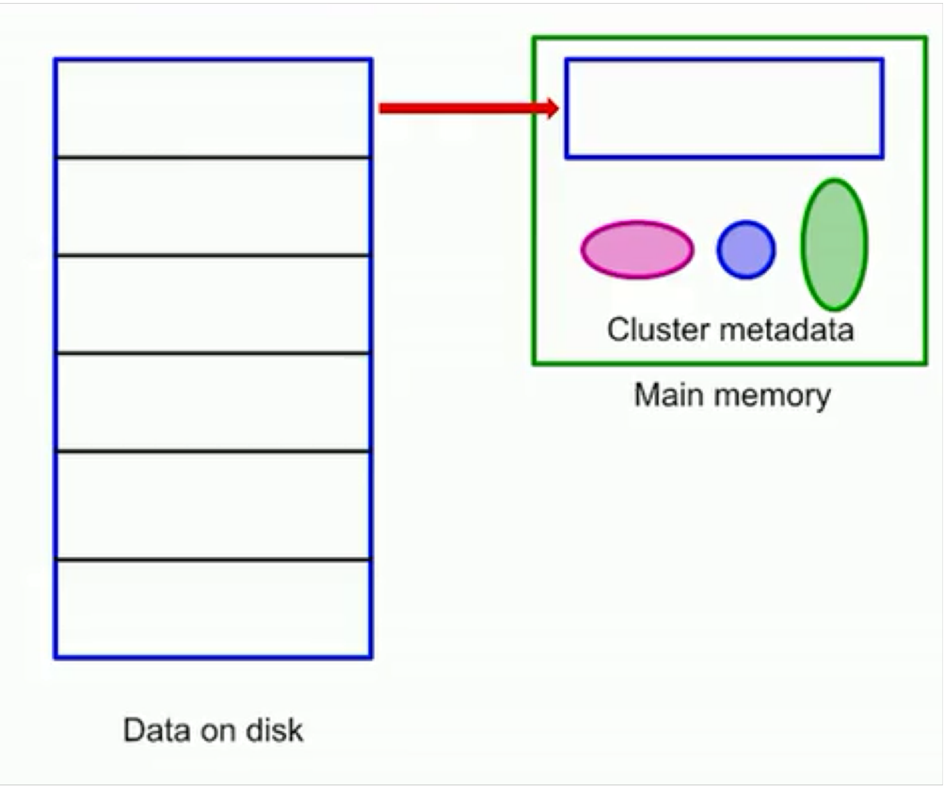
Roadmap

- Problem, types, and distance functions
- Hierarchical clustering
- Point assignment
  - K-means
  - BFR
  - CURE
- Curse of dimensionality

USC Viterbi
School of Engineering
Information Sciences Institute

BFR [Bradley-Fayyad-Reina] Algorithm

- Extend k-means to handle large data set
  - So large that can not be fit in main memory
  - Need to process one chunk at a time



BFR Algorithm

- Select k points as initial centroids
  - using methods discussed before

- Load one chunk of data into memory at a time

- For each chunk, its points are either:
  a) assigned to existing clusters, or
  b) used to form new mini-clusters, or
  c) retained

  Points in case a and b are not retained in main memory

Case a

- Case a: assigned to an existing cluster
  - if point is sufficiently close to the centroid of the cluster

- Update summary of cluster C_i
  - N: # of points in C_i
  - SUM_i: Sum of values of points in each dim
  - $SUMSQ_i$:Sum of squared values of points in each dim

  => 2d + 1 values, where d = # of dimensions

Cluster Summary

- Points in cluster: (5, 1), (6, -2), (7, 0)
- N = 3, SUM = [18, -1], SUMSQ = [110, 5]

$\Rightarrow$Centroid = SUM/N = [6, -1/3]

$\Rightarrow$Variance = SUMSQ/N – $(\text{SUM/N})^2$

= $[110/3 - 6^2, 5/3 - (-1/3)^2]$ = [.667, 1.56]

=> Standard deviation = [.816, 1.25]

Variance

- $Var(X) = E\left[\left(X - E(X)\right)^2\right]$

$$= E[X^2 - 2XE[X] + (E[X])^2)]$$

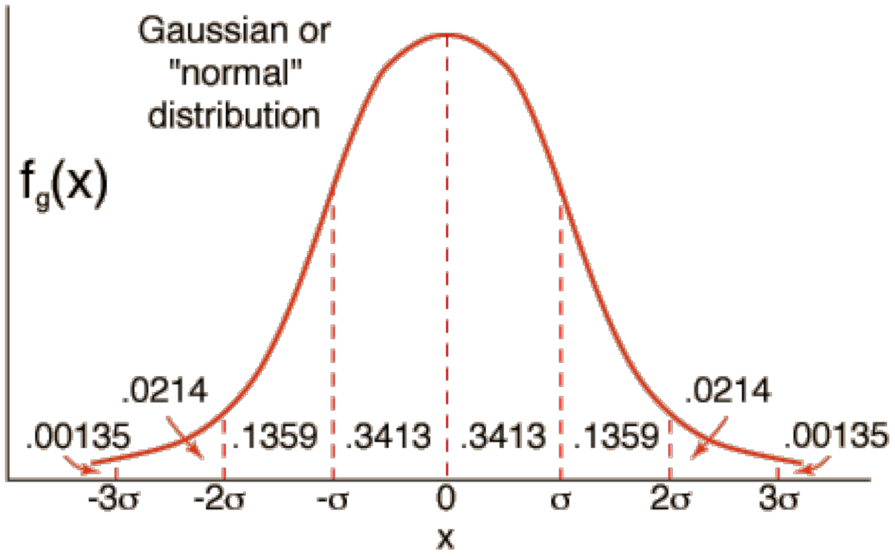
$$= E[X^2] - 2E[X]E[X] + (E[X])^2$$

$$= E[X^2] \ - (E[X])^2$$

USC Viterbi School of Engineering Information Sciences Institute

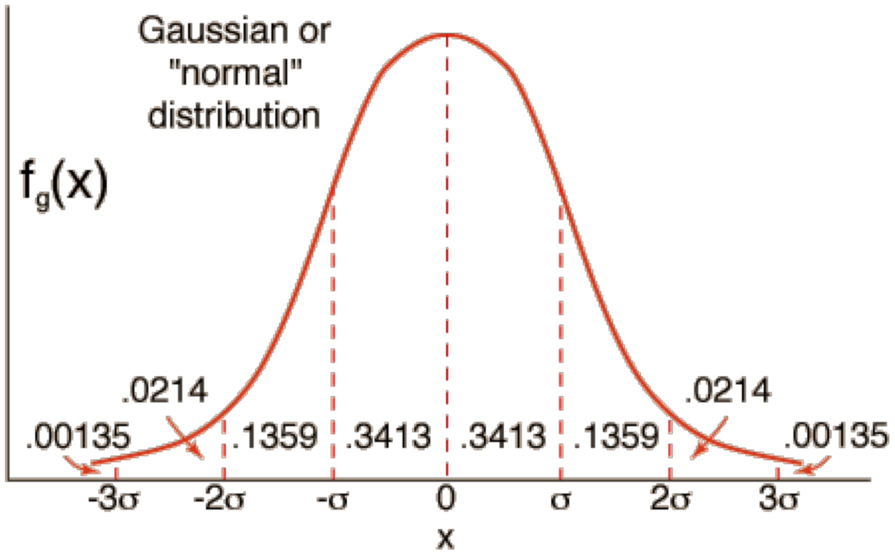
Define “Sufficiently Close”

- Assume points in cluster are normally distributed
  => we know prob. of particular distance from mean



Define “Sufficiently Close”

- ~68% of points: 1 σ away from mean
- ~95% of points: 2 σ away
- ~99% of points: 3 σ away



USC Viterbi School of Engineering
Information Sciences Institute

Mahalanobis Distance

- Normalized distance for multi-dimensional data
  - How many σ away from centroid
  - This assumes no-correlation among diff. dimensions

Combine dist. in Euclidean space

$$\sqrt{\sum_{i=1}^{d}\left(\frac{p_i - c_i}{\sigma_i}\right)^2}$$

Normalized distance in i-th dim

- Point p: $[p_1, \ldots, p_d]$; centroid: $[c_1, \ldots, c_d]$

USC Viterbi School of Engineering Information Sciences Institute

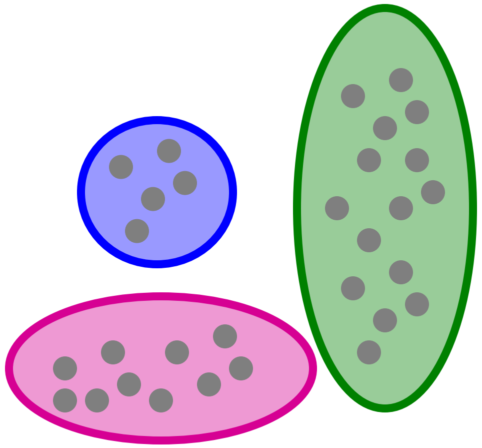
Define "Sufficiently Close"

- Use Mahalanobis to measure distance
- Pick centroid with smallest distance
- If distance < threshold (e.g., 4), add point to cluster
  - Prob. of 4σ away from mean is less than 10^{-6}

USC Viterbi School of Engineering Information Sciences Institute

Assumptions

- Axes of cluster align with axes of space



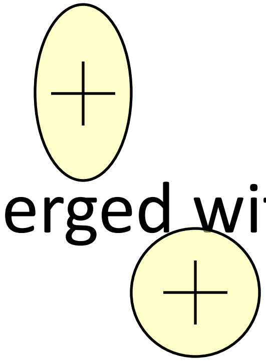
Case b: form new mini-clusters

- Use
  - points not assigned to existing clusters
  - points retained from last rounds

- Can use a hierarchical clustering algorithm with proper stopping condition

Merge Mini-Clusters

- Merge new and existing mini-clusters
  - If variance of merged cluster is small enough
  - Variance can be computed from: N, SUM, SUMSQ

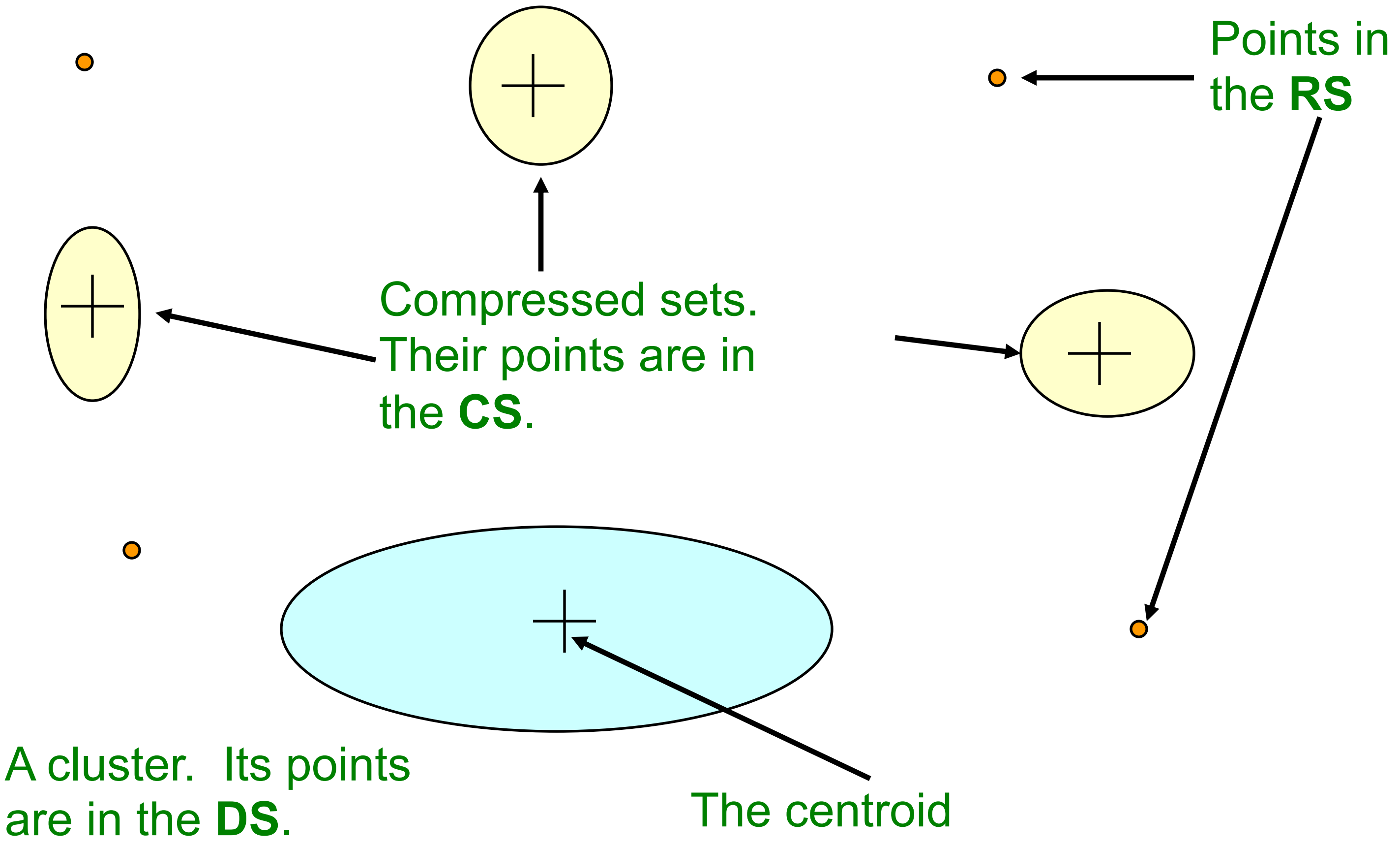


- Note that none of mini-clusters can be merged with existing (non-mini) clusters

Classification of Points

- Discard set (DS)
  - Points close enough to an existing cluster
- Compression set (CS)
  - Points close to each other to form mini-clusters
  - But not close enough to any cluster
- Retained set (RS)
  - Isolated points retained for next rounds

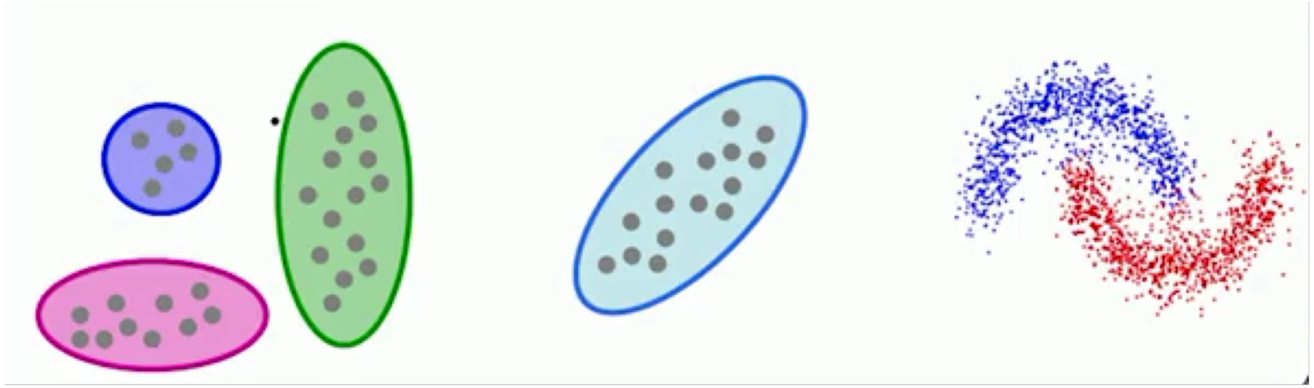
BFR: “Galaxy” Picture



USC Viterbi
School of Engineering
Information Sciences Institute

Limitations of BFR

- Strong assumptions about clusters
  - Normally distributed in each dimension
  - Axis-parallel: not ok to have ellipses at an angle



OK Not OK Not OK

Roadmap

- Problem, types, and distance functions
- Hierarchical clustering
- Point assignment
  - K-means
  - BFR
  - CURE
- Curse of dimensionality

CURE (Clustering Using REpresentatives)

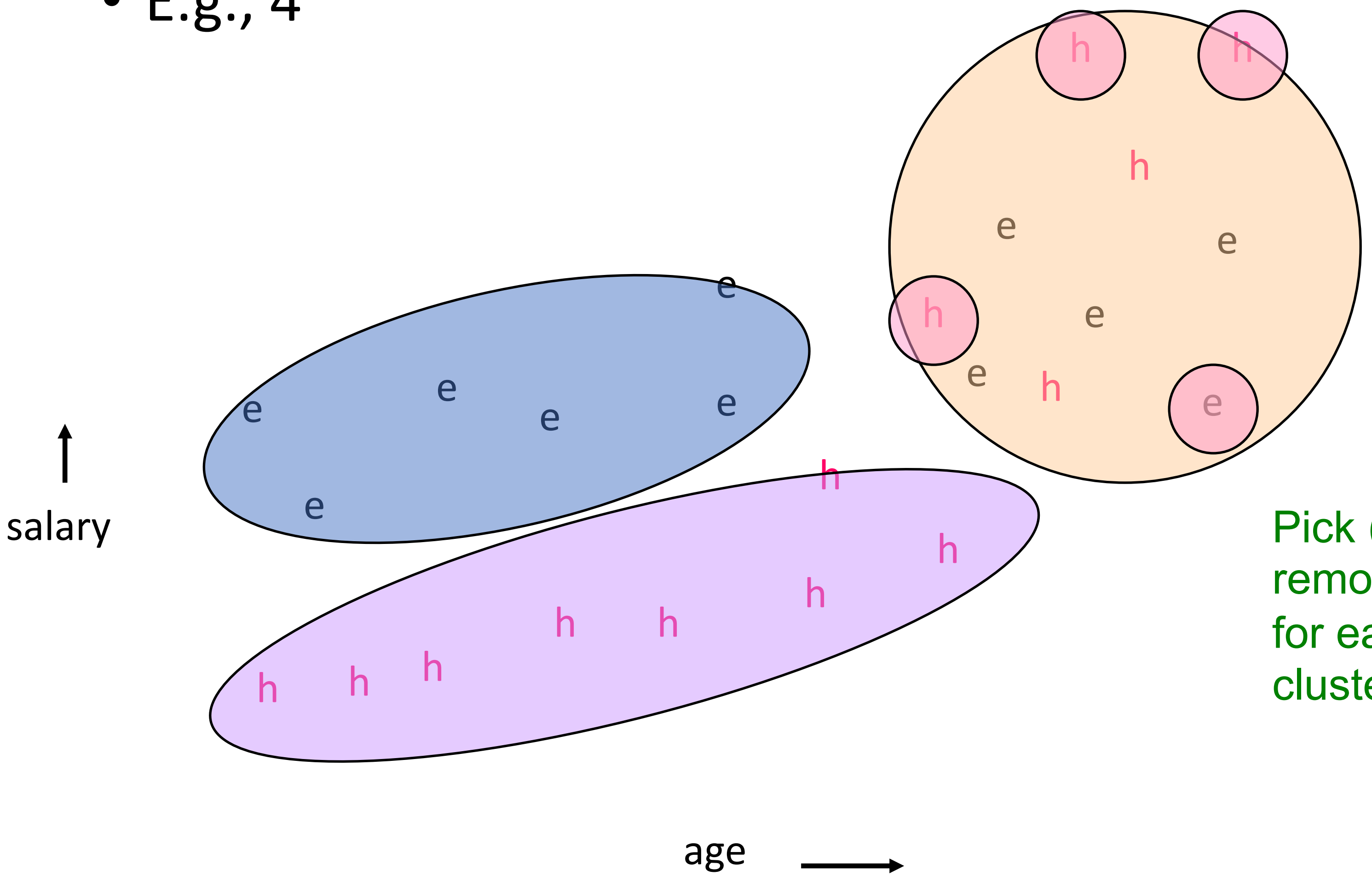
- Also handle large-scale data

- Two passes:
  1. Pick a sample and cluster it hierarchically
  2. Scan data and assign points to closest cluster

- Distance between point p and cluster C
  - Distance of p from the closest representative in C

Cluster Representatives

- A small set of points far away from each other
  - E.g., 4

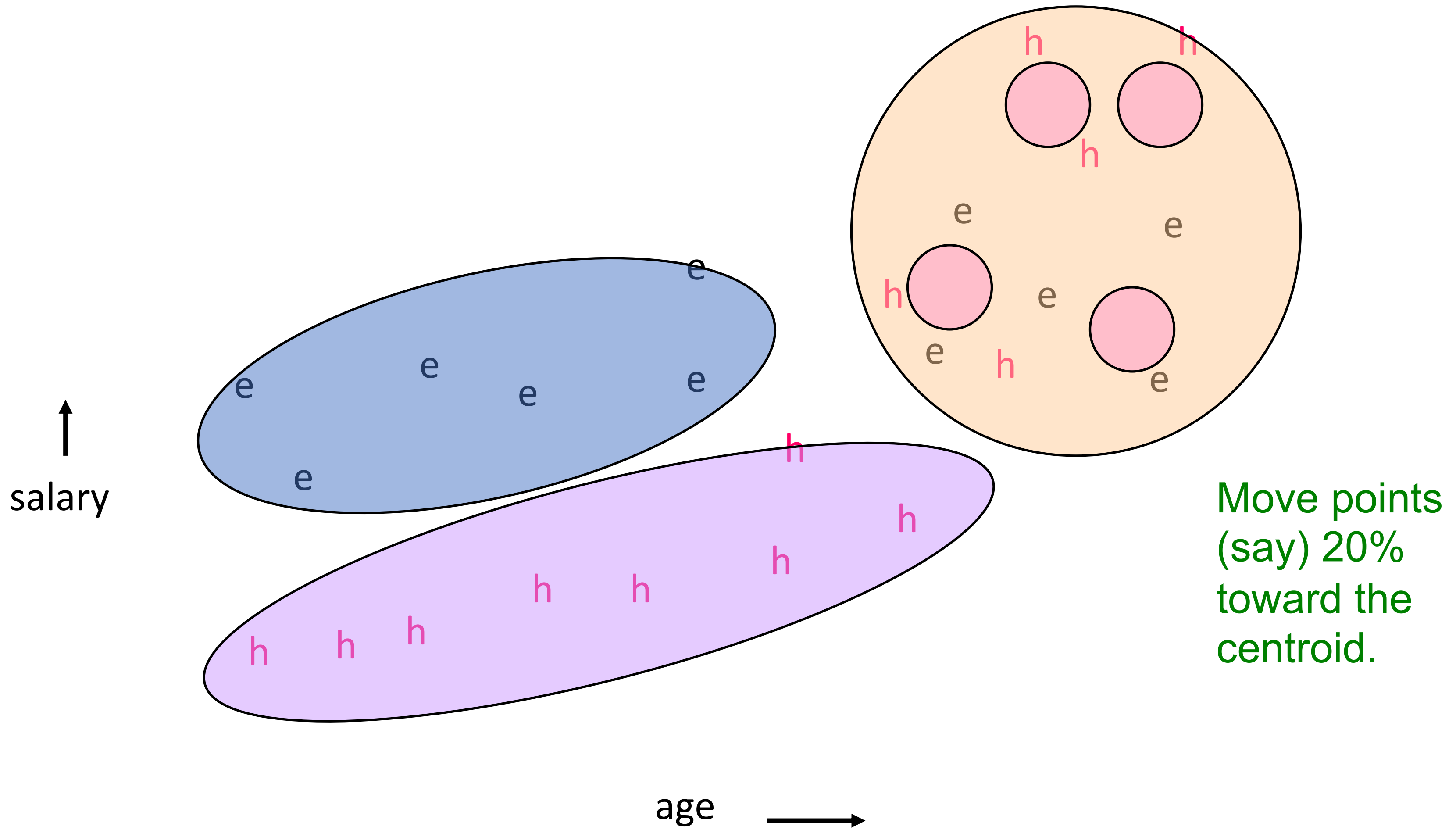


Pick (say) 4 remote points for each cluster.

USC Viterbi School of Engineering Information Sciences Institute

Moving Representatives

- A fixed fraction of distance toward centroid
  - E.g., 20%



Compare CURE with BFR

- Distribution of data
  - CURE: do not assume any particular distribution
  - BFR: data should be normally distributed

- Representation of cluster
  - CURE: a set of representatives
  - BFR: centroid

- Common: both assume data in Euclidean space

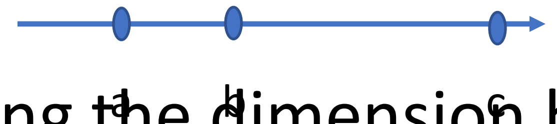
Roadmap

- Problem, types, and distance functions
- Hierarchical clustering
- Point assignment
  - K-means
  - BFR
  - CURE
- Curse of dimensionality

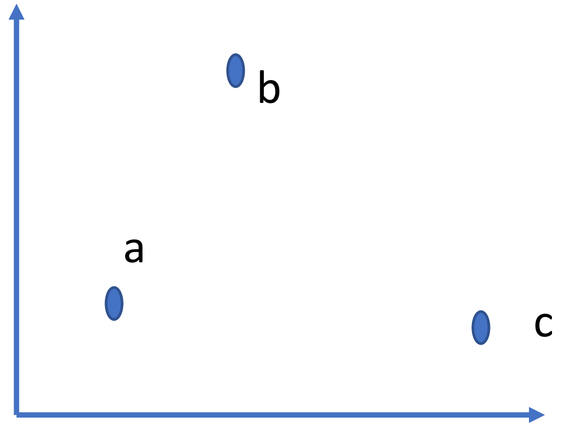
USC Viterbi School of Engineering
Information Sciences Institute

High Dimension: Euclidean

- Consider a set of data points on a line
  - dist(a, b) < dist(a, c)



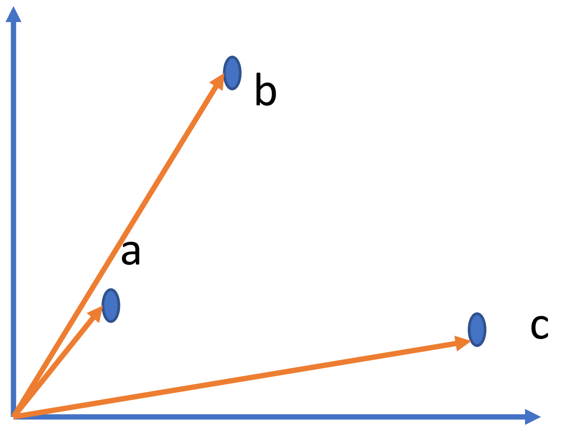
- Consider increasing the dimension by 1
  - dist(a, b) ~ dist(a, c)

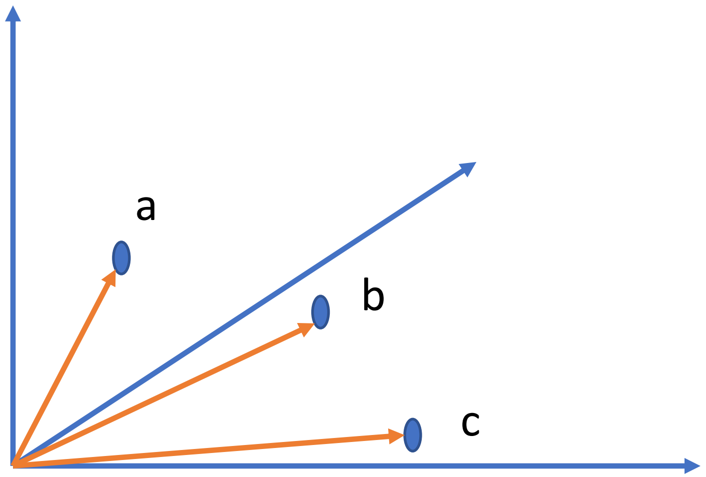


USC Viterbi School of Engineering Information Sciences Institute

High Dimension: Cosine

- Cosine(a, b) > Cosine(a, c)
- Increase d to 3
  - Cosine(a, b) ~ Cosine(a, c)
- Higher d
  - Angle -> 90^o
  - Cosine -> 0





Curse of Dimensionality

- Data points have similar distance btw each other
  - Euclidean distance breaks

- Data vectors become orthogonal
  - Cosine function breaks

Subspace Clustering