Machine Learning
Lecture 9: Clustering

Jie Li

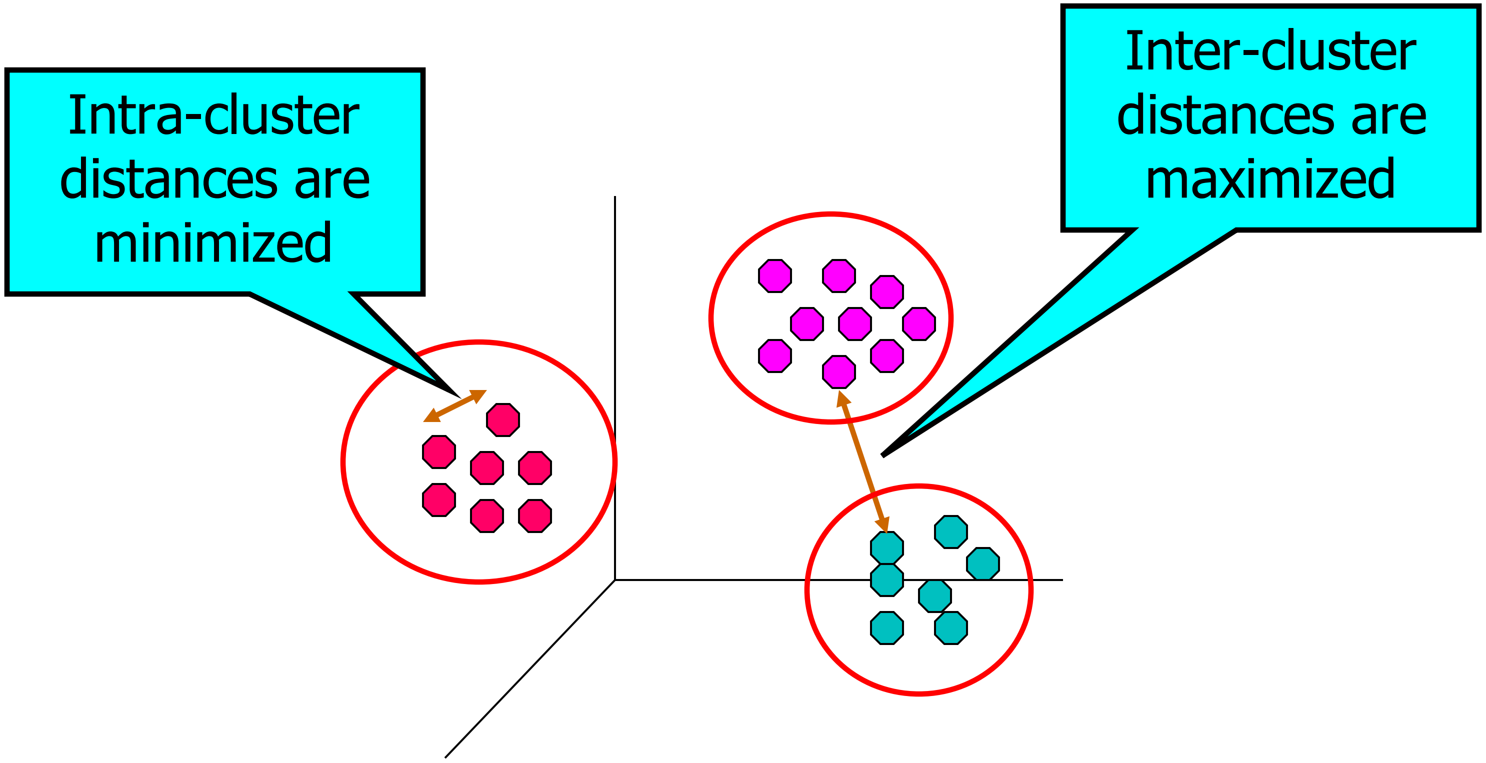
nijanice@163.com

Machine Learning Problems

	Supervised Learning	*Unsupervised Learning*
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

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



Clustering Applications

Clustering Applications

Recommender systems and advertising

- Cluster users for item/ad recommendation
- Cluster items for related item suggestion

Text mining

- Cluster documents for related search
- Cluster words for query suggestion

Image search

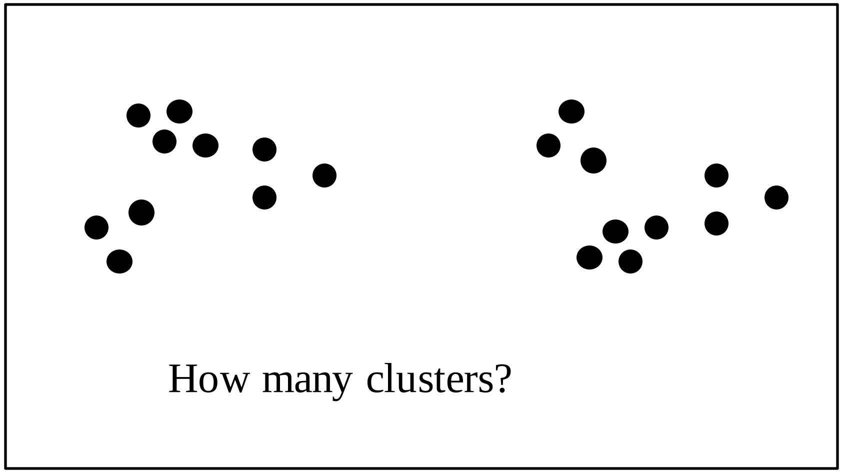
- Cluster images for similar image search and duplication detection

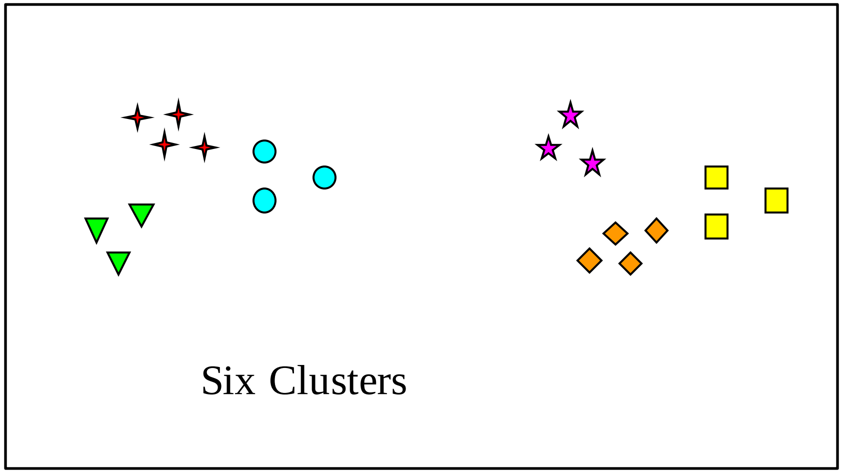
Speech recognition or separation

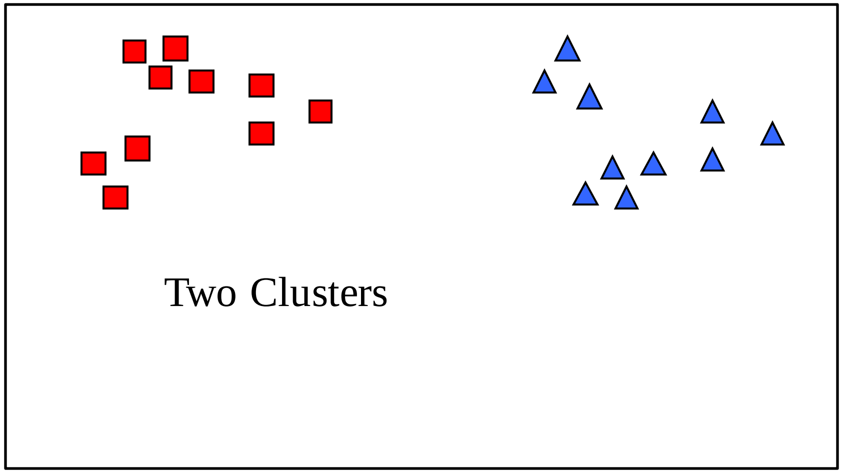
- Cluster phonetical features

…

Notion of a Cluster can be Ambiguous







Quality: What Is Good Clustering?

A good clustering method will produce high quality clusters with

- high intra-class similarity
- low inter-class similarity

The quality of a clustering result depends on both the similarity measure used by the method and its implementation

The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns

Major Clustering Approaches

Partitioning approach:

- Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
- Typical methods: k-means, ISODATA ,k-medoids, Kernel K-means , CLARANS

Hierarchical approach:

- Create a hierarchical decomposition of the set of data (or objects) using some criterion
- Typical methods: Agnes, Diana, BIRCH, ROCK, CAMELEON

Density-based approach:

- Based on connectivity and density functions
- Typical methods: DBSCAN, OPTICS, DenClue

Major Clustering Approaches

Grid-based approach:

- based on a multiple-level granularity structure
- Typical methods: STING, WaveCluster, CLIQUE

Model-based:

- A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
- Typical methods: EM, SOM, COBWEB

Frequent pattern-based:

- Based on the analysis of frequent patterns
- Typical methods: pCluster

…

Other Distinctions Between Sets of Clusters

- Exclusive versus non-exclusive
  - In non-exclusive clusterings, points may belong to multiple clusters.
  - Can represent multiple classes or 'border' points
- Fuzzy versus non-fuzzy
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
  - Probabilistic clustering has similar characteristics
- Partial versus complete
  - In some cases, we only want to cluster some of the data
- Heterogeneous versus homogeneous
  - Cluster of widely different sizes, shapes, and densities

Types of Clusters

Well-separated clusters

Center-based clusters

Contiguous clusters

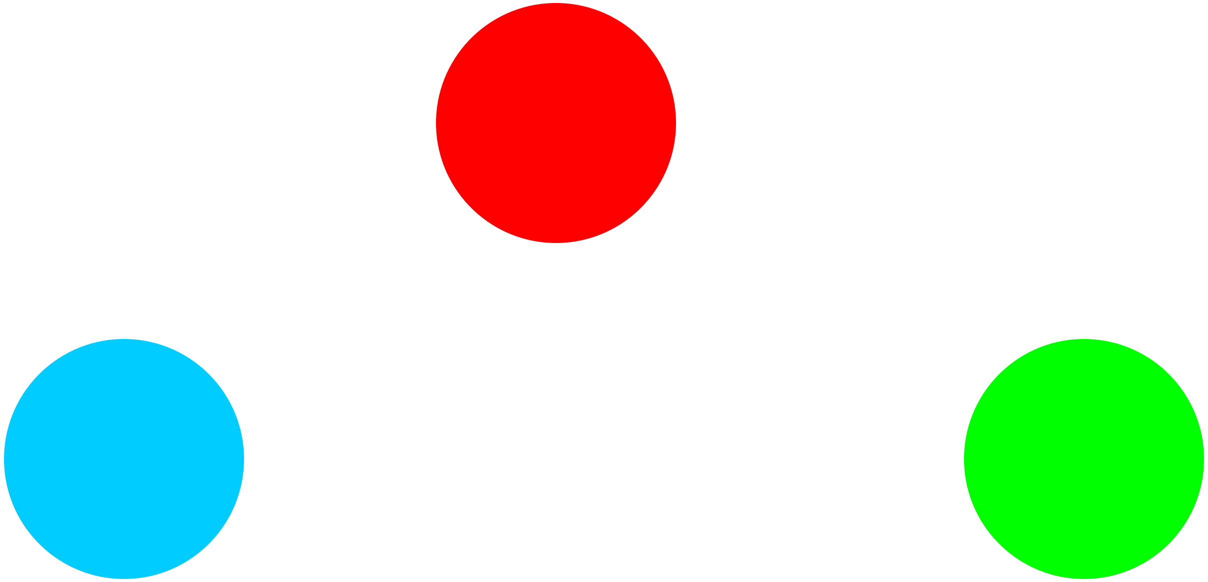
Density-based clusters

Property or Conceptual

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

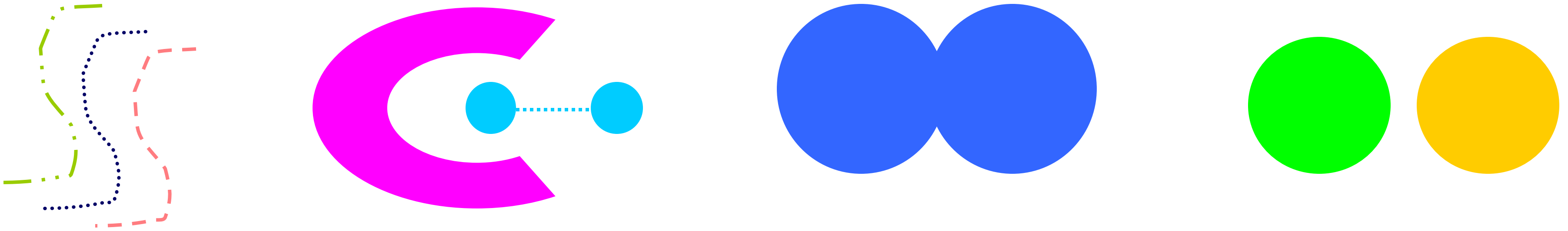
- A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster
- The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the most "representative" point of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

4 center-based clusters

Types of Clusters: Contiguity-Based

Contiguous Cluster (Nearest neighbor or Transitive)

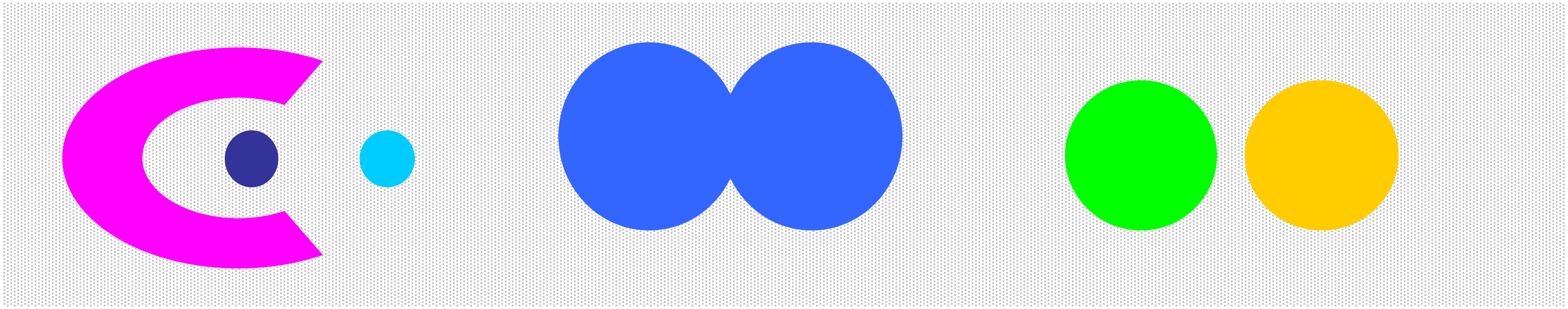
- A cluster can be defined as a connected component: a group of objects that are connected to each other but not to objects outside the group.
- A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in 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.



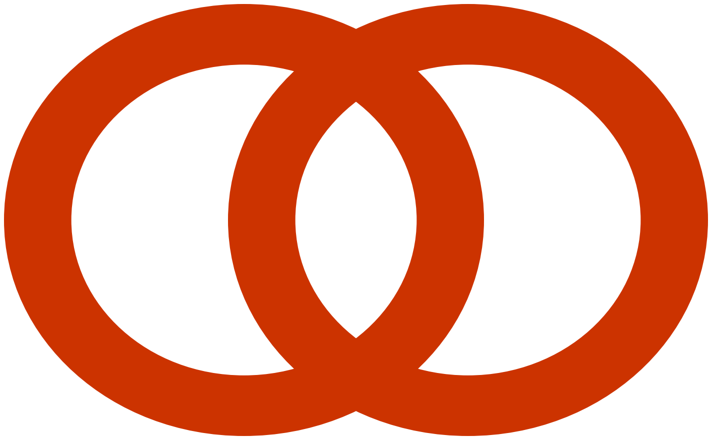
6 density-based clusters

Types of Clusters: Conceptual Clusters

Shared Property or Conceptual Clusters

- Finds clusters that share some common property or represent a particular concept.

.



2 Overlapping Circles

Clustering Algorithms

Partitional Clustering

Hierarchical clustering

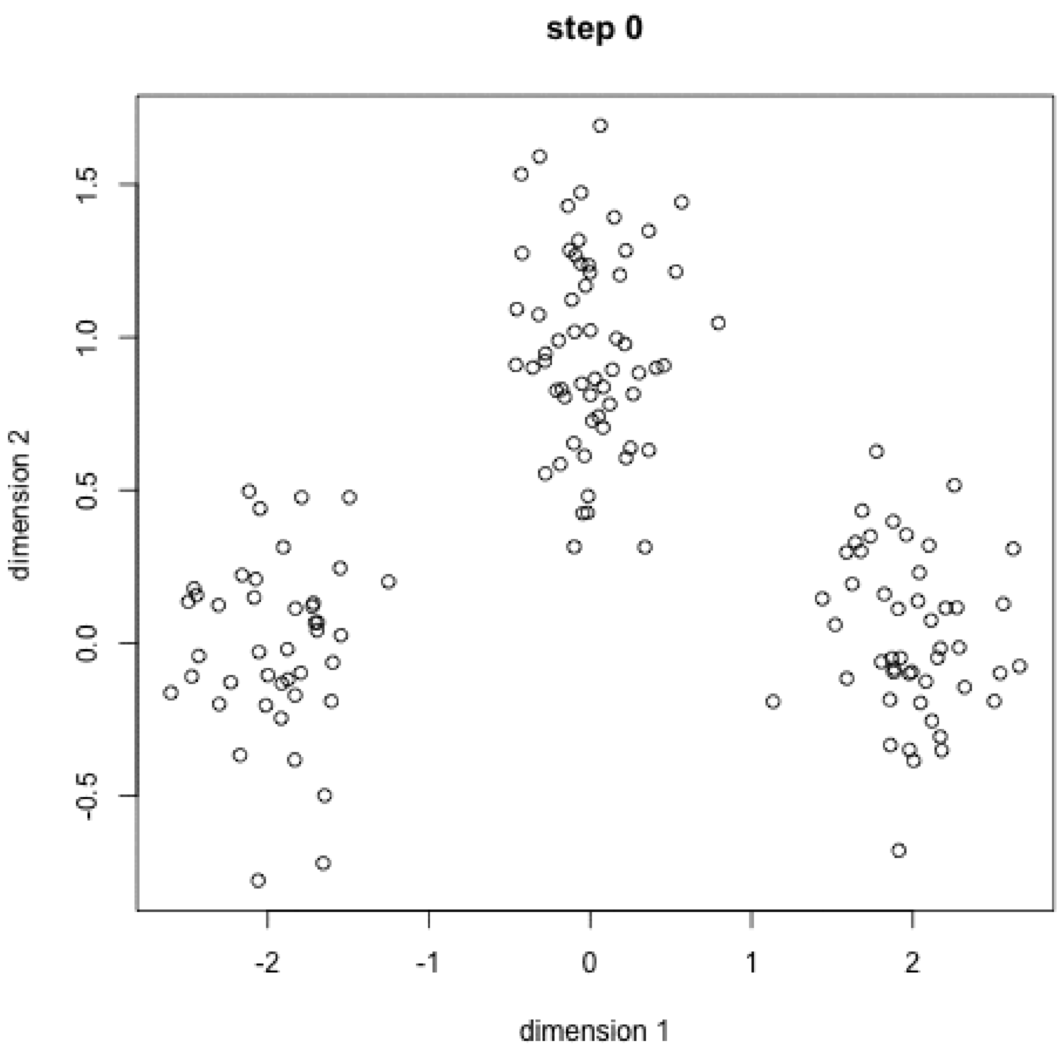
Density-based clustering

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

```
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



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, 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

Euclidean Distance

Euclidean Distance

$$d(x, y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

cosine similarity

If d_1 and d_2 are two document vectors, then

$$\cos(x, y) = (x \bullet y) / ||x||\ ||y||\ ,$$

Example:

x **= 3 2 0 5 0 0 0 2 0 0**

y **= 1 0 0 0 0 0 0 1 0 2**

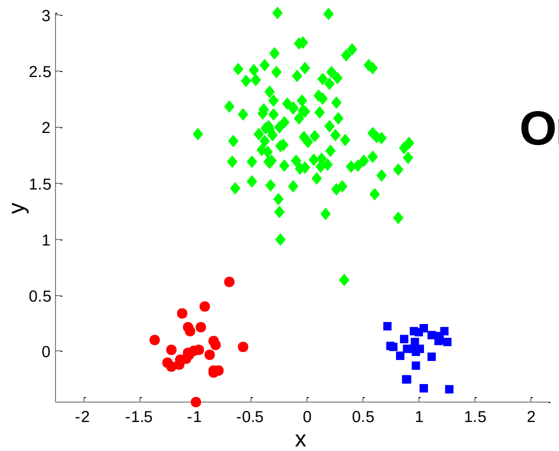
$x \bullet y$ = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5

$||x|| = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.481$

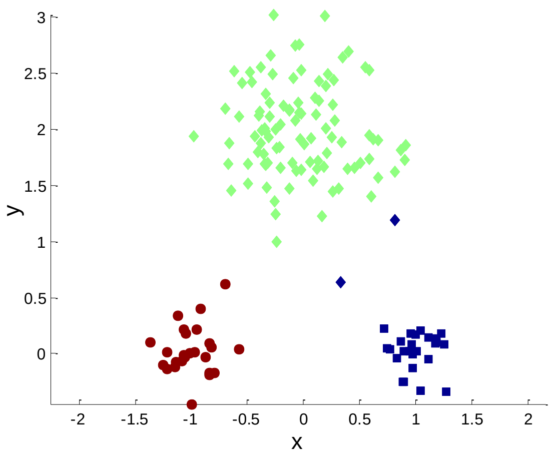
$||y|| = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)^{0.5} = (6)^{0.5} = 2.245$

$\cos(d_1, d_2) = 0.3150$

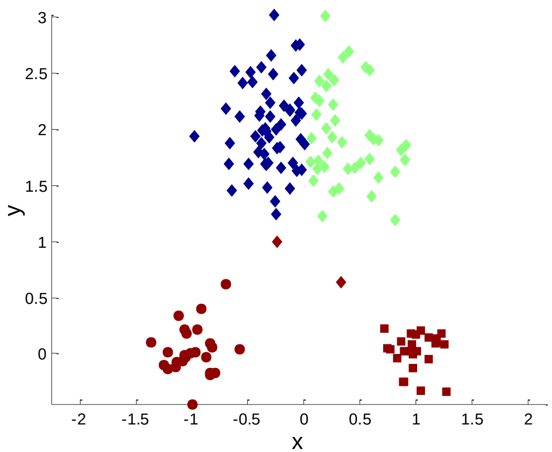
Two different K-means Clusterings



Original Points

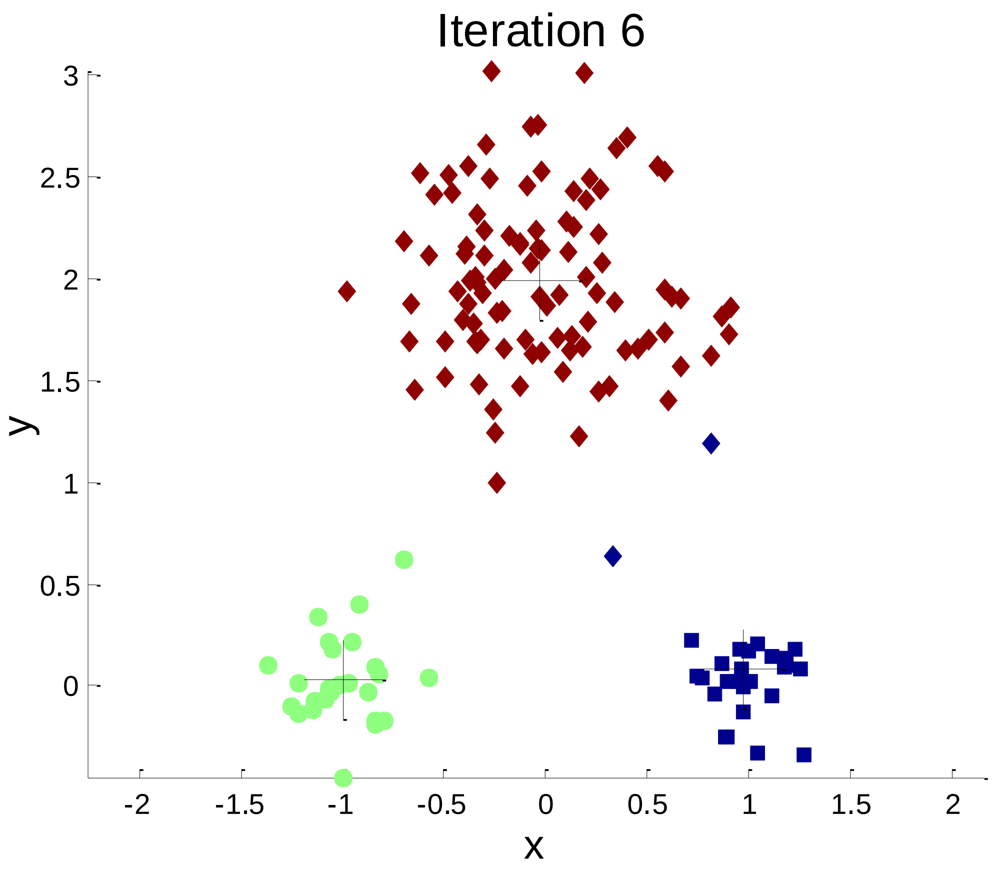


Optimal Clustering

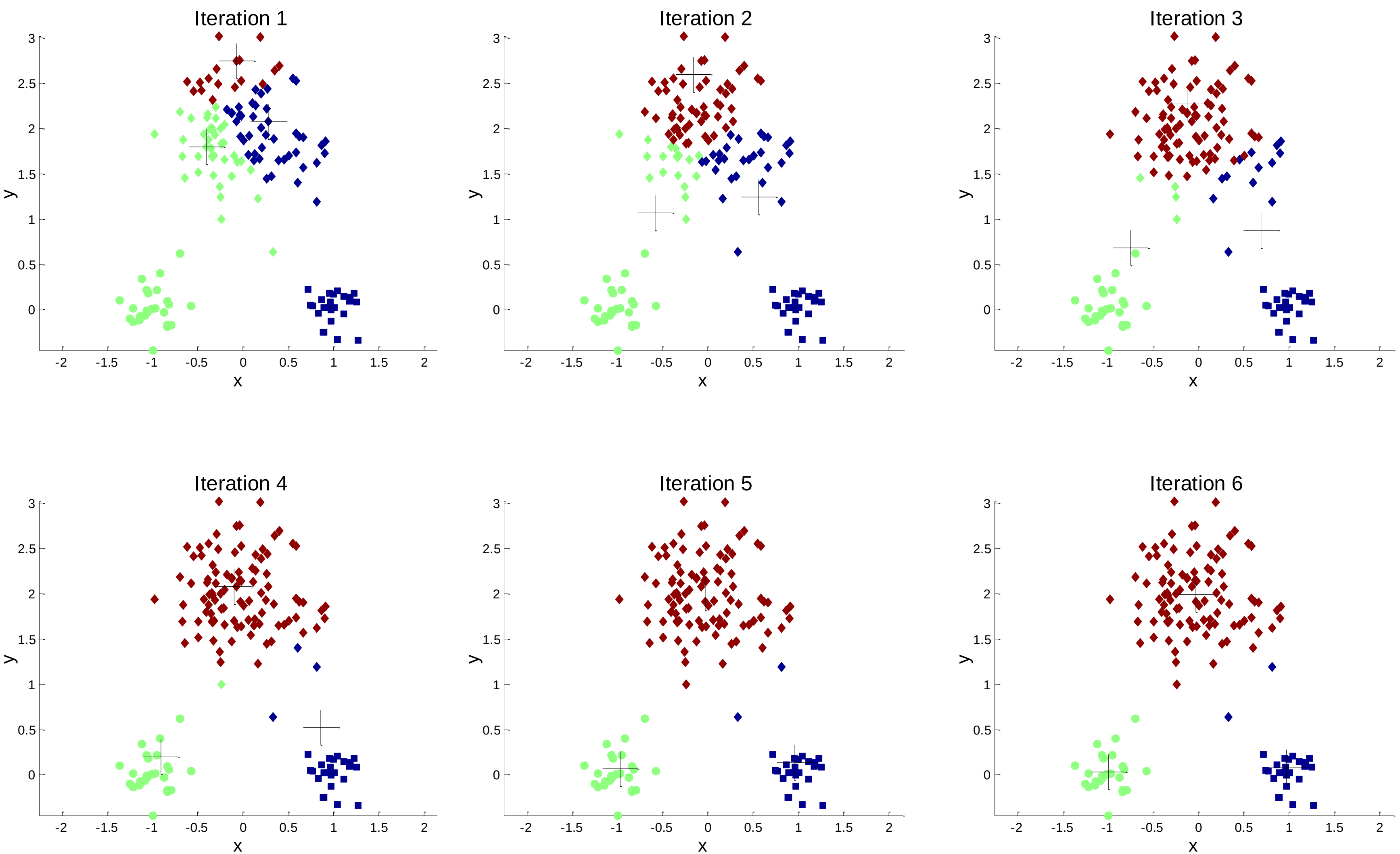


Sub-optimal Clustering

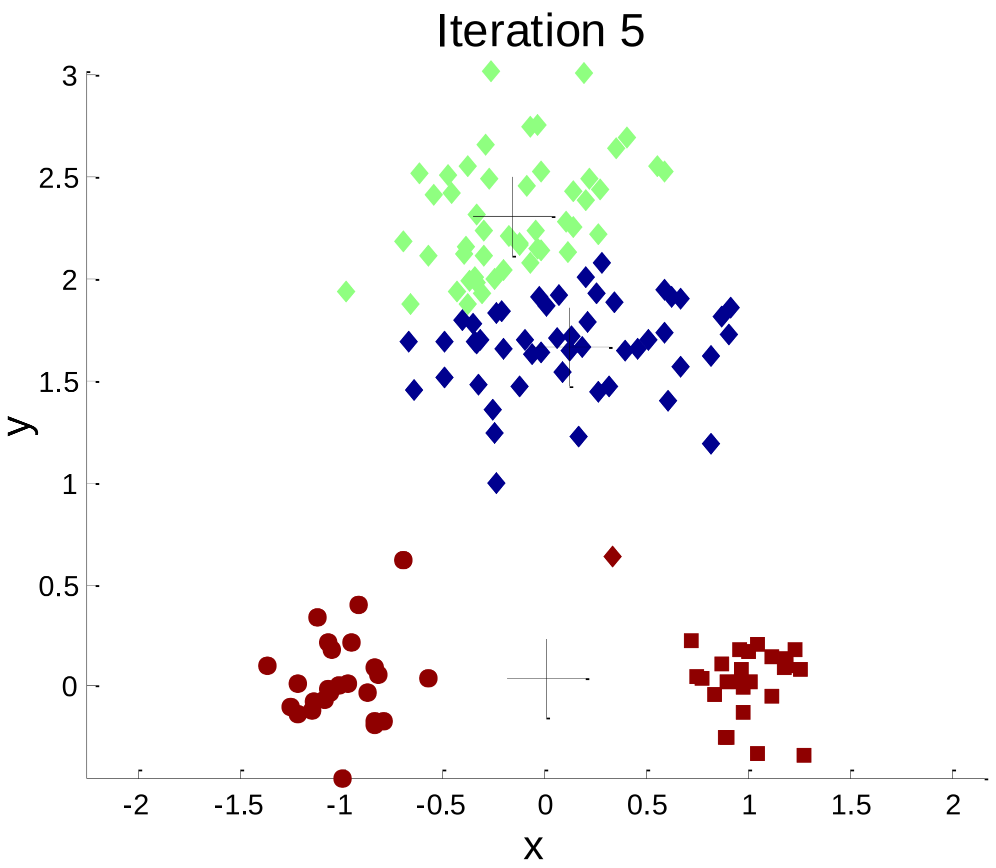
Importance of Choosing Initial Centroids



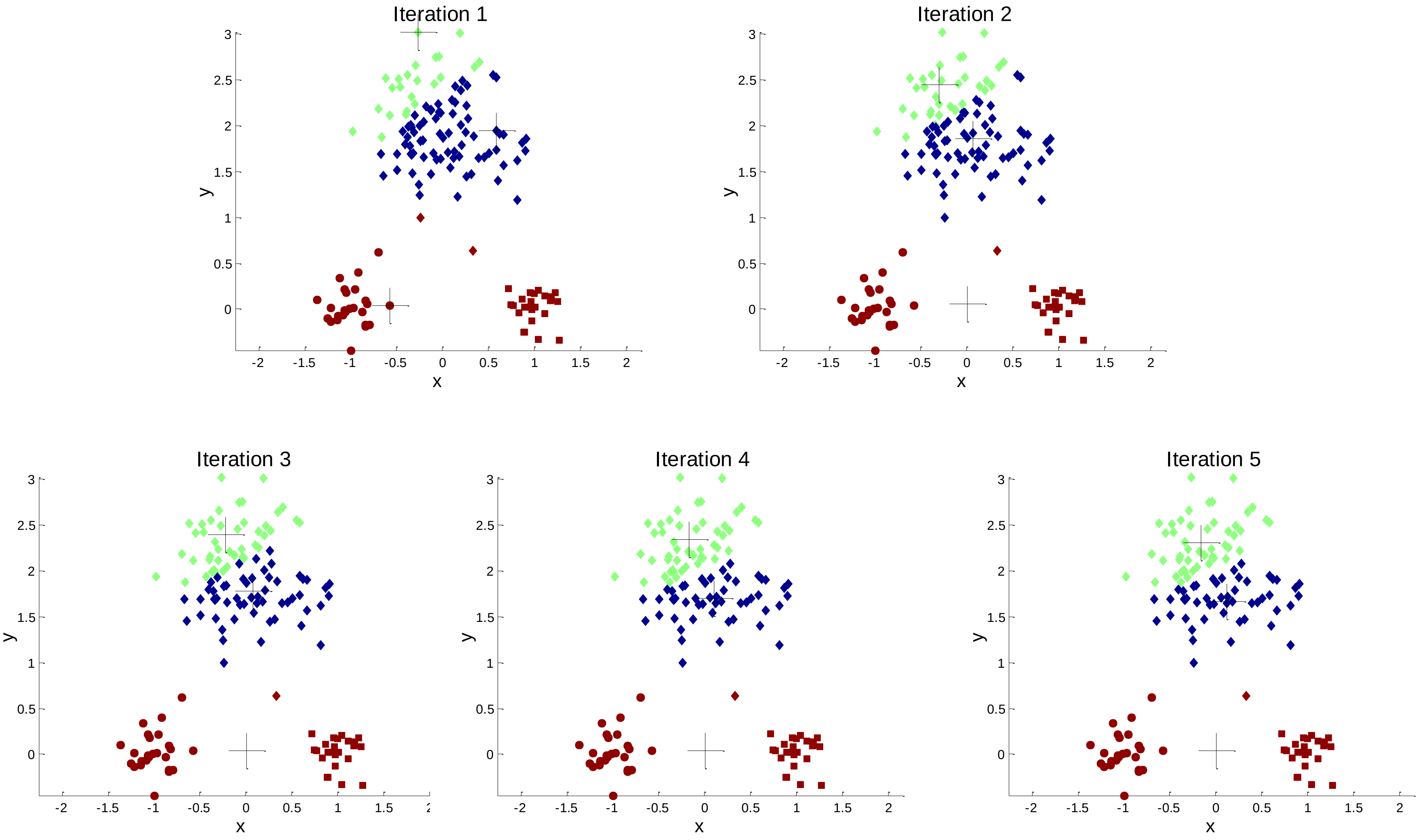
Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids …



Importance of Choosing Initial Centroids …



Problems with Selecting Initial Points

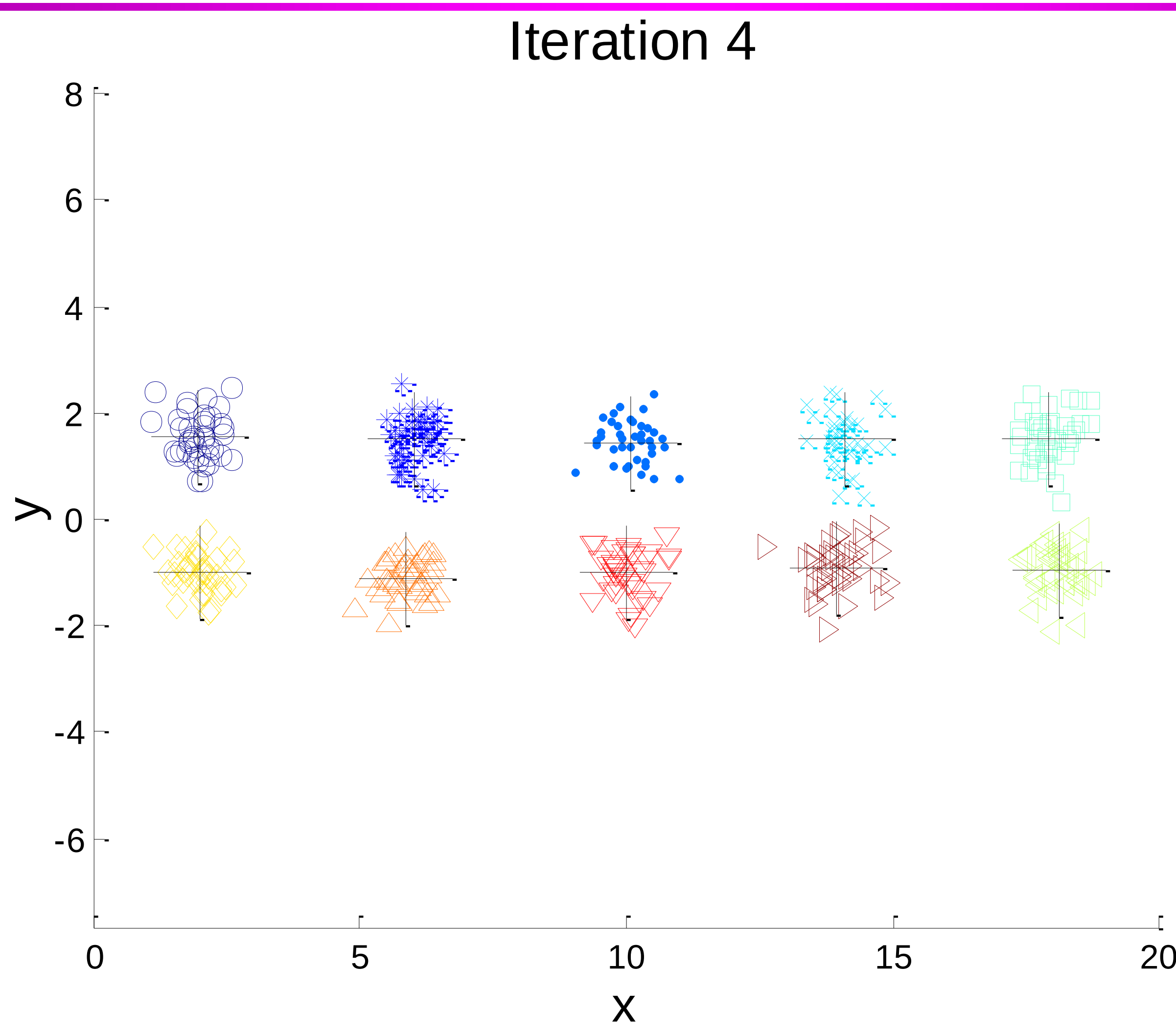
If there are K 'real' clusters then the chance of selecting one centroid from each cluster is small.

- Chance is relatively small when K is large
- If clusters are the same size, n, then

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

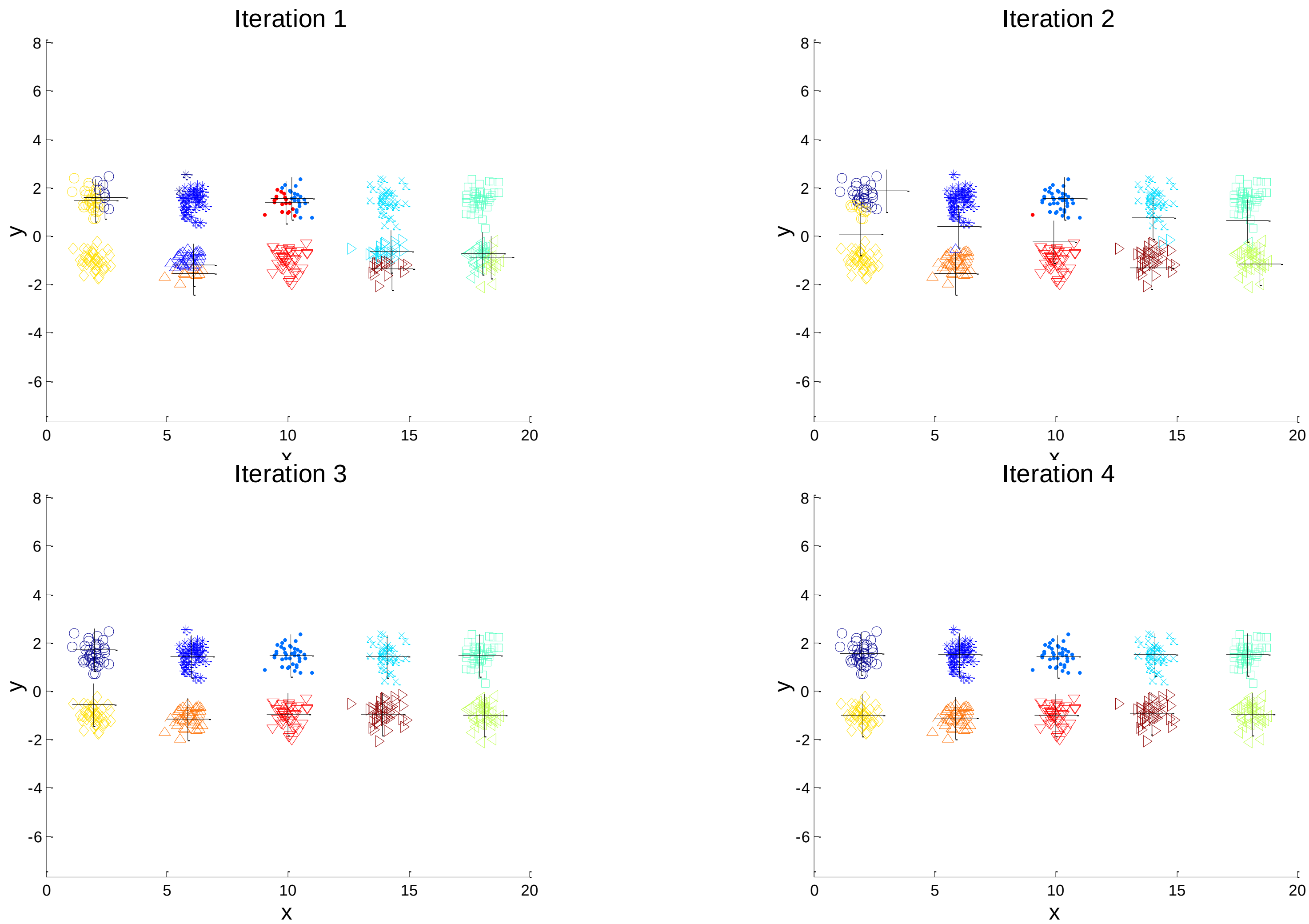
- For example, if K = 10, then probability = $10!/10^{10}$ = 0.00036
- Sometimes the initial centroids will readjust themselves in 'right' way, and sometimes they don't
- Consider an example of five pairs of clusters

10 Clusters Example



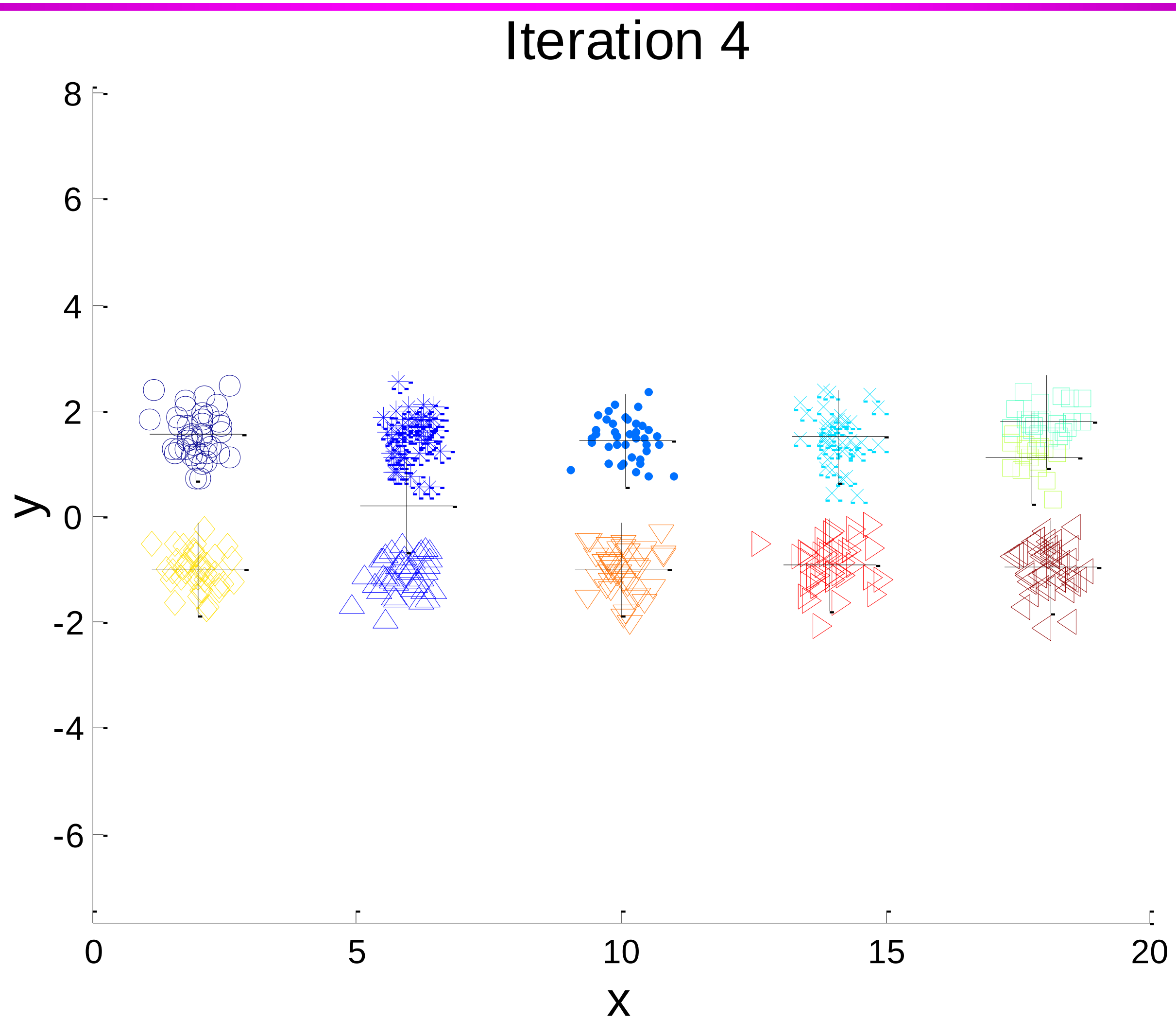
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



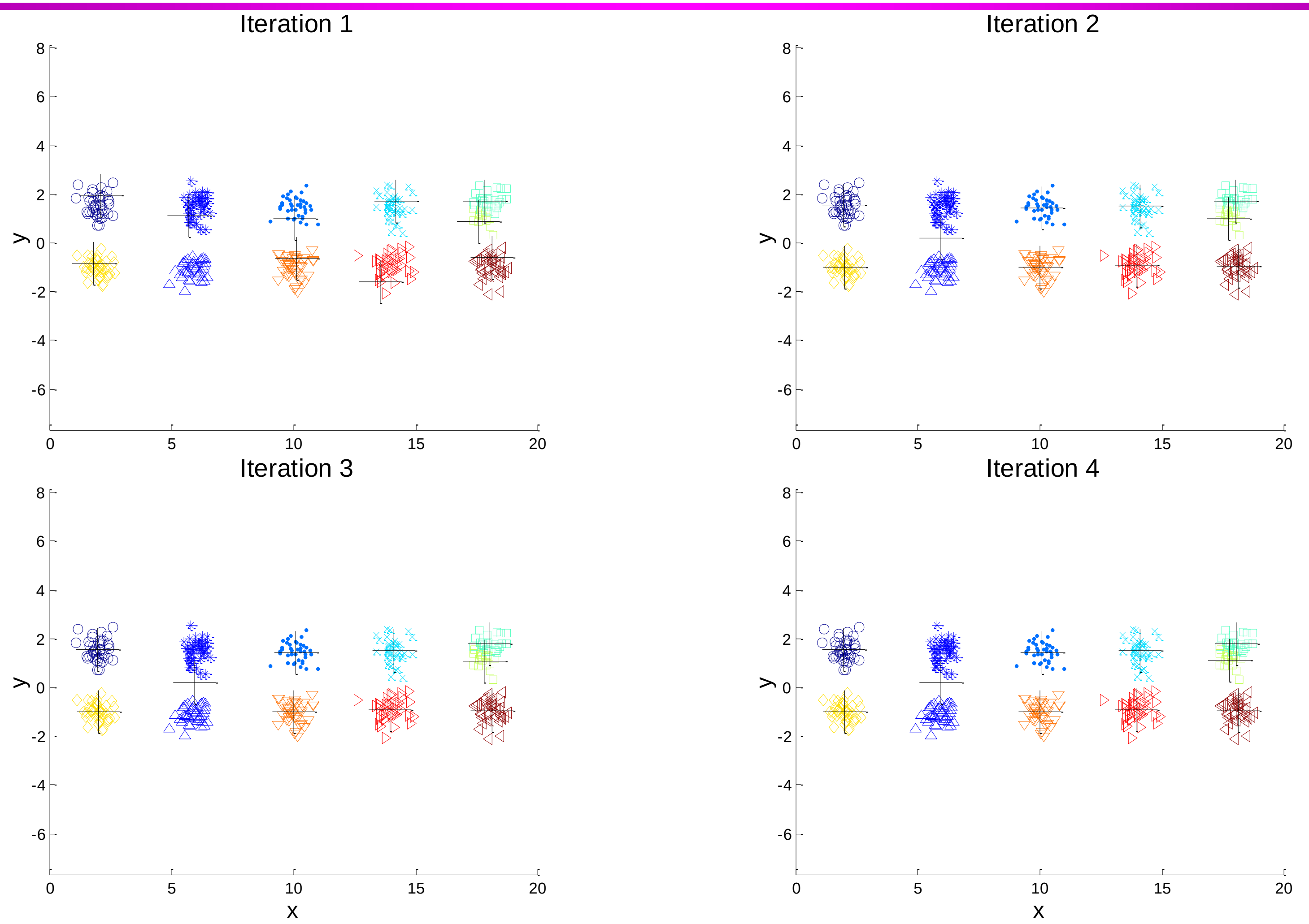
Starting with two initial centroids in one cluster of each pair of clusters

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
  - Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- Select more than k initial centroids and then select among these initial centroids
  - Select most widely separated
- K-means and its variants
- Bisecting K-means
  - Not as susceptible to initialization issues
- Postprocessing

Bisecting K-means

Bisecting K-means algorithm

– Variant of K-means that can produce a partitional or a hierarchical clustering

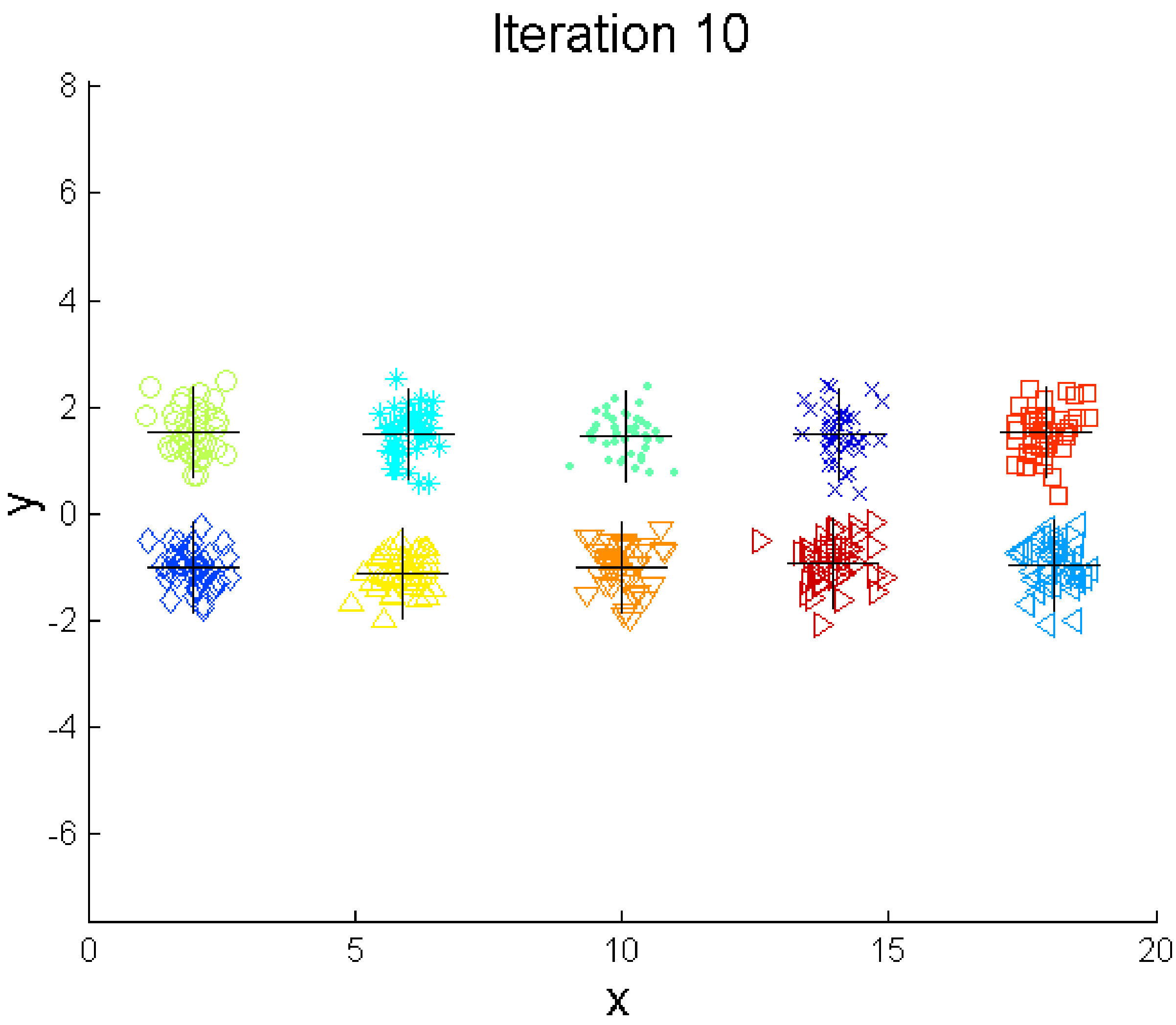
---

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

Sum of Squared Error (SSE)

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

Bisecting K-means Example



Evaluating K-means Clusters

Most common measure is Sum of Squared Error (SSE)

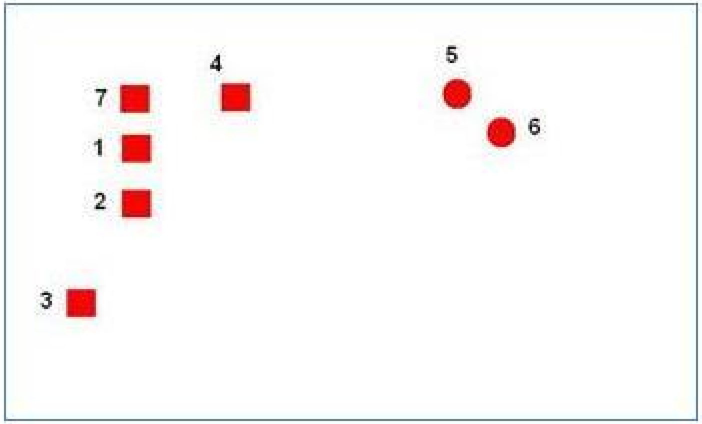
- For each point, the error is the distance to the nearest cluster
- To get SSE, we square these errors and sum them.

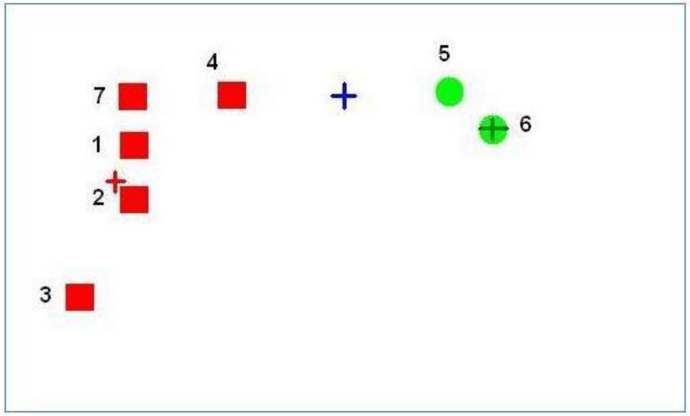
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

- x is a data point in cluster C_i and m_i is the representative point for cluster C_i
  - can show that m_i corresponds to the center (mean) of the cluster
- Given two clusters, we can choose the one with the smallest error
- One easy way to reduce SSE is to increase K, the number of clusters
  - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

Handling Empty Clusters

Basic K-means algorithm can yield empty clusters





Handling Empty Clusters

Basic K-means algorithm can yield empty clusters

Several strategies

- Choose the point that contributes most to SSE
- Choose a point from the cluster with the highest SSE
- If there are several empty clusters, the above can be repeated several times.

Updating Centers Incrementally

In the basic K-means algorithm, centroids are updated after all points are assigned to a centroid

An alternative is to update the centroids after each assignment (incremental approach)

- Each assignment updates zero or two centroids
- More expensive
- Introduces an order dependency
- Never get an empty cluster
- Can use “weights” to change the impact

Pre-processing and Post-processing

- Pre-processing
  - Normalize the data
  - Eliminate outliers
- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are ‘close’ and that have relatively low SSE
  - Can use these steps during the clustering process.

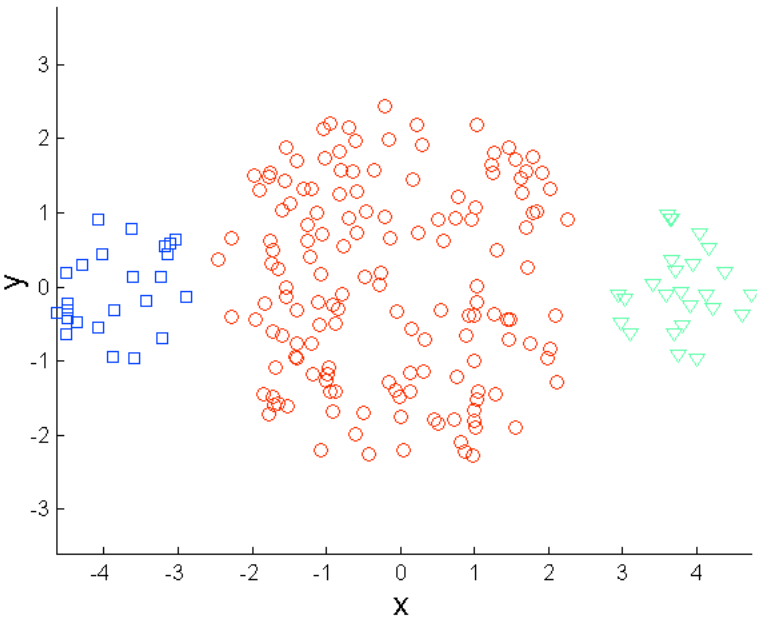
Limitations of K-means

K-means has problems when clusters are of differing

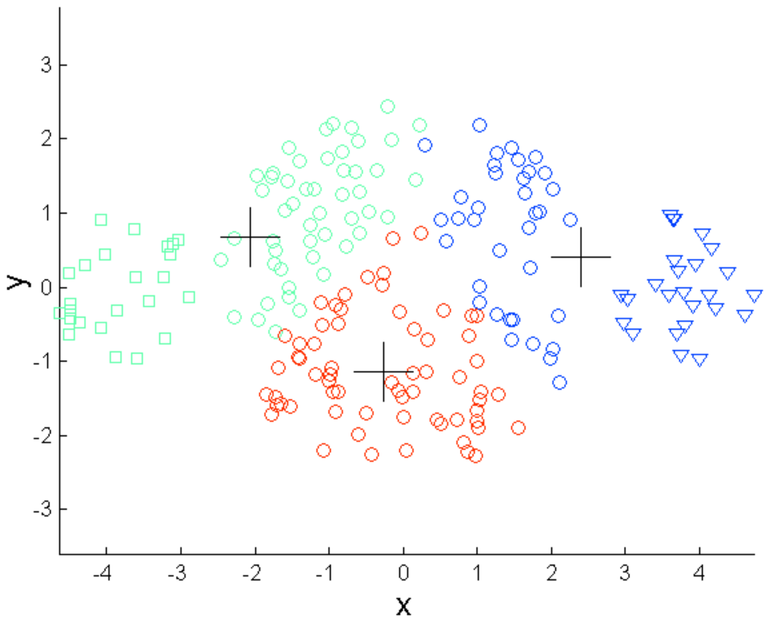
- Sizes
- Densities
- Non-globular shapes

K-means has problems when the data contains outliers.

Limitations of K-means: Differing Sizes

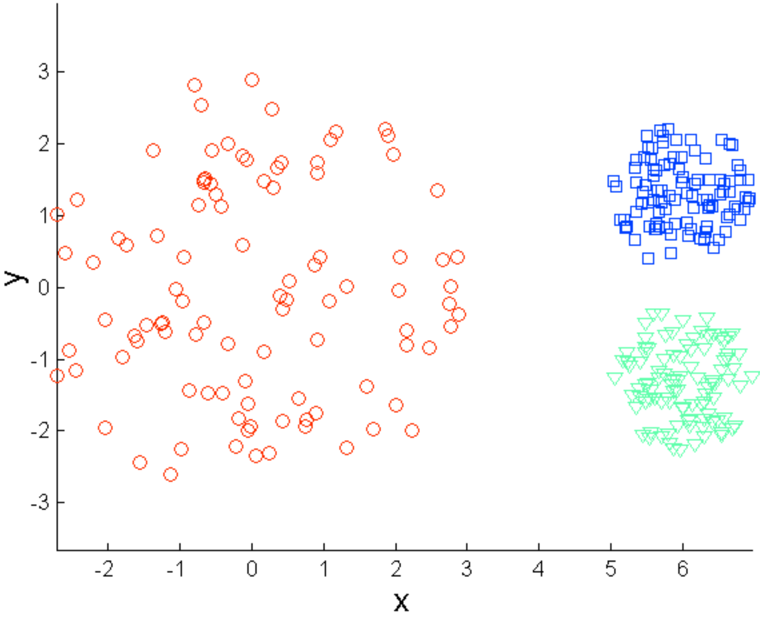


Original Points

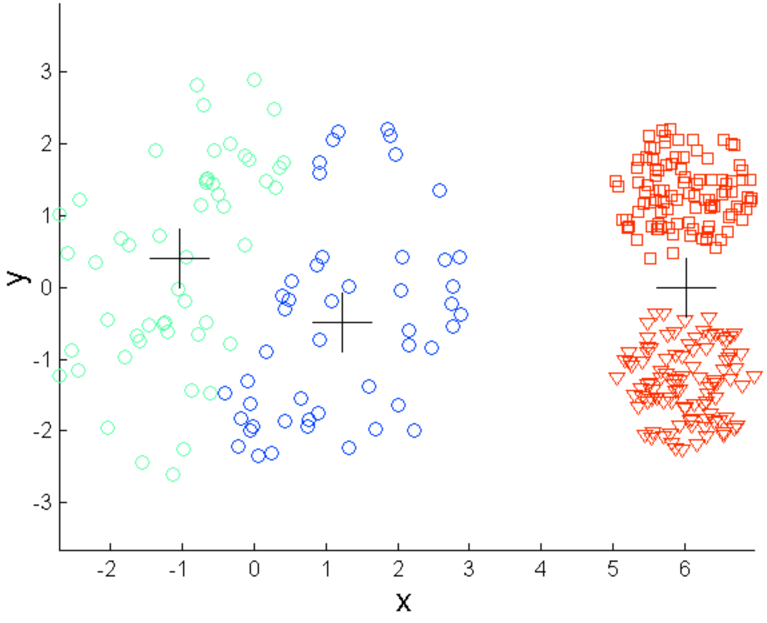


K-means (3 Clusters)

Limitations of K-means: Differing Density

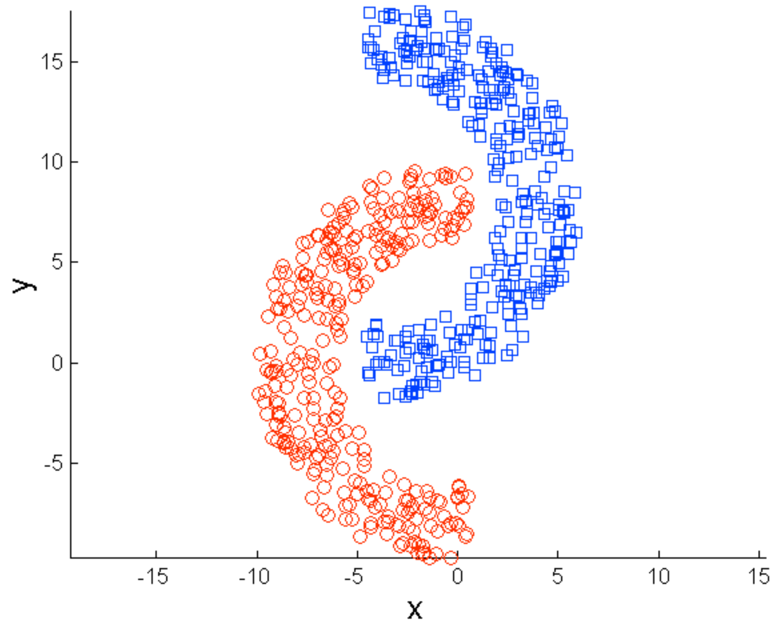


Original Points

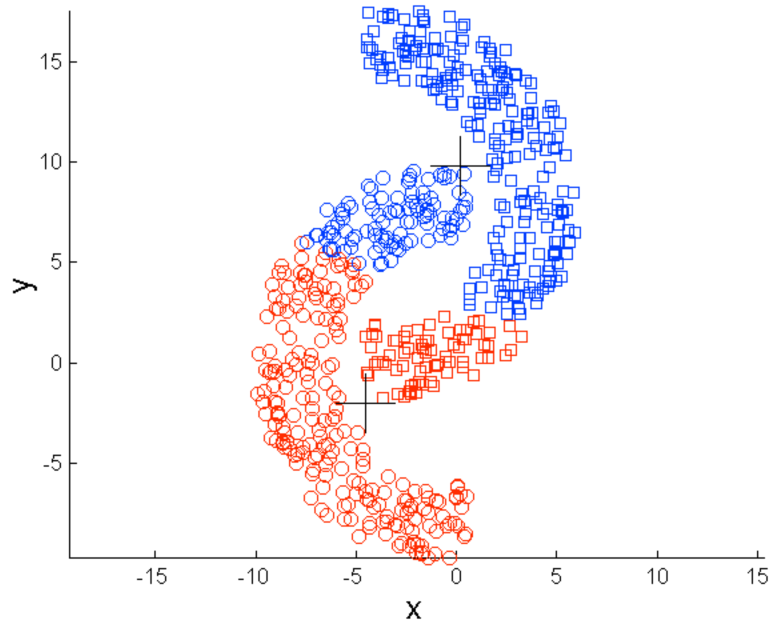


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes

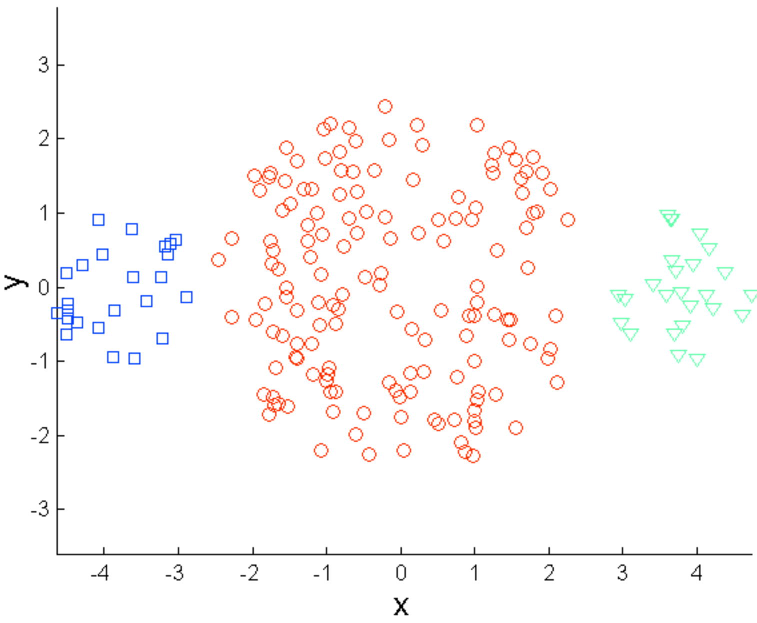


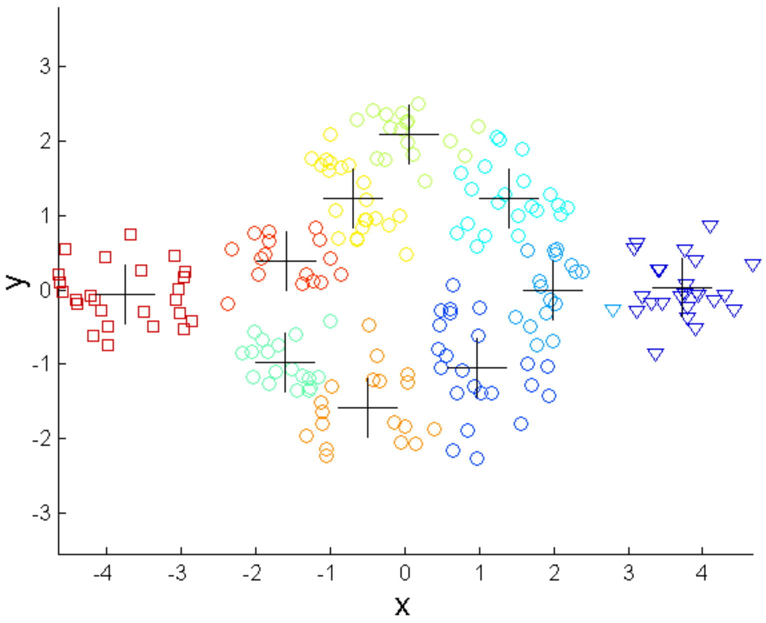
Original Points



K-means (2 Clusters)

Overcoming K-means Limitations



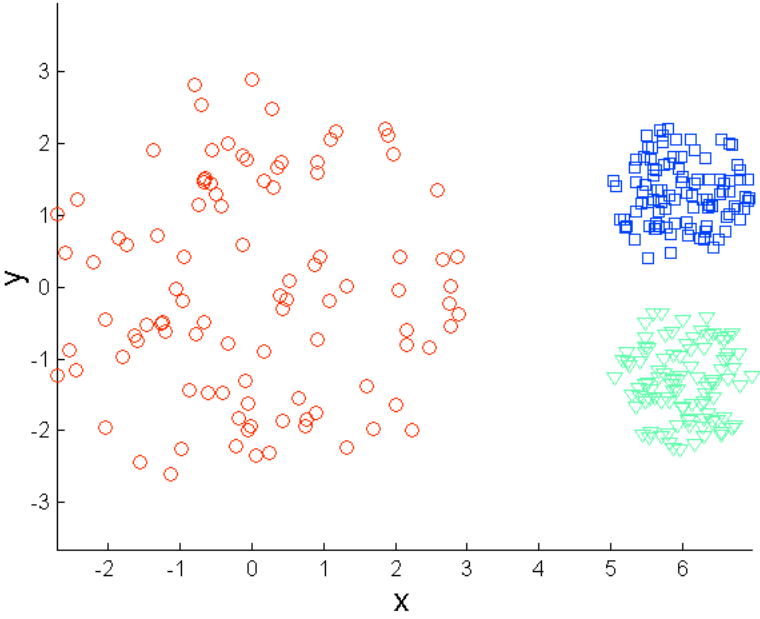


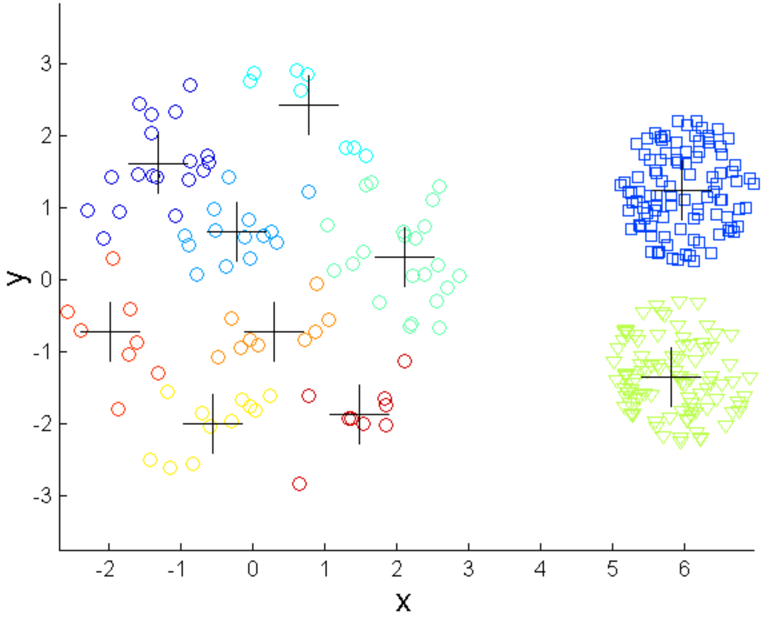
Original Points **K-means Clusters**

One solution is to use many clusters.

Find parts of clusters, but need to put together.

Overcoming K-means Limitations

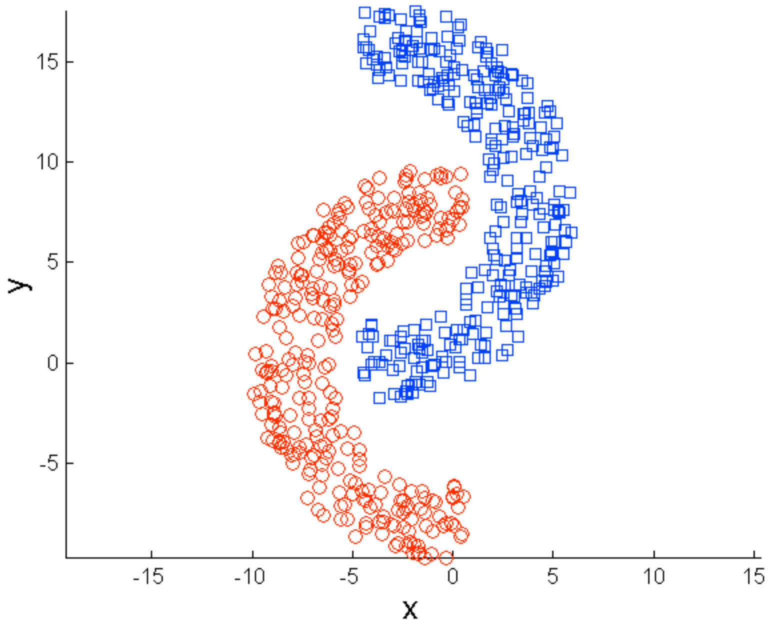


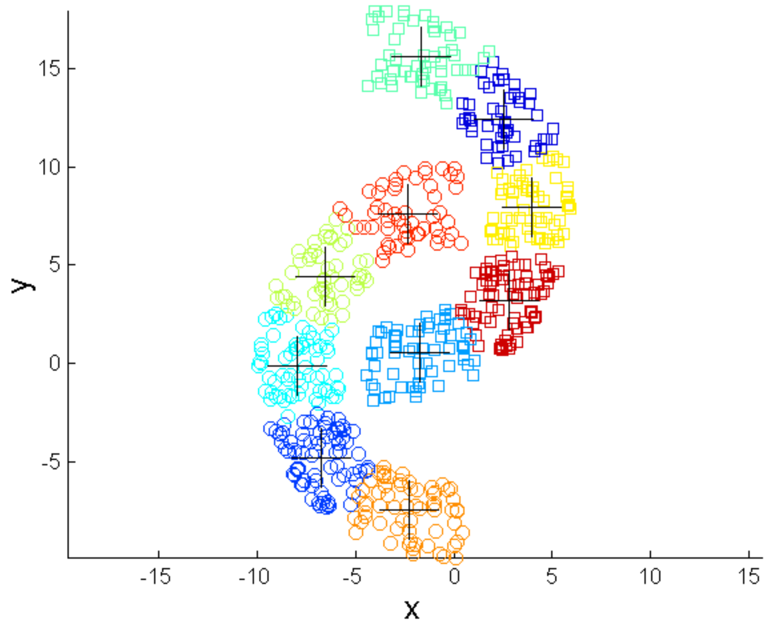


Original Points

K-means Clusters

Overcoming K-means Limitations





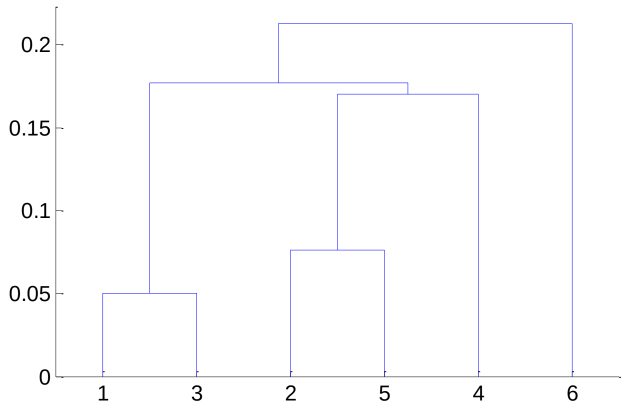
Original Points **K-means Clusters**

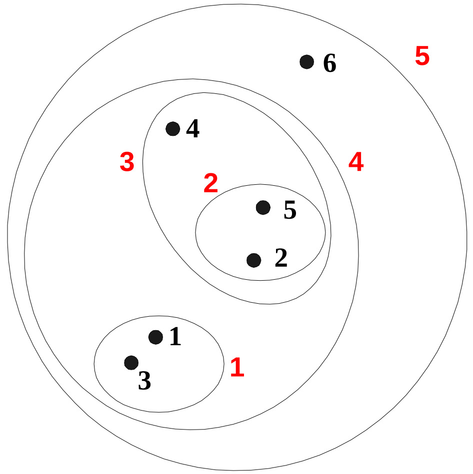
Hierarchical Clustering

Produces a set of nested clusters organized as a hierarchical tree

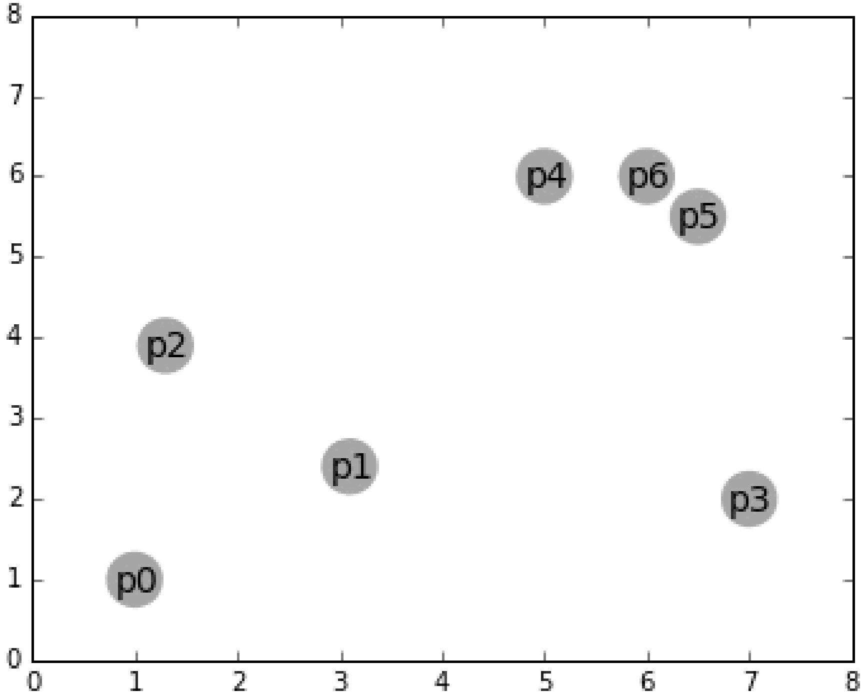
Can be visualized as a dendrogram(树图)

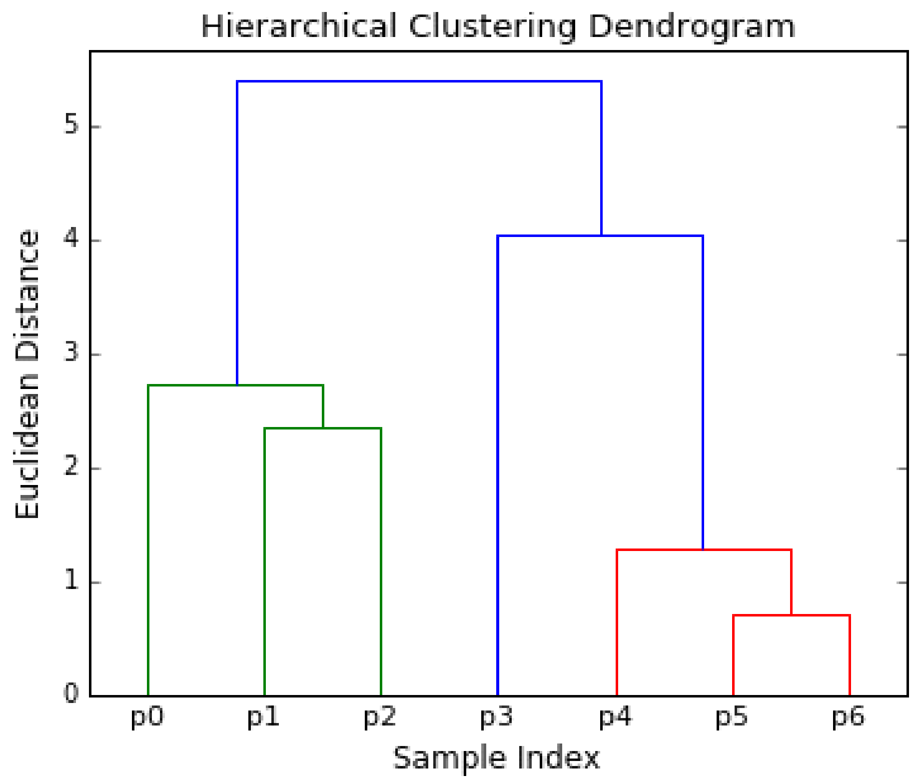
– A tree like diagram that records the sequences of merges or splits





Hierarchical Clustering





Strengths of Hierarchical Clustering

Do not have to assume any particular number of clusters

- Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level

They may correspond to meaningful taxonomies

- Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Hierarchical Clustering

Two main types of hierarchical clustering

- Agglomerative (凝聚):
  - Start with the points as individual clusters
  - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
- Divisive(分裂):
  - Start with one, all-inclusive cluster
  - At each step, split a cluster until each cluster contains a point (or there are k clusters)

Traditional hierarchical algorithms use a similarity or distance matrix

- Merge or split one cluster at a time

Agglomerative Clustering Algorithm

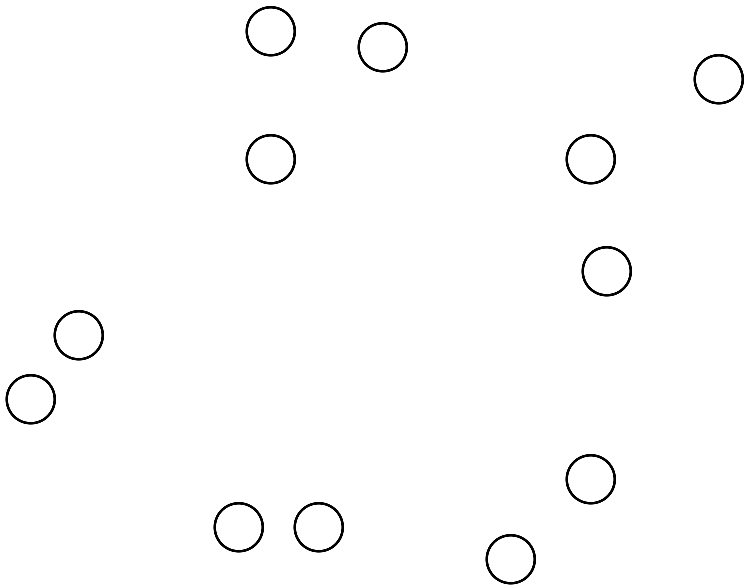
- More popular hierarchical clustering technique
- Basic algorithm is straightforward

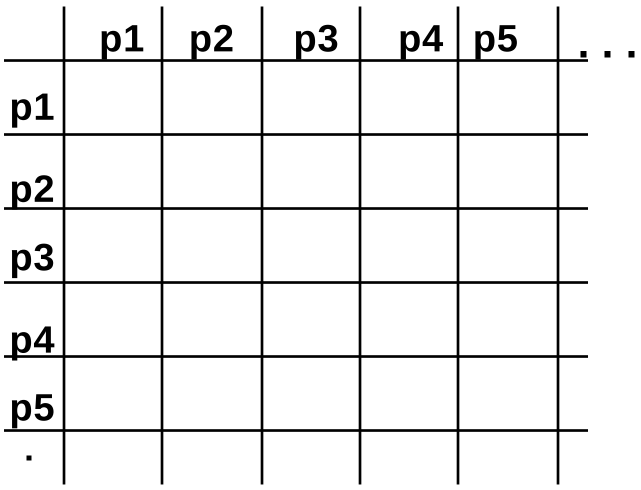
1. Compute the proximity matrix
2. Let each data point be a cluster
3. **Repeat**
4. Merge the two closest clusters
5. Update the proximity matrix
6. **Until** only a single cluster remains

- Key operation is the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms

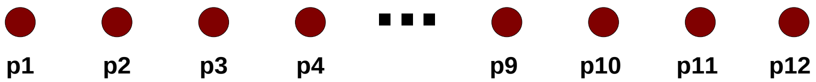
Starting Situation

Start with clusters of individual points and a proximity matrix





Proximity Matrix

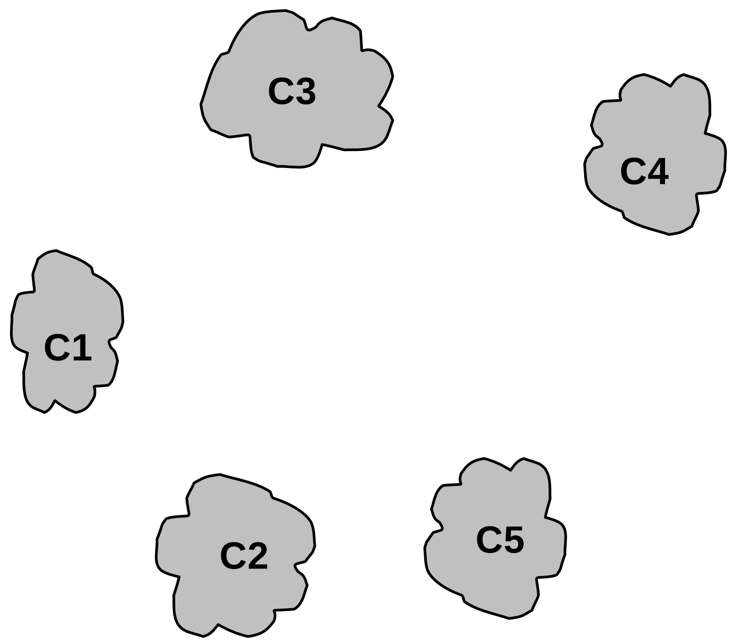


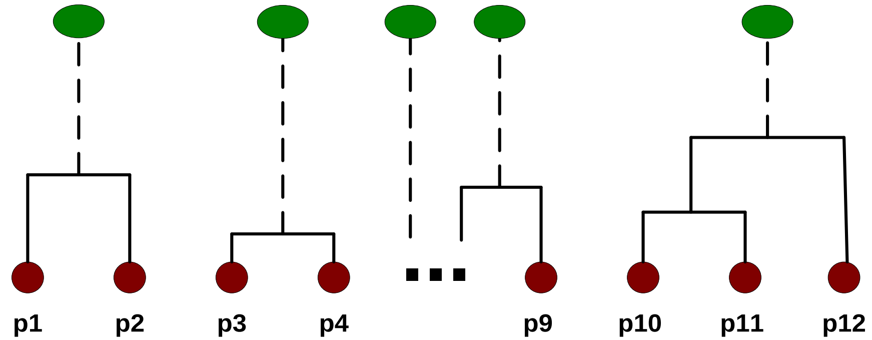
Intermediate Situation

After some merging steps, we have some clusters

	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

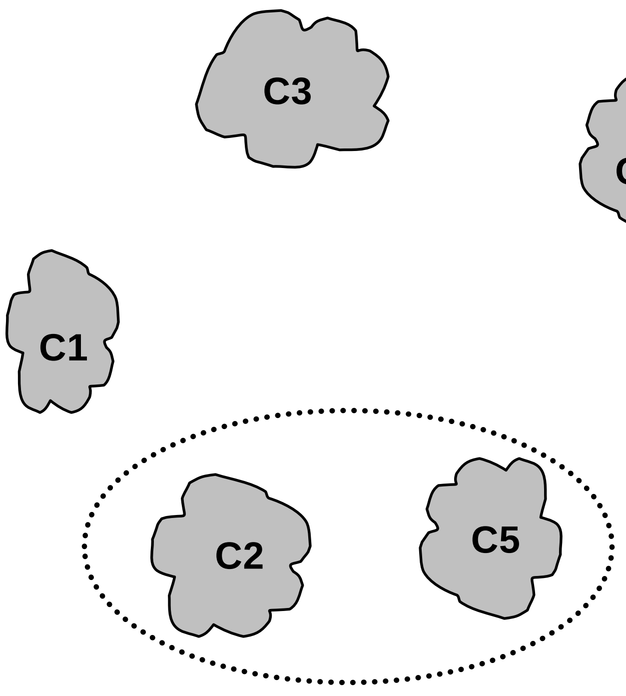
Proximity Matrix

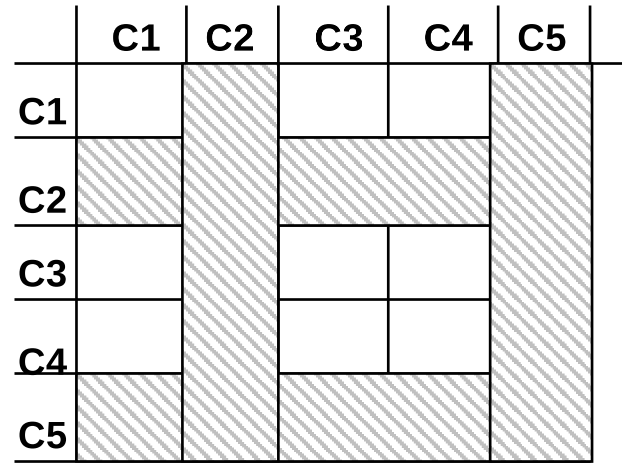




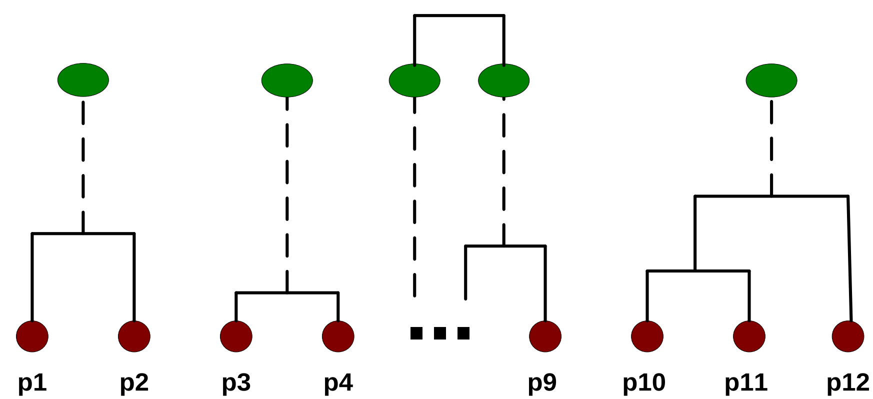
Intermediate Situation

We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.



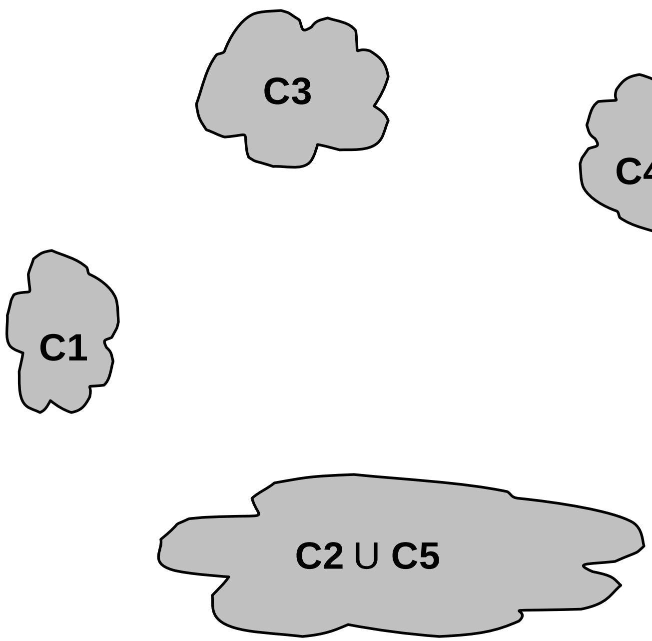


Proximity Matrix



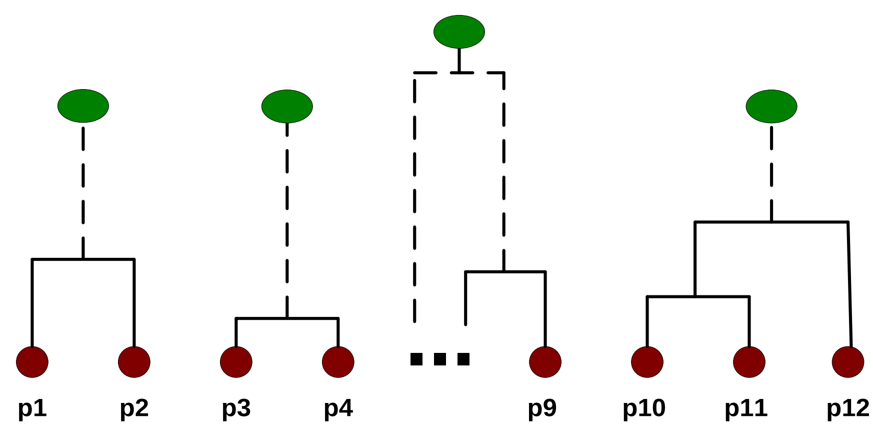
After Merging

The question is “How do we update the proximity matrix?”

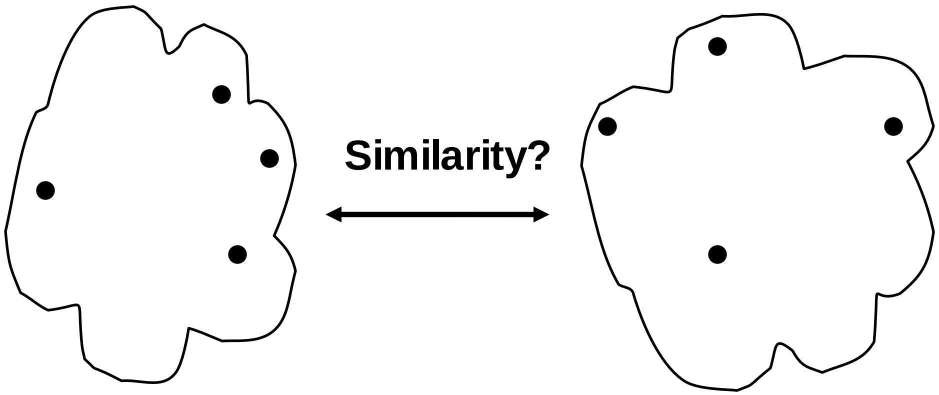


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Similarity

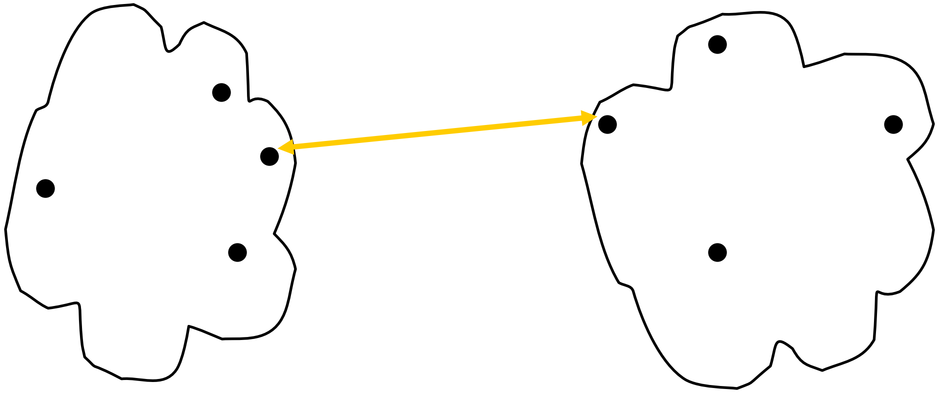


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

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

How to Define Inter-Cluster Similarity

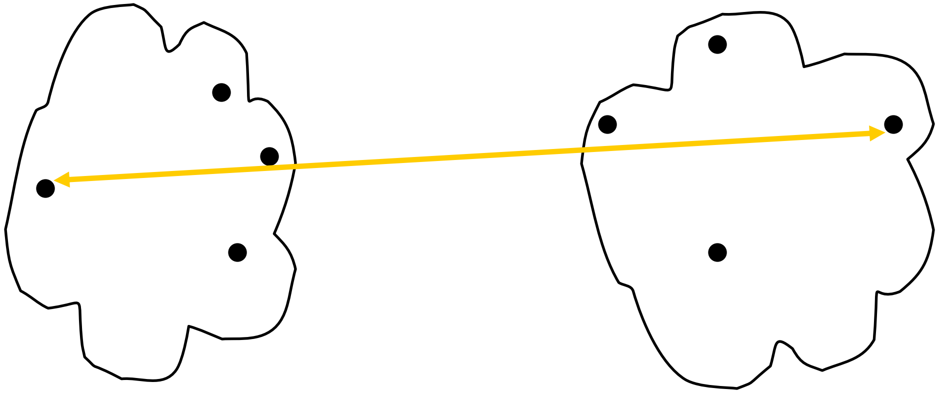


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

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

How to Define Inter-Cluster Similarity



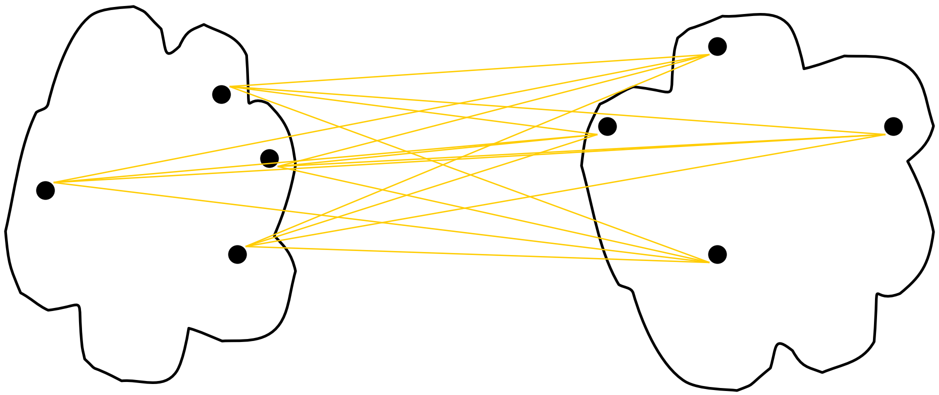
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						

.
.
.

Proximity Matrix

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

How to Define Inter-Cluster Similarity

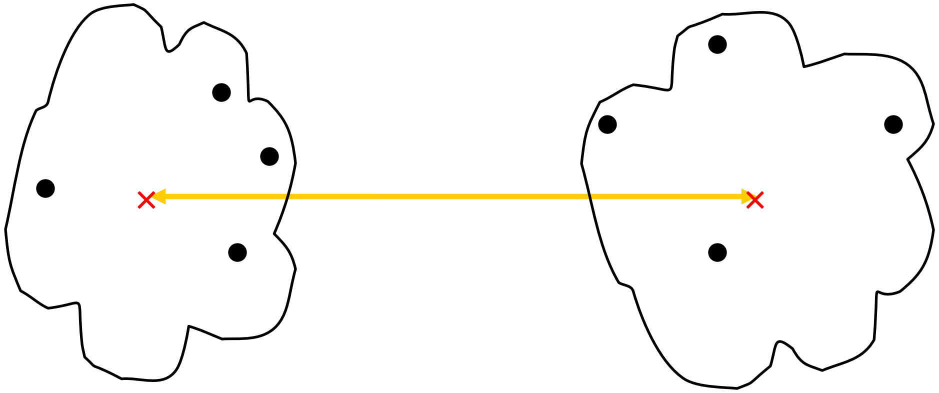


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

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

How to Define Inter-Cluster Similarity



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

Proximity Matrix

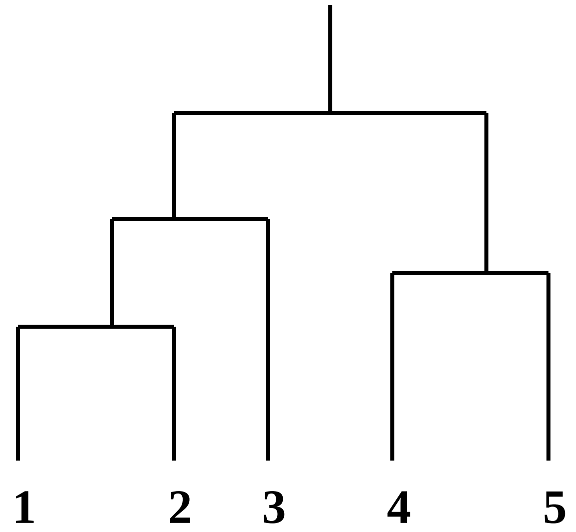
- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

Cluster Similarity: MIN or Single Link

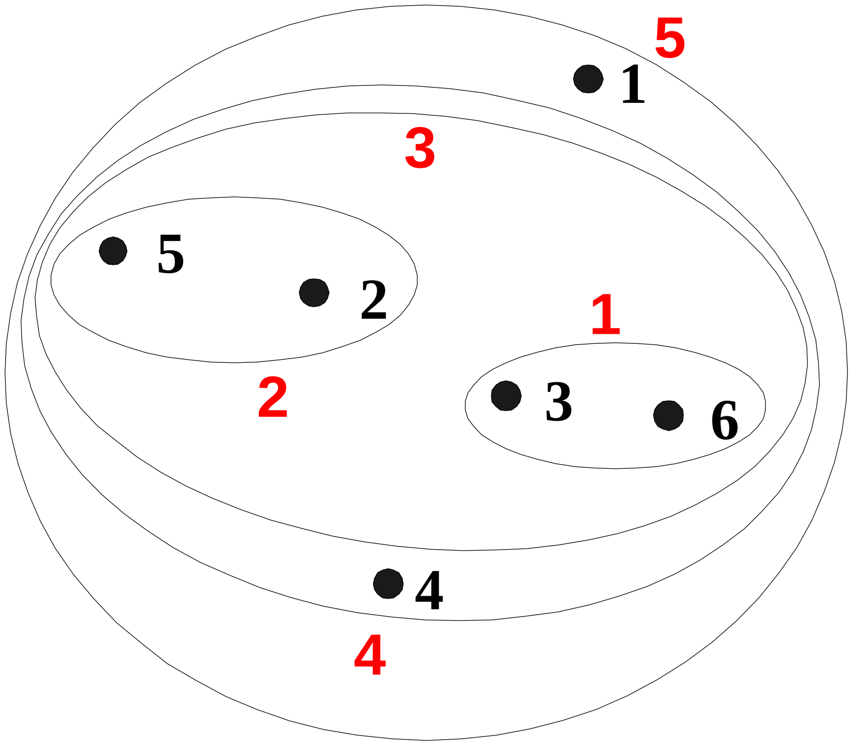
Similarity of two clusters is based on the two most similar (closest) points in the different clusters

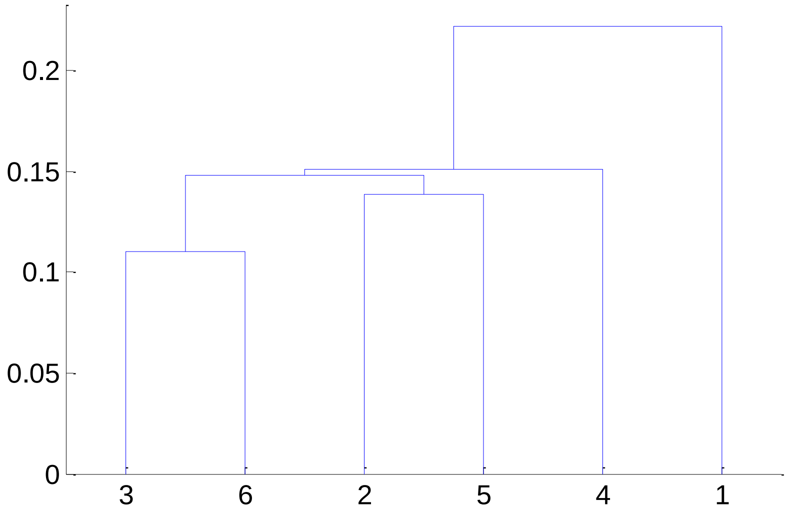
– Determined by one pair of points, i.e., by one link in the proximity graph.

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MIN

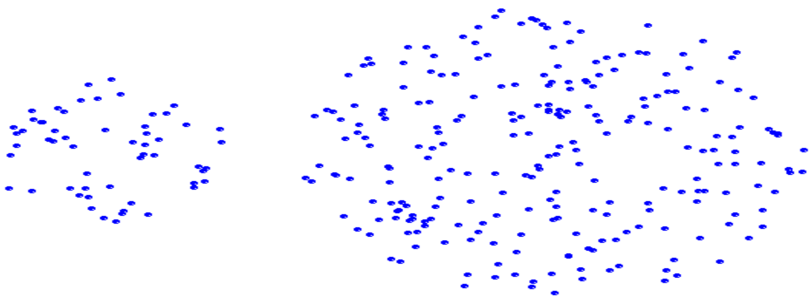




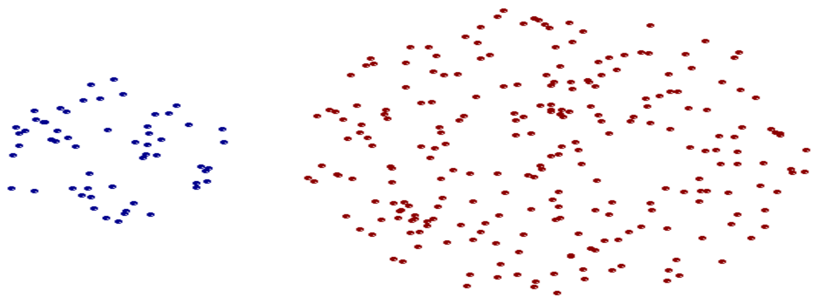
Nested Clusters

Dendrogram

Strength of MIN



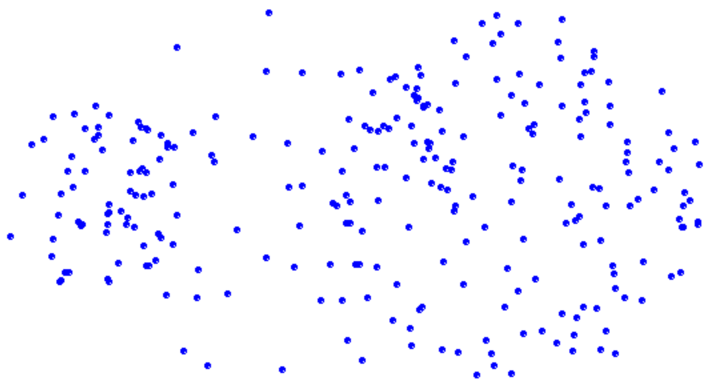
Original Points



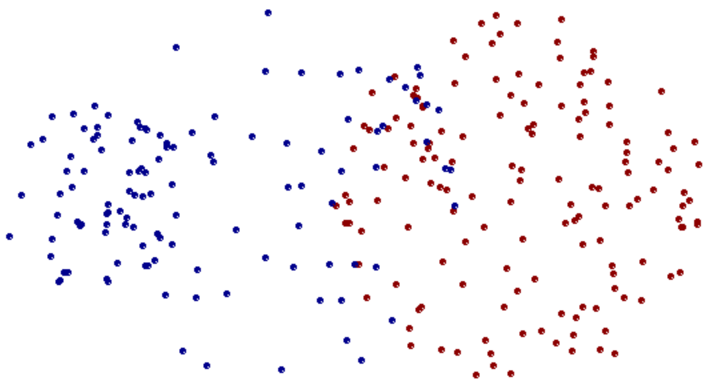
Two Clusters

• Can handle non-elliptical shapes

Limitations of MIN



Original Points



Two Clusters

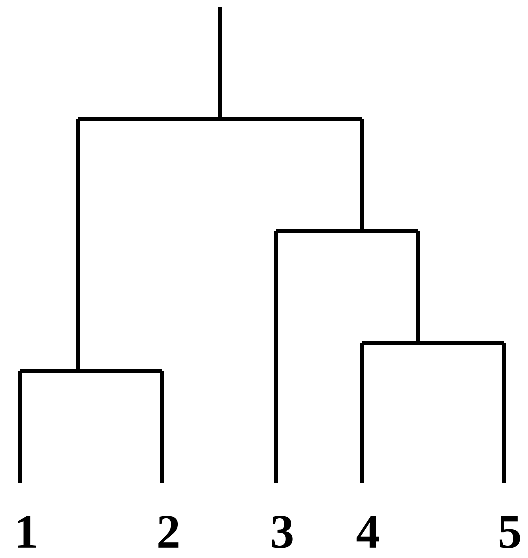
• **Sensitive to noise and outliers**

Cluster Similarity: MAX or Complete Linkage

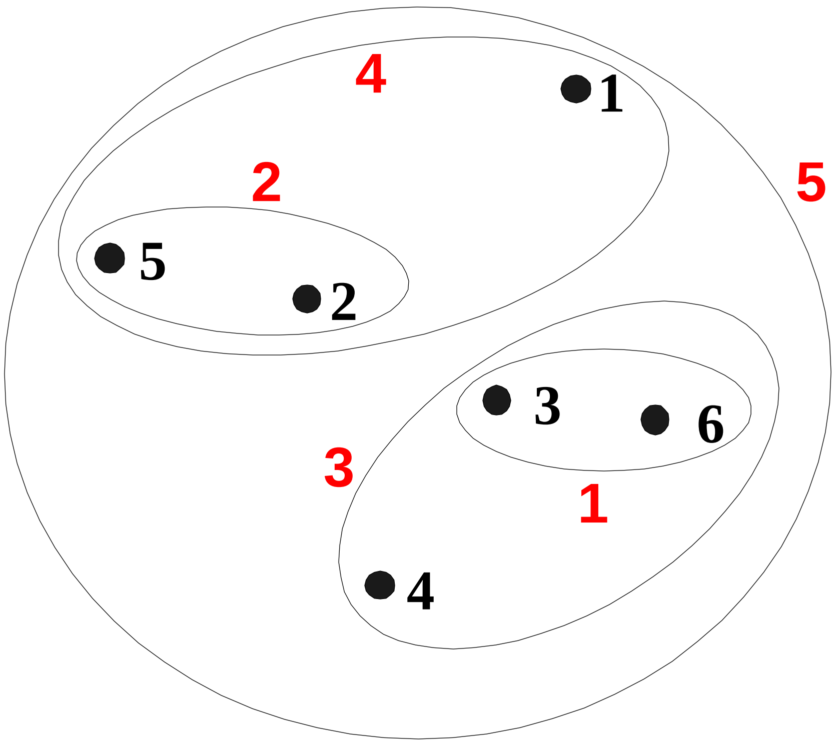
Similarity of two clusters is based on the two least similar (most distant) points in the different clusters

– Determined by all pairs of points in the two clusters

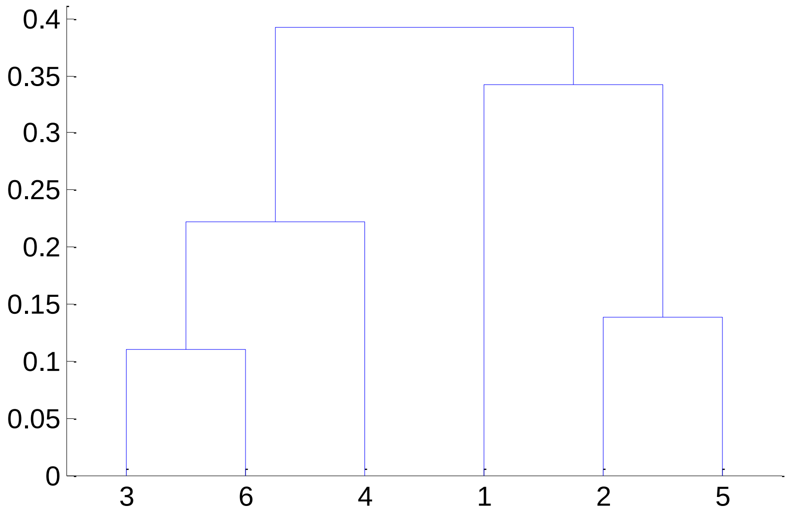
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MAX

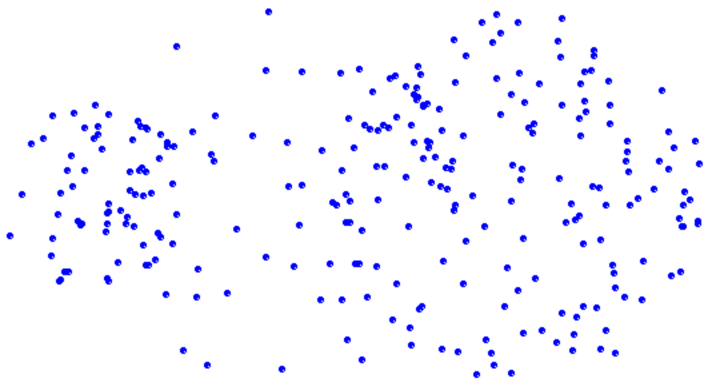


Nested Clusters

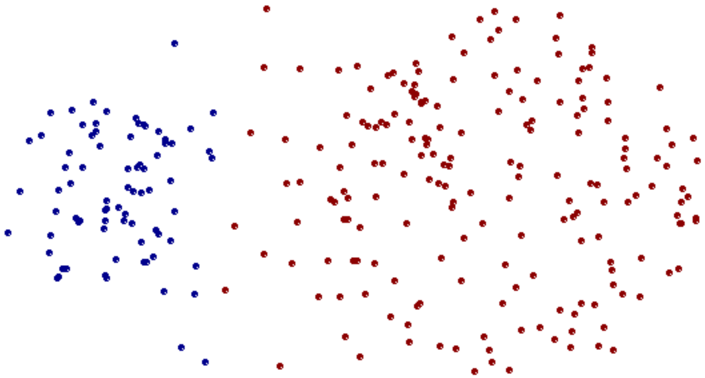


Dendrogram

Strength of MAX



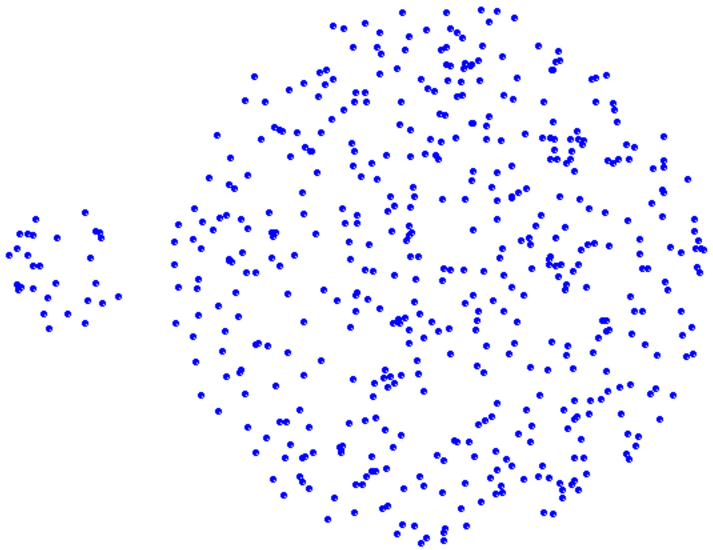
Original Points



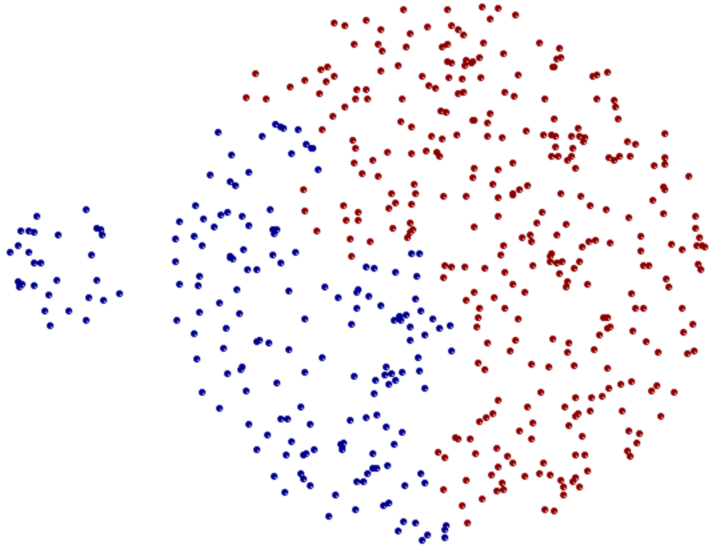
Two Clusters

- Less susceptible to noise and outliers

Limitations of MAX



Original Points



Two Clusters

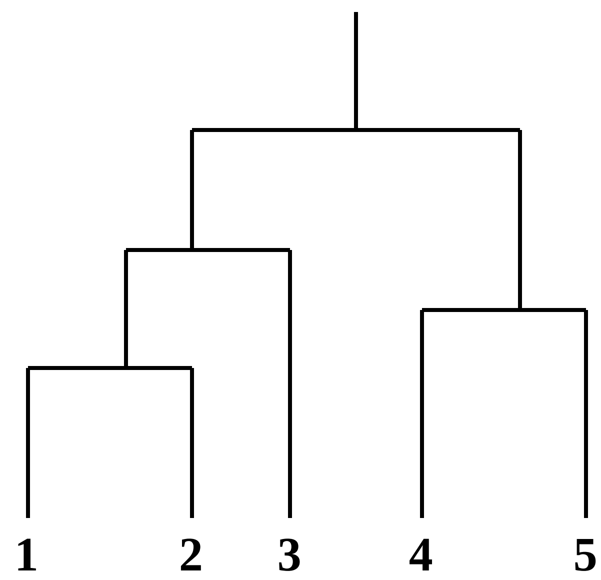
- Tends to break large clusters
- Biased towards globular clusters

Cluster Similarity: Group Average

Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

$$\mathbf{proximity}(\mathbf{Cluster}_i, \mathbf{Cluster}_j) = \frac{\sum\limits_{\substack{p_i \in \mathbf{Cluster}_i \\ p_j \in \mathbf{Cluster}_j}} \mathbf{proximity}(p_i, p_j)}{|\mathbf{Cluster}_i| * |\mathbf{Cluster}_j|}$$

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: Group Average

Compromise between Single and Complete Link

Strengths

– Less susceptible to noise and outliers

Limitations

– Biased towards globular clusters

Cluster Similarity: Ward's Method

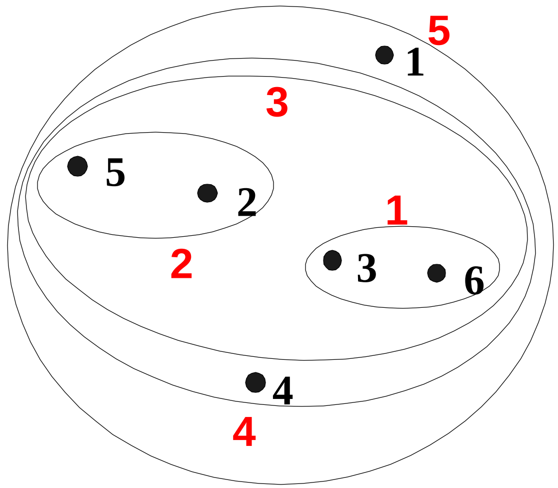
- Similarity of two clusters is based on the increase in squared error when two clusters are merged
  - Similar to group average if distance between points is distance squared

- Less susceptible to noise and outliers

- Biased towards globular clusters

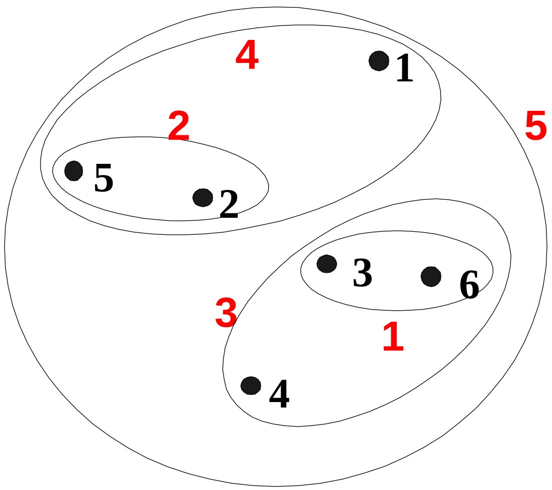
- Hierarchical analogue of K-means
  - Can be used to initialize K-means

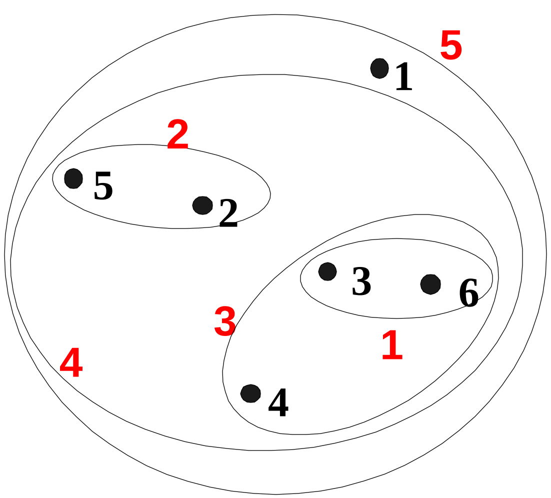
Hierarchical Clustering: Comparison



MIN

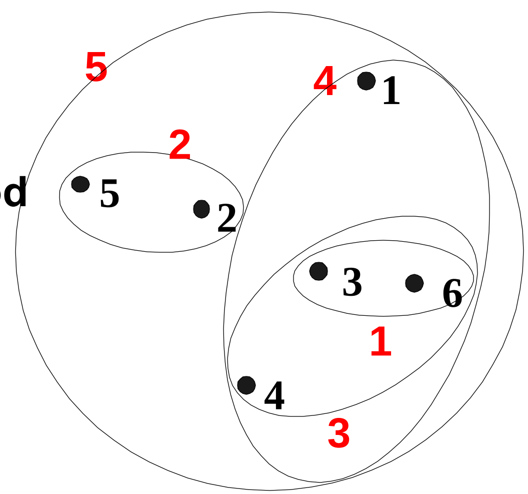
MAX





Group Average

Ward's Method



Hierarchical Clustering: Time and Space requirements

$O(N^2)$ space since it uses the proximity matrix.

- N is the number of points.

$O(N^3)$ time in many cases

- There are N steps and at each step the size, N^2, proximity matrix must be updated and searched
- Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches

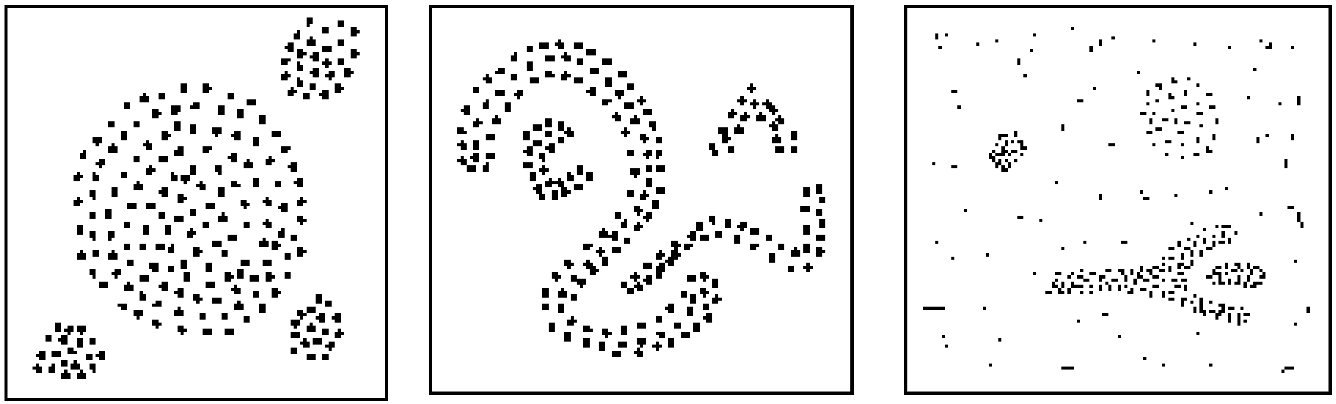
Hierarchical Clustering: Problems and Limitations

Once a decision is made to combine two clusters, it cannot be undone

Different schemes have problems with one or more of the following:

- Sensitivity to noise and outliers
- Difficulty handling different sized clusters and convex shapes
- Breaking large clusters

Density-Based Clustering



Clustering based on density (local cluster criterion), such as density-connected points

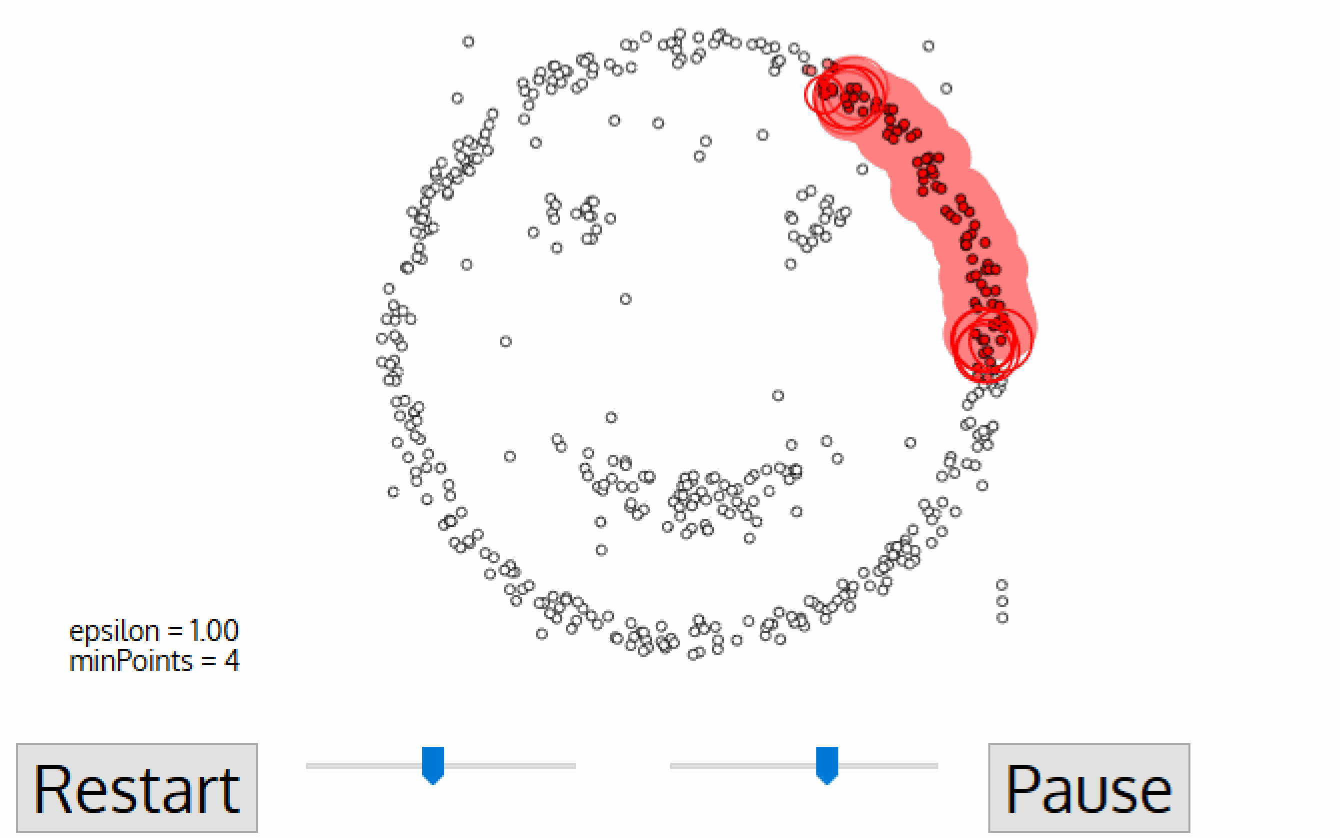
Each cluster has a considerable higher density of points than outside of the cluster

Density-Based Clustering

Major features:
- Discover clusters of arbitrary shape
- Handle noise
- One scan

Several interesting studies:
- **DBSCAN**: **Ester, et al.** (KDD'**96**)
- GDBSCAN: Sander, et al. (KDD'98)
- OPTICS: Ankerst, et al (SIGMOD'99).
- DENCLUE: Hinneburg & D. Keim (KDD'98)
- CLIQUE: Agrawal, et al. (SIGMOD'98)

Density-Based Clustering



DBSCAN

DBSCAN is a density-based algorithm.

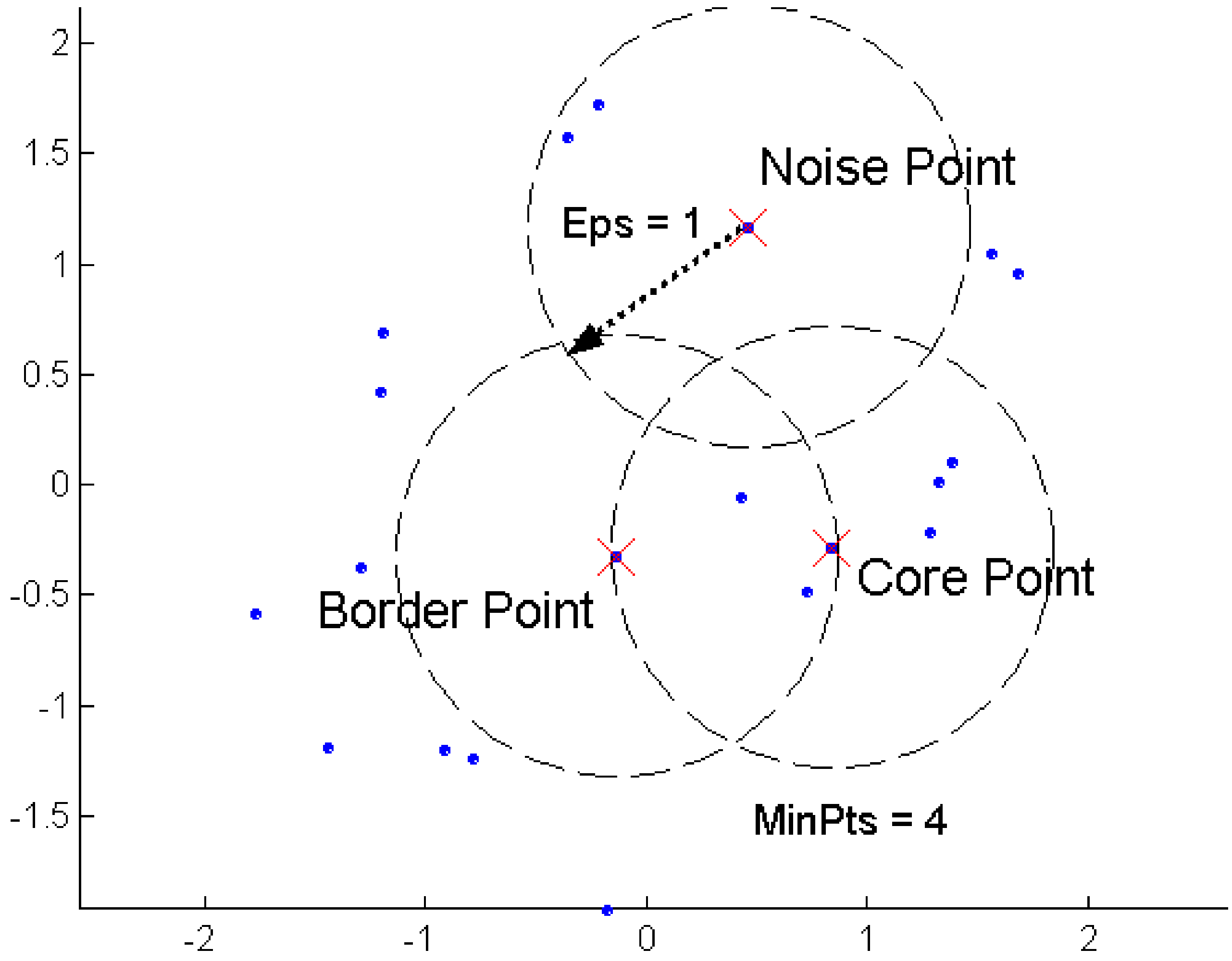
- Density = number of points within a specified radius (Eps)

- A point is a core point if it has more than a specified number of points (MinPts) within Eps
  - These are points that are at the interior of a cluster
- A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point

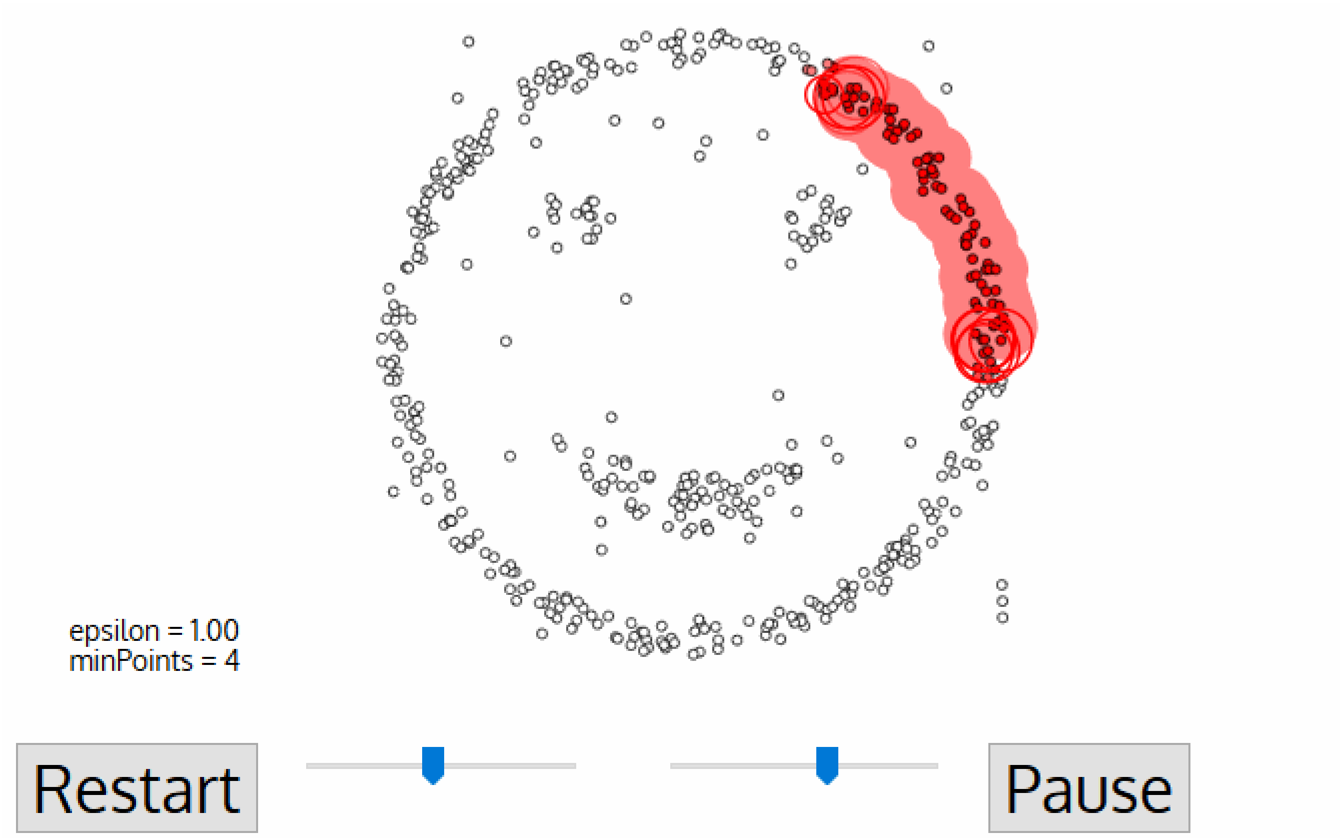
- A noise point is any point that is not a core point or a border point.

- does not require one to specify the number of clusters.
- requires few parameters .

DBSCAN: Core, Border, and Noise Points



Density-Based Clustering



DBSCAN Algorithm

Eliminate noise points

Perform clustering on the remaining points

$current\_cluster\_label \leftarrow 1$

for all core points **do**

  if the core point has no cluster label **then**

    $current\_cluster\_label \leftarrow current\_cluster\_label + 1$

    Label the current core point with cluster label $current\_cluster\_label$

  end if

  for all points in the Eps-neighborhood, except i^{th} the point itself **do**

    if the point does not have a cluster label **then**

      Label the point with cluster label $current\_cluster\_label$

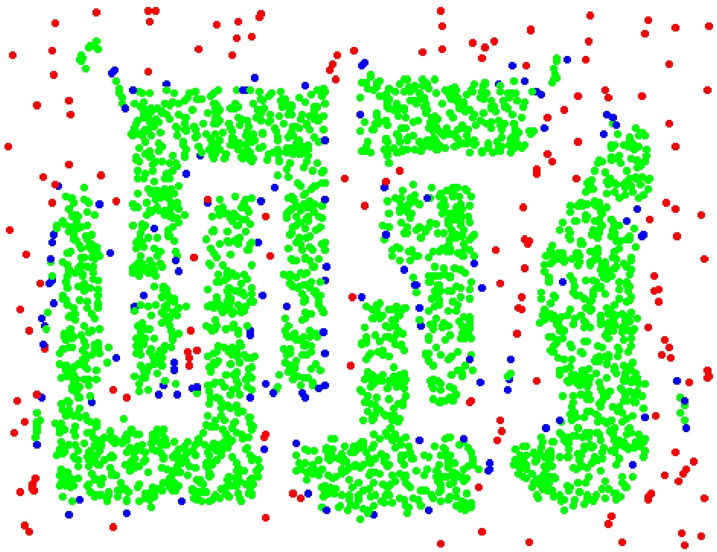
    end if

  end for

end for

DBSCAN: Core, Border and Noise Points

Original Points

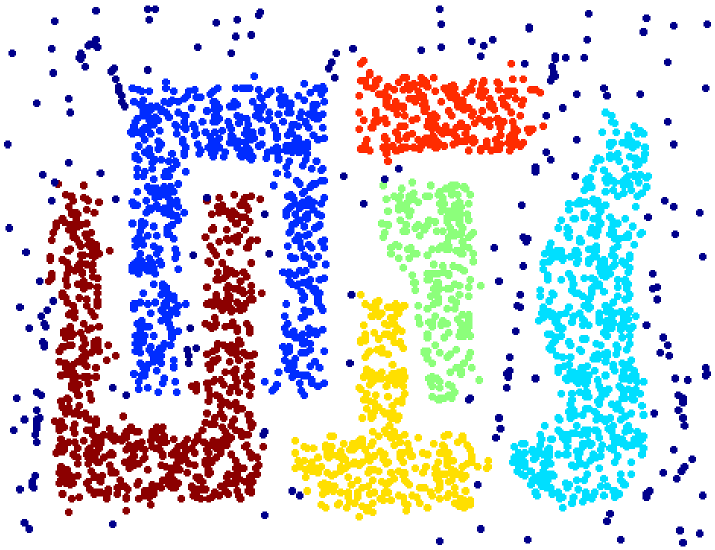


Point types: core, border and noise

Eps = 10, MinPts = 4

When DBSCAN Works Well

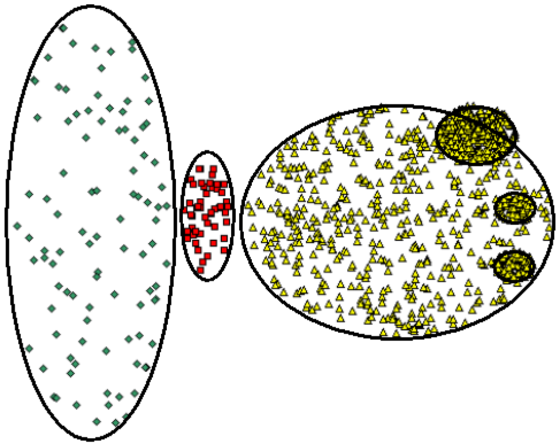
Original Points



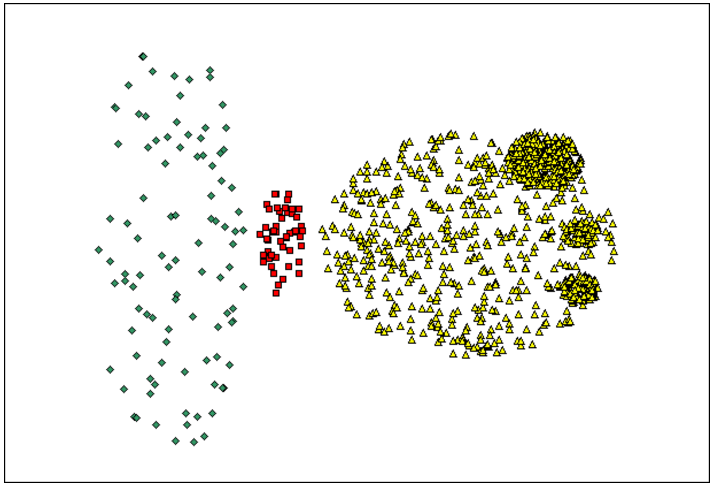
Clusters

- Resistant to Noise
- Can handle clusters of different shapes and sizes

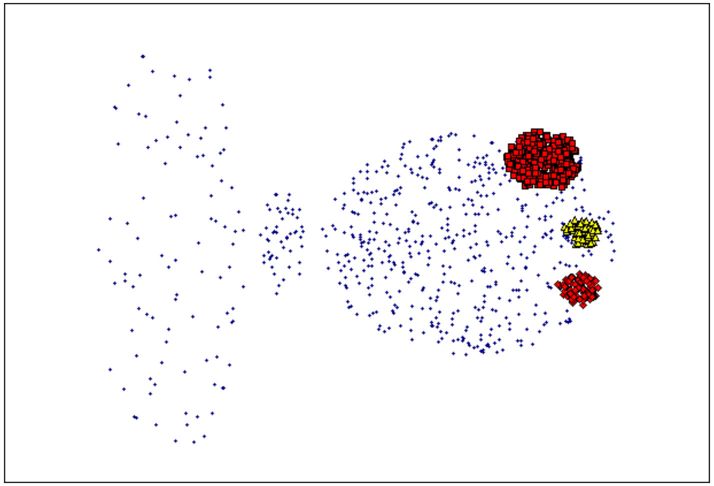
When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

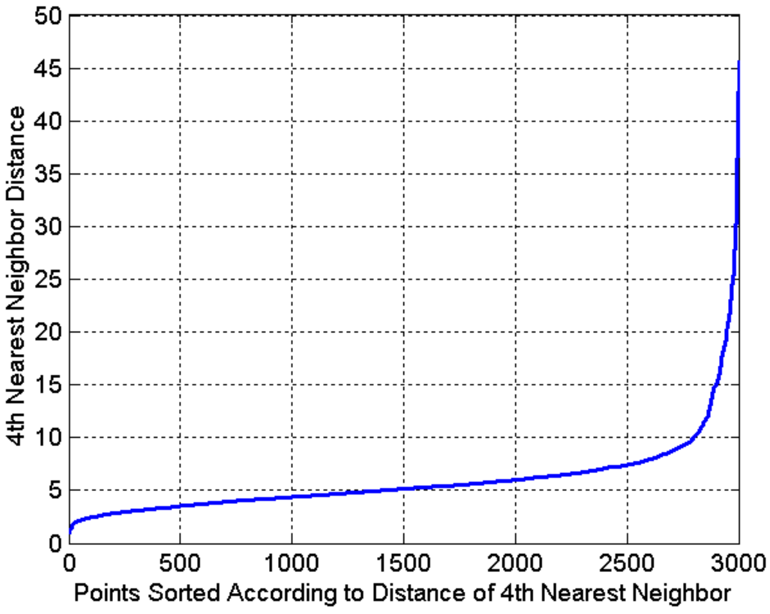
- **Varying densities**
- **High-dimensional data**

DBSCAN: Determining EPS and MinPts

Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance

Noise points have the k^{th} nearest neighbor at farther distance

So, plot sorted distance of every point to its k^{th} nearest neighbor



Cluster Validity

For supervised classification we have a variety of measures to evaluate how good our model is

- Accuracy, precision, recall

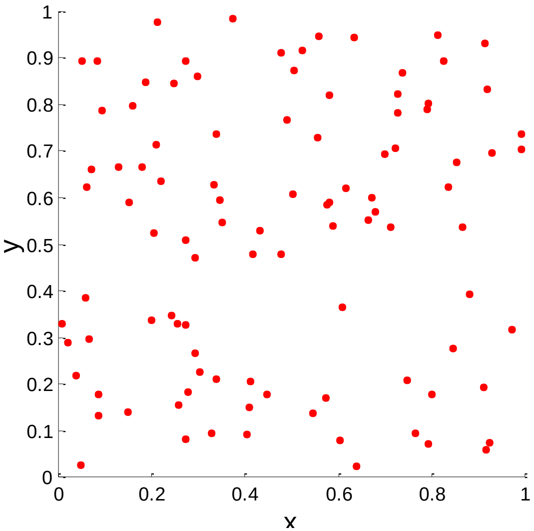
For cluster analysis, the analogous question is how to evaluate the "goodness" of the resulting clusters?

Then why do we want to evaluate them?

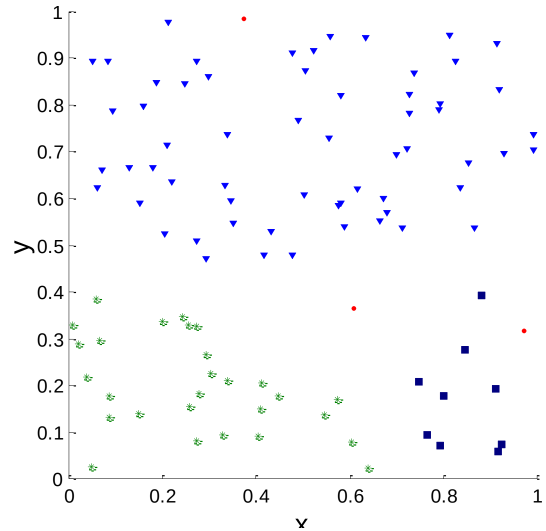
- To compare clustering algorithms
- To compare two sets of clusters
- To compare two clusters

Clusters found in Random Data

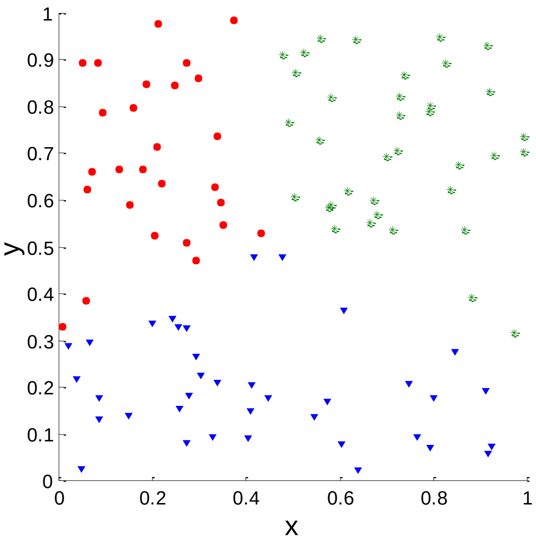
Random Points



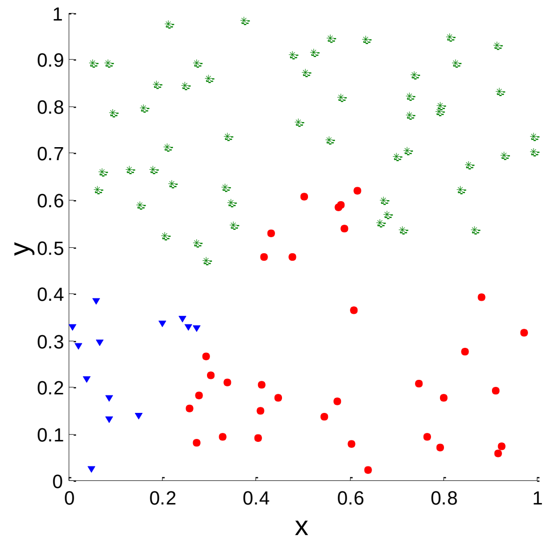
DBSCAN



K-means



Complete Link



Measures of Cluster Validity

Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.

- External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
  - Jaccard Coefficient, Mutual Information, Fowlkes and Mallows Index, Rand Index, Entropy
- Internal Index: Used to measure the goodness of a clustering structure *without* respect to external information.
  - 轮廓系数Silhouette Coefficient, Calinski-Harabasz, Sum of Squared Error (SSE), Davies-Bouldin Index, Dunn Index
- Relative Index: Used to compare two different clusterings or clusters.
  - Often an external or internal index is used for this function, e.g., SSE or entropy

Measures of Cluster Validity

轮廓系数（**Silhouette Coefficient**）：衡量一个数据点与其所属聚类内部的相似度相对于最近的邻居聚类的不相似度。范围在**[-1, 1]**之间，越接近**1**表示聚类质量越好。

Calinski-Harabasz指数：通过聚类间的方差与聚类内的方差的比值来评估聚类的紧密度，指数值越大越好。

Davies-Bouldin指数：度量不同聚类之间的相似性，指数值越小表示聚类越好。

互信息（**Mutual Information**）：度量两个集合之间的相似性，用于比较聚类结果与真实标签的一致性。

Fowlkes-Mallows指数：评估两个聚类结果的相似性，是精度和召回率的调和平均值。

Jaccard系数：比较两个聚类集合的相似性，通过交集与并集的比值计算。

DB指数（**Davies-Bouldin Index**）：通过聚类内部的最大距离和聚类间的最小距离的比值来评估聚类的紧密度。

Measuring Cluster Validity Via Correlation

Two matrices

- Proximity Matrix (邻接矩阵)– actual similarity matrix
- incidence Matrix(关联矩阵)
  - One row and one column for each data point
  - An entry is 1 if the associated pair of points belong to the same cluster
  - An entry is 0 if the associated pair of points belongs to different clusters

Compute the correlation between the two matrices

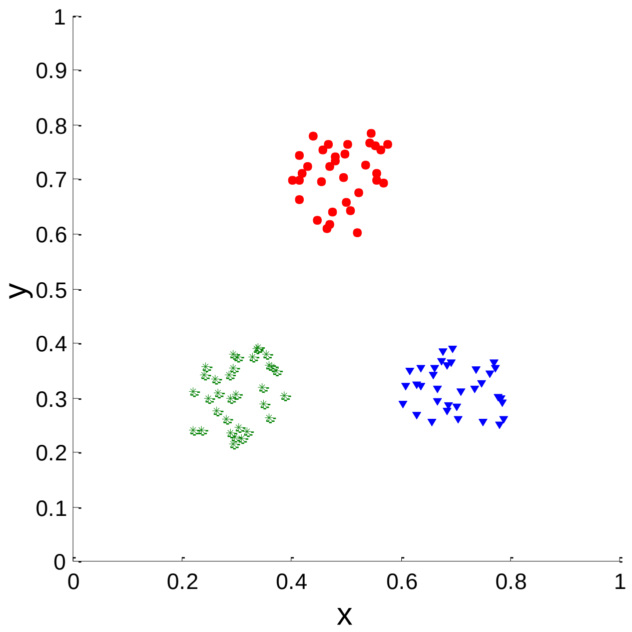
- Since the matrices are symmetric, only the correlation between n(n-1) / 2 entries needs to be calculated.

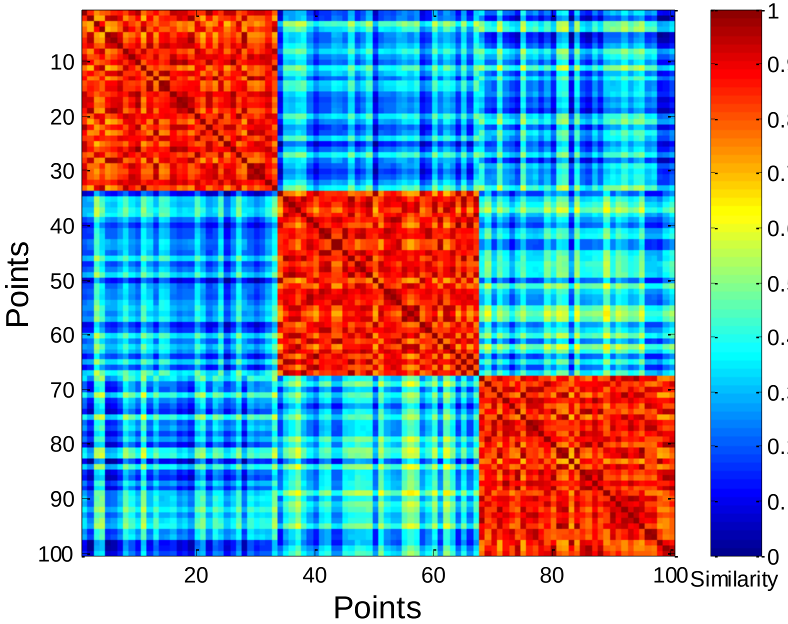
High correlation indicates that points that belong to the same cluster are close to each other.

Not a good measure for some density or contiguity based clusters.

Using Similarity Matrix for Cluster Validation

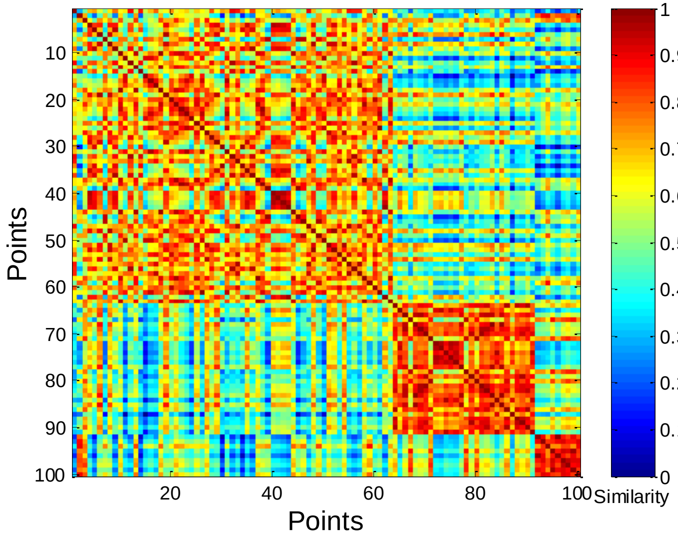
Order the similarity matrix with respect to cluster labels and inspect visually.

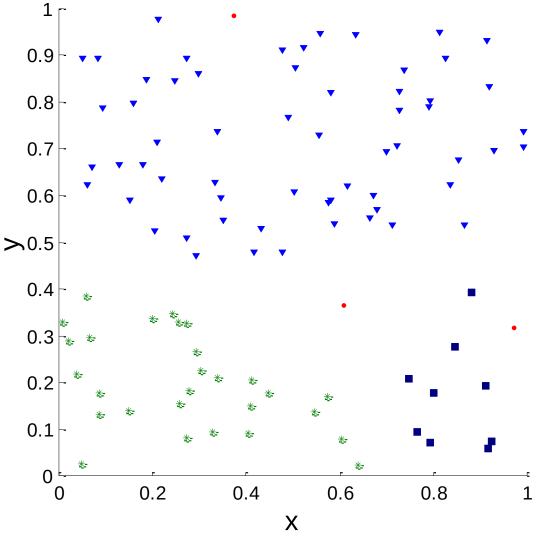




Using Similarity Matrix for Cluster Validation

Clusters in random data

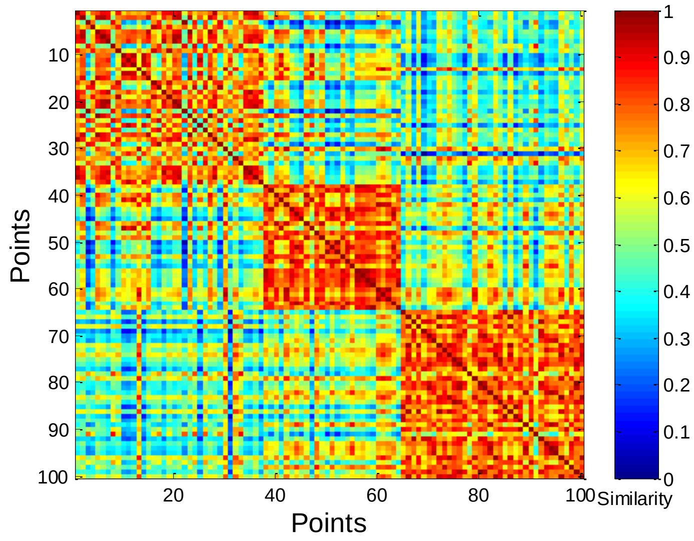


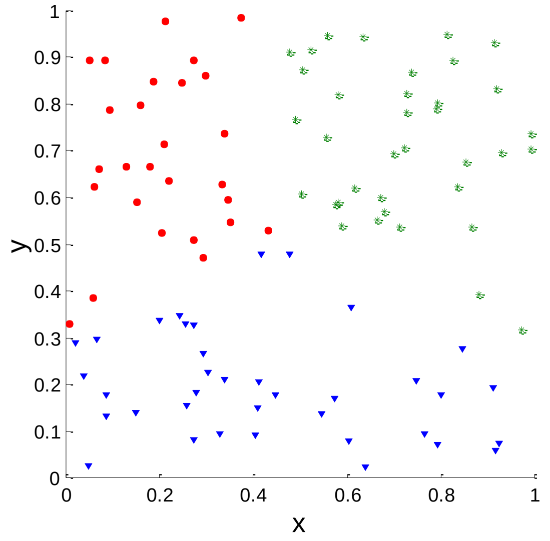


DBSCAN

Using Similarity Matrix for Cluster Validation

Clusters in random data

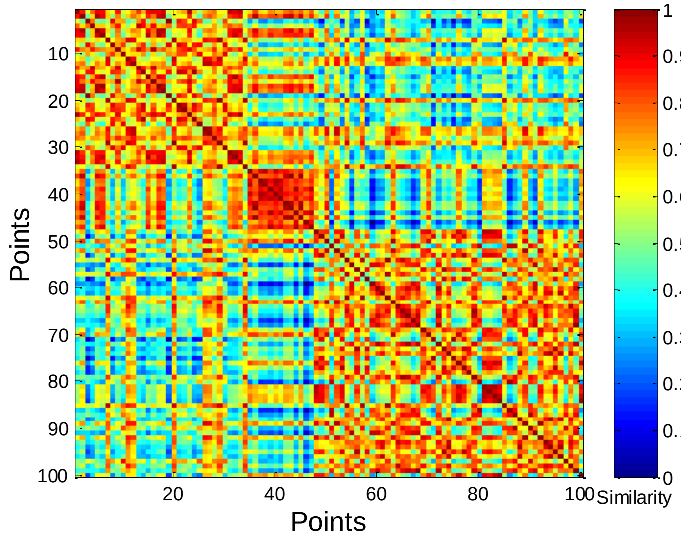


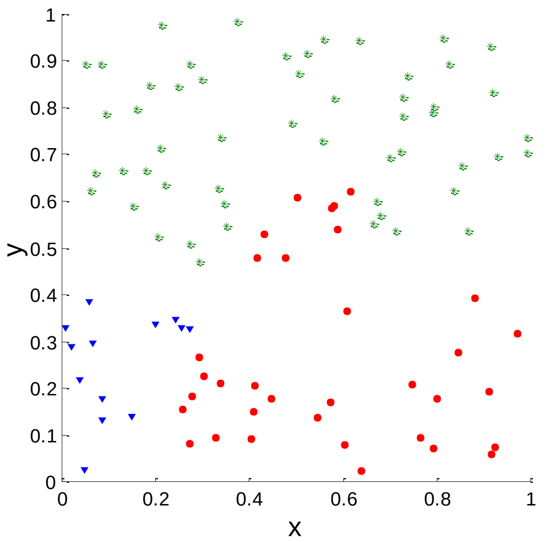


K-means

Using Similarity Matrix for Cluster Validation

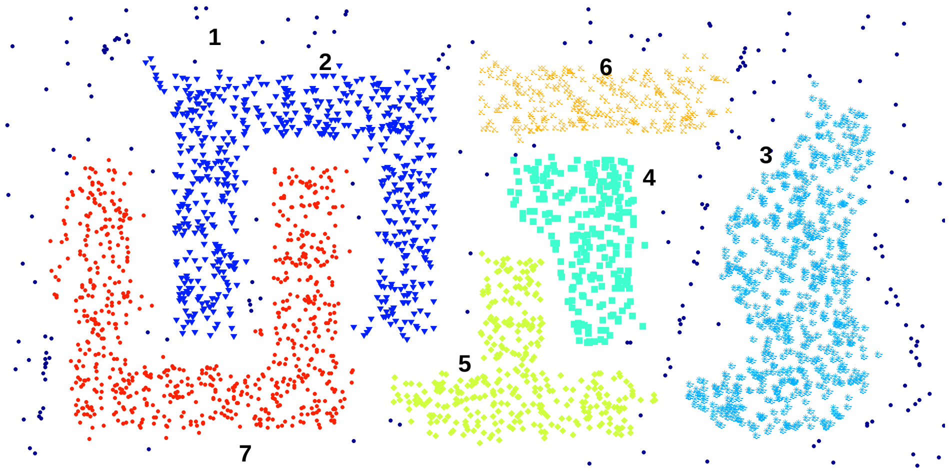
Clusters in random data

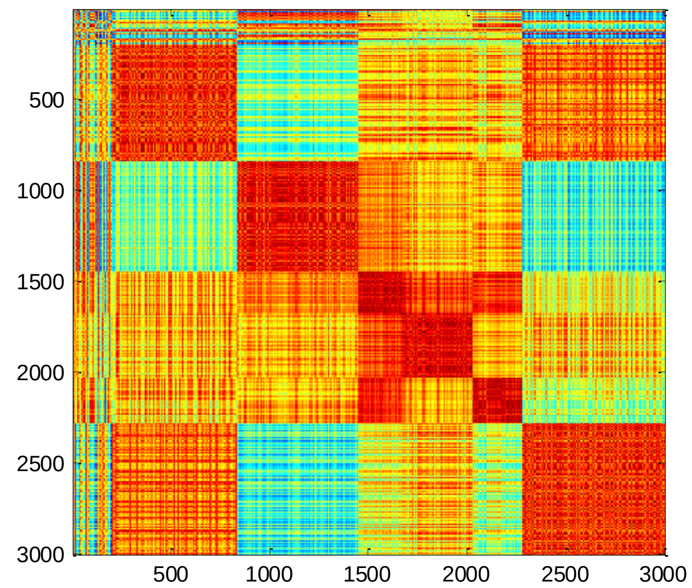




Complete Link

Using Similarity Matrix for Cluster Validation

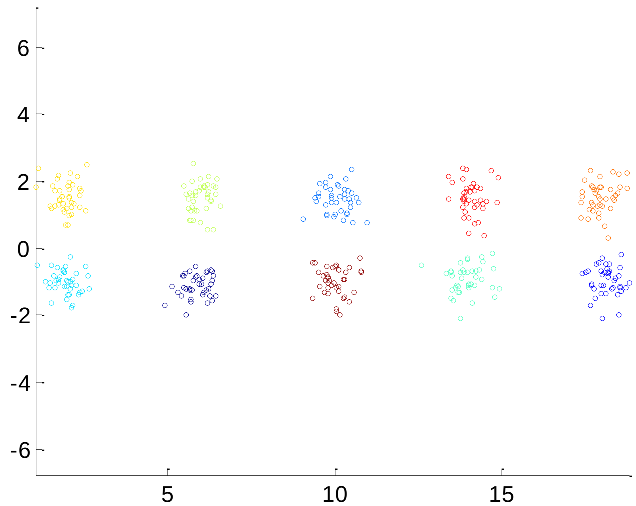


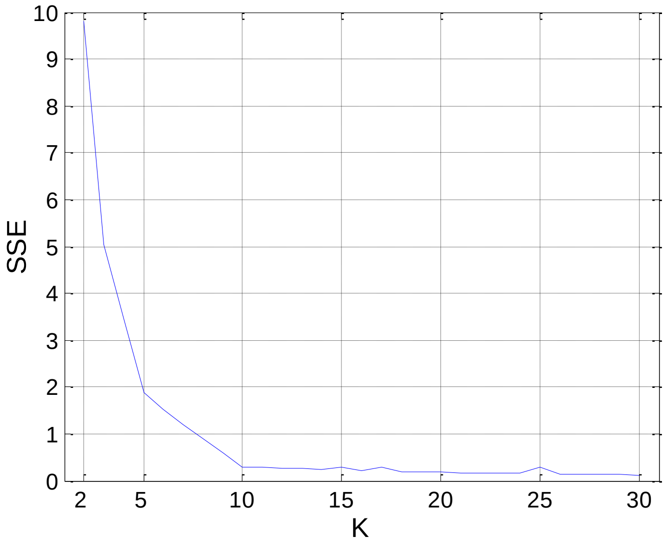


DBSCAN

Internal Measures: SSE

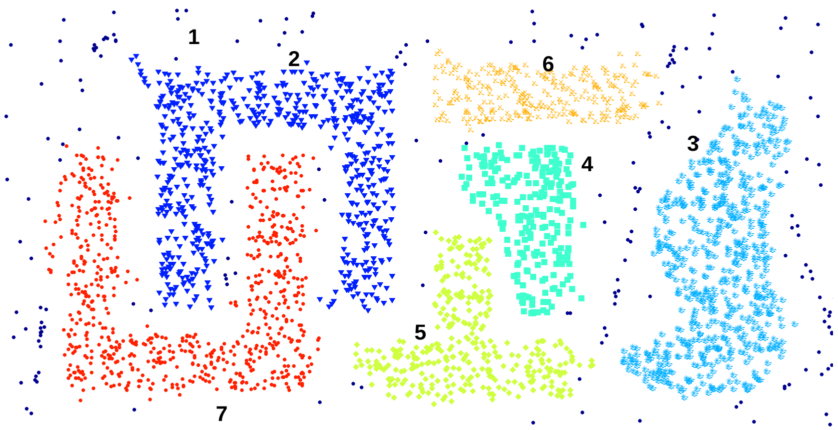
- Clusters in more complicated figures aren’t well separated
- Internal Index: Used to measure the goodness of a clustering structure without respect to external information
  – SSE
- SSE is good for comparing two clusterings or two clusters (average SSE).
- Can also be used to estimate the number of clusters

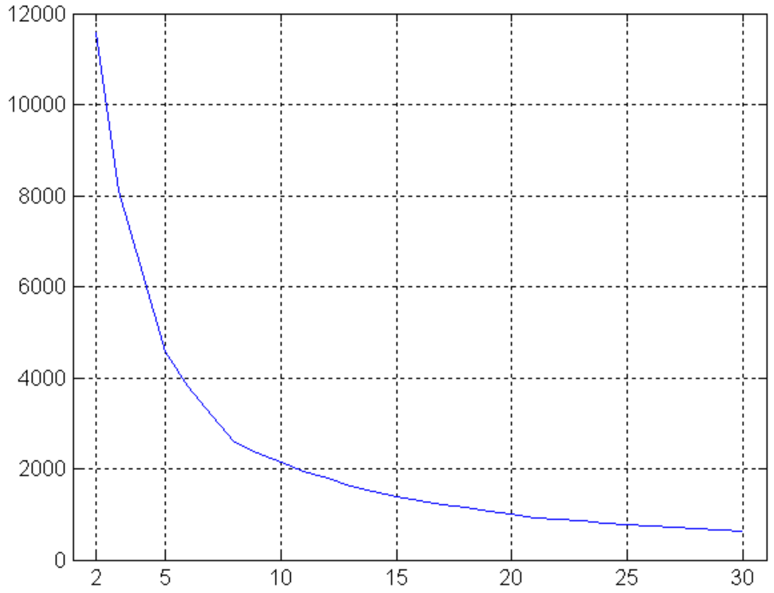




Internal Measures: SSE

SSE curve for a more complicated data set





SSE of clusters found using K-means

Conclusion

- Clustering Method
  - Unsupervised, discrete
- Algorithm
  - Partitioning approach: CATA ,k-medoids, Kernel K-means , CLARANS
  - Hierarchical approach: Agnes, Diana, BIRCH, ROCK, CAMELEON
  - Density-based approach: DBSCAN, OPTICS, DenClue
- Measures of Cluster Validity
  - External Index: Jaccard Coefficient, Fowlkes and Mallows Index, Rand Index, Entropy
  - Internal Index: Silhouette Coefficient , Sum of Squared Error (SSE)， Davies-Bouldin Index， Dunn Index
  - Relative Index: SSE or entropy

思考题

列举出这次课主要给出的三类聚类算法的优缺点?