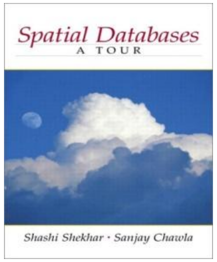


Spatial Data Mining

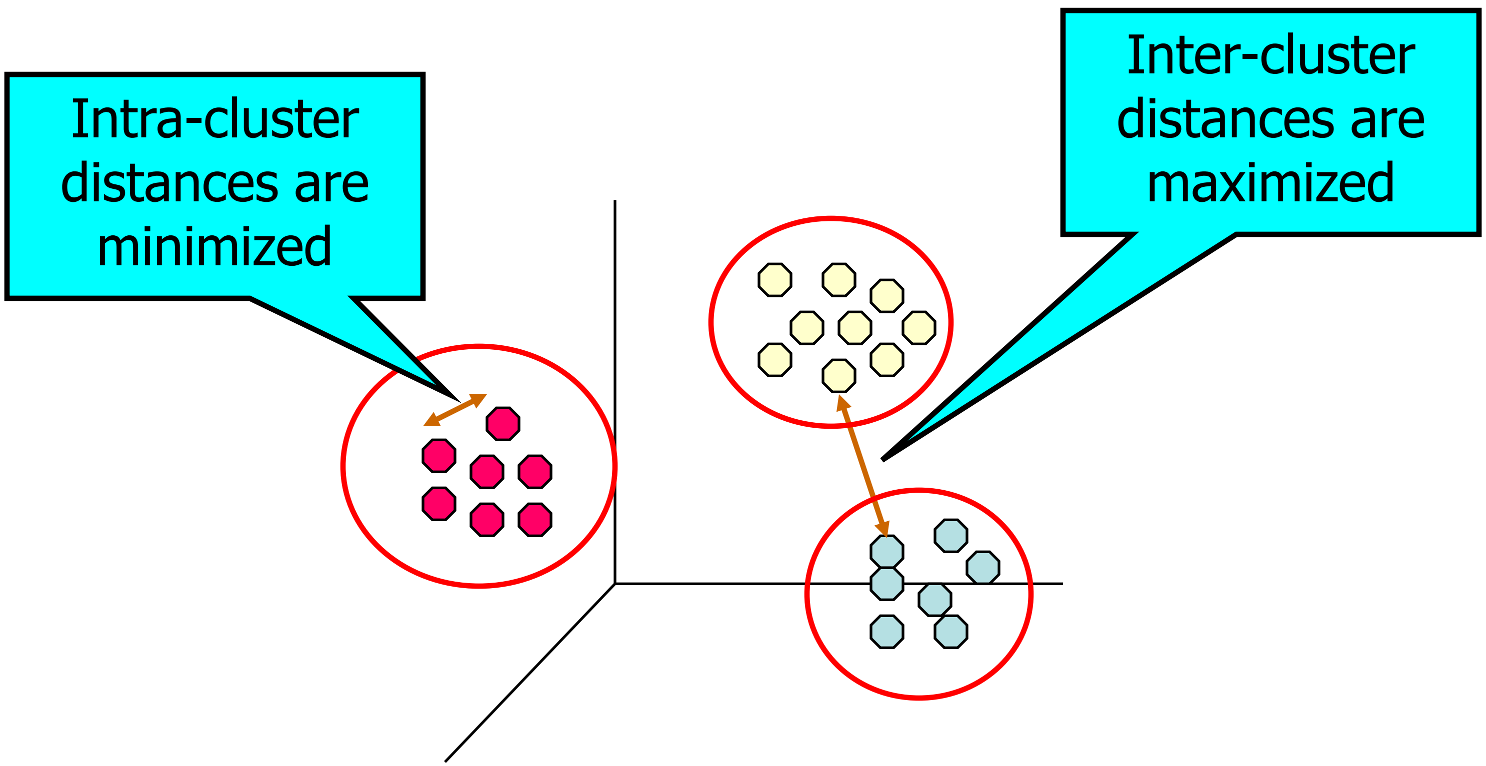
Clustering Techniques

Overview

- What is clustering?
- What kind of clusters?
- Which clustering algorithms?
- How to measure the quality of clustering result?

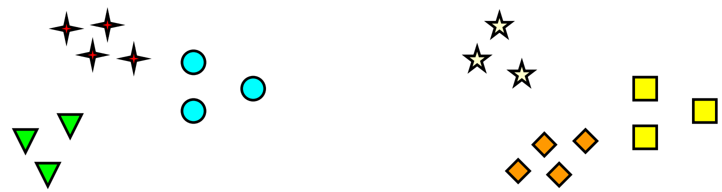
What is Cluster Analysis?

- 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



Notion of a Cluster can be Ambiguous

How many clusters?



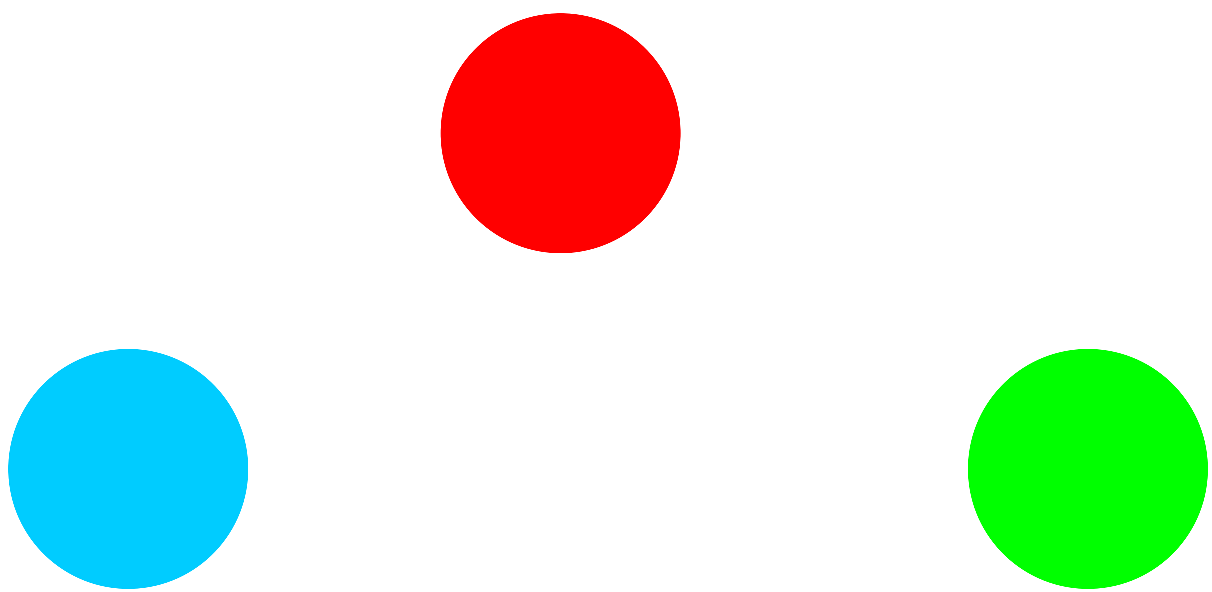
Six Clusters

Two Clusters

Four Clusters

Types of Clusters: Well-Separated

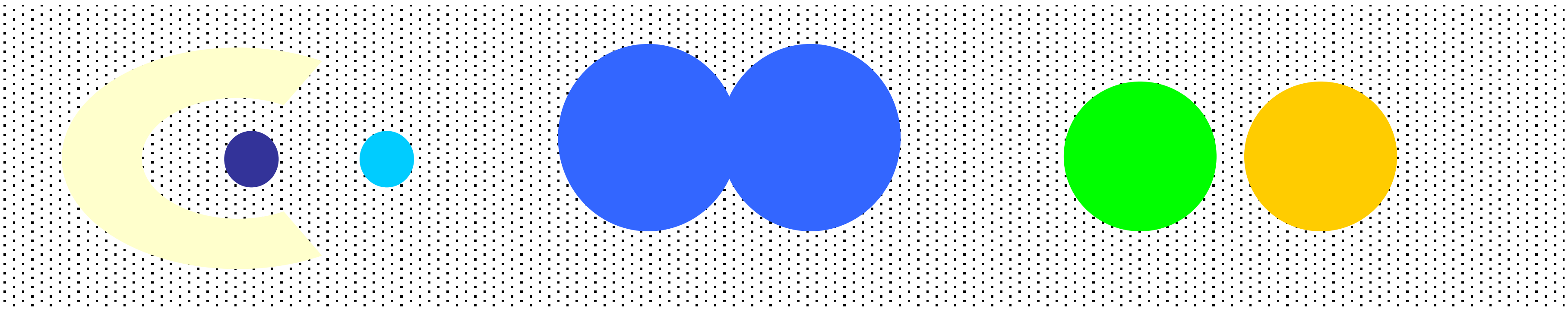
- Well-Separated Clusters:
  - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



3 well-separated clusters

Types of Clusters: Density-Based

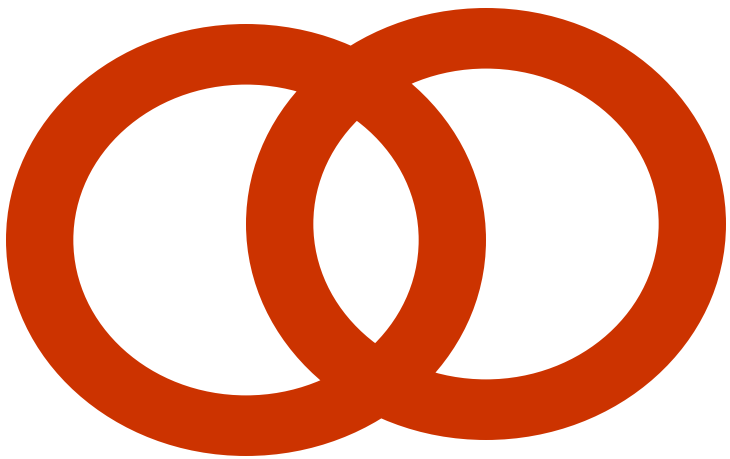
- Density-based
  - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
  - Used when the clusters are irregular or intertwined, and when noise and outliers are present.



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.

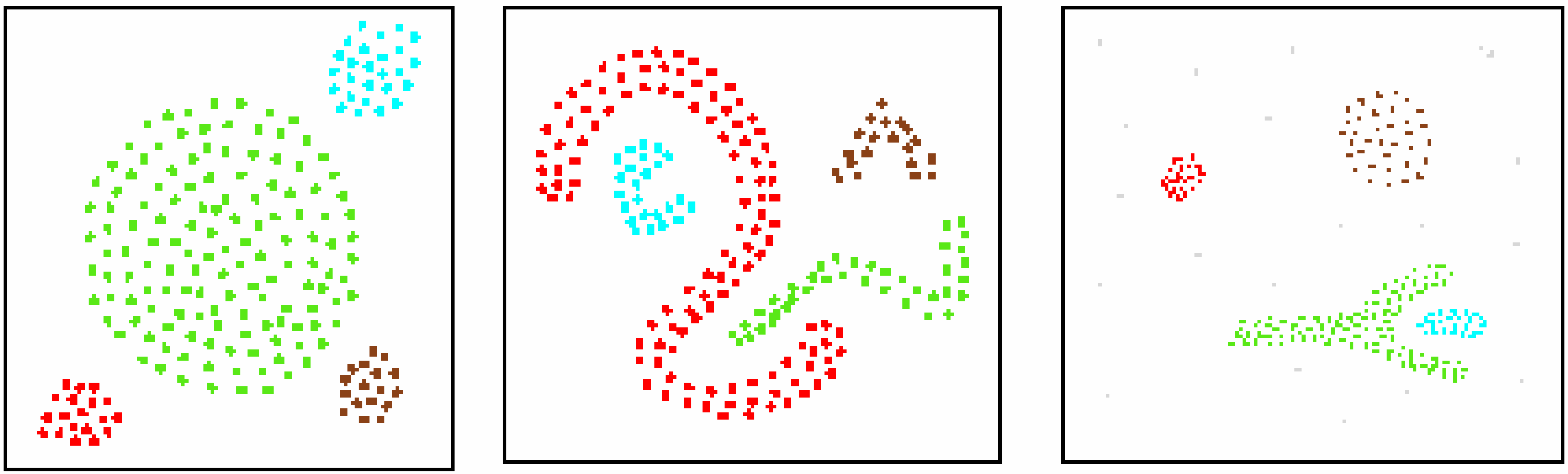


2 Overlapping Circles

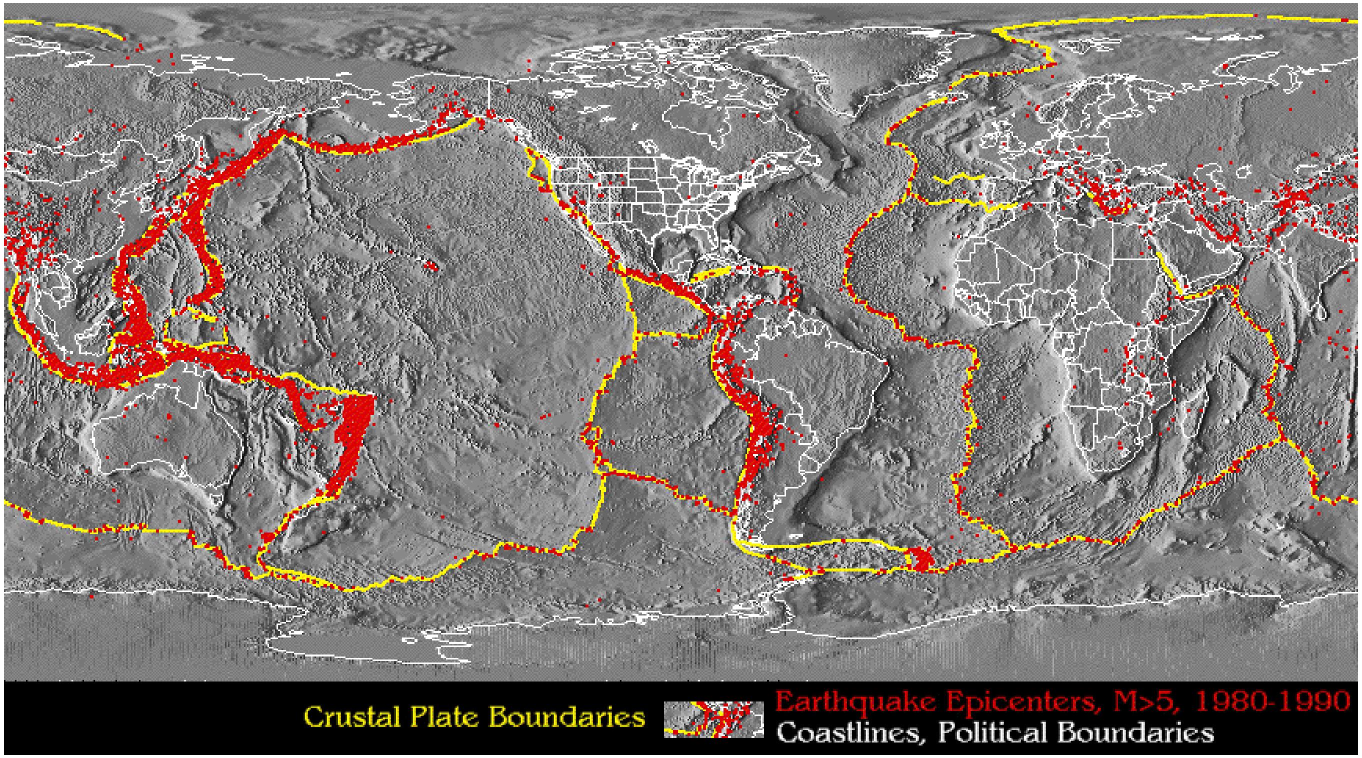
Requirements for Clustering Algorithms

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

Arbitrary shapes



Example: Geology

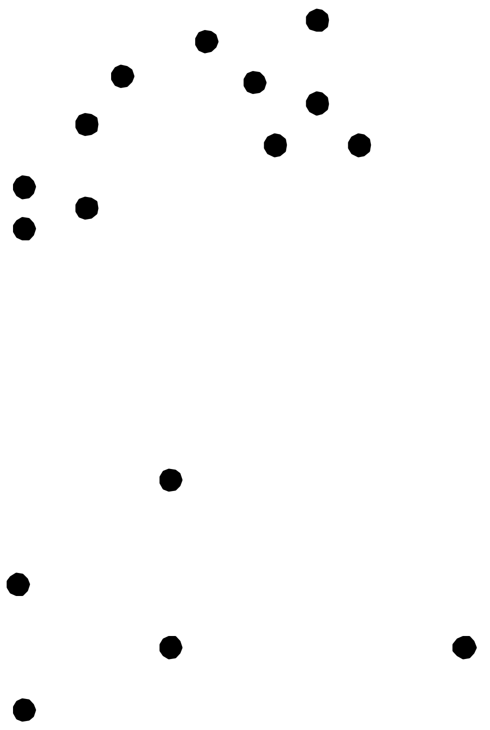


clustering observed earthquake epicenters to identify dangerous zones;

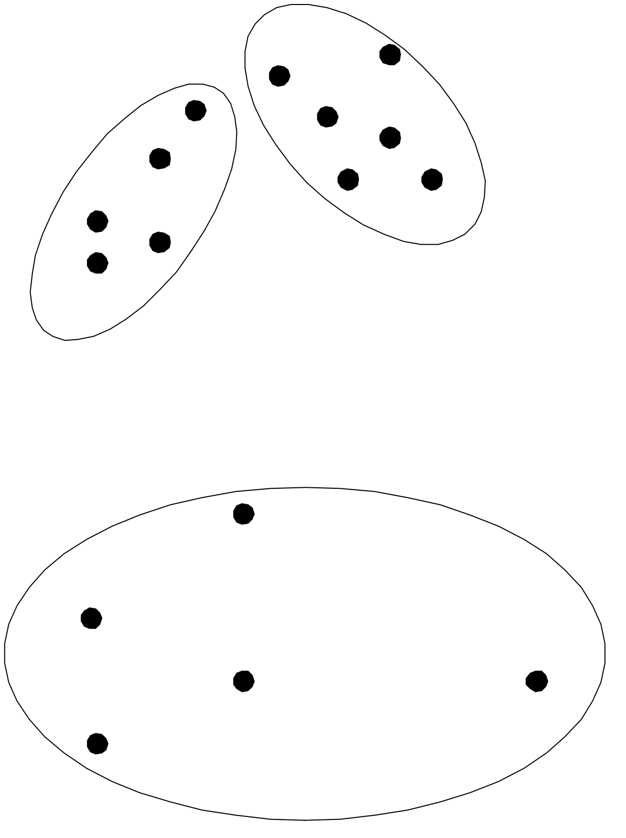
Major Clustering Approaches

- Partitioning algorithms: Construct various partitions and then evaluate them by some criterion
- Hierarchy algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion
- Density-based: based on connectivity and density functions
- Grid-based: based on a multiple-level granularity structure
- Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other

Partitional Clustering



Original Points



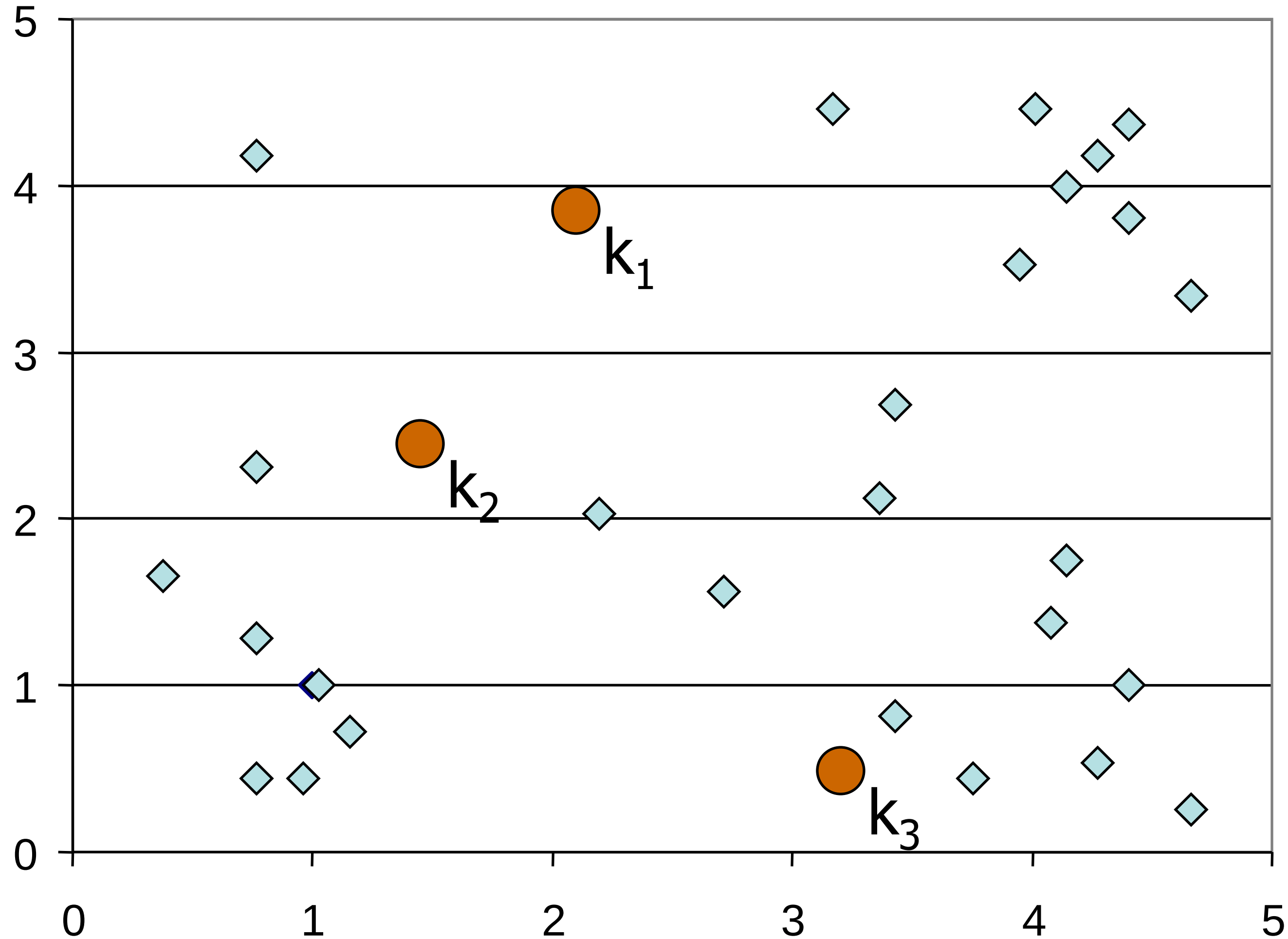
A Partitional Clustering

K-means Clustering

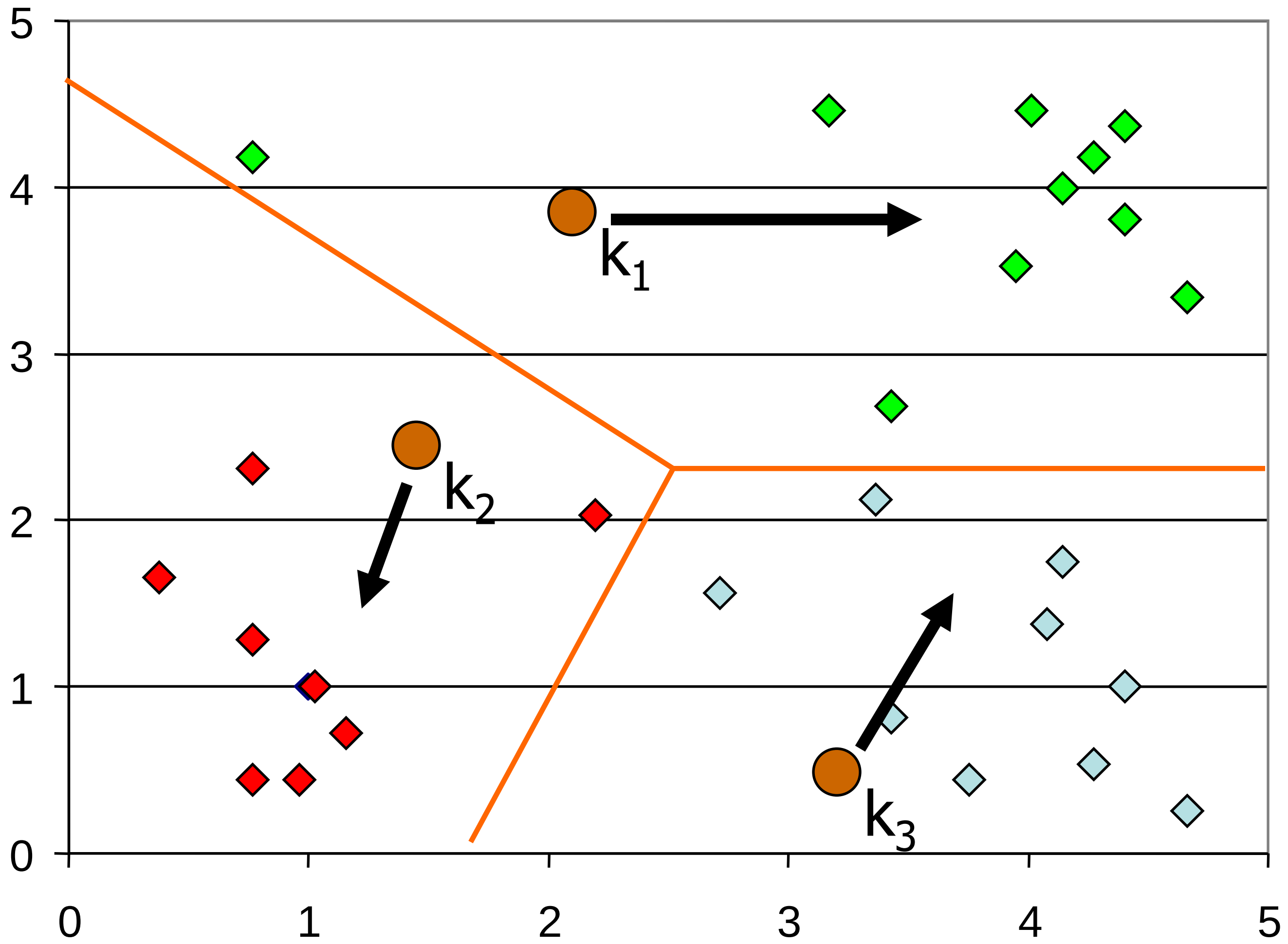
- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- The basic algorithm is very simple

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

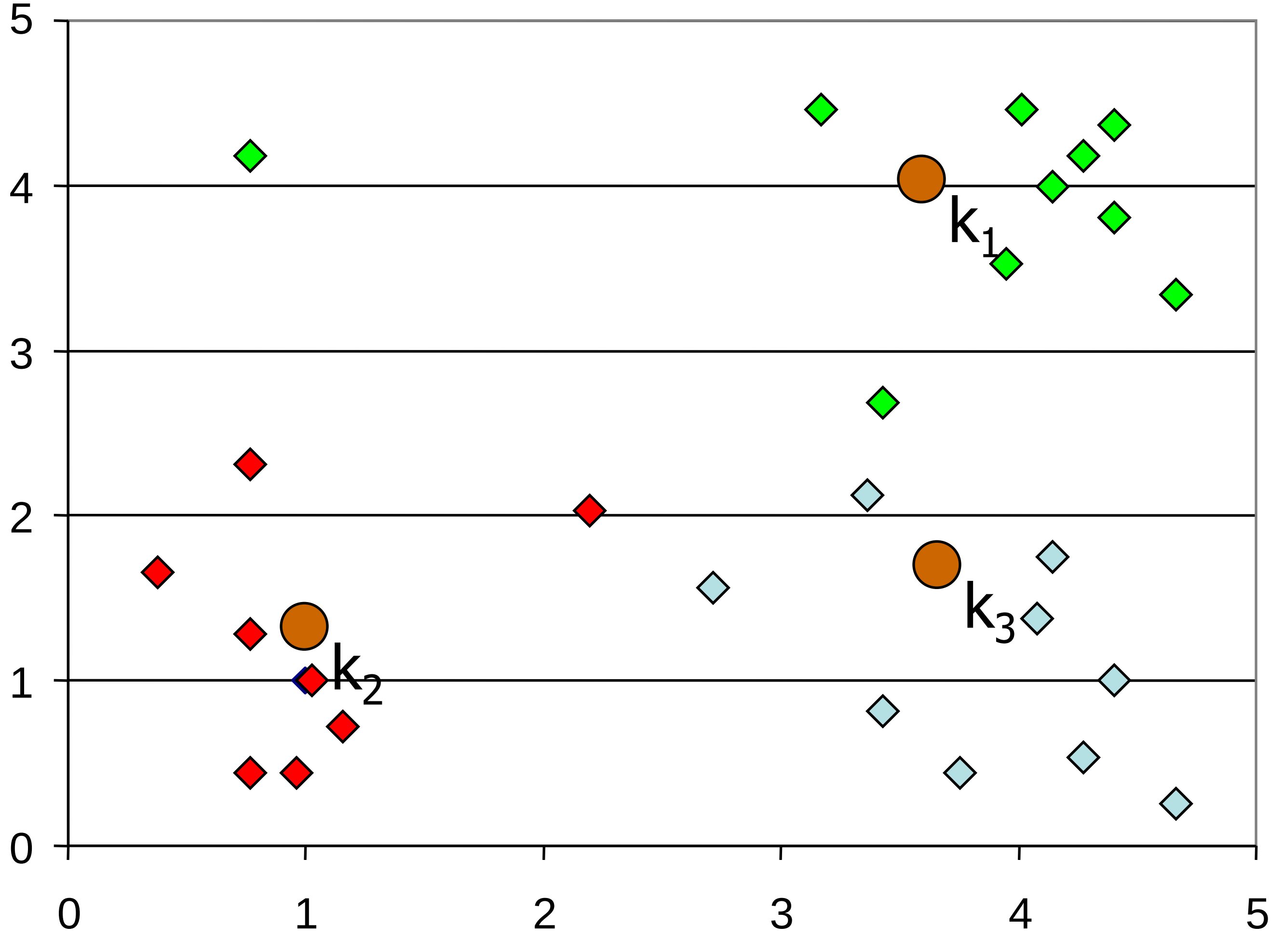
K-means Clustering: Step 1



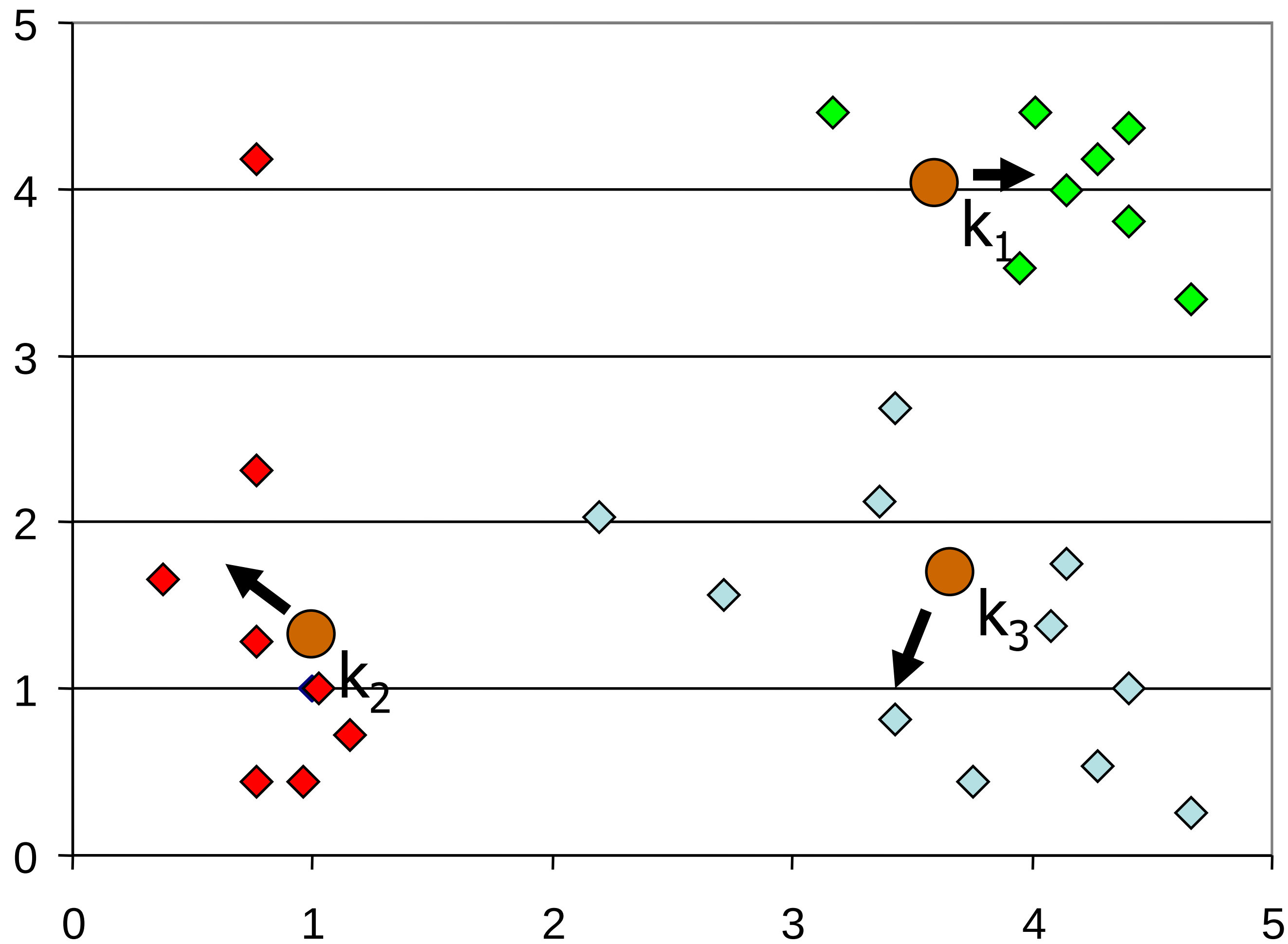
K-means Clustering: Step 2



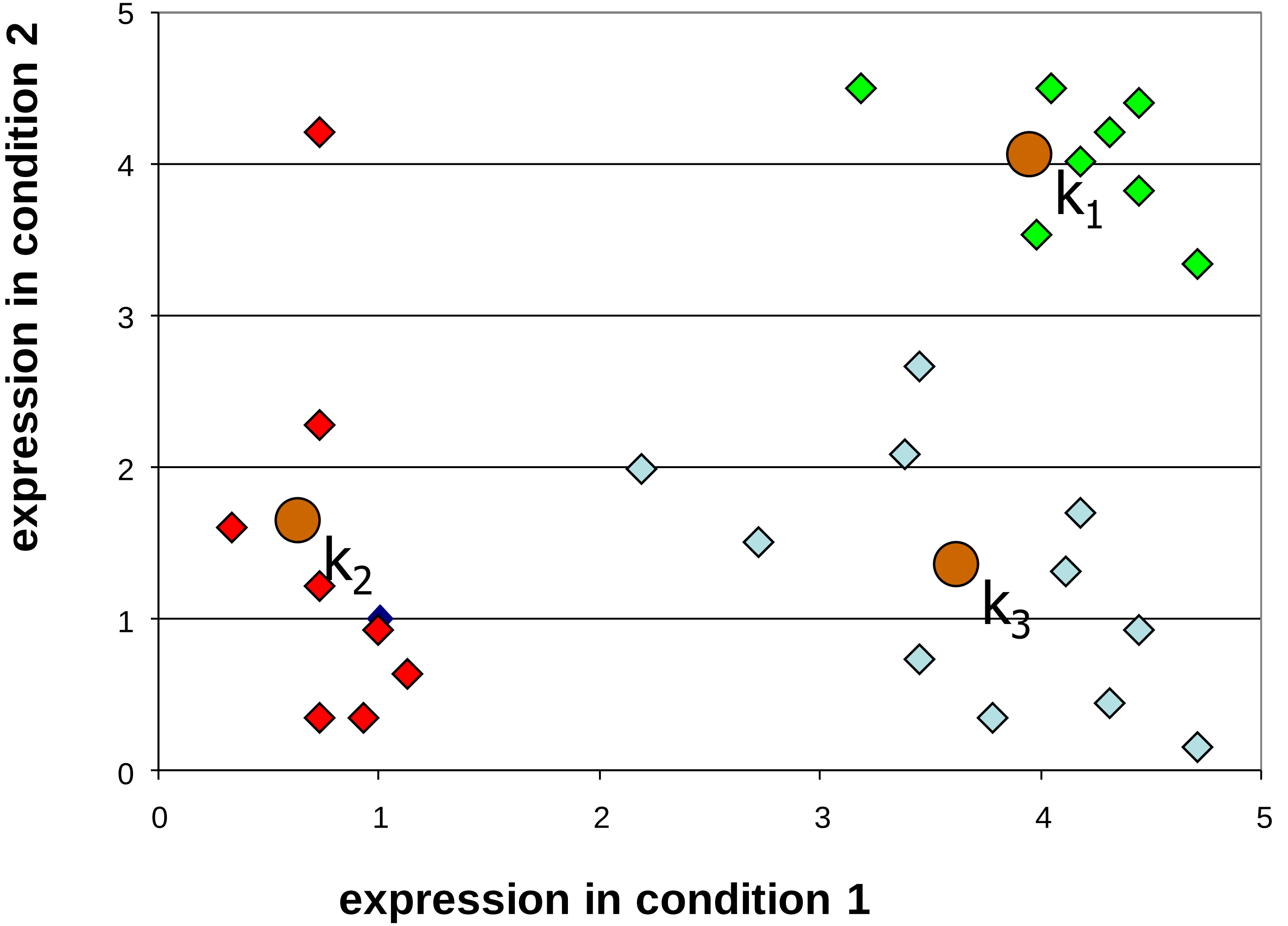
K-means Clustering: Step 3



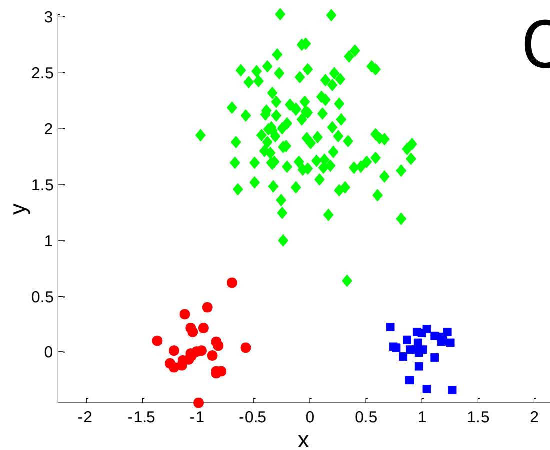
K-means Clustering: Step 4



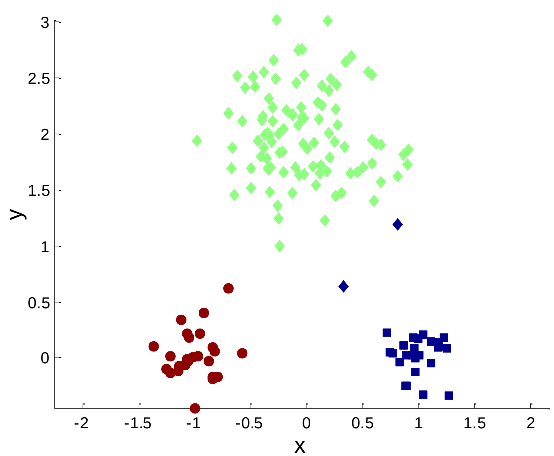
K-means Clustering: Step 5



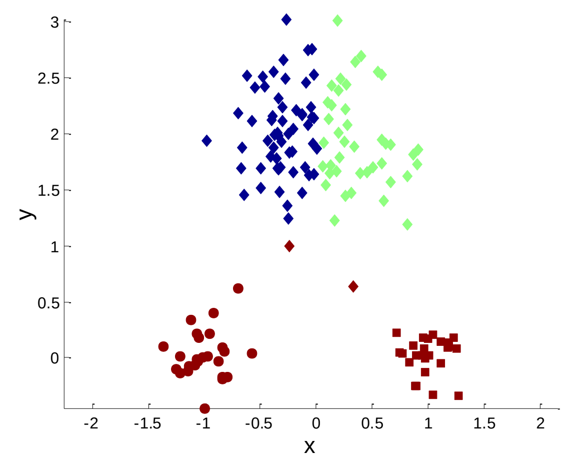
How stable is K-Means?



Original Points

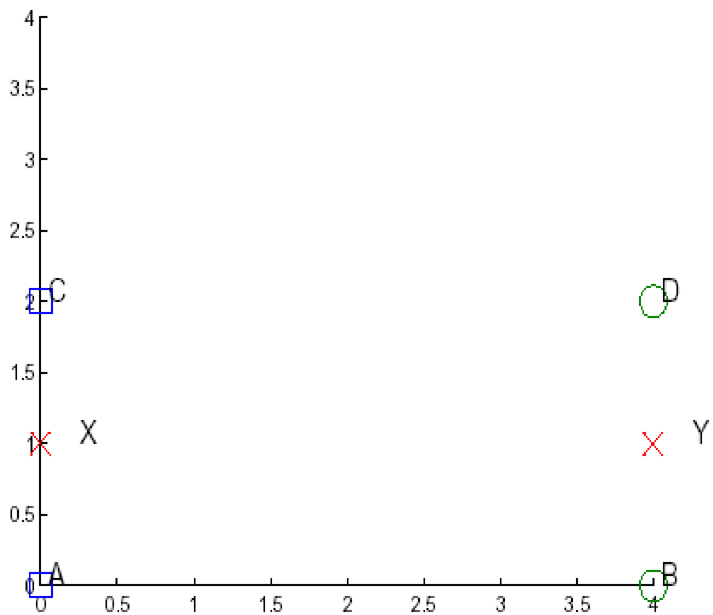


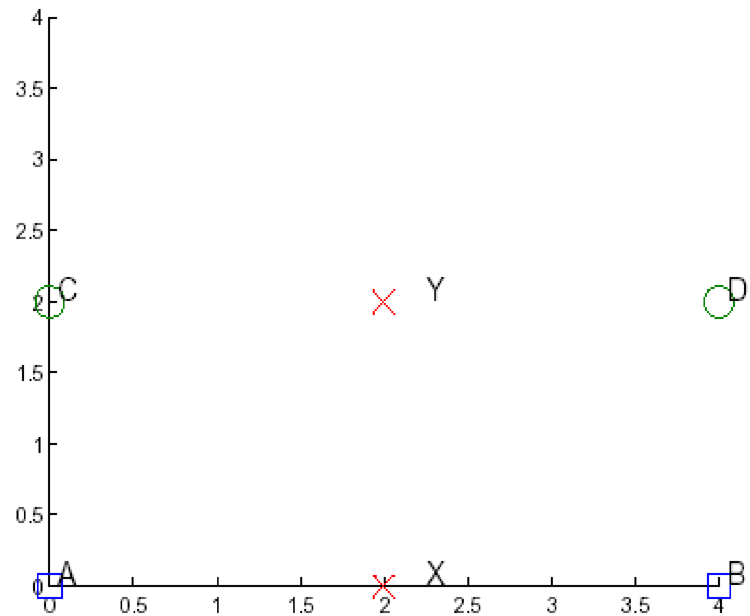
Optimal Clustering



Sub-optimal Clustering

Local minima



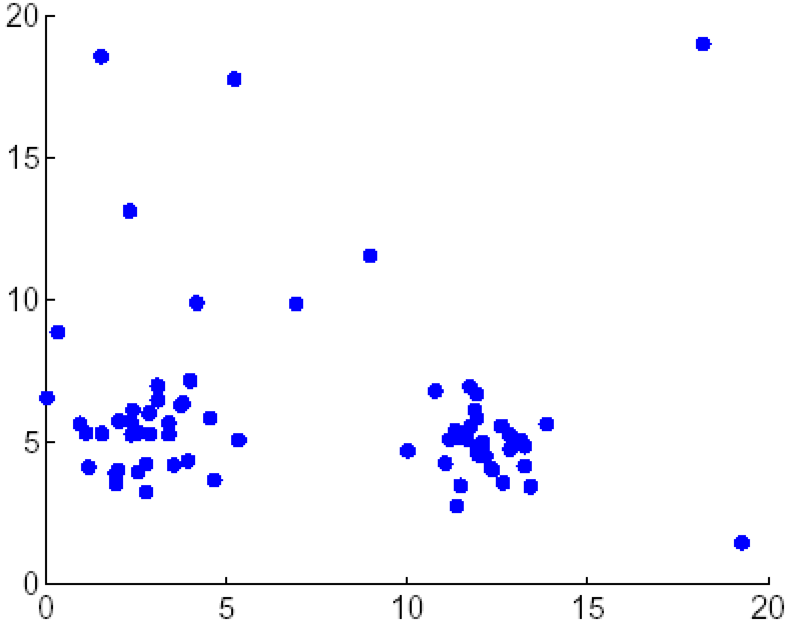


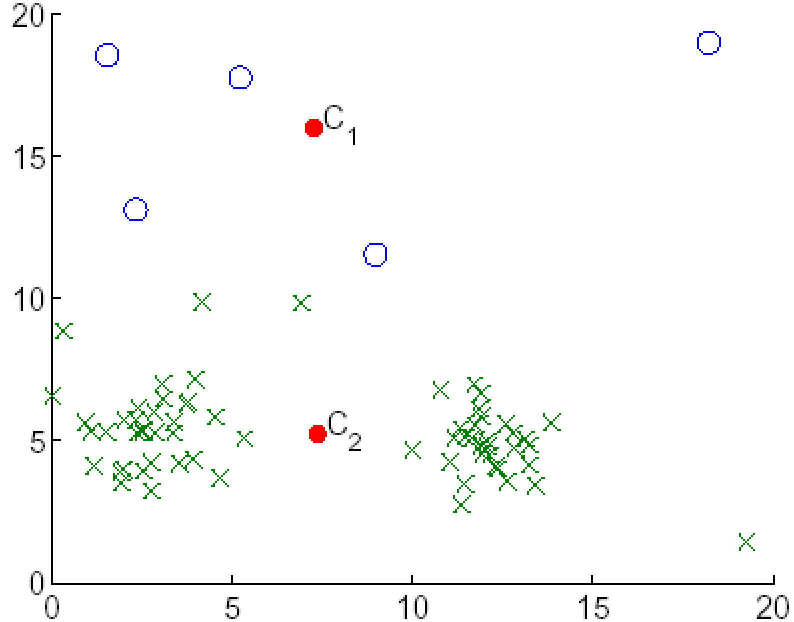
SSE = 4

SSE = 16

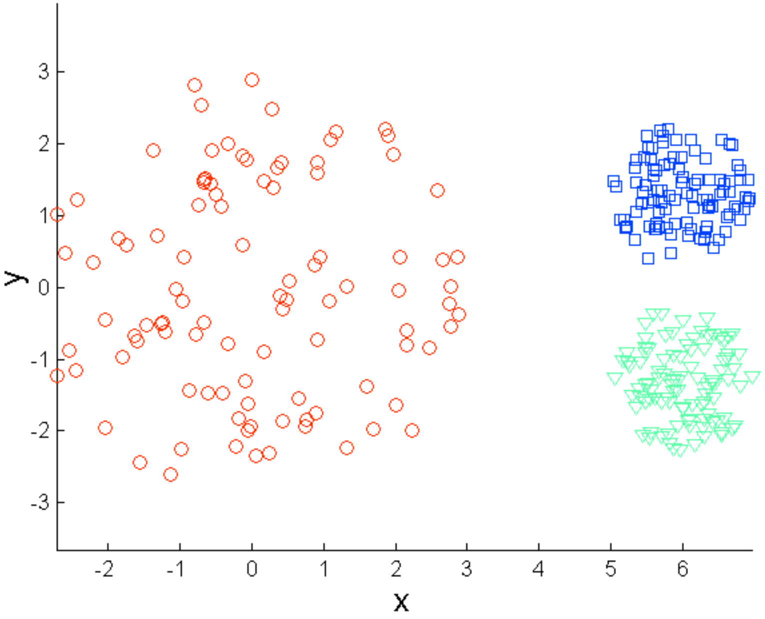
Repeat with different random initial centers

Sensitivity to noise/outliers

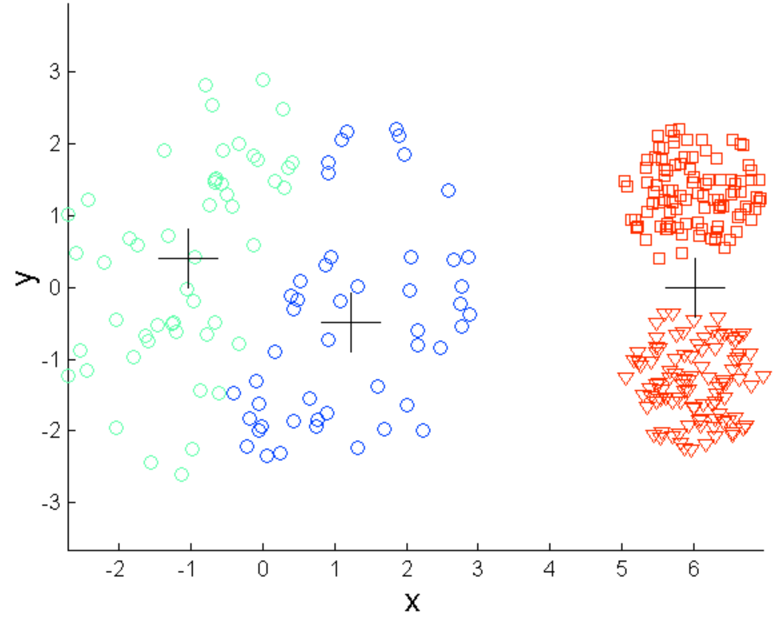




Limitations of K-means: Differing sizes/density

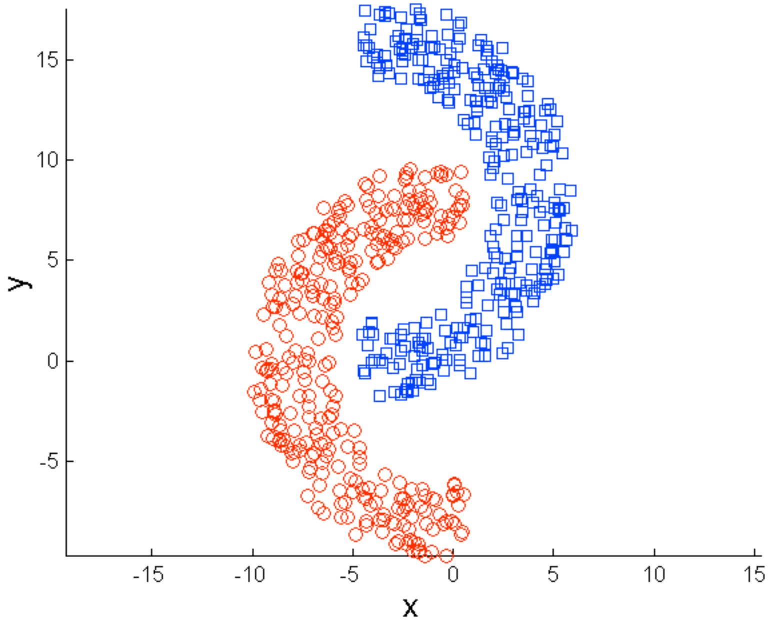


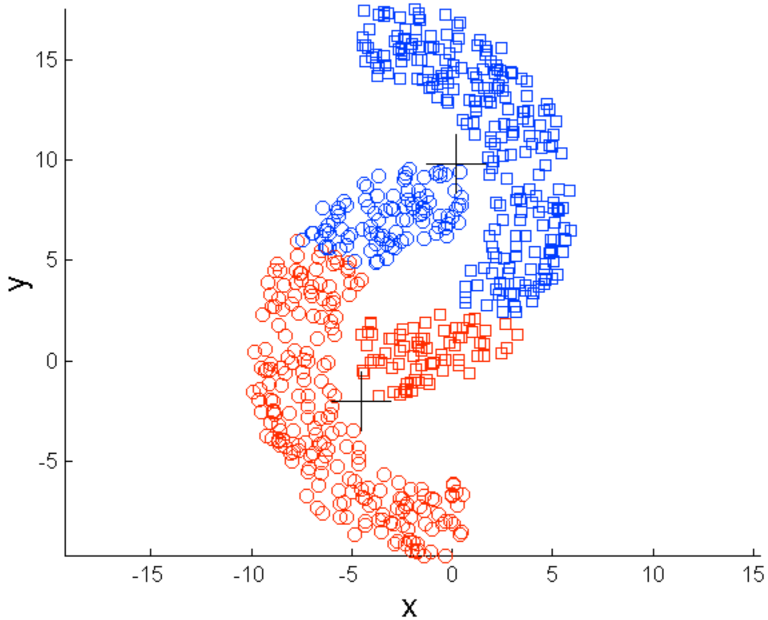
Original Points



K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes





Original Points

K-means (2 Clusters)

K-means characteristics

Pros

- Simple
- Fast (O(nd))

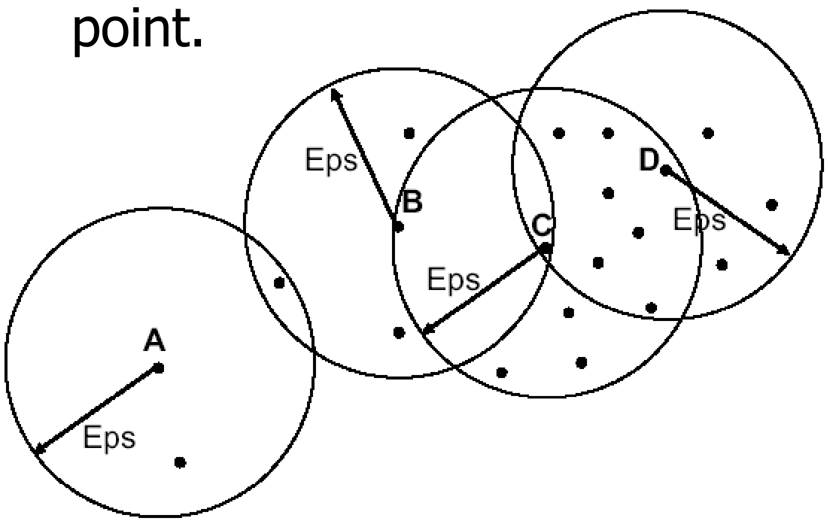
Cons

- Selection of k
- Local minima
- Sensitive to noise, sizes, shapes

DBSCAN: 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
- 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.



MinPts = 10

C,D core

B border

A noise

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**

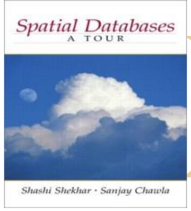
        if the point does not have a cluster label **then**

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

        end if

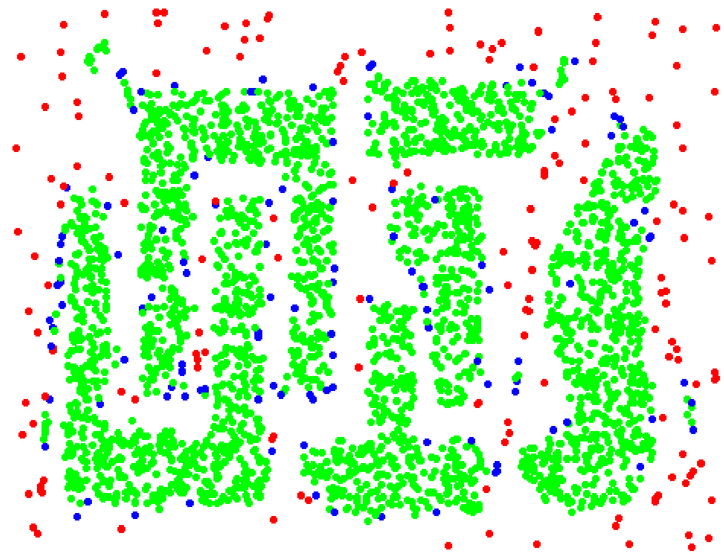
    end for

end for



DBSCAN: Core, Border and Noise Points

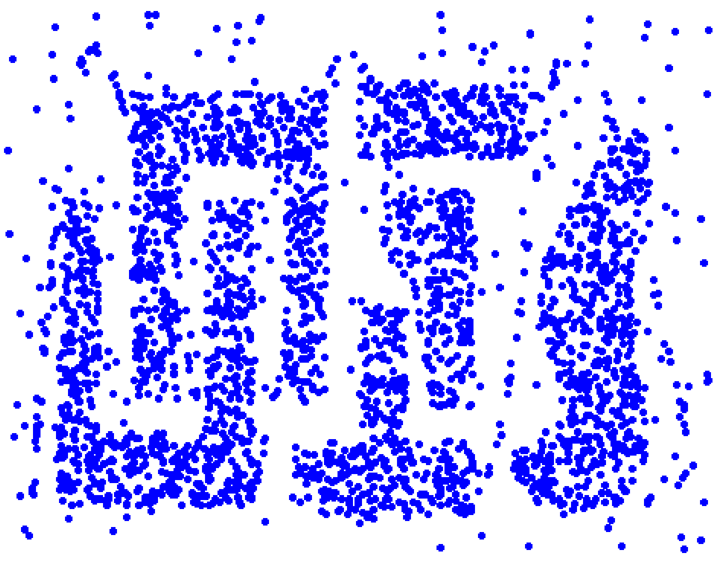
Original Points



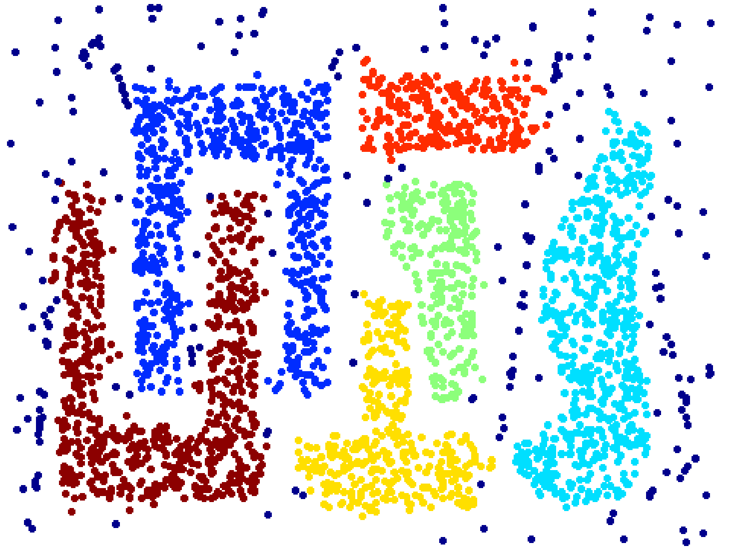
Point types: core, border and noise

Eps = 10, MinPts = 4

When DBSCAN Works Well



Original Points

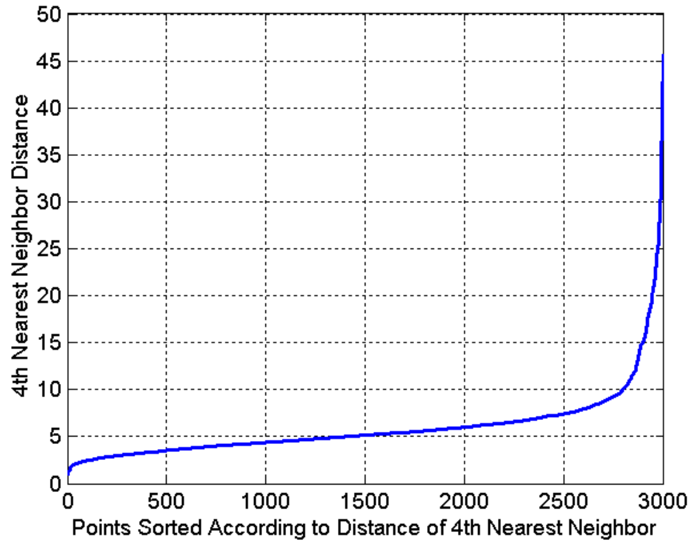


Clusters

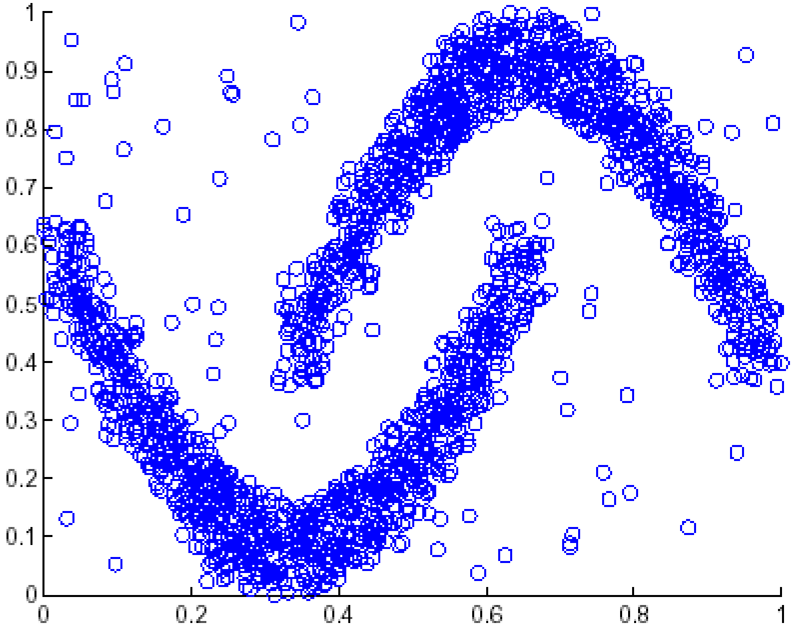
- Resistant to Noise
- Can handle clusters of different shapes and sizes

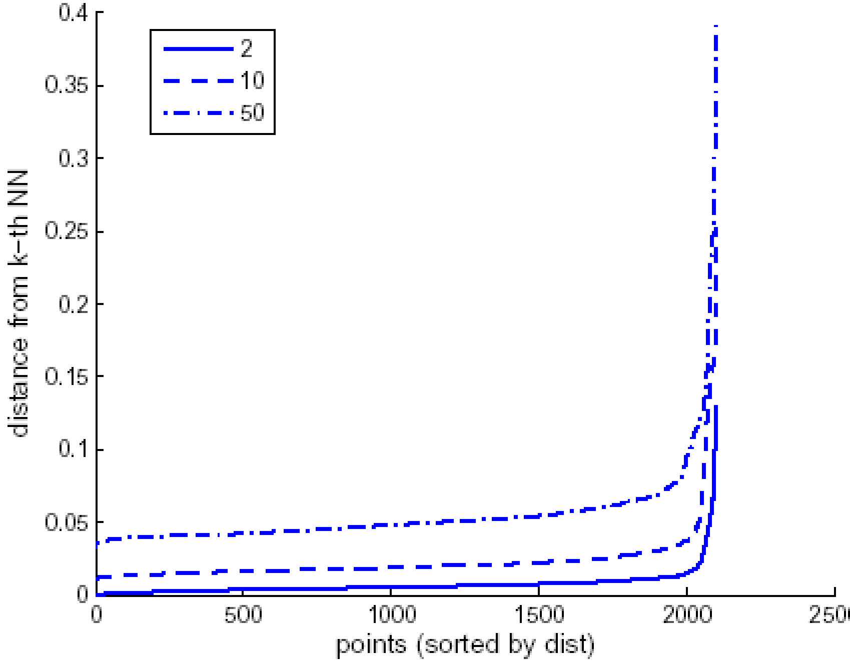
DBSCAN: Determining EPS and MinPts

- 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



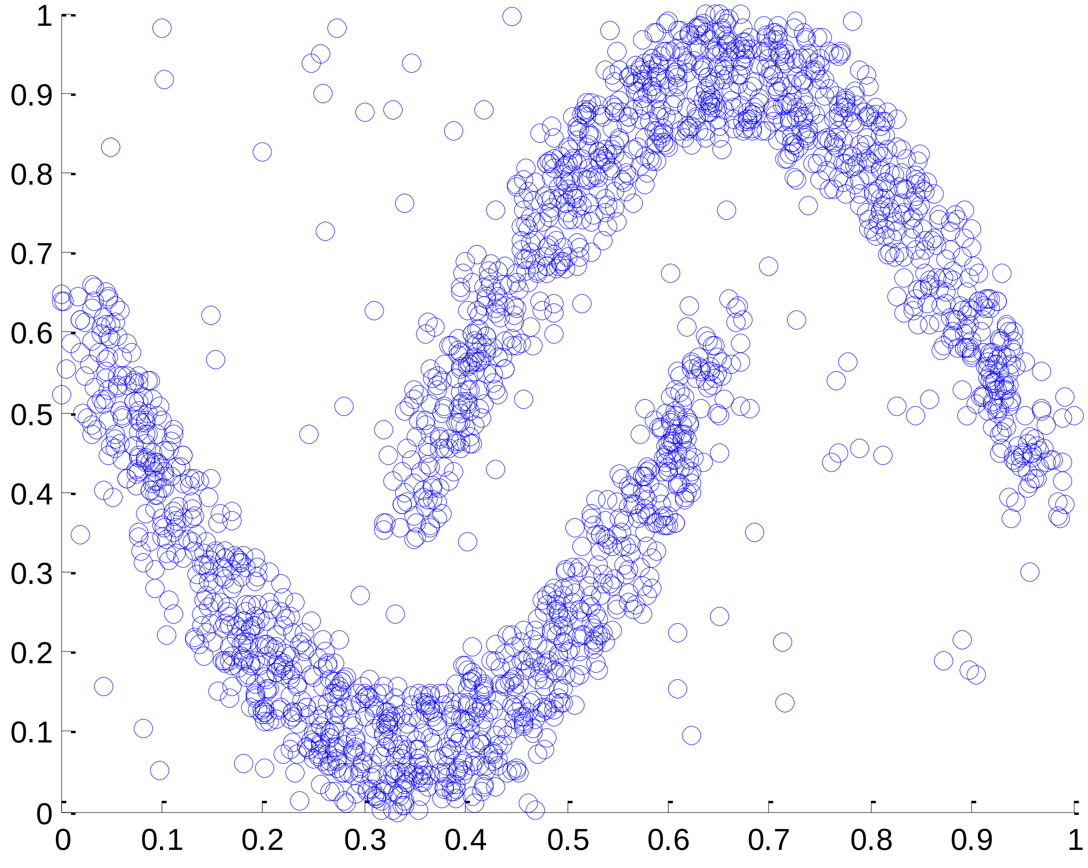
Example



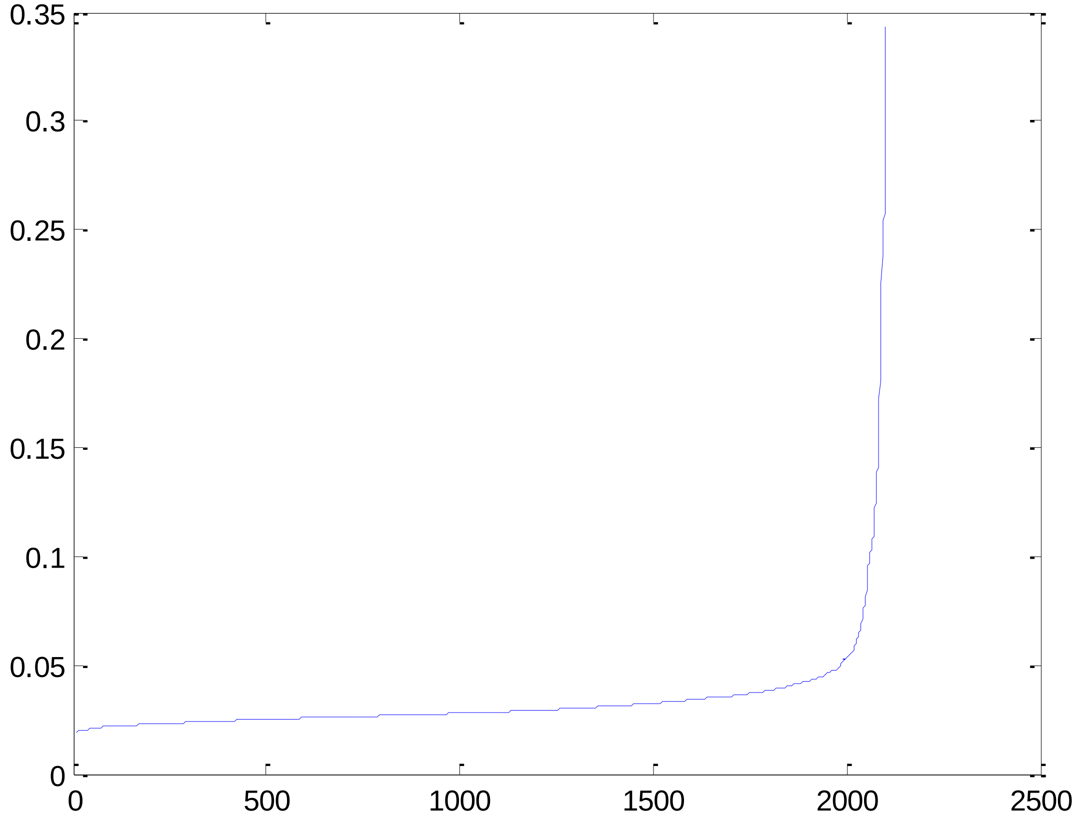


- MinPts = 10
- Eps = 0.04

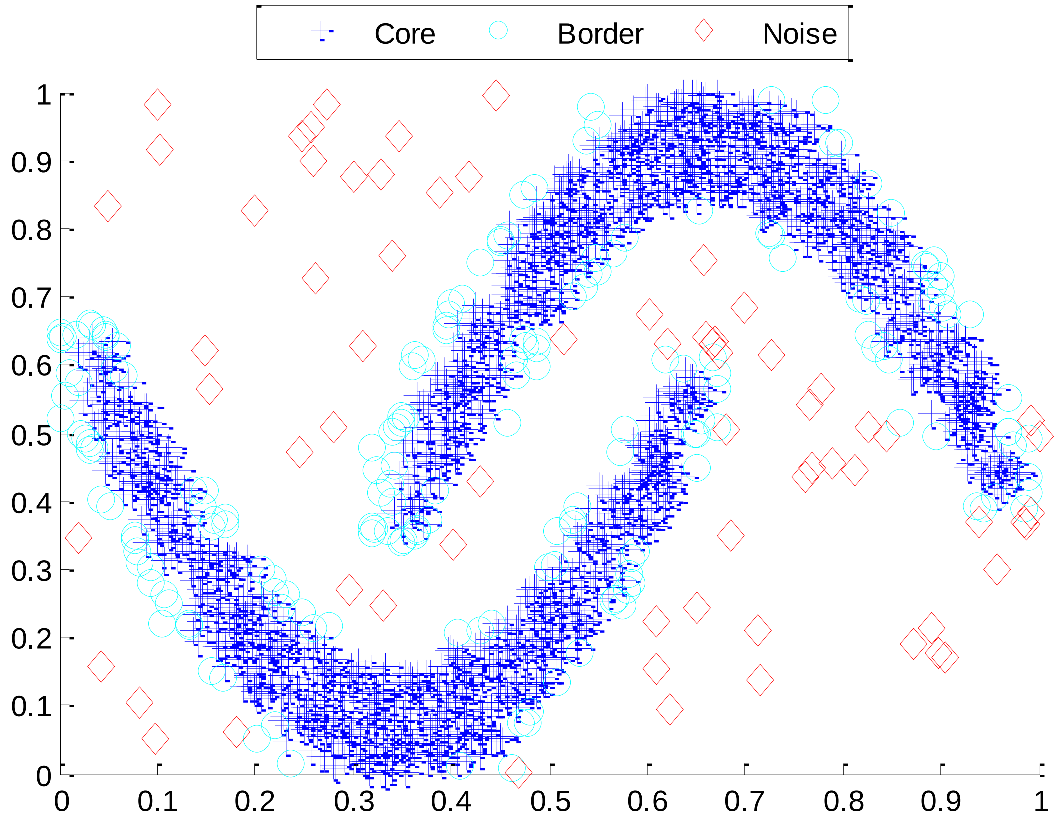
Sensitivity to Eps, MinPts (Example)



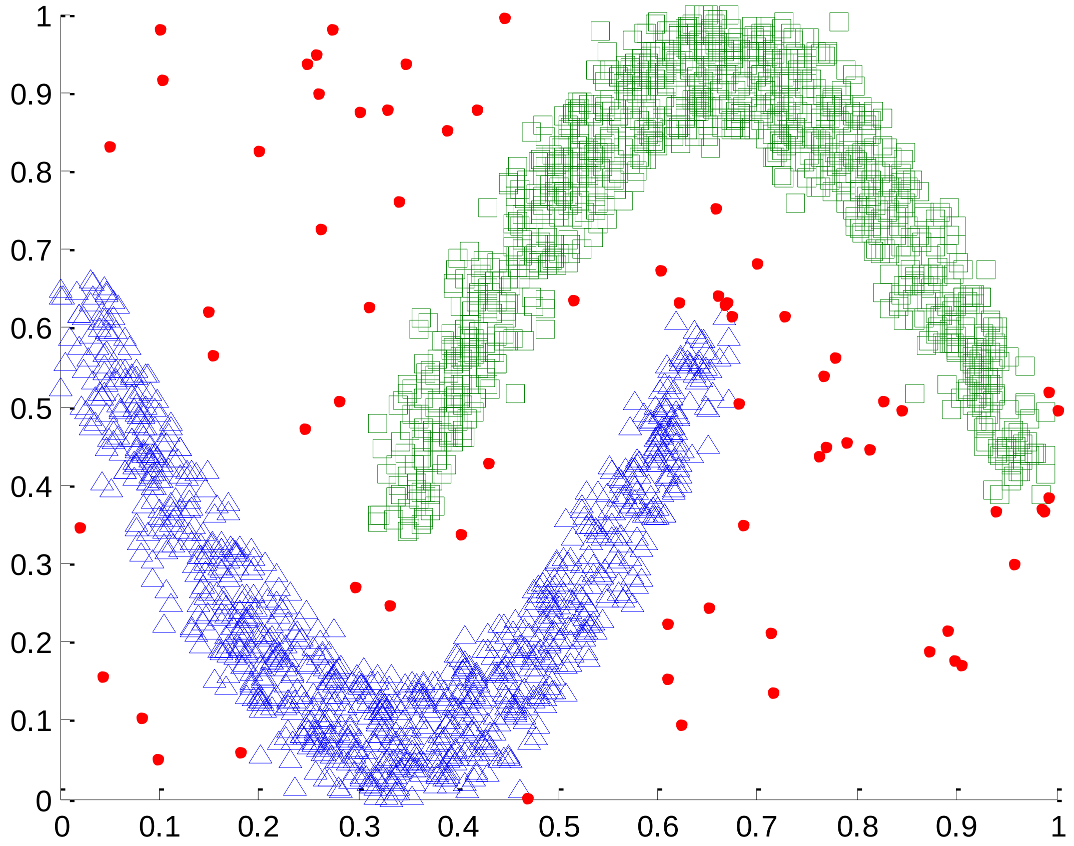
kNN Plot (k=20)



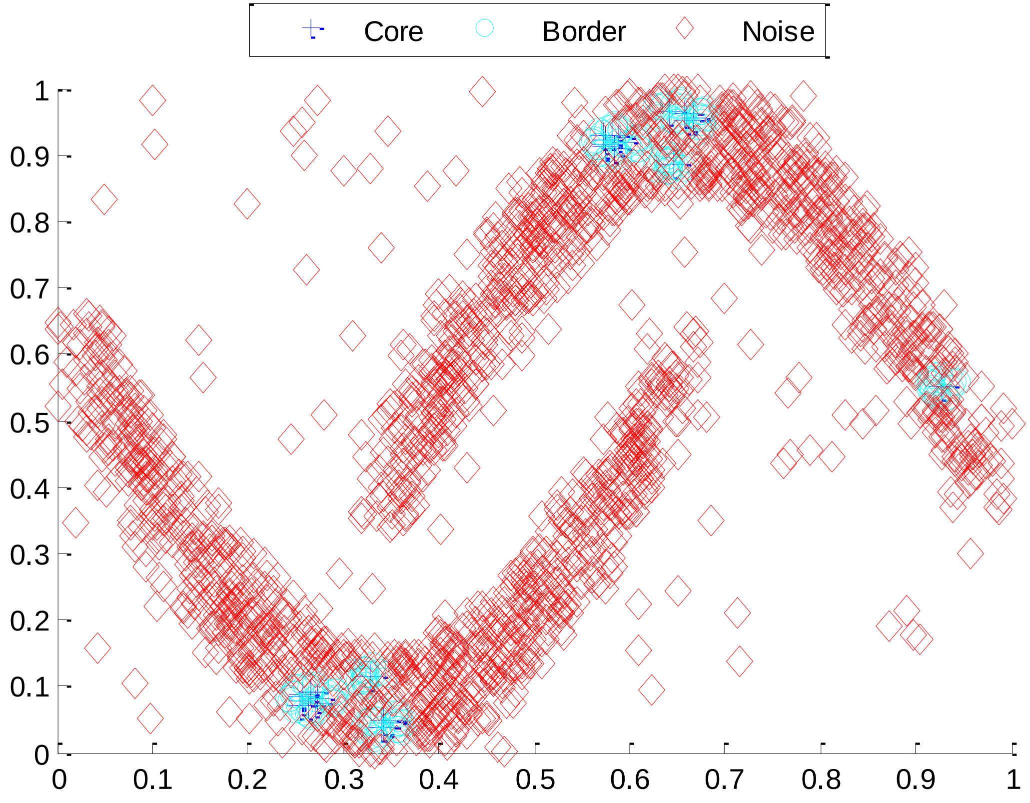
Point classification (Eps = 0.04, MinPts = 20)



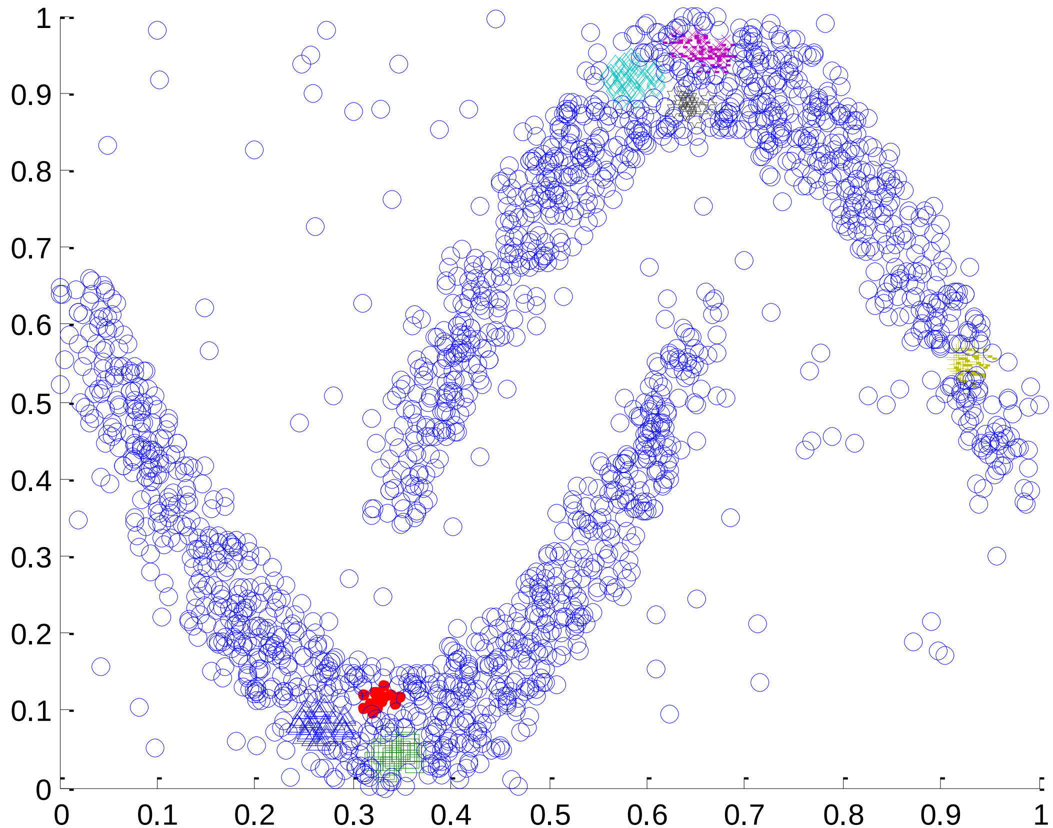
Clustering result (Eps = 0.04, , MinPts = 20)



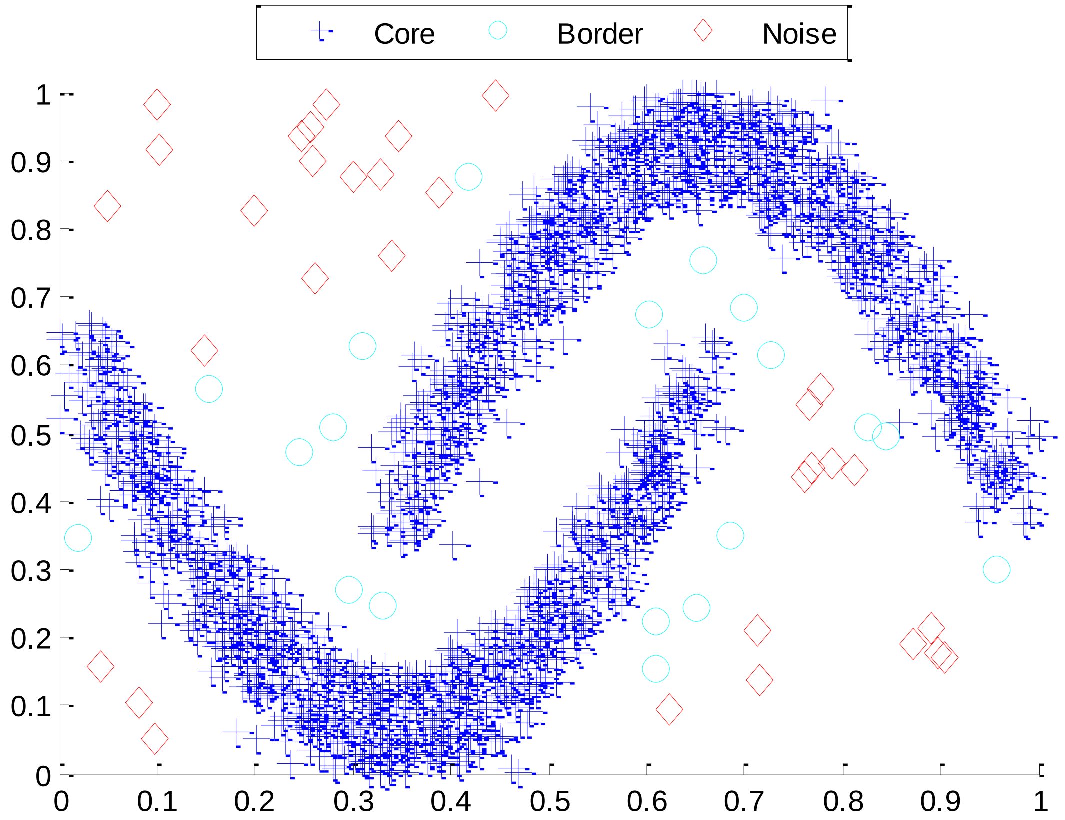
Point classification (Eps = 0.02, MinPts = 20)



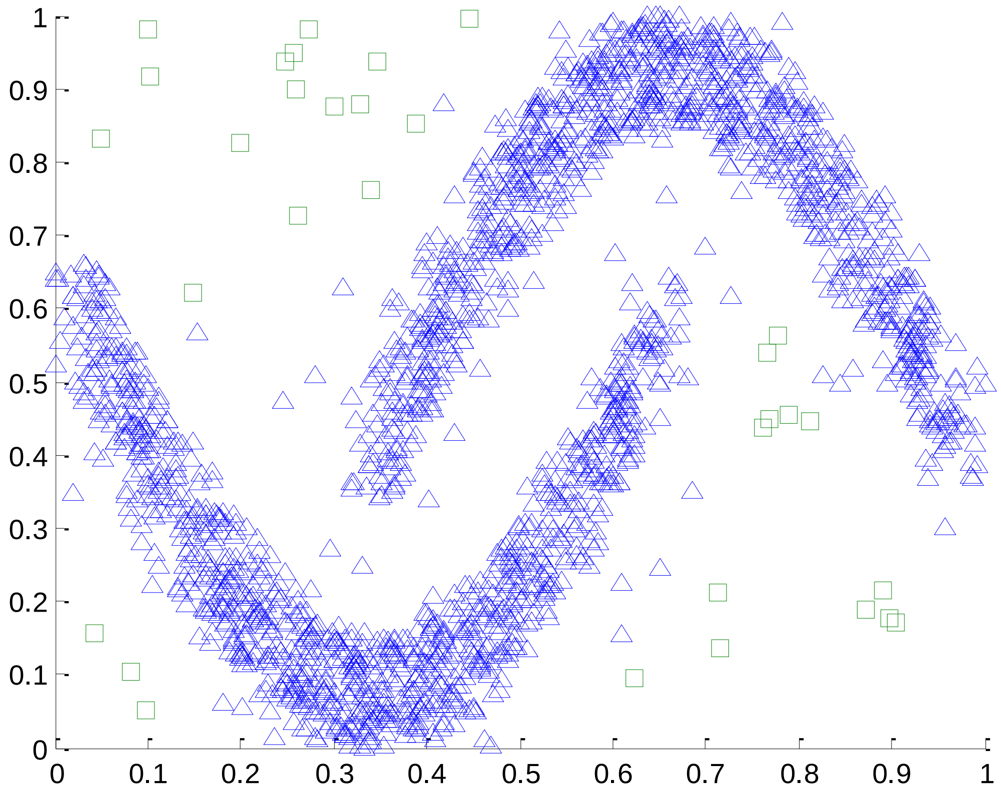
Clustering result (Eps = 0.02, , MinPts = 20)



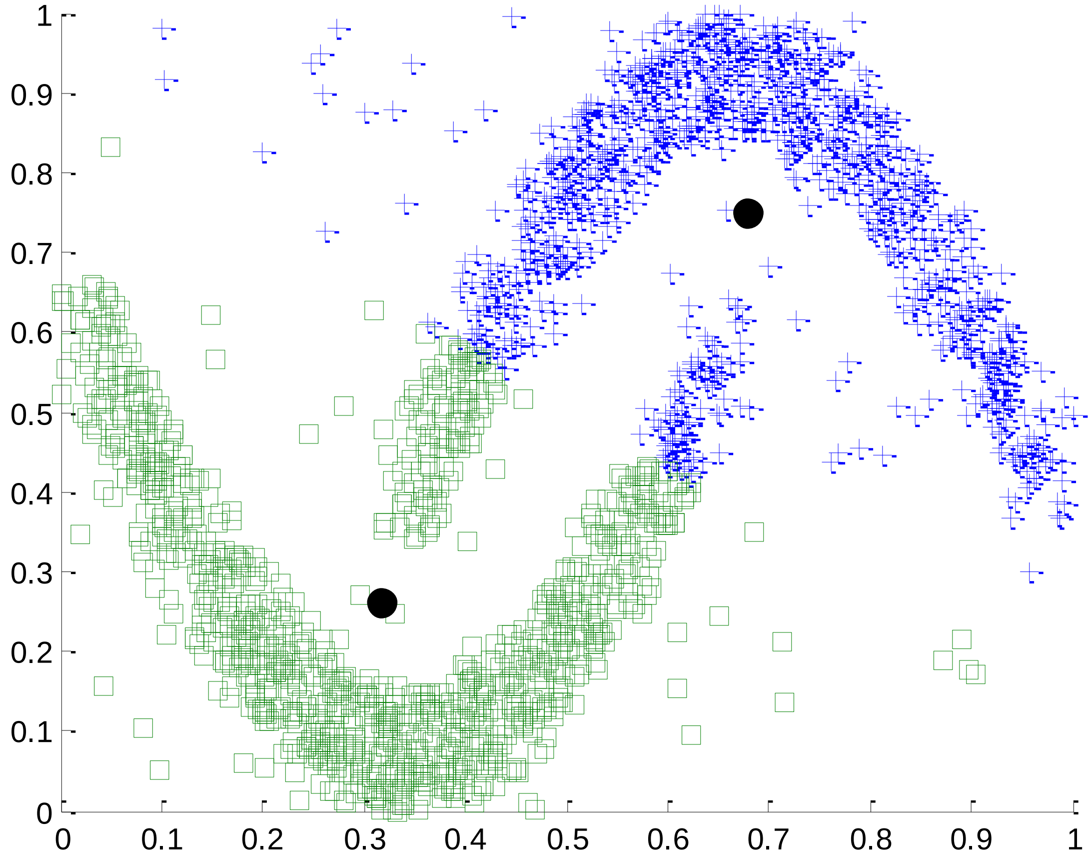
Point classification (Eps = 0.08, MinPts = 20)



Clustering result (Eps = 0.08, MinPts = 20)



DBScan vs. k-means



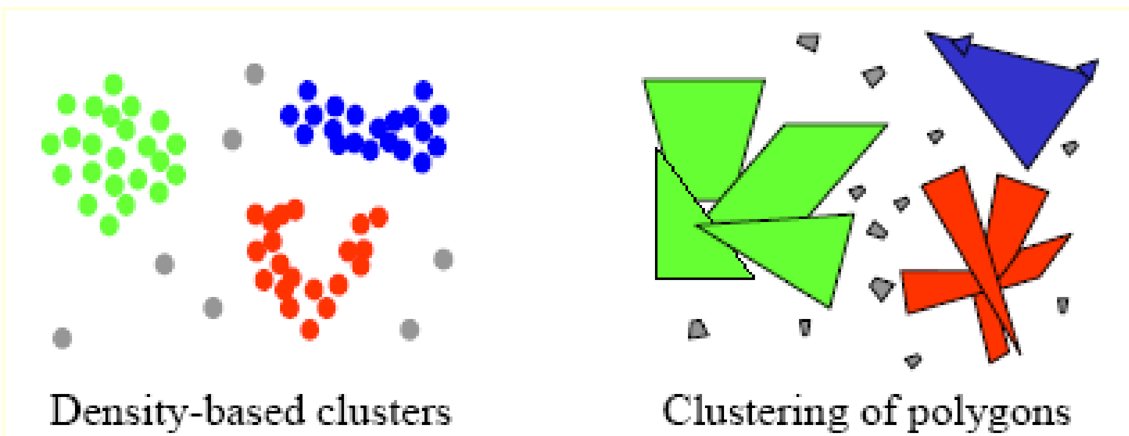
DBScan

- **Advantages**
- DBScan does not require you to know the number of clusters in the data a priori. Compare this with k-means.
- BScan does not have a bias towards a particular cluster shape or size. Compare this with k-means.
- DBScan is resistant to noise and provides a means of filtering for noise if desired.

- **Disadvantages**
- DBScan does not respond well to data sets with varying densities so called hierarchical data sets.

GDBSCAN

- generalized algorithm
- can cluster point objects as well as spatially extended objects
  - e.g., 2D polygons used in geography



Cluster Validity

- For supervised classification we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall

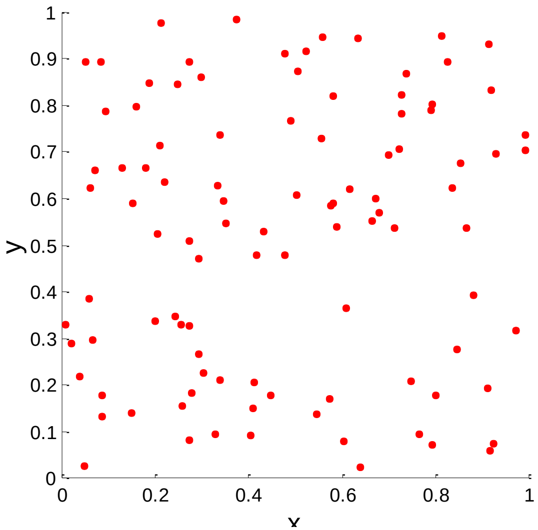
- For cluster analysis, the analogous question is how to evaluate the "goodness" of the resulting clusters?

- But "clusters are in the eye of the beholder"!

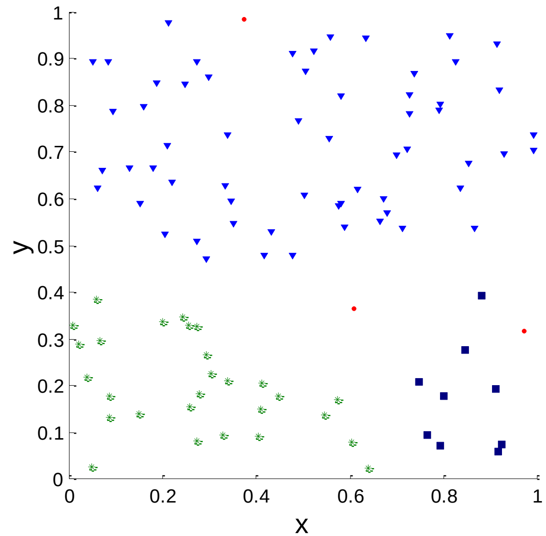
- Then why do we want to evaluate them?
  - To avoid finding patterns in noise
  - To compare clustering algorithms

Clusters found in Random Data

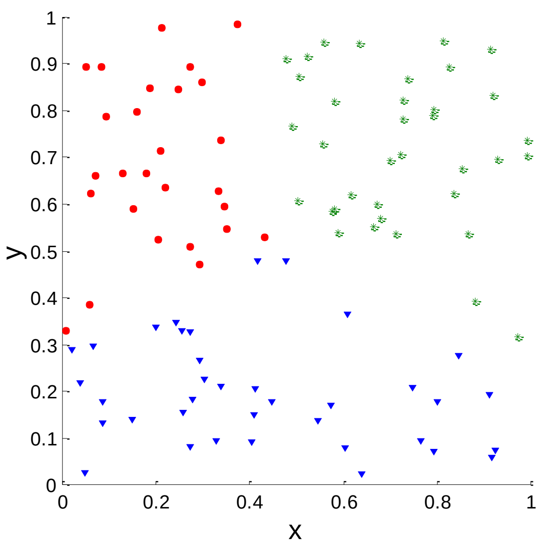
Random Points



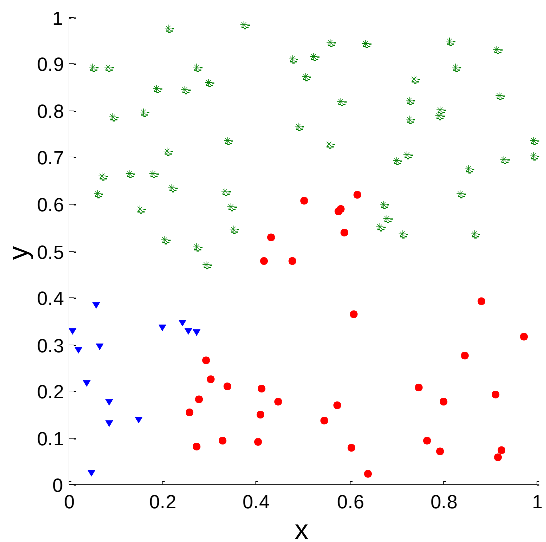
DBSCAN



K-means



Complete Link



Different Aspects of Cluster Validation

1. Determining the clustering tendency of a set of data, i.e., distinguishing whether non-random structure actually exists in the data.
2. Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels.
3. Evaluating how well the results of a cluster analysis fit the data *without* reference to external information.
   - Use only the data
4. Determining the 'correct' number of clusters.

Measures of Cluster Validity

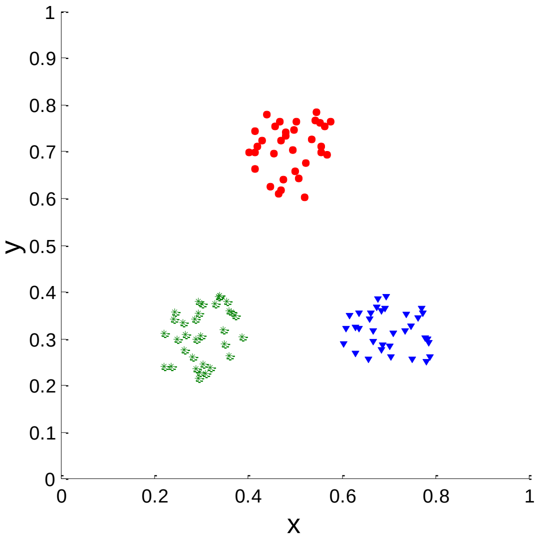
- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy
  - Internal Index: Used to measure the goodness of a clustering structure *without* respect to external information.
    - Sum of Squared Error (SSE)
  - 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

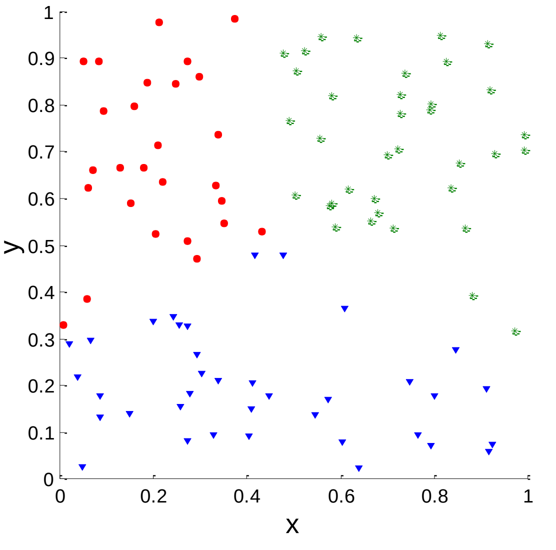
Measuring Cluster Validity Via Correlation

- Two matrices
  - Proximity Matrix
  - "Incidence" Matrix
    - 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.

Measuring Cluster Validity Via Correlation

- Correlation of incidence and proximity matrices for the K-means clusterings of the following two data sets.



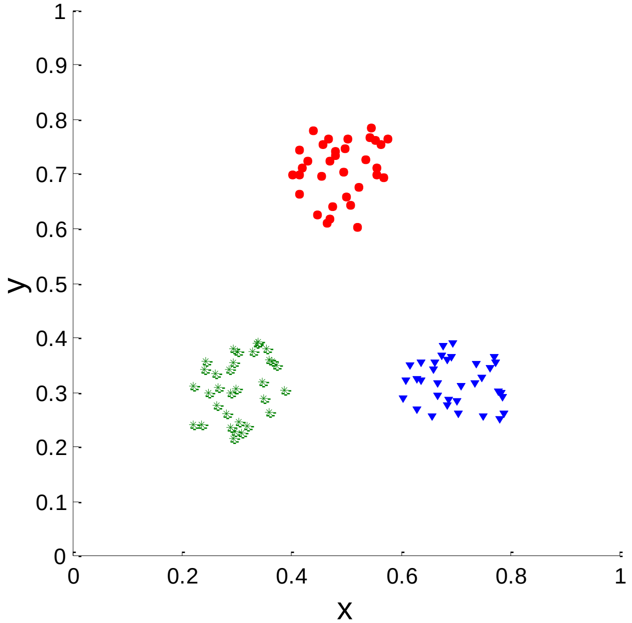


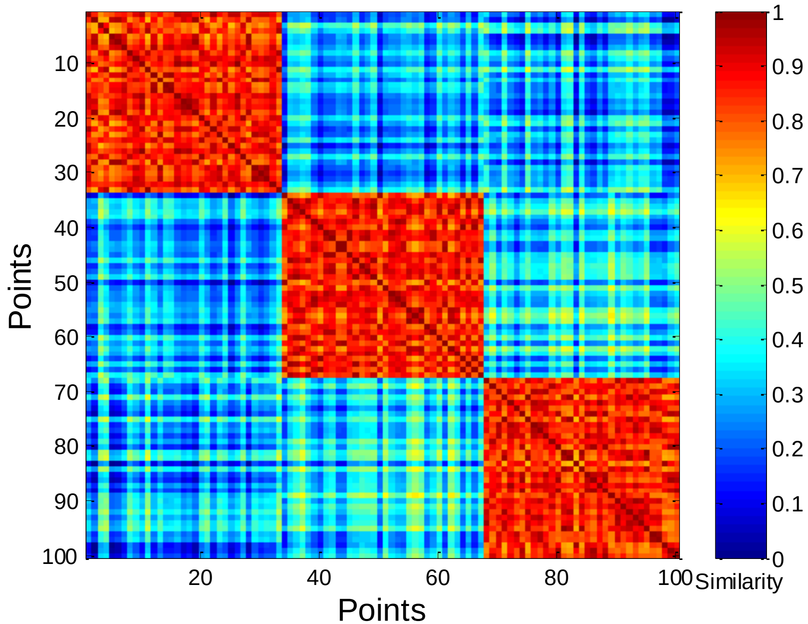
Corr = -0.9235

Corr = -0.5810

Using Similarity Matrix for Cluster Validation

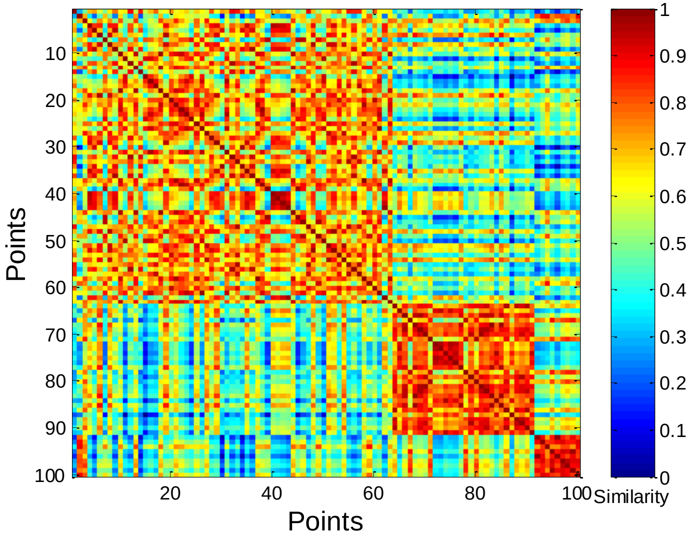
- Order the similarity matrix with respect to cluster labels and inspect visually.

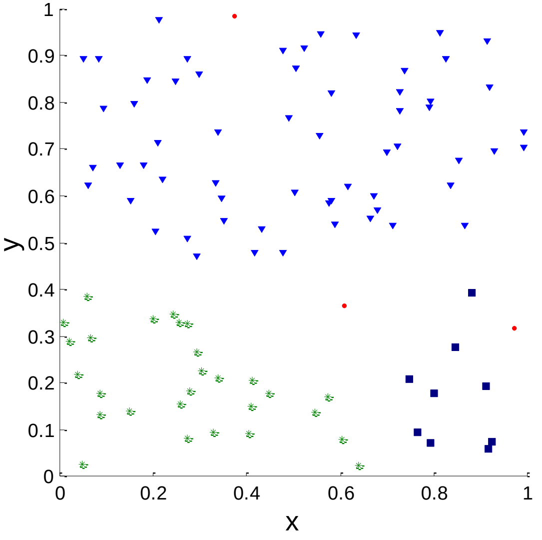




Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

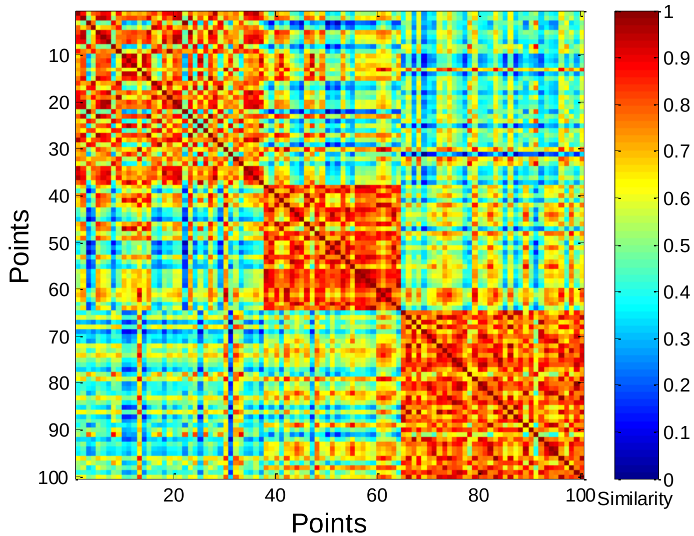


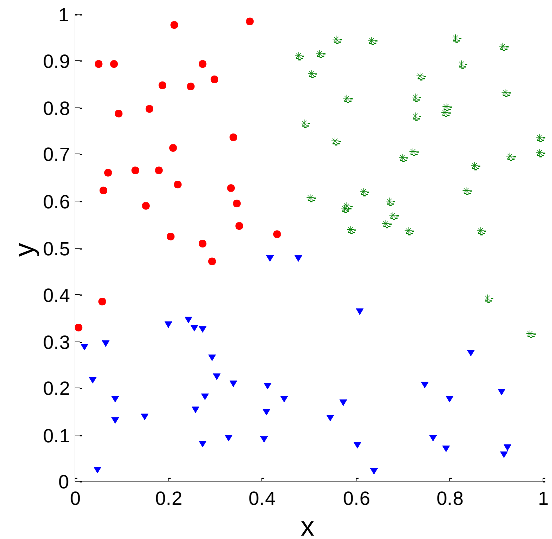


DBSCAN

Using Similarity Matrix for Cluster Validation

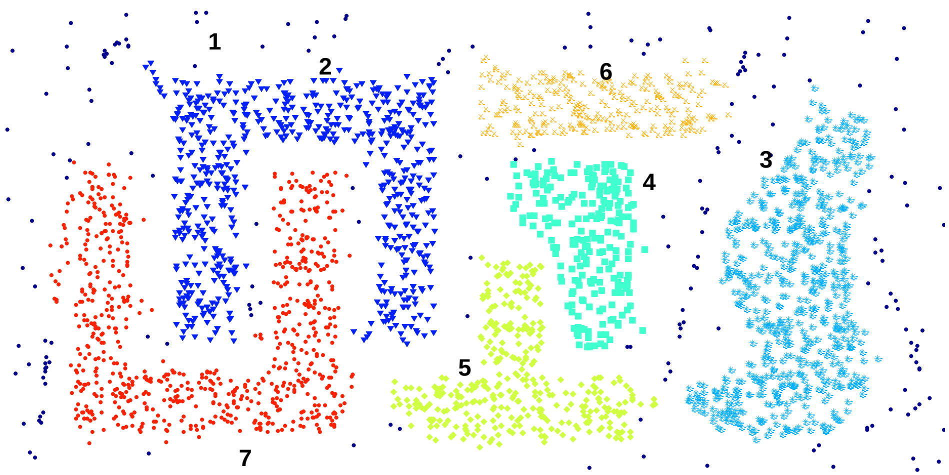
- Clusters in random data are not so crisp

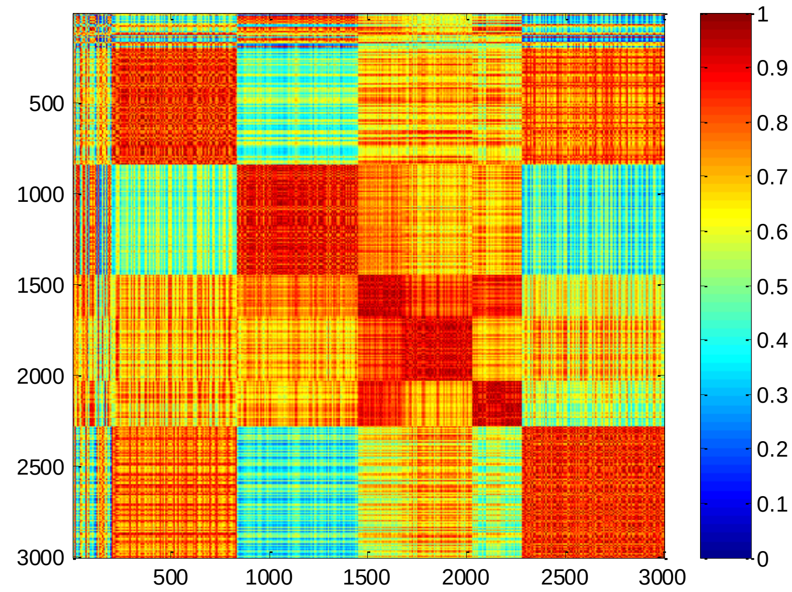




K-means

Using Similarity Matrix for Cluster Validation

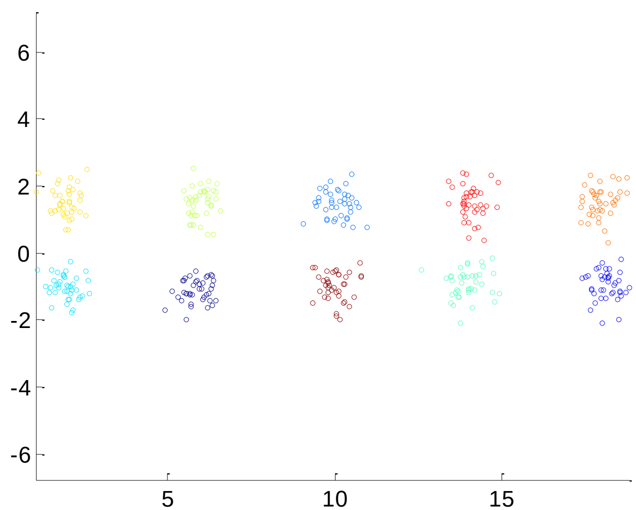


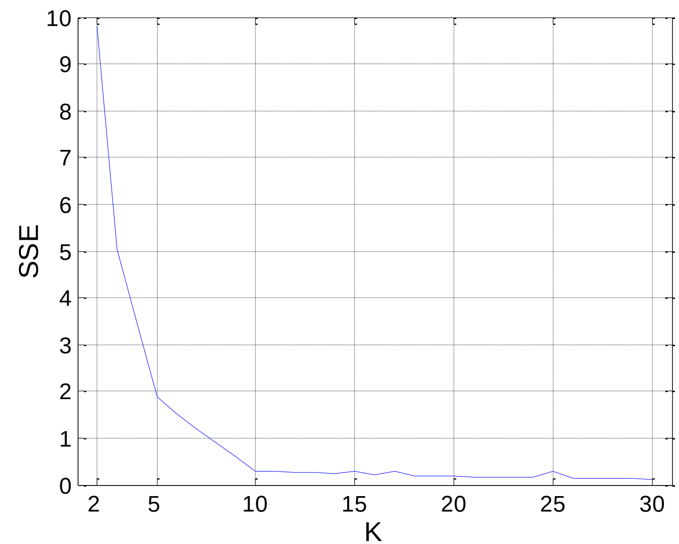


DBSCAN

Internal Measures: SSE

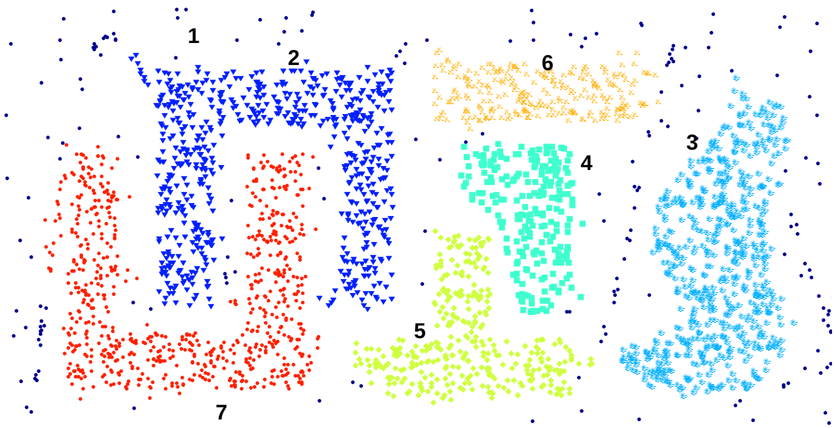
- Clusters in more complicated figures aren't well separated
- Internal Index: Used to measure the goodness of a clustering structure without respect to external information
  - SSE
- SSE is good for comparing two clusterings or two clusters (average SSE).
- Can also be used to estimate the number of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

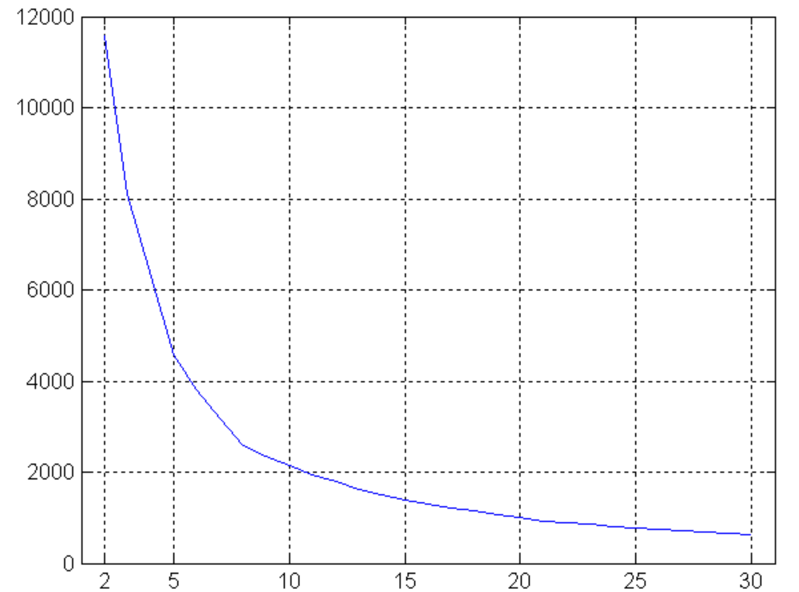




Internal Measures: SSE

- SSE curve for a more complicated data set

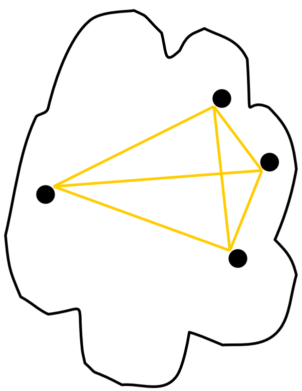




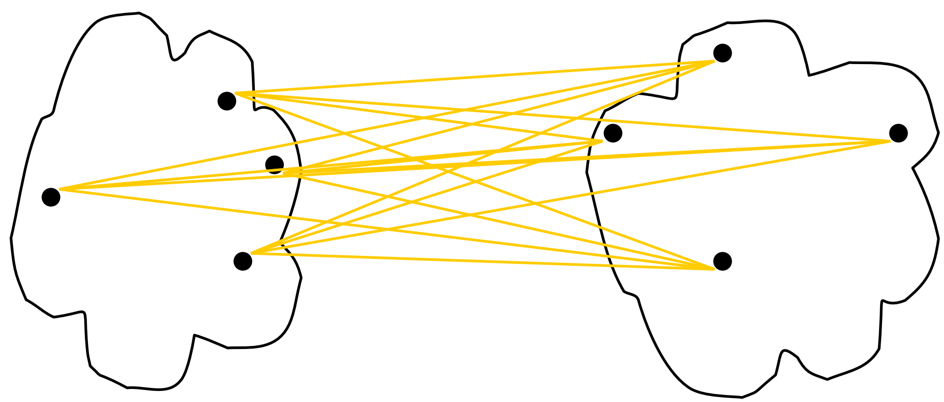
SSE of clusters found using K-means

Internal Measures: Cohesion and Separation

- A proximity graph based approach can also be used for cohesion and separation.
  - Cluster cohesion is the sum of the weight of all links within a cluster.
  - Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster.



cohesion



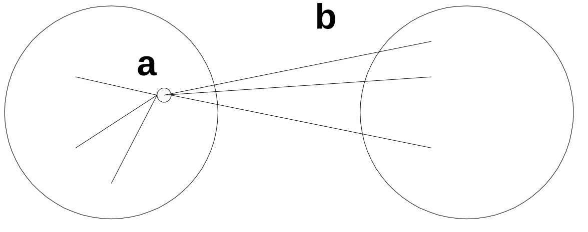
separation

Internal Measures: Silhouette Coefficient

- Silhouette Coefficient combine ideas of both cohesion and separation, but for individual points, as well as clusters and clusterings
- For an individual point, *i*
  - Calculate ***a*** = average distance of *i* to the points in its cluster
  - Calculate ***b*** = min (average distance of *i* to points in another cluster)
  - The silhouette coefficient for a point is then given by

    $s = 1 – a/b$ if $a < b$, (or $s = b/a - 1$ if $a \geq b$, not the usual case)

  - Typically between 0 and 1.
  - The closer to 1 the better.



- Can calculate the Average Silhouette width for a cluster or a clustering

Final Comment on Cluster Validity

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

Algorithms for Clustering Data, Jain and Dubes