Clustering

CS 550: Machine Learning

This slide set mainly uses the slides given in the following links:

http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf

http://www-users.cs.umn.edu/~kumar/dmbook/dmslides/chap8_basic_cluster_analysis.pdf

Clustering / Unsupervised Learning

- In **SUPERVISED** learning
  - There is a teacher providing labels (outputs) for training samples
  - The task is to map an input space to an output space

- In **UNSUPERVISED** learning
  - There is not an explicit teacher providing outputs
  - Task is to find regularities (clusters) in the input space
    - e.g., cluster customers based on their demographic information and past transactions for developing marketing strategies
    - e.g., cluster pixels based on their colors for image compression
  - Unsupervised learning can be used in
    - Understanding the data (e.g., group related documents for browsing; group genes and proteins with similar functionality)
    - Summarizing the data

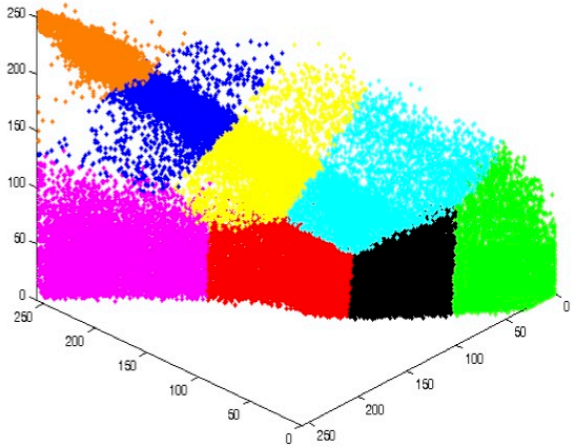
Example: Image Compression

Image compression to reduce the no of bits to be transferred

Originally in RGB space
→24 bits for each pixel

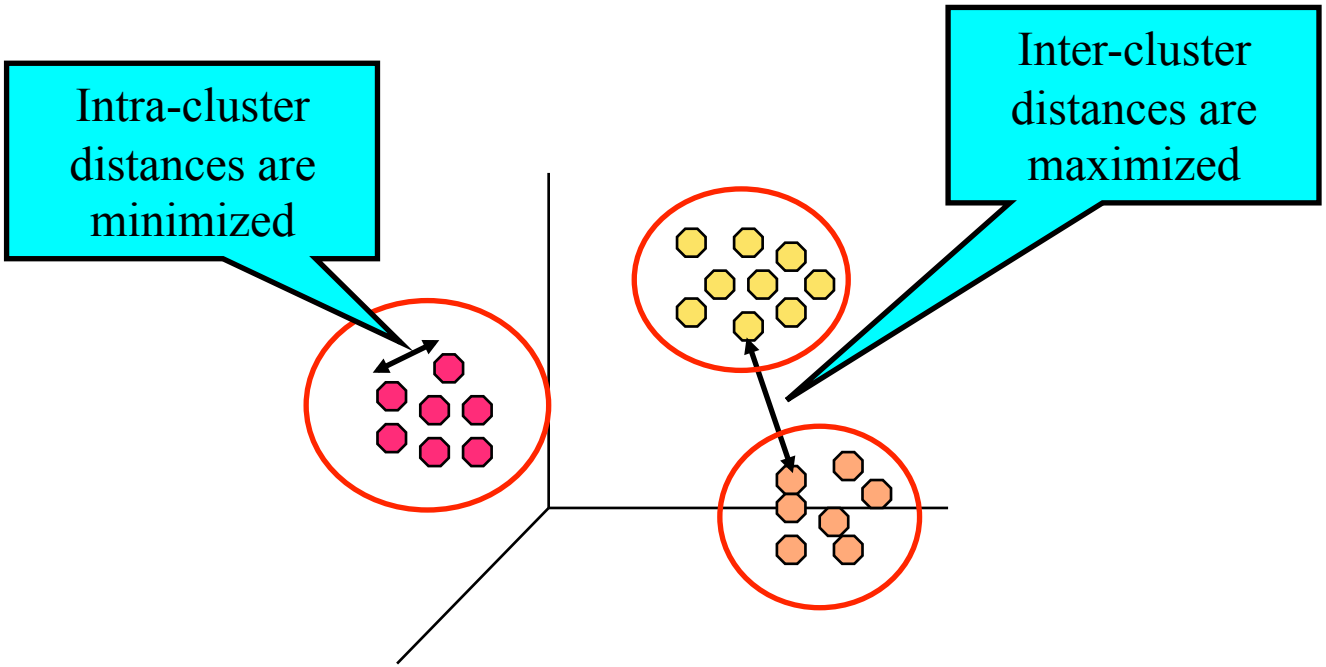
8 (= 2^3) clusters (colors)
→3 bits for each

32 (= 2^5) clusters (colors)
→5 bits for each pixel

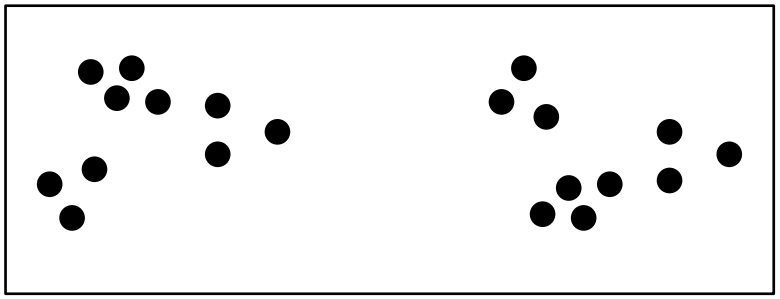


What is Cluster Analysis?

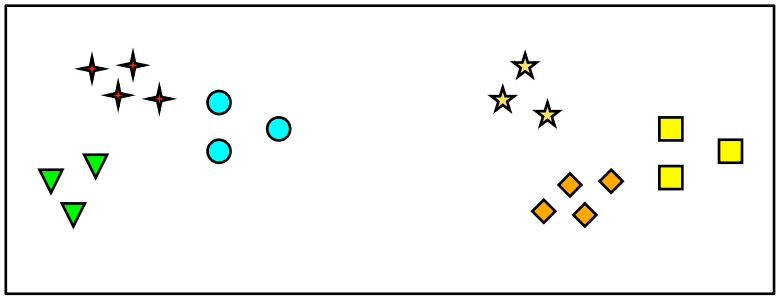
- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



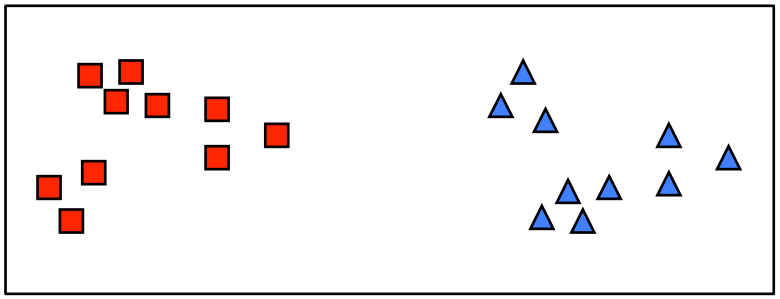
Notion of a Cluster can be Ambiguous



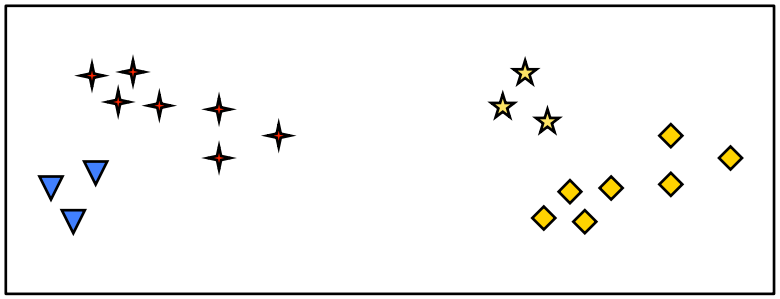
How many clusters?



Six clusters



Two clusters



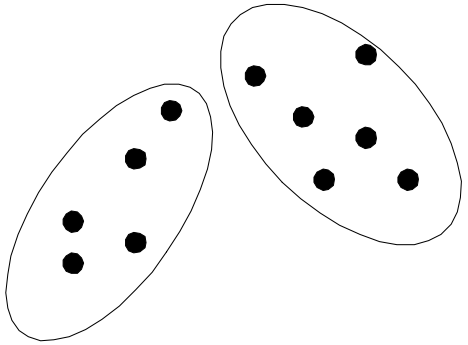
Four clusters

Types of Clusterings

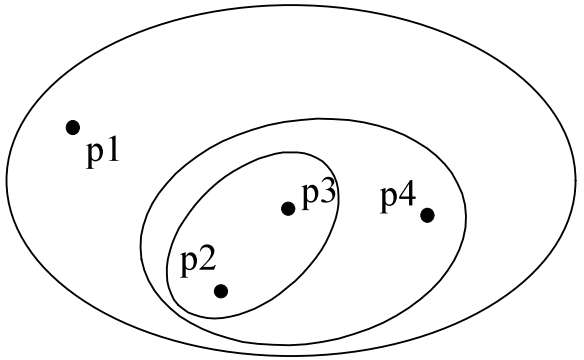
A clustering is a set of clusters

1. **Partitional clustering**
   - A division of objects into non-overlapping subsets (clusters) such that each object is in exactly one subset

2. **Hierarchical clustering**
   - A set of nested clusters



Partitional clustering



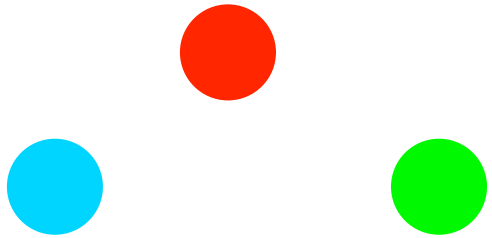
Hierarchical clustering

Characteristics of Clusters

- **Exclusive versus non-exclusive:** a point may belong to multiple clusters in non exclusive clustering
- **Fuzzy versus non-fuzzy:** a point belongs to every cluster with some weight between 0 and 1 (weights must sum up to 1)
- **Partial versus complete:** in some cases, we may want to cluster only some of the data
- **Heterogeneous vs homogeneous:** clusters may have different sizes, shapes, and/or characteristics

Types of Clusters

- **Well-separated clusters**
  - A cluster is a set of points such that any point in a cluster is closer (more similar) to every other point in the cluster than to any point not in the cluster



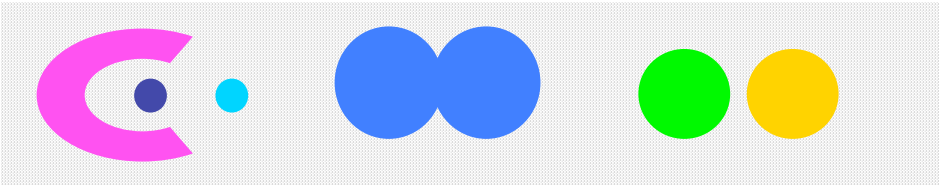
- **Center-based clusters**
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster than to the center of any other cluster
    - Centroid (center of a cluster) is the average of all points in the cluster
    - Medoid is the most "representative" point of a cluster

Types of Clusters

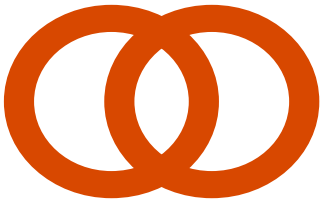
- **Contiguity-based clusters**
  - A cluster is a set of points such that a point in a cluster is closer (more similar) to one or more other points in the cluster than to any point not in the cluster

Types of Clusters

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



- **Conceptual clusters**
  - A cluster shares some common property or represents a particular concept



Two overlapping circles

Clusters Defined by an Objective Function

- Find clusters that minimize/maximize an objective function
- Consider all possible clusterings and evaluate the *goodness* of each using the objective function (NP-hard)
- Can use global or local objective functions
  - Hierarchical clustering algorithms typically have local objectives
  - Partitional clustering algorithms typically have global objectives
- A variation of the global objective function approach is to fit the data to a parameterized model
  - Parameters for the model are determined from the data
  - Mixture models assume that the data is a *mixture* of a number of statistical distributions

Clusters Defined by an Objective Function

- Map the clustering problem to a different domain and solve a related problem in that domain
  - Similarity matrix defines a weighted graph, where the nodes are the points being clustered, and the weighted edges represent the similarities between the points
  - Clustering is equivalent to breaking the graph into connected components (one for each cluster)
  - Minimize the sum of the edge weights between clusters and maximize the sum of the edge weights within clusters

Clustering Algorithms

1. K-means clustering
2. Hierarchical clustering
3. Density-based clustering

K-means Clustering

- Example of a partitional clustering
- Each cluster is represented with a mean vector (centroid)

start with randomly initialized cluster (mean) vectors m_i

do

for each cluster, estimate samples that belong to the mean vector m_i (estimate D_i) } **E-step**

for each cluster, compute the mean vector m_i that minimizes/maximizes the criterion function } **M-step**

until there is no (or small) change in m_i

This is an example of the expectation-maximization (EM) algorithm

K-means Clustering

D_i estimation

Each cluster contains samples that are most similar to m_i

$$D_i = \left\{ x \;\middle|\; \underbrace{\| x - m_i \|^2}_{\textit{Euclidean distance}} = \min_j \| x - m_j \|^2 \right\}$$

m_i computation

Set a value as to minimize/maximize a criterion function

$$\frac{\partial E}{\partial m_i} = 0$$

$$\frac{\partial E}{\partial m_i} = \frac{1}{2} \sum_{x \in D_i} -2\,(x - m_i) = -\sum_{x \in D_i} x + \sum_{x \in D_i} m_i$$

$$m_i = \frac{\sum_{x \in D_i} x}{n_i}$$

Number of samples in D_i

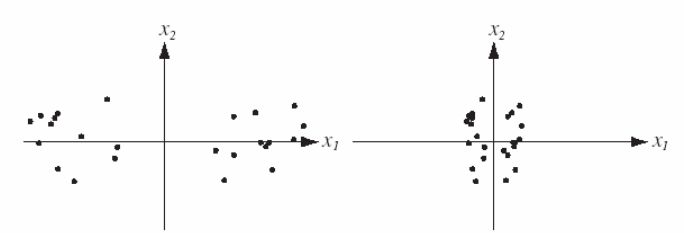
$$E = \underbrace{\frac{1}{2} \sum_{i=1}^{k} \sum_{x \in D_i} \| x - m_i \|^2}_{\textit{Sum of squared error}}$$

K-means Clustering

How to measure the similarity between samples?

- Use a distance metric as the dissimilarity between samples
  - Euclidean distance

$$dist(x, y) = \| x - y \|^2 = \sqrt{\sum_{j=1}^{d} (x_j - y_j)^2} = \sqrt{(x-y)^T (x-y)}$$



Invariant to translation and rotation, but not to scaling

  - Mahalanobis distance

$$dist(x, y) = \sqrt{(x-y)^T \Sigma^{-1} (x-y)}$$

Also scale invariant

- Define a similarity function
  - Normalized inner product

$$sim(x, y) = \frac{x^T y}{\|x\| \|y\|}$$

Invariant to rotation, but not to translation and scaling

K-means Clustering

How to evaluate the goodness of clustering?

- Sum of squared error

$$E = \frac{1}{2} \sum_{i=1}^{k} \sum_{x \in D_i} \| x - m_i \|^2$$

Could lead to incorrect clusters especially when there are great differences in clusters' sizes and when there are outliers

- Related minimum variance criterion

$$E = \frac{1}{2} \sum_{i=1}^{k} n_i \ \sigma_i$$

where

$$\sigma_i = \underbrace{\frac{1}{n_i^2} \sum_{x \in D_i} \sum_{y \in D_i} \| x - y \|^2}$$

Average squared distance between every pair of the samples in the i-th cluster

It is also possible to use a similarity measure instead of a distance

K-means Clustering

How to evaluate the goodness of clustering?

- Scatter criteria

$$S_T = \underbrace{S_W}_{\text{Within cluster scatter}} + \underbrace{S_B}_{\text{Between cluster scatter}}$$

Total scatter

$$S_W = \sum_{i=1}^{k} \sum_{x \in D_i} (x - m_i)\,(x - m_i)^{\mathsf{T}}$$

$$S_B = \sum_{i=1}^{k} n_i\,(m_i - m)\,(m_i - m)^{\mathsf{T}}$$

Mean of all samples

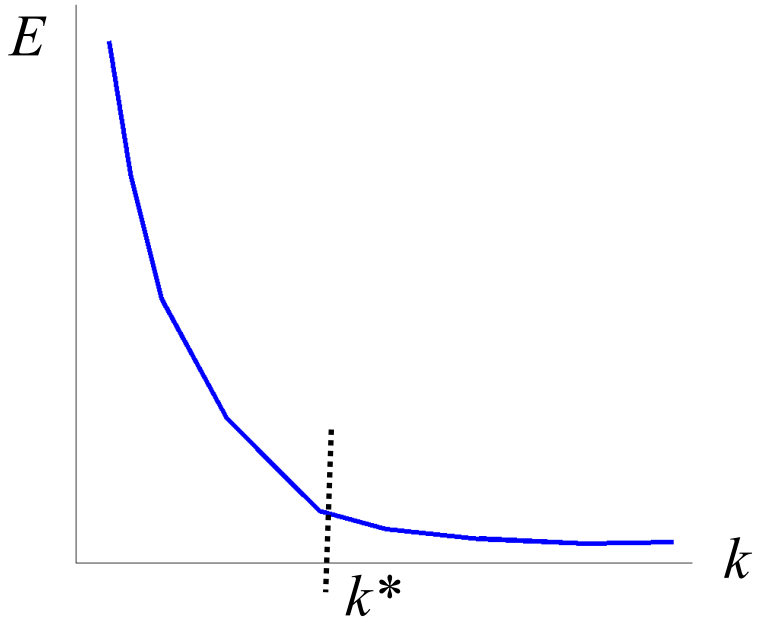
Maximize the between-cluster scatter or minimize the within-cluster scatter

1. Trace criterion: Minimize the trace of S_W (sum of the diagonal elements of S_W)
2. Determinant criterion: Minimize the determinant of S_W
3. Invariant criterion: Maximize the trace of $S_W^{-1} S_B$

K-means Clustering

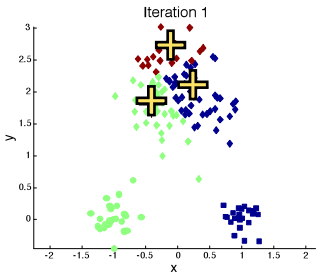
How to select the number of clusters?

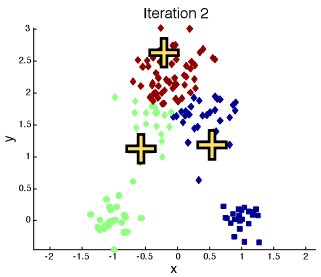
- We expect a rapid decrease in the criterion function until k is equal to the number of natural clusters in the data and more slow decreases thereafter

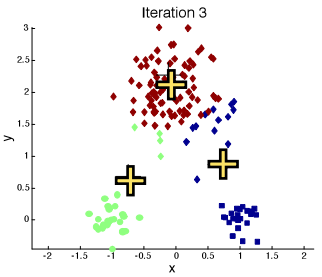


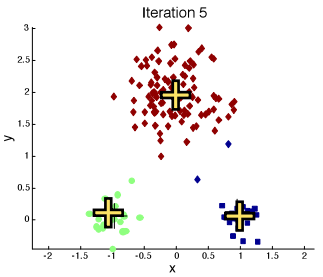
K-means Clustering

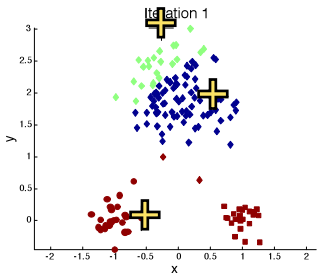
Final clustering highly depends on the initial mean vectors

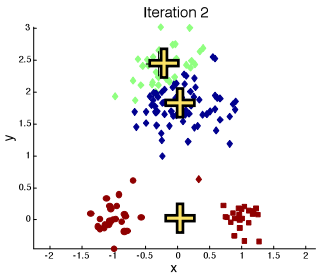


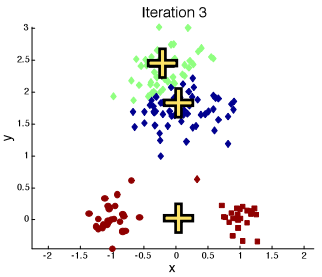


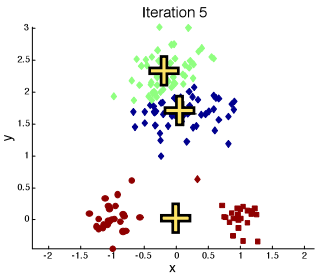












How to Select Initial Mean Vectors?

- Use hierarchical clustering to determine initial centroids (mean vectors)
- Select more than k initial centroids, run k-means, and select the "best" centroids as the initial centroids of the final k-means clustering
- Use postprocessing
- Use bisecting k-means algorithm, which is not as susceptible to initialization issues

Preprocessing and Postprocessing

- Preprocessing
  - Data normalization
  - Outlier elimination
- Postprocessing
  - Eliminate small clusters that may represent outliers
  - Split "loose" clusters with relatively high error
  - Merge "close" clusters with relatively low error
  - These steps can be used also during the clustering process
- Basic k-means algorithm may yield empty clusters
  - Choose the point that contributes most to the error function
  - Choose a point from the cluster with the highest error

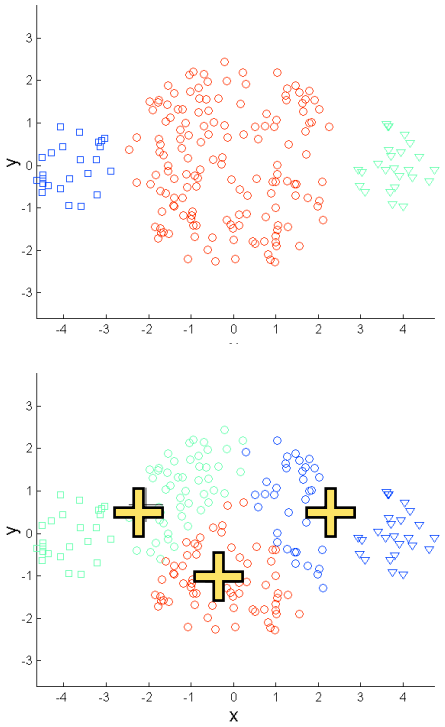
Bisecting K-means Clustering

```
start with a single cluster containing all
samples (this is the initial list of clusters)

repeat
      pick a cluster to split from the list

      for i = 1 to N          (bisecting step)
            bisect the selected cluster using
            the basic k-means
      end for

      add the two clusters corresponding
      to the "best" split into the list

until the list contains k clusters
```

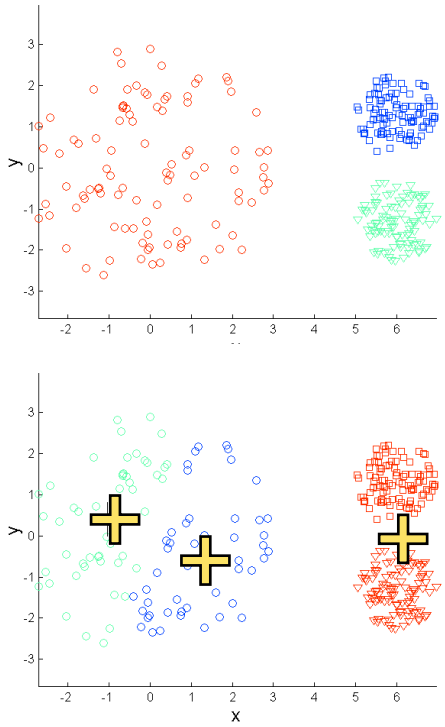
Limitations of K-means Clustering

- K-means may have problems when clusters are of different sizes, densities, and/or non-globular shapes and when data contain some outliers

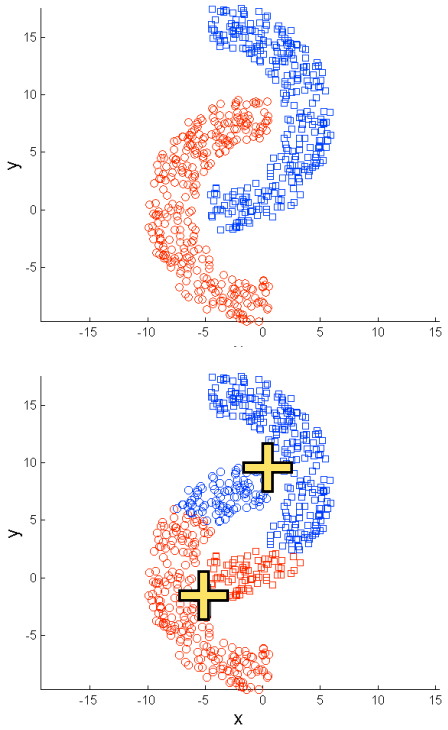
Different sizes



Different densities



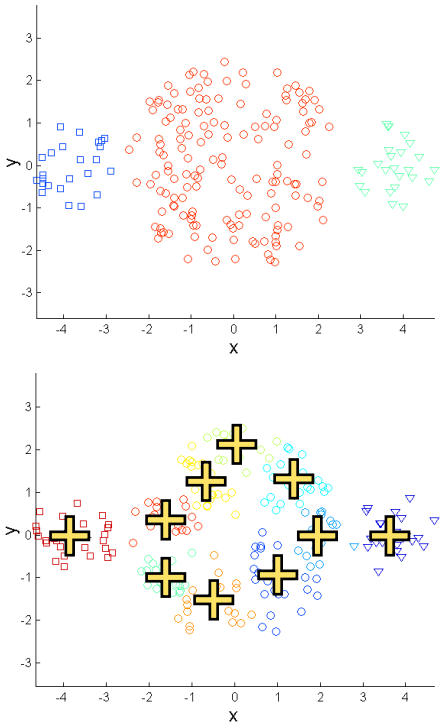
Non-globular shapes



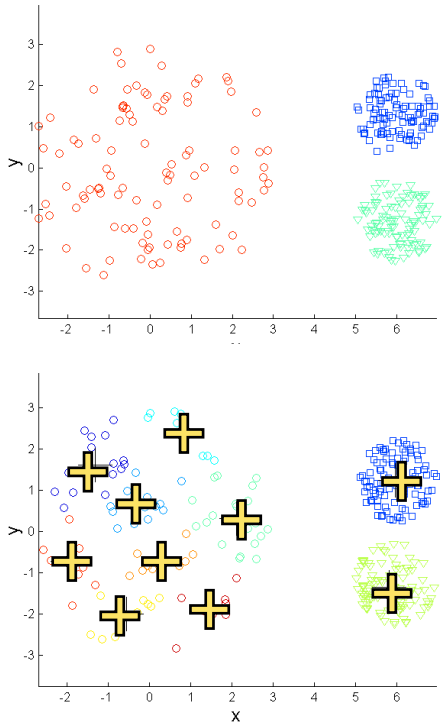
Overcoming K-means Limitations

- One solution is to use many clusters and then put them together to identify the final clusters

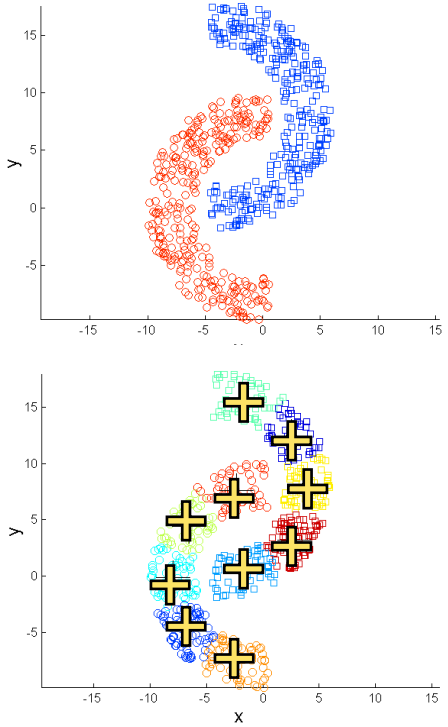
Different sizes



Different densities



Non-globular shapes



Fuzzy K-means Clustering

Each sample has "fuzzy" membership in every cluster c_i

$$E_{fuzzy} = \sum_{i=1}^{k} \sum_{t=1}^{N} \left[P(c_i \mid x^t) \right]^b \left\| x^t - m_i \right\|^2$$

Blending parameter

Memberships quantified as posteriors

$$\frac{\partial E_{fuzzy}}{\partial m_i} = 0 \quad \text{and} \quad \frac{\partial E_{fuzzy}}{\partial P_i} = 0$$

It may improve convergence compared to k-means. However, serious problems may arise when k is incorrectly specified.

initialize mean vectors m_i and posteriors $P(m_i, x^t)$

normalize $P(m_i, x^t)$

do

recompute m_i

recompute $P(m_i, x^t)$

until there is small change in m_i and $P(m_i, x^t)$

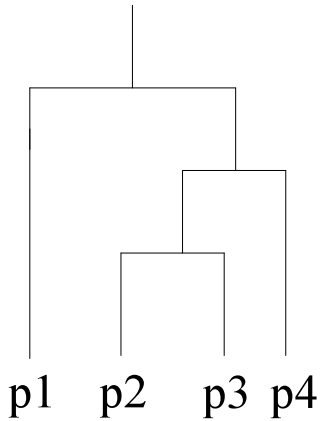
$$\sum_i P(c_i \mid x^t) = 1$$

$$m_i = \frac{\sum_t \left[P(c_i \mid x^t) \right]^b x^t}{\sum_t \left[P(c_i \mid x^t) \right]^b}$$

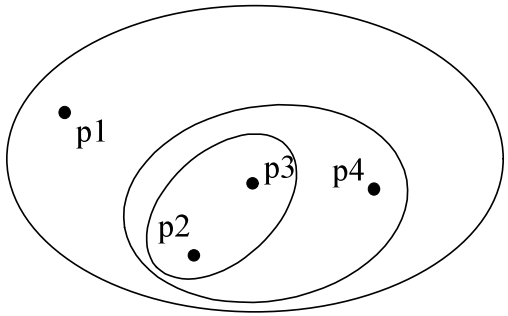
$$P(c_i \mid x^t) = \frac{\left(1 / \left\| x^t - m_i \right\|^2 \right)^{1/(b-1)}}{\sum_k \left(1 / \left\| x^t - m_k \right\|^2 \right)^{1/(b-1)}}$$

Hierarchical Clustering

- Produces a set of nested clusters organized in a hierarchy
- Can be represented with dendrograms or sets



Dendrogram representation



Set representation

- Strengths
  - 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

Hierarchical Clustering

- Two main types of hierarchical clustering

1. Agglomerative (bottom-up)
   - Start with N singleton clusters and merge the clusters successively with respect to their (dis)similarities

2. Divisive (top-down)
   - Start with a single cluster containing all samples and split the clusters successively with respect to their (dis)similarities

Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward

```
let each sample be a cluster
compute the dissimilarity (distance) matrix
repeat
      merge the two most similar cluster
      update the distance matrix
until only a single cluster remains
```

- Key operation is the definition of (dis)similarity between two clusters

How to Define (Dis)similarity

- Similarity between clusters c_i and c_j can be defined as

$$s_{\min}(c_i, c_j) = \min_{\substack{x \in D_i \\ y \in D_j}} \| x - y \|^2$$

$$s_{\max}(c_i, c_j) = \max_{\substack{x \in D_i \\ y \in D_j}} \| x - y \|^2$$

Results are more sensitive to outliers

$$s_{\text{avg}}(c_i, c_j) = \frac{1}{n_i n_j} \sum_{x \in D_i} \sum_{y \in D_j} \| x - y \|^2$$

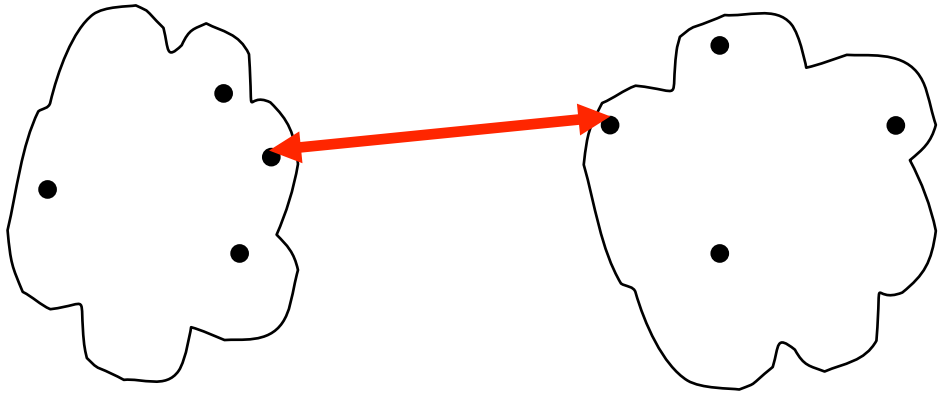
$$s_{\text{mean}}(c_i, c_j) = \| m_i - m_j \|^2$$

- Instead of using the Euclidean distance, one can use another distance metric or a similarity measure

How to Define (Dis)similarity

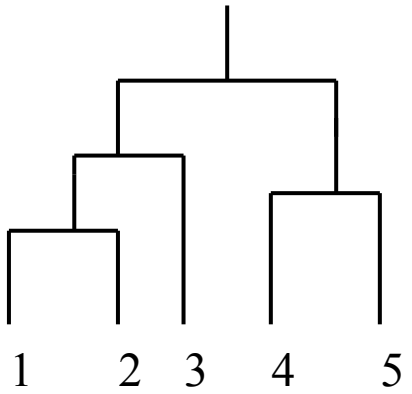
Minimum similarity (or single linkage algorithm)

- Similarity of two clusters is based on the two most similar samples in different clusters

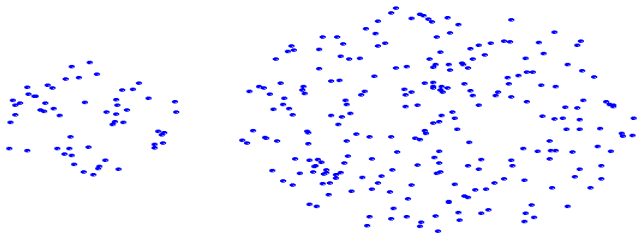


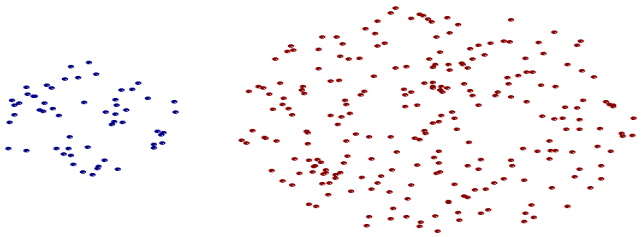
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00

Similarity matrix

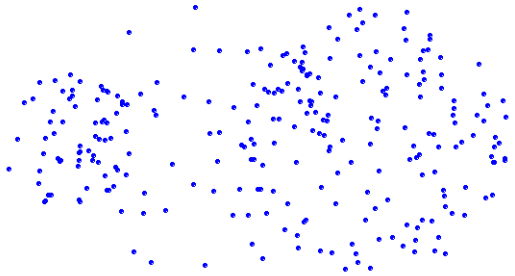


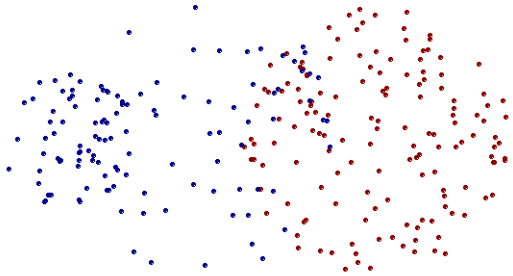
Minimum Similarity (Single Linkage)





Strength: *It can handle non-elliptical shapes*



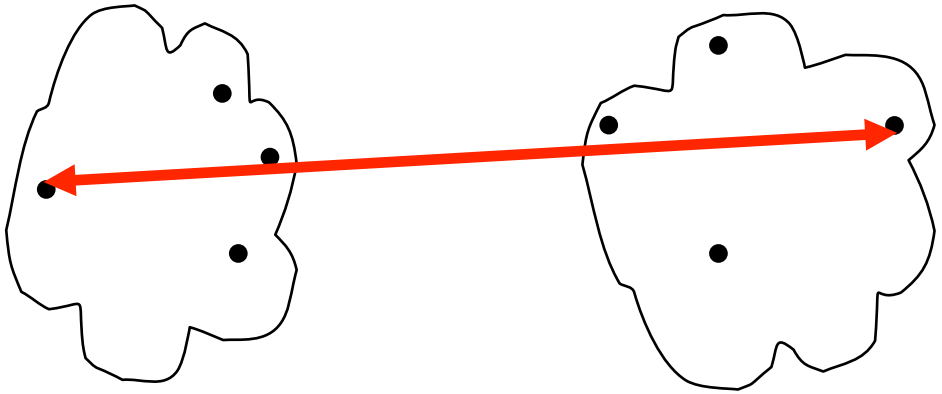


Limitation: *It is sensitive to noise and outliers*

How to Define (Dis)similarity

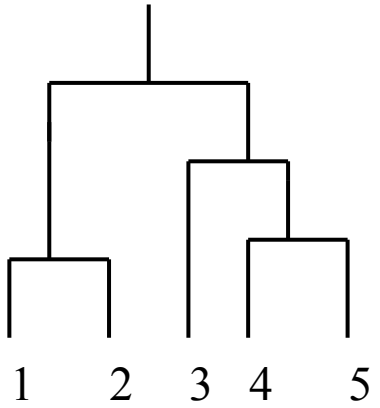
Maximum similarity (or complete linkage algorithm)

- Similarity of two clusters is based on the two least similar samples in different clusters

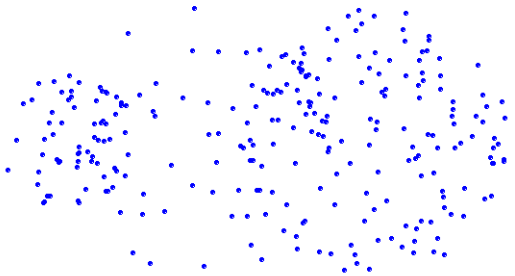


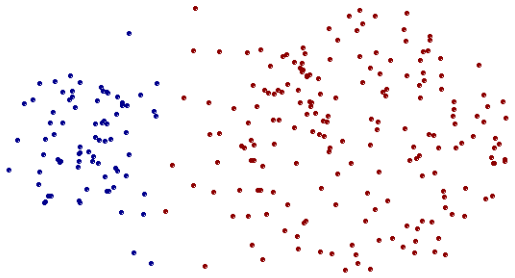
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00

Similarity matrix

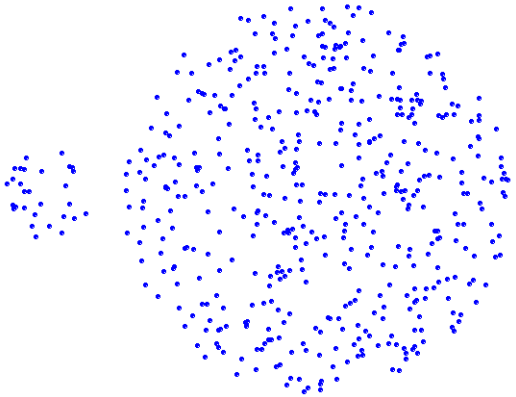


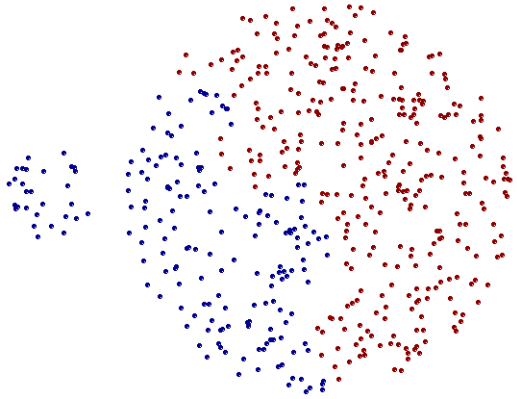
Maximum Similarity (Complete Linkage)





Strength: It is less susceptible to noise and outliers



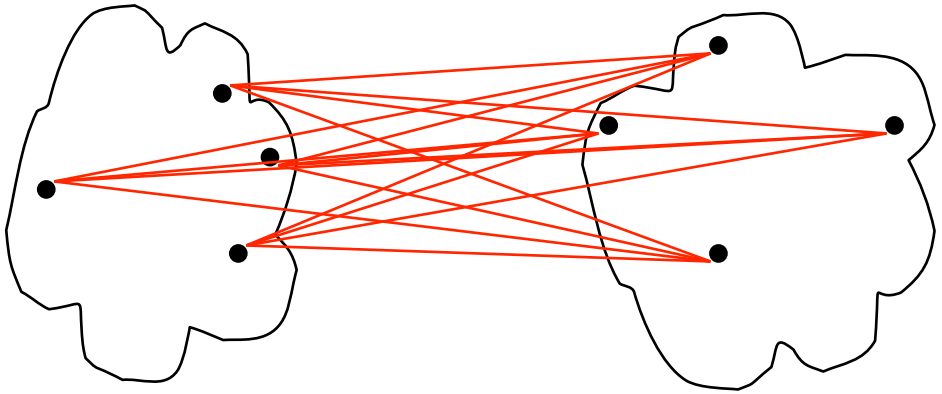


Limitation: It tends to break large clusters
Biased towards globular clusters

How to Define (Dis)similarity

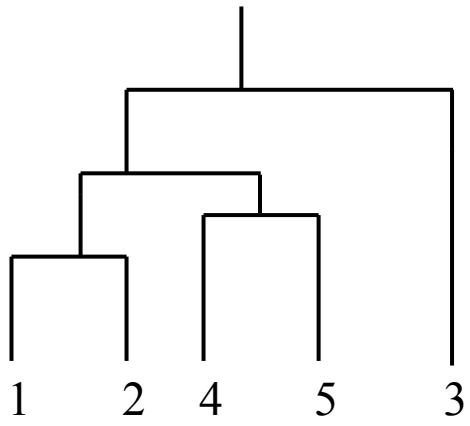
Group average

- Similarity of two clusters is the average of pairwise similarities of samples in different 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

Similarity matrix



Group Average Similarity

- Compromise between single linkage and complete linkage

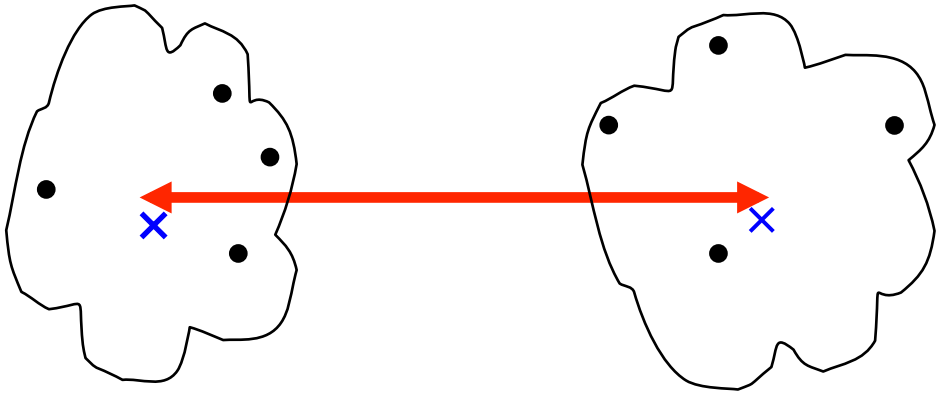
- Strength
  - It is less susceptible to noise and outliers

- Limitation
  - It is still biased towards globular cluster

How to Define (Dis)similarity

Similarity between the mean vectors

- Similarity of two clusters is the similarity of their centroids



Hierarchical Clustering

Time and space requirements

- $O(N^2)$ space since it uses the proximity matrix, where N is the number of samples

- $O(N^3)$ time in many cases
  - There are N steps and at each step the proximity matrix must be updated and searched

Hierarchical Clustering

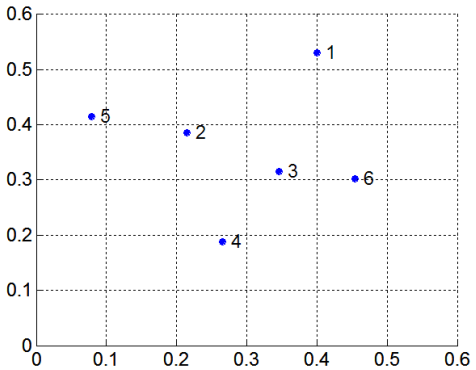
Problems and Limitations

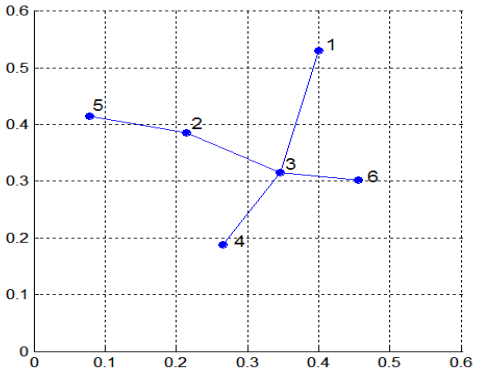
- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- Different schemes have problems with one or more of the following
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters and convex shapes
  - Breaking large clusters

Divisive Hierarchical Clustering

Can be achieved using a graph

```
compute a minimum spanning tree for the
similarity (distance) graph

repeat
        create a new cluster by breaking
        the edge corresponding to the
        smallest similarity (largest distance)

until only singleton clusters remain
```

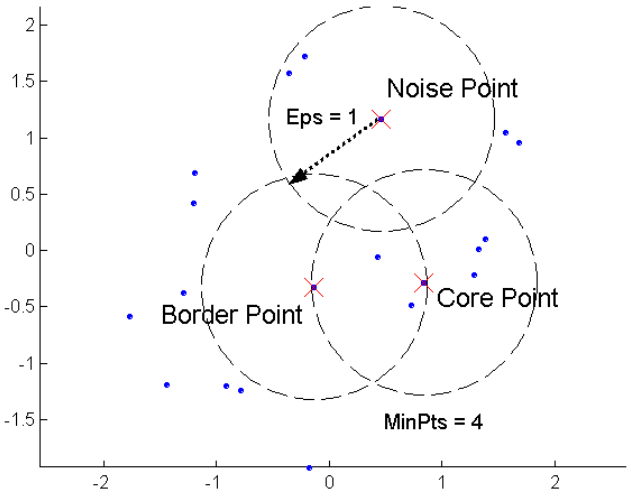




Density-Based Clustering

DBSCAN (Density-based spatial clustering of applications with noise)

- Density is defined as the number of points within a specified radius *Eps*
- A point is a core point if it has more than a specified number *MinPts* of points within *Eps*
  - These are points 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



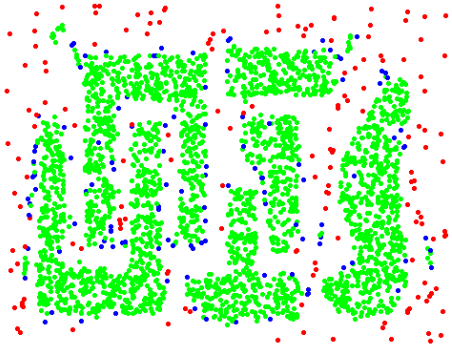
DBSCAN Algorithm

- Eliminate noise points
- Perform clustering on the remaining points

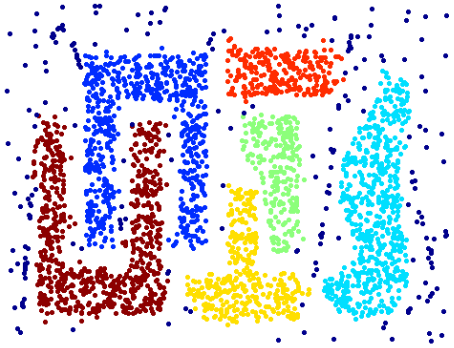
```
current_cluster_label ← 1
for all core points do
  if the core point has no cluster label then
    current_cluster_label ← current_cluster_label + 1
    Label the current core point with cluster label current_cluster_label
  end if
  for all points in the Eps-neighborhood, except i^th the point itself 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 Algorithm

Original points



Point types: core, border and noise
Eps = 10, MinPts = 4



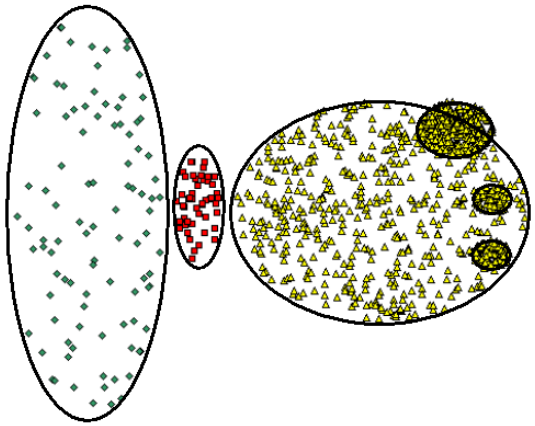
Clusters

Strength:

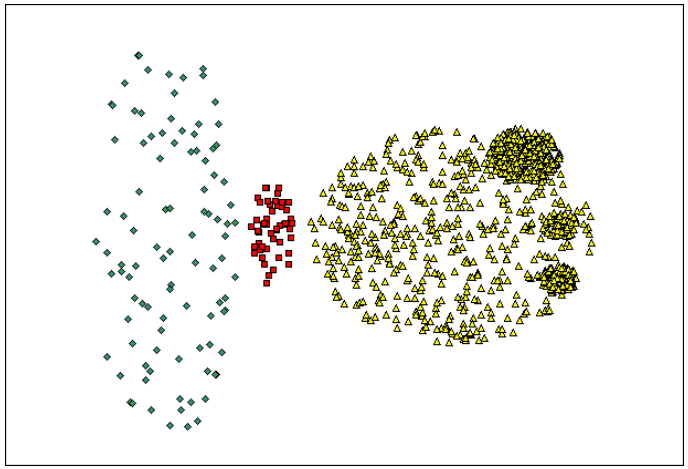
Resistant to noise

Can handle clusters of different shapes and sizes

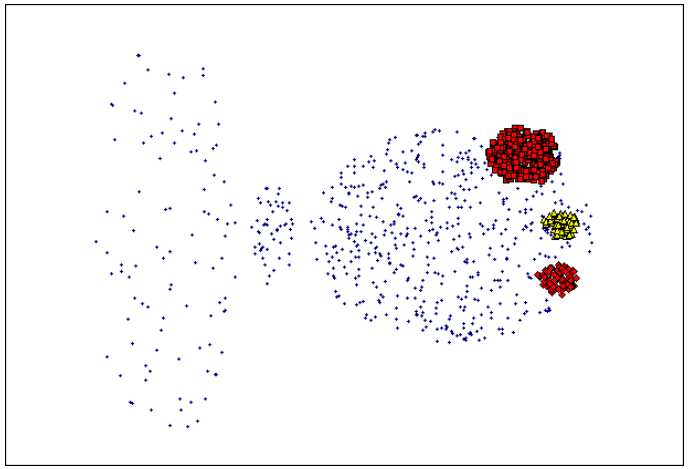
DBSCAN Algorithm



Original points



Eps = 9.75
MinPts = 4



Eps = 9.92
MinPts = 4

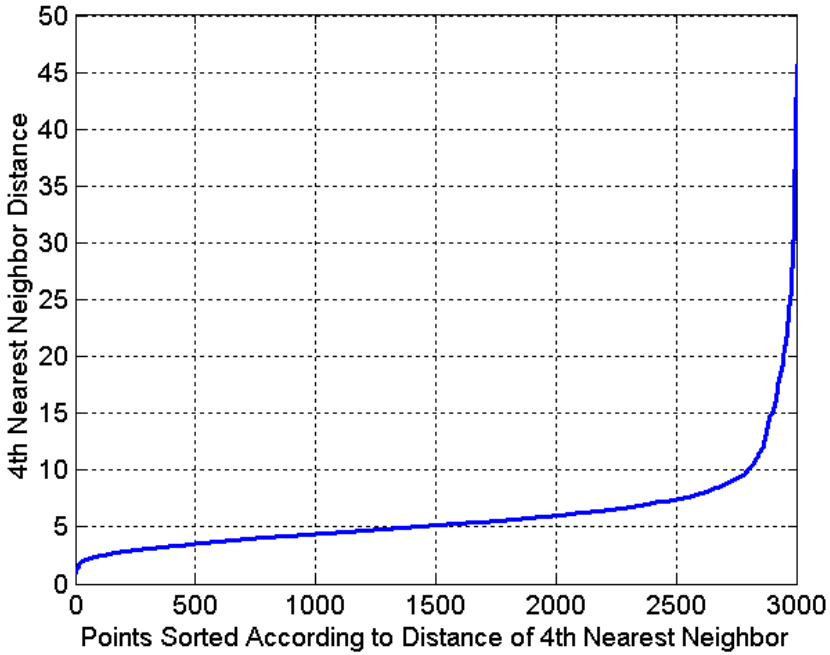
Difficulty:

When there exist clusters with differing densities

DBSCAN Algorithm

How to determine *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?
- But "clusters are in the eye of the beholder"!
- Then why do we want to evaluate them?
  - To compare clustering algorithms
  - To compare two clusters
  - To determine the cluster number

Measures of Cluster Validity

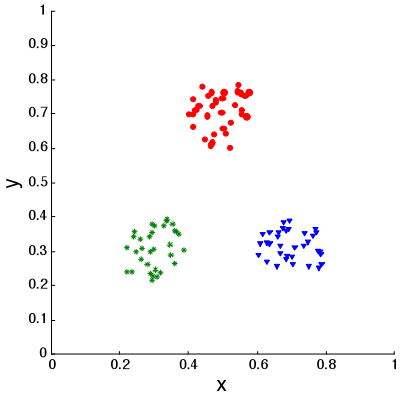
- External Measure: Used to measure the extent to which cluster labels match externally supplied class labels
  - Correlation, entropy
- Internal Measure: Used to measure the goodness of a clustering structure *without* respect to external information
  - Sum of squared error

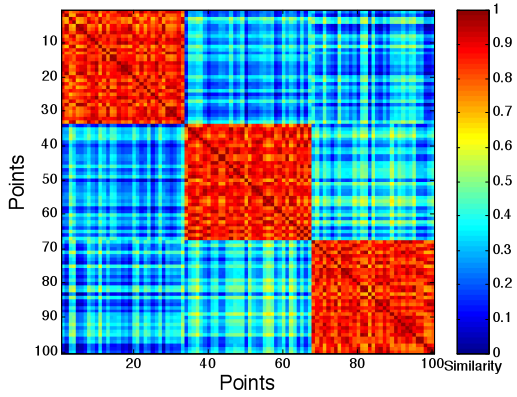
External Measure: Correlation

- Compute the correlation between the similarity (distance) matrix and the ideal version of the similarity matrix
- Ideal version
  - One row and one column for each point
  - An entry is 1 if the corresponding points belong to the same cluster (according to the externally supplied labels)
  - An entry is 0 if the corresponding points belong to different clusters
- High correlation indicates that points belonging to the same cluster are close to each other

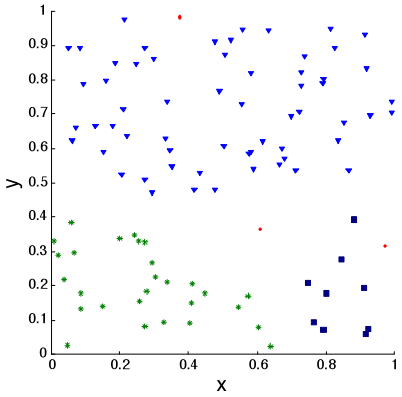
Similarity Matrix for Cluster Validation

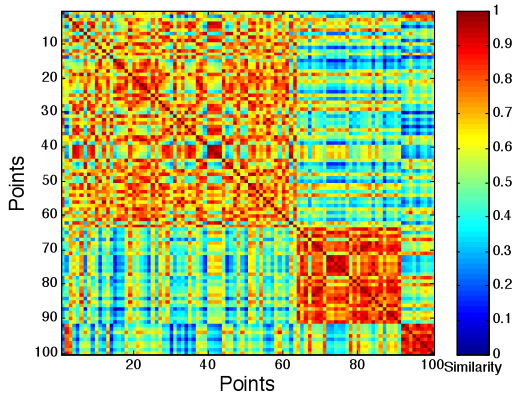
- Order the similarity matrix with respect to the externally provided cluster labels and inspect it visually





Ordered similarity matrix for well-separated clusters





Ordered similarity matrix for random data

External Measure: Entropy and Purity

Entropy

- For each cluster, calculate the distribution of the externally provided labels and calculate entropy
- Take the weighted sum to calculate the entropy of the clustering

$$E_i = -\sum_{j=1}^{m} p_{ij} \log p_{ij}$$

$$E = \sum_{i=1}^{k} n_i E_i$$

Purity

- For each cluster, the probability of the majority label
- Take the weighted sum to calculate the purity of the clustering

Table 5.9. K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

$$P_i = \max_j p_{ij}$$

$$P = \sum_{i=1}^{k} n_i P_i$$

Internal Measure

- Remember what we have seen before
  - Sum of squared error
  - Related minimum variance criterion
  - Scatter criteria
    (within cluster scatter, between cluster scatter)

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