



Unsupervised Learning

Jayanta Mukhopadhyay
Dept. of Computer Science and Engg.
IIT Kharagpur



Supervised learning

- Learning with labeled data.
 - To learn a mapping from the input to an output
 - labels provided by a supervisor.
- Classification
 - Classify digits from hand written numerals.
- Regression
 - Predict the price of a car given a set of its attributes (brand, year, mileage, engine capacity, etc.).



Unsupervised learning

- Learning from only input data.
 - No labels of instances available.
 - no supervisor to provide mapping between input and output.
- The aim is to find the regularities / structures / patterns in the input.
 - Number of clusters?
 - Any hierarchy present among them?
 - How to attribute them with semantics?

Clustering

Clustering: the task of organizing objects into groups whose members are *similar in some way*.

Cluster: a collection of objects *similar to each other*, but dissimilar to the objects belonging to other clusters .

- Regions of homogeneity in an image.
 - Segments.
- Grouping of similar components.





Class and cluster

A class: well studied group of objects identified by their common properties or characteristics.

A cluster: a group with 'loosely' defined similarity among the objects.

- Potential to form a class.



Clustering: Motivation

- finding representatives for homogeneous groups
 - to reduce data.
- discovering natural groups or categories.
 - to describe by their unknown properties.
- finding relevant groups.
 - major groups in the given context.
 - segments of an image.
- detecting unusual data objects
 - outliers.



K-means clustering

- Given N d -dimensional data points,
 - compute K partitions (clusters) in them
 - so that it minimizes the sum of square of distances between a data point and the center of its respective partition (cluster).

Optimization problem

Minimization of Sum of Squared Errors (SSE)

$$E = \sum_k \sum_{\forall x \in C_k} \|x - c_k\|^2$$

where

$$c_k = \frac{1}{|C_k|} \sum_{\forall x \in C_k} x$$



Exhaustive K-Means!

- The number of ways a set of N objects partitioned into K non-empty groups?

$$S(N, K) = \frac{1}{K!} \sum_{i=0}^K (-1)^{K-i} \binom{K}{i} i^N$$

Stirling numbers of the second kind.

$$\approx K^N / K!$$

- Checking all possible combinations prohibitive!
 - Of exponential order with input size
- An NP-hard problem ($K > 1$).

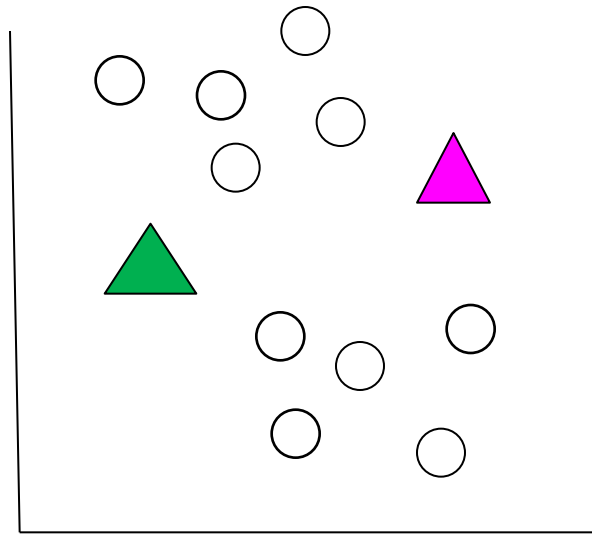


The Lloyd algorithm (1957) (Batch K-Means)

- Given K initial centers, assign a point to the cluster represented by its center, if it is the closest among them.
- Update the centers.
- Iterate above two steps, till the centers do not change their positions.



K-means: example ($k=2$)

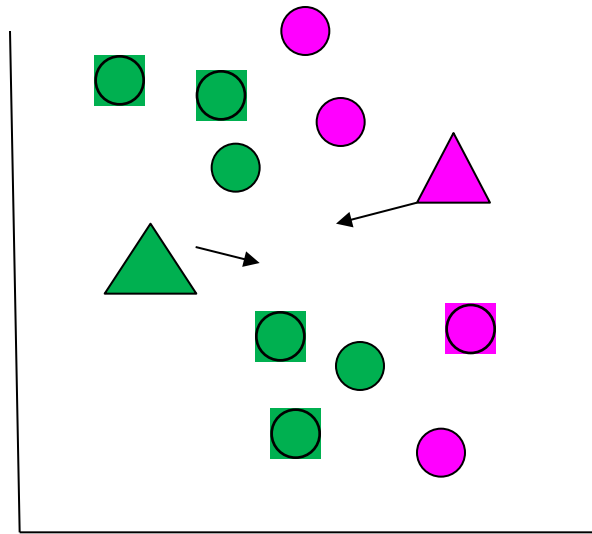


Choose initial centers.

Compute partitions.



K-means: example ($k=2$)

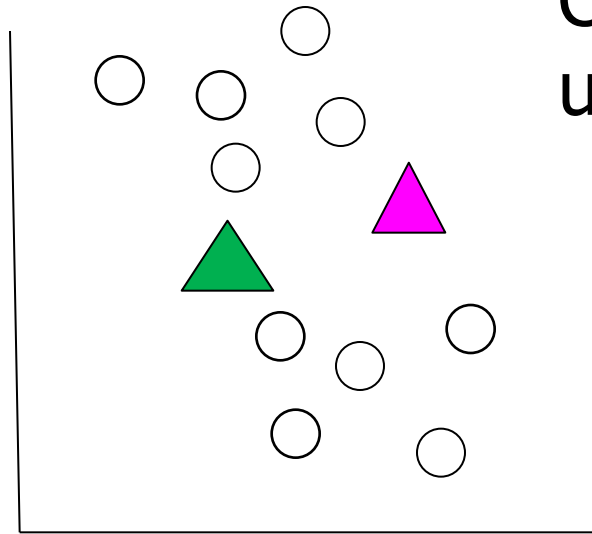


Compute partitions.

Update centers.



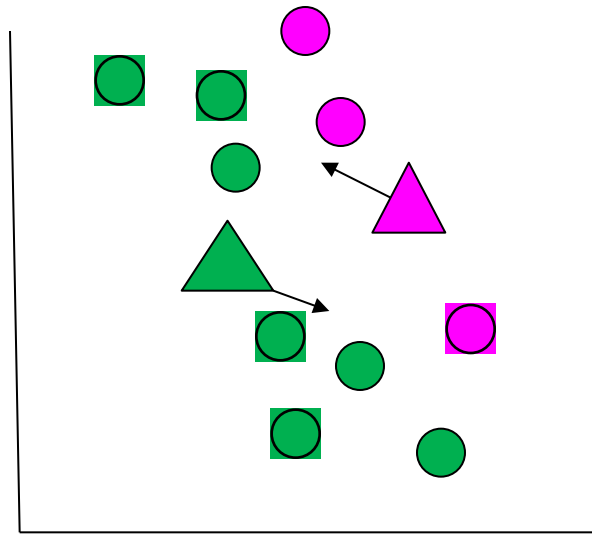
K-means: example ($k=2$)



Compute new partitions with updated centers.



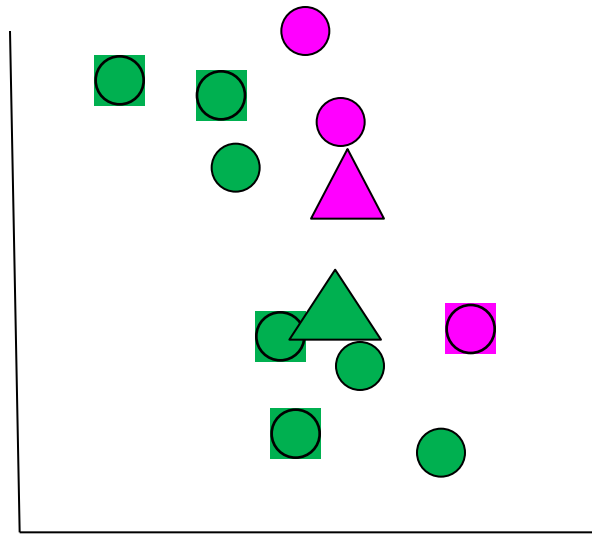
K-means: example ($k=2$)



Update centers.

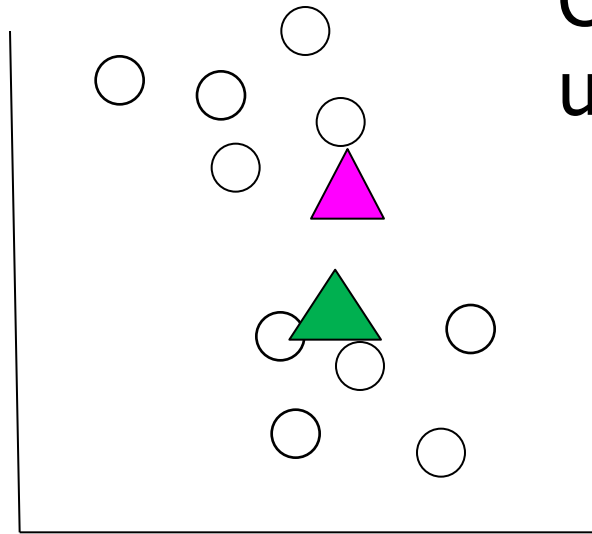


K-means: example (k=2)





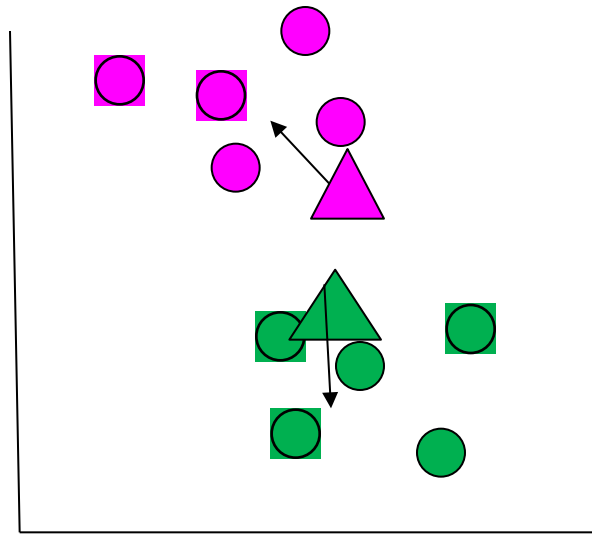
K-means: example ($k=2$)



Compute new partitions with updated centers.



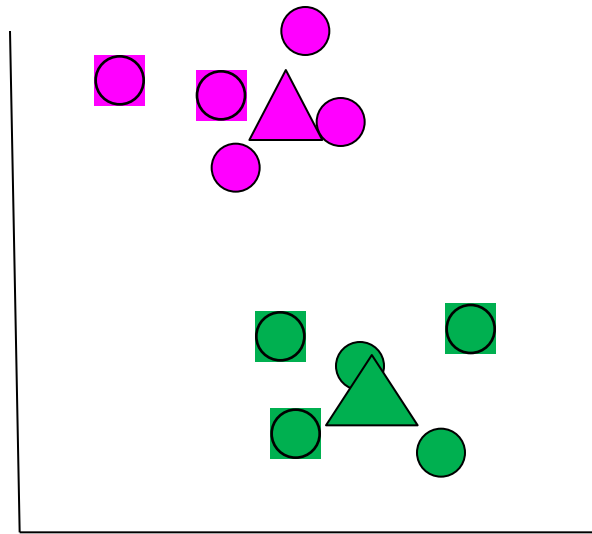
K-means: example ($k=2$)



Update centers.



K-means: example ($k=2$)



Stop at no change
(or a very little
change in cluster
centers).



A more conservative approach

- Lloyd algorithm fast but not necessarily causing better convergence.
- A more conservative approach to move one data point at a time provided overall cost gets reduced.
- A greedy approach by choosing the transfer of a data point from a class (say, i) to another class (say, j), which causes the best (maximal) cost reduction at that step.



Strength

- Trying to minimize the energy function SSE defined by the sum of divergences of each cluster from its center.

$$E = \sum_k \sum_{\forall x \in C_k} \|x - c_k\|^2$$

- Convergence guaranteed at a quadratic rate.
- Linear time complexity in N , d and K .
- Versatile, simple, and invariant to data ordering.



Weakness

- Only detects well separated, compact, hyperspherical clusters.
 - Value of K ?
 - Sensitive to noise and outlier points
 - Due to squared Euclidean distance.
 - May get stuck at local minima.
 - Highly sensitive to the selection of the initial centers
 - Improper initialization
 - empty clusters,
 - slower convergence, and
 - a higher chance of getting stuck in bad local minima
- Use of an adaptive initialization method!



Various Initialization Approaches

- Each point randomly to one of the clusters (Forgy 1965)
- First K points as the centers. (McQueen 1967)
 - Sensitive to data ordering
 - Choose them randomly
 - Outliers still may get selected.
- Repeated K-means. (Bradley & Fayyad, ICML'98)
 - K-means on J random subsets.
 - Merge all centers and run K-means repeatedly on them
 - Choose the best set of centers minimizing the error, and use them for iterative convergence.



K-means++

- The first center c_1 chosen randomly.
- The i -th ($i=2,3,\dots,K$) center c_i chosen as x' with a probability proportional to square of the minimum distance from the selected $i-1$ centers.

$$p(x') = \frac{\min_{j=1,2,\dots,i-1} \|x' - c_j\|^2}{\sum_x \min_{j=1,2,\dots,i-1} \|x - c_j\|^2}$$



How do you determine K?

- Use of cluster validity index.
 - Maximize or minimize depending upon the nature of metric.
- Check for stable clustering results with random initialization.
 - Use of different measures of stability.



Cluster validity indices

- External indices using a reference partitioning information, e.g. class labels.

$$NMI = 2I(Y;C)/(H(Y)+H(C))$$

- Normalized Mutual Information (NMI)
- Fraction of same pairs in same clusters (FM index)
- Set matching measures

Y: Cluster Label

C: Class Label

$$I(Y;C) = H(Y) - H(Y|C)$$

- Finding matching partition pairs and maximal common coverage

- Internal indices by Looking at variance distribution, structure of clusters

$$s(x) = (a(x) - b(x)) / \max(a(x), b(x))$$

Avg. of $s(x)$'s.

$a(x)$: Avg. dist. of points within the cluster from x

$b(x)$: Min. avg. dist. of points of other clusters from x .

- Silhouette index

- Higher better in $[-1, 1]$

- Calinski-Harabasz (CH) Index

$$CH(K) = \frac{(J(1) - J(K)) / (K - 1)}{J(K) / (n - K)}$$

$J(i)$: SSE with $K=i$

- Higher better.



Stability check based clustering

- Repeated clustering should have similar partitioning
 - for an appropriate K .
- Wang's method of cross-validation (2010)
 - Permute the input data c times.
 - Each time divide into three parts,
 - S_1, S_2 , and S_3 , such that $|S_1|=|S_2|=m$
 - Perform k-means on S_1 and S_2 .
 - Test on S_3 both cases to find the cluster numbers.
 - Compute the number of disagreement
 - a pair being in the same or different clusters.
 - Take the average over c observations

Choose K
minimizing avg.
number of
disagreements.

Generalizing K-Means: Mixture densities

Number of components

$$P(\mathbf{x}) = \sum_{i=1}^K P(\mathbf{x}|G_i)P(G_i)$$

Component density

Component proportion

- G_i defines the i th group or cluster.
- K is a hyper-parameter and should be known.
- For multivariate Gaussian distribution:
 - $P(\mathbf{x}|G_i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$
- To estimate $\boldsymbol{\mu}_i$, Σ_i and $P(G_i)$ for all i . from the set of *iid.* input samples: $X = \{\mathbf{x}^t\}$, $t = 1, 2, \dots, N$



Mixture of Gaussians

- Each cluster center is augmented by a covariance matrix, whose values are re-estimated from corresponding samples.
 - Mahalanobis distance function:

$$d(x, \mu_k; \Sigma_k) = (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)$$

Cluster center

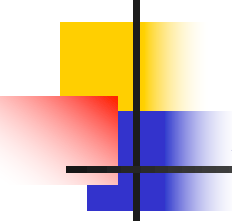
Covariance matrix

Technique could be refined by computing probabilities of belongingness to a cluster.

Parametric PDF:

$$p(x|\{\pi_k, \mu_k, \Sigma_k\}) = \sum_k \pi_k N(x|\mu_k, \Sigma_k)$$

Mixing coefficients



Expectation Maximization (EM) Algorithm

$$z_{ik} = \frac{1}{Z_i} \pi_k N(x_i | \mu_k, \Sigma_k)$$

Normalizing
factor

$$Z_i = \sum_k \pi_k N(x_i | \mu_k, \Sigma_k)$$

- Start with initial set : $\{\pi_k, \mu_k, \Sigma_k\}$.
- E-Step (Expectation stage)
 - Compute **probability** (z_{ik}) of x belonging to k th Gaussian cluster.
 - Assign x to the m th cluster whose **probability** is maximum.
- M-Step (Maximization Stage)
 - Re-estimate parameters ($\{\pi_k, \mu_k, \Sigma_k\}$) from class distribution
- Iterate above two steps till it converges.

Optional step.
Decision to be
taken at the end. →



Parameter re-estimation

$$z_{ik} = \frac{1}{Z_i} \pi_k N(x | \mu_k, \Sigma_k)$$

← Normalizing factor

$$N_k = \sum_i z_{ik}$$

Expected number of pixels in class k .

$$\mu_k = \frac{1}{N_k} \sum_i z_{ik} x_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_i z_{ik} (x_i - \mu_k)(x_i - \mu_k)^T$$

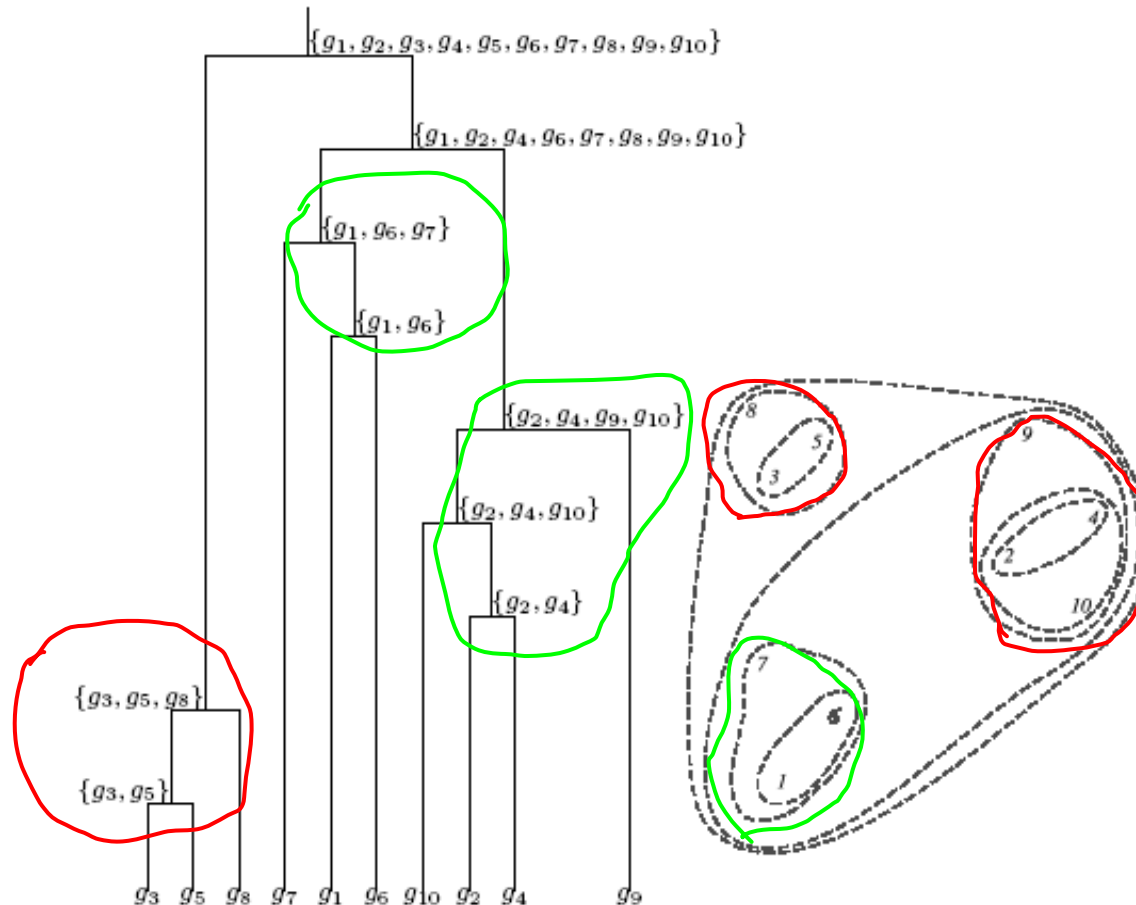
$$\pi_k = \frac{N_k}{N}$$



Hierarchical clustering

- Builds hierarchy of groups.
 - Usually a bottom-up approach.
- Uses a distance matrix among the samples.
- Explicit feature representation may not be required.
- A non-probabilistic approach.

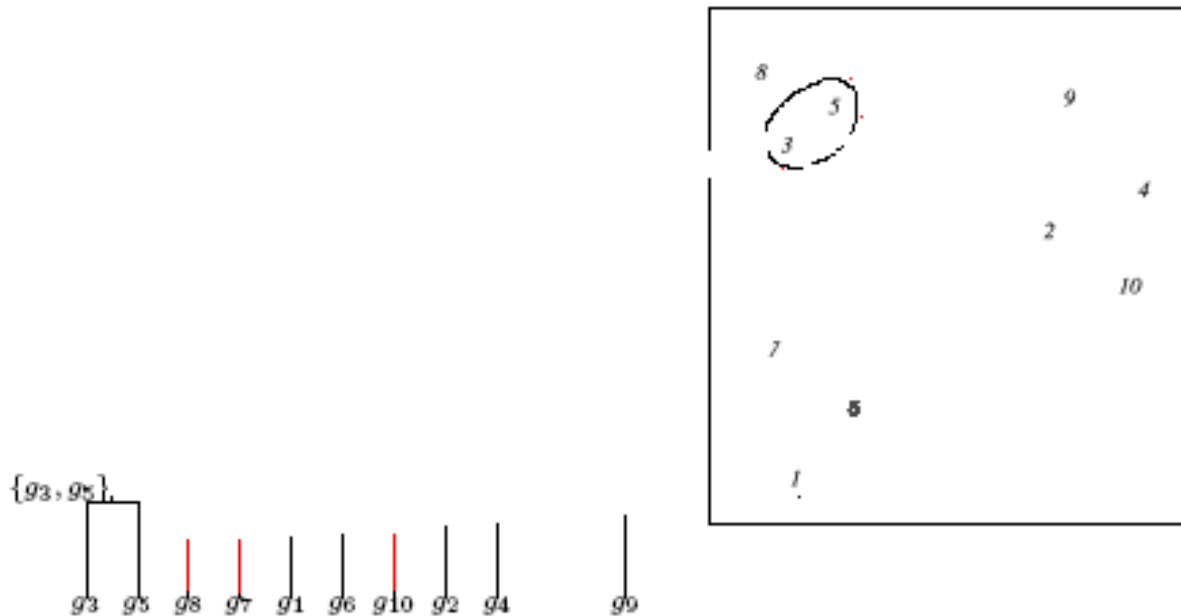
Hierarchical Clustering: An example



Courtesy: www.bioalgorithms.info (Pavel Pevzner)

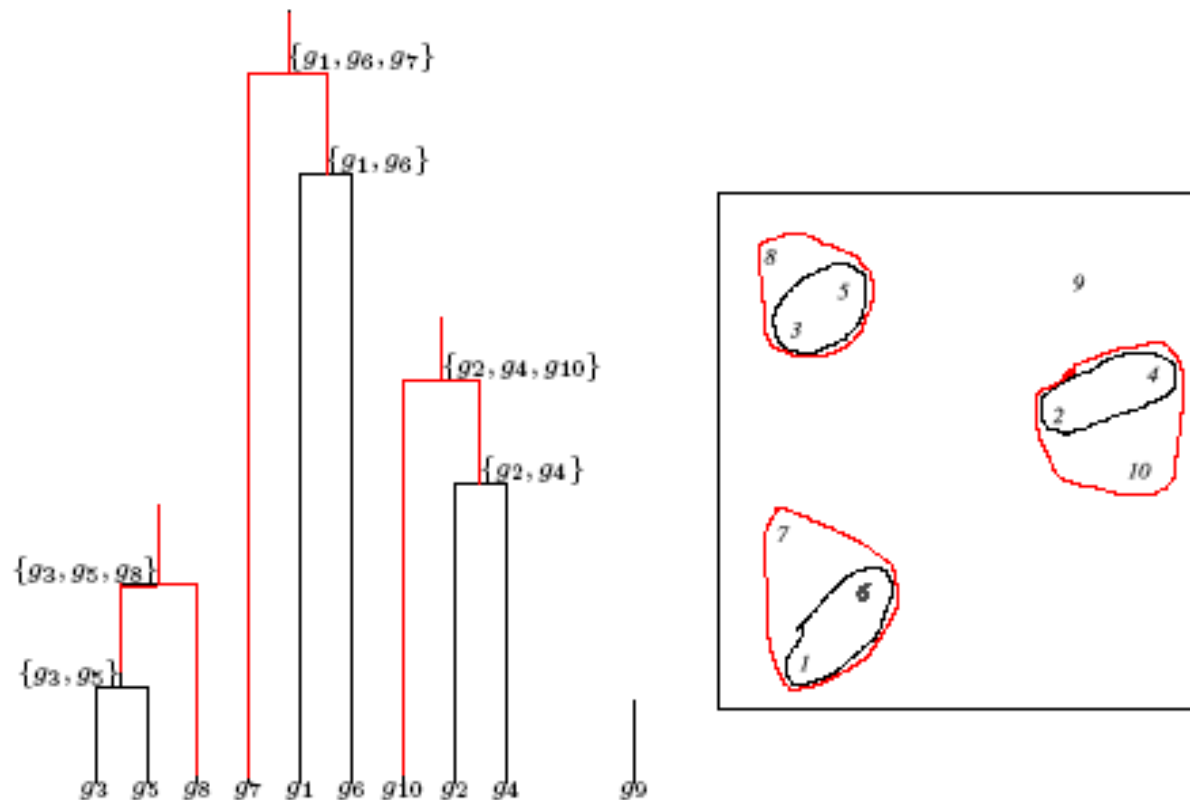
Courtesy: www.bioalgorithms.info (Pavel Pevzner)

Hierarchical Clustering: Example



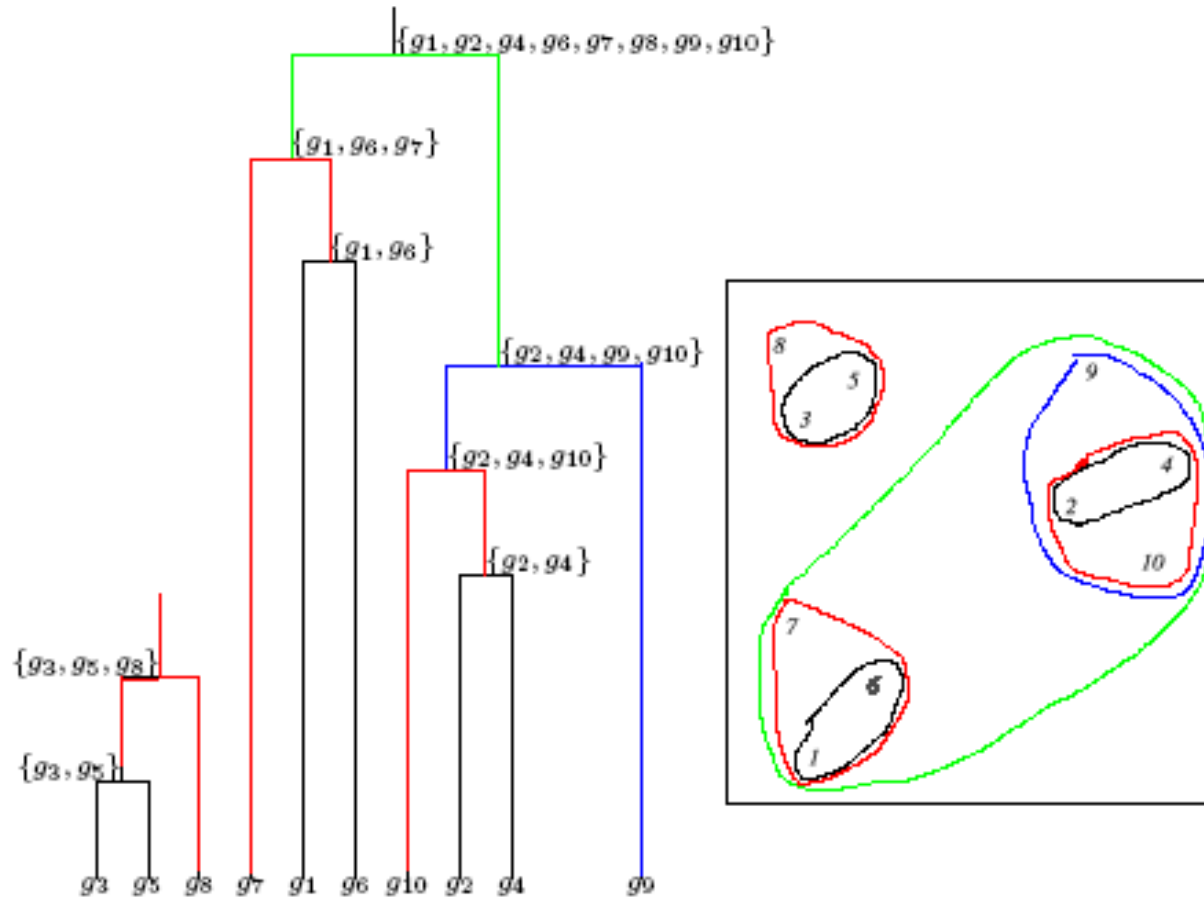
Courtesy: www.bioalgorithms.info (Pavel Pevzner)

Hierarchical Clustering: Example



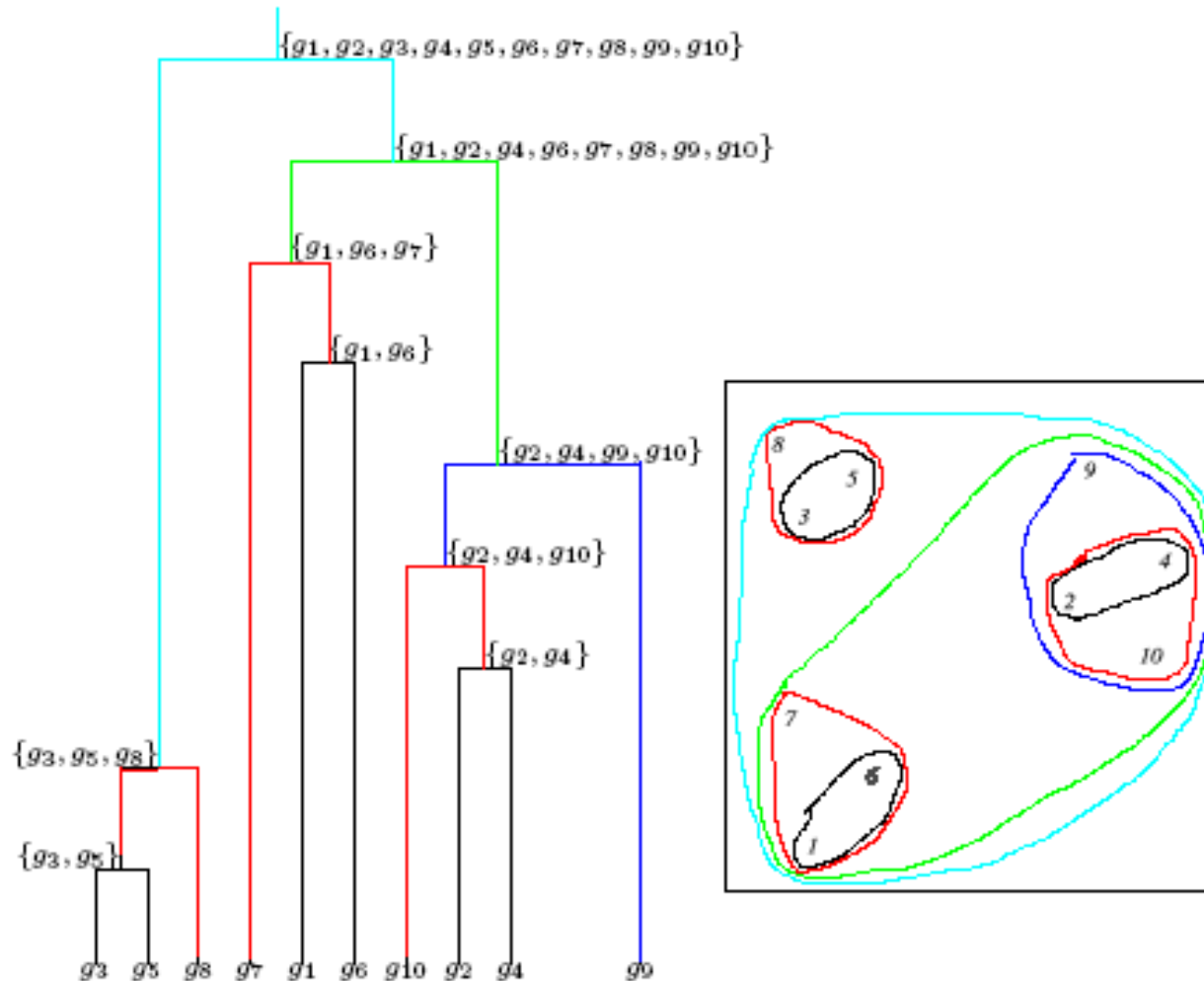
Courtesy: www.bioalgorithms.info (Pavel Pevzner)

Hierarchical Clustering: Example



Courtesy: www.bioalgorithms.info (Pavel Pevzner)

Hierarchical Clustering:



Courtesy: www.bioalgorithms.info (Pavel Pevzner)



Hierarchical Clustering Algorithm

The algorithm takes a $n \times n$ distance matrix d of pairwise distances between points as an input.

Hierarchical Clustering (d, n)

Form n clusters each with one element.

Initialize a graph T with a vertex for each cluster.

while there is more than one cluster

Find the two closest clusters C_1 and C_2 .

Merge C_1 and C_2 into C with $|C_1| + |C_2|$ elements.

Compute distance from C to all other clusters.

Add a new vertex C to T and connect to vertices C_1 and C_2 .

Remove rows and columns of d corresponding to C_1 and C_2 .

Add a row and column to d corresponding to the new cluster C .

return T

Different ways to define distances between clusters may lead to different clustering.



Computing distance between a pair of clusters.

- $d_{min}(C, C^*) = \min d(x, y)$
*for all elements x in C and y in C^**
- Distance between two clusters is the **smallest** distance between any pair of their elements.
- $d_{avg}(C, C^*) = (1 / (|C^*||C|)) \sum d(x, y)$
*for all elements x in C and y in C^**
- Distance between two clusters is the **average** distance between all pairs of their elements.

Courtesy: www.bioalgorithms.info (Pavel Pevzner)

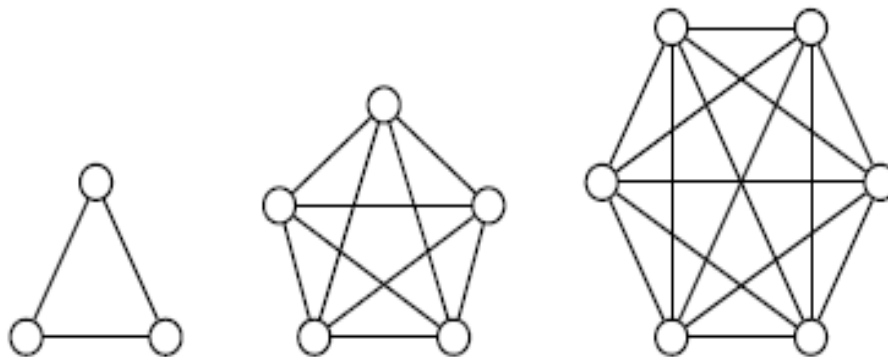


Graph based approaches

- Form a graph from the input data.
 - May not be explicit.
- Compute cliques, connected components, etc.

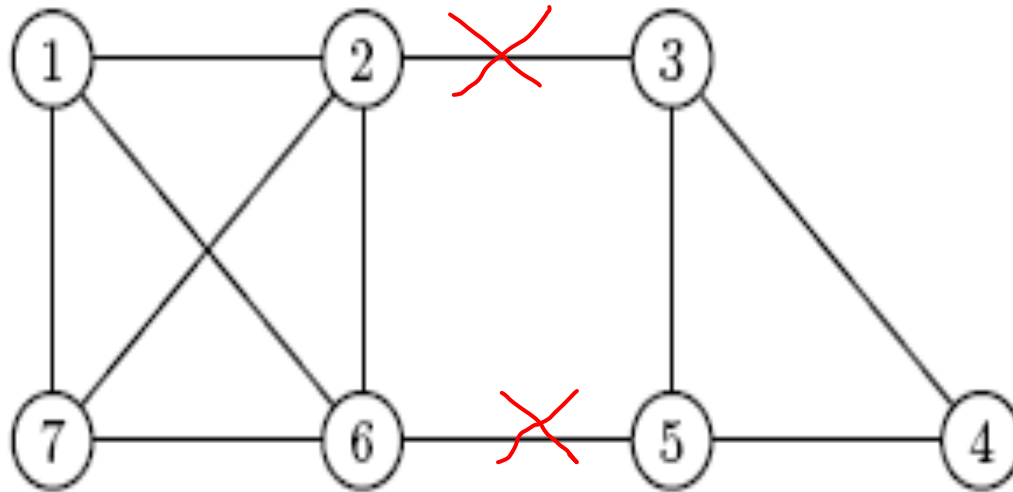
Clique Graphs

- A **clique** is a graph with every vertex connected to every other vertex.
- A **clique graph** is a graph where each connected component is a clique.



Transforming an Arbitrary Graph into a Clique Graphs

- A graph can be transformed into a clique graph by adding or removing edges.





Corrupted Cliques Problem

Input: A graph G

Output: The smallest number of additions and removals of edges that will transform G into a clique graph.



Distance Graphs

- Feature vectors represented as vertices in the graph.
- Choose a distance threshold θ .
- If the distance between two vertices is below θ , draw an edge between them.
- The resulting graph may contain cliques.
- These cliques represent clusters of closely located data points!

Courtesy: www.bioalgorithms.info (Pavel Pevzner)

Transforming into Clique Graph

The distance graph (threshold $\theta=7$) is transformed into a clique graph after removing the two highlighted edges

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}
g_1	0.0	8.1	9.2	7.7	9.3	2.3	5.1	10.2	6.1	7.0
g_2	8.1	0.0	12.0	0.9	12.0	9.5	10.1	12.8	2.0	1.0
g_3	9.2	12.0	0.0	11.2	0.7	11.1	8.1	1.1	10.5	11.5
g_4	7.7	0.9	11.2	0.0	11.2	9.2	9.5	12.0	1.6	1.1
g_5	9.3	12.0	0.7	11.2	0.0	11.2	8.5	1.0	10.6	11.6
g_6	2.3	9.5	11.1	9.2	11.2	0.0	5.6	12.1	7.7	8.5
g_7	5.1	10.1	8.1	9.5	8.5	5.6	0.0	9.1	8.3	9.3
g_8	10.2	12.8	1.1	12.0	1.0	12.1	9.1	0.0	11.4	12.4
g_9	6.1	2.0	10.5	1.6	10.6	7.7	8.3	11.4	0.0	1.1
g_{10}	7.0	1.0	11.5	1.1	11.6	8.5	9.3	12.4	1.1	0.0

(a) Distance matrix, d (distances shorter than 7 are shown in bold).

After transforming the distance graph into the clique graph, the dataset is partitioned into three clusters

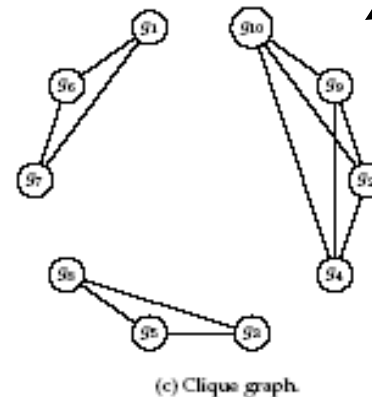
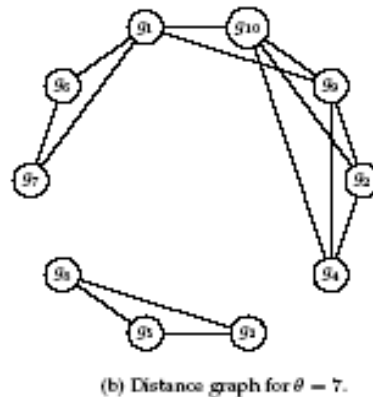


Figure 10.6 The distance graph (b) for $\theta = 7$ is not quite a clique graph. However, it can be transformed into a clique graph (c) by removing edges (g_1, g_{10}) and (g_1, g_9) .



Corrupted Clique Problem

- Corrupted Cliques problem is NP-Hard, some heuristics exist to approximately solve it:
- **Two approximate methods:**
 1. **Parallel Classification with Cores (PCC).**
 - (Amir Ben-Dor et. al (1999))
 2. **Cluster Affinity Search Technique (CAST)**

Courtesy: www.bioalgorithms.info (Pavel Pevzner)



Parallel Classification with cores (PCC)

- Suppose S' is a subset of S .
- Let, $\{C_1, C_2, \dots, C_k\}$ be a clustering on S' .
- How do you extend the clustering to S ?
- Let $j \in S - S'$ and $N(j, C_i)$ be no. of edges from j to C_i .
- $Affinity(j, C_i) = N(j, C_i) / |C_i|$
- Assign j to the cluster which has maximum affinity.

Algorithm for PCC

Algorithm PCC(S, G, k)

S : Set of n elements (say, feature vectors forming vertices of G).

G : Distance graph , k : No. of clusters

1. Randomly select S' , a subset from S , and S'' , a subset from $S-S'$,

s.t. $|S'| = \log(\log(n))$ and $|S''| = \log(n)$.

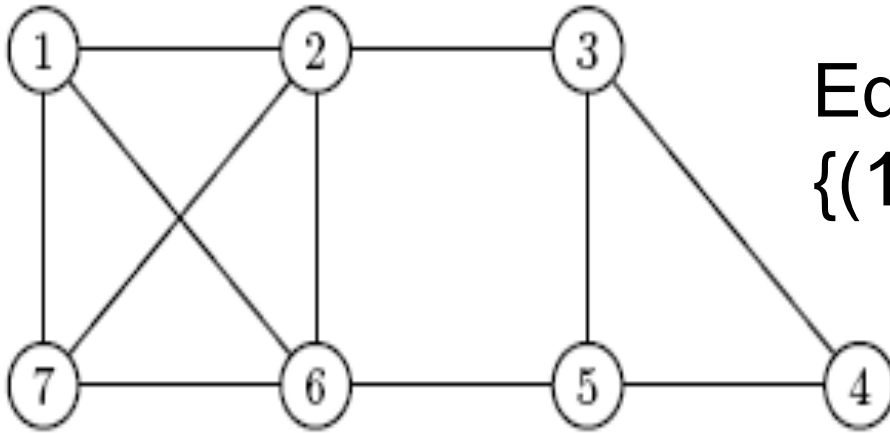
2. For all k partitions in S'

2.1 Obtain extended partition in S through two stages of extensions
i.e. $S' \rightarrow S'' \rightarrow (S - (S' \cup S''))$

2.2 Choose the one which has minimum score , i.e. the no. of edges reqd. to add or remove from G to get a Clique graph as per the partition.

End PCC.

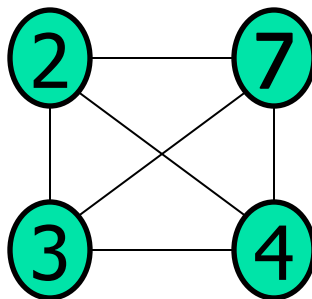
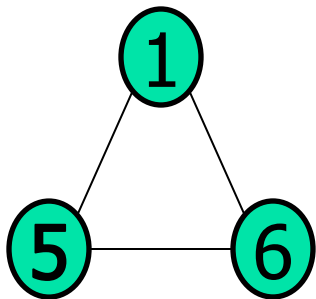
Example of computing score



Edges to be added in G:
 $\{(1,5), (2,4), (3,7), (4,7)\}$

Edges to be deleted in G:
 $\{(1,2), (1,7), (6,7), (4,5), (2,6), (3,5)\}$

Let a partition P with $k=2$:
 $\{1,5,6\}, \{2,3,7,4\}$.



Score: 10

Clique graph w.r.t . P



PCC: Time complexity

- No. of partitions in $S' = k^{|S'|}$
$$= k^{\log(\log(n))}$$
$$= (\log(n))^{\log_2(k)}$$
- In each iteration $O(n^2)$ operations for extension and score computation.
- Total time complexity: $O(n^2 (\log(n))^{\log_2(k)})$



CAST

- **CAST** (Cluster Affinity Search Technique): a practical and fast algorithm:
 - **CAST** is based on the notion of features *close* to cluster C or *distant* from cluster C .
 - Distance between feature i and cluster C :
 $d(i, C)$ = average distance between feature i and all other features in C

Gene i is *close* to cluster C , if $d(i, C) < \theta$
and *distant* otherwise.

Courtesy: www.bioalgorithms.info (Pavel Pevzner)

CAST(S, G, θ)

***S – set of elements, G – distance graph,
 θ – distance threshold***

$$P \leftarrow \emptyset$$

while $S \neq \emptyset$

$V \leftarrow$ vertex of maximal degree in the distance graph G .

$$C \leftarrow \{v\}$$

while a **close** feature i *not in* C or **distant** feature i *in* C exists

$$\{$$

Find the nearest close feature i not in C and add it to C .

Remove the farthest distant feature i in C .

}

Add cluster C to partition P .

$$S \leftarrow S \setminus C$$

Remove vertices of cluster C from the distance graph G .

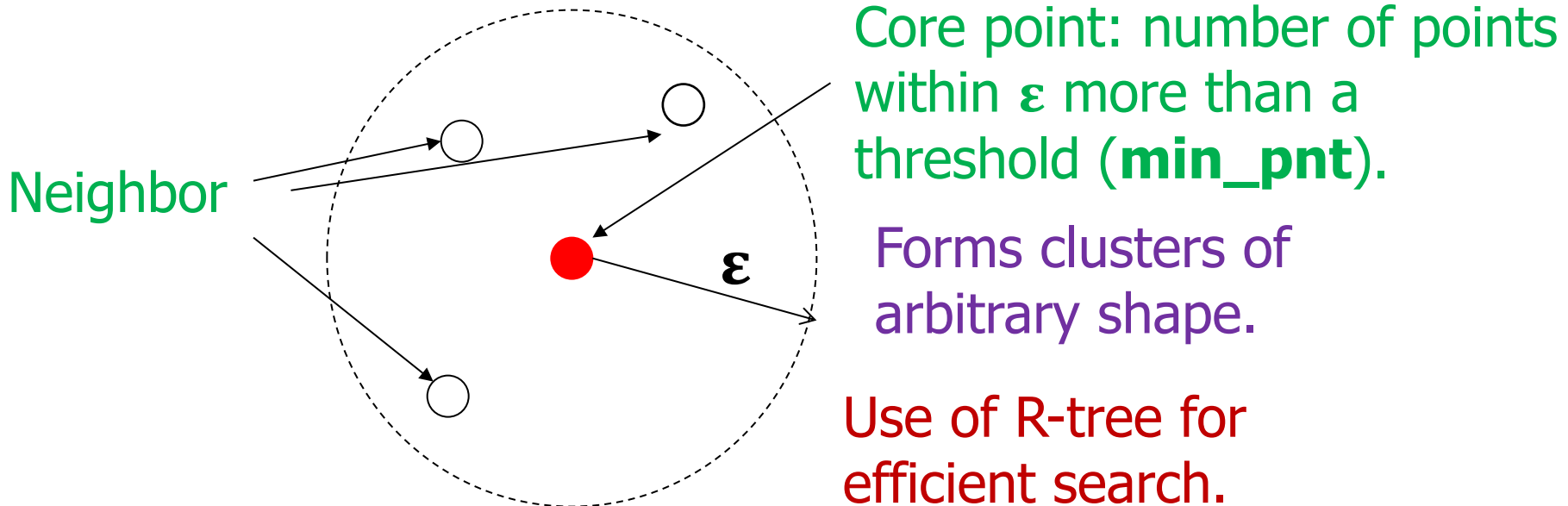
return P

Courtesy: www.bioalgorithms.info (Pavel Pevzner)

DBSCAN

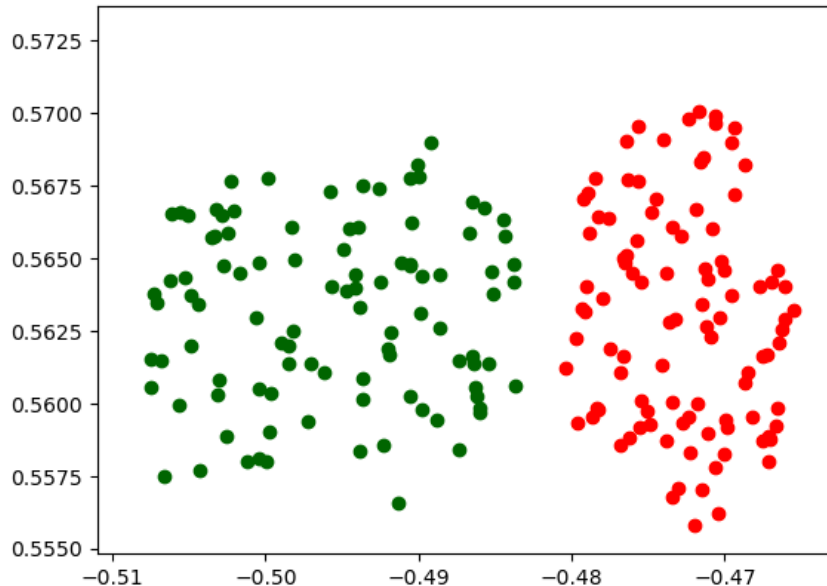
Density-based spatial clustering of applications with noise (DBSCAN)
(Ester, Kriegel, Sander and Xu'96)

- No explicit computation of distance graph.

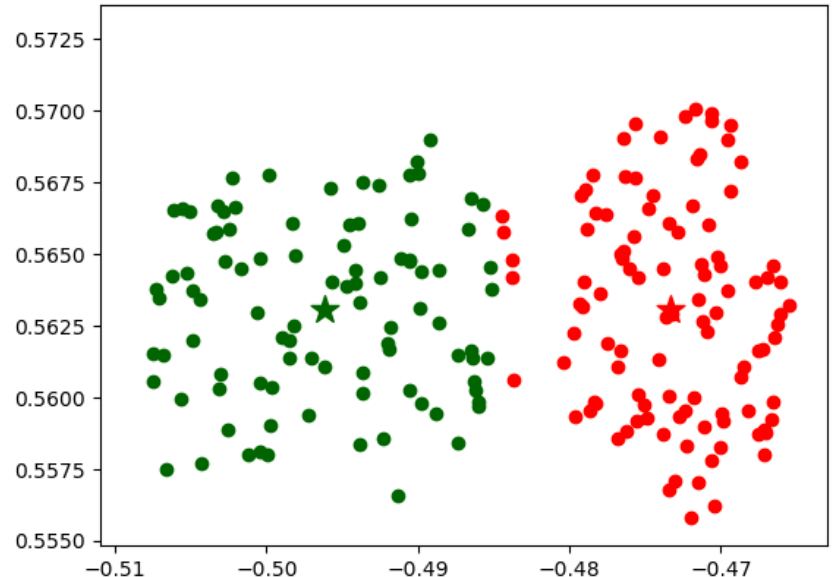


- Grow regions of connected core points from a seed.
 - A neighbor but not a core point called a **border point**.
 - A point not reachable from any core point is **noise / outlier**.

DBSCAN and K-means: A typical example



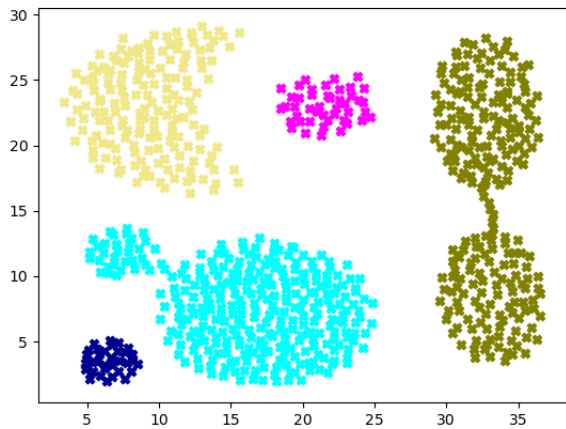
DBSCAN



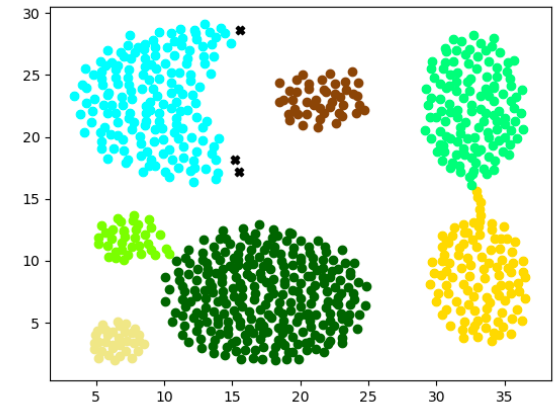
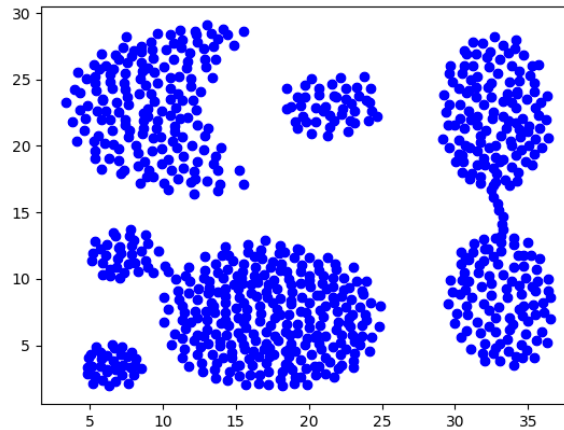
K-Means

Courtesy: Jayasree Saha

DBSCAN: Effect of varying parameters (An example)



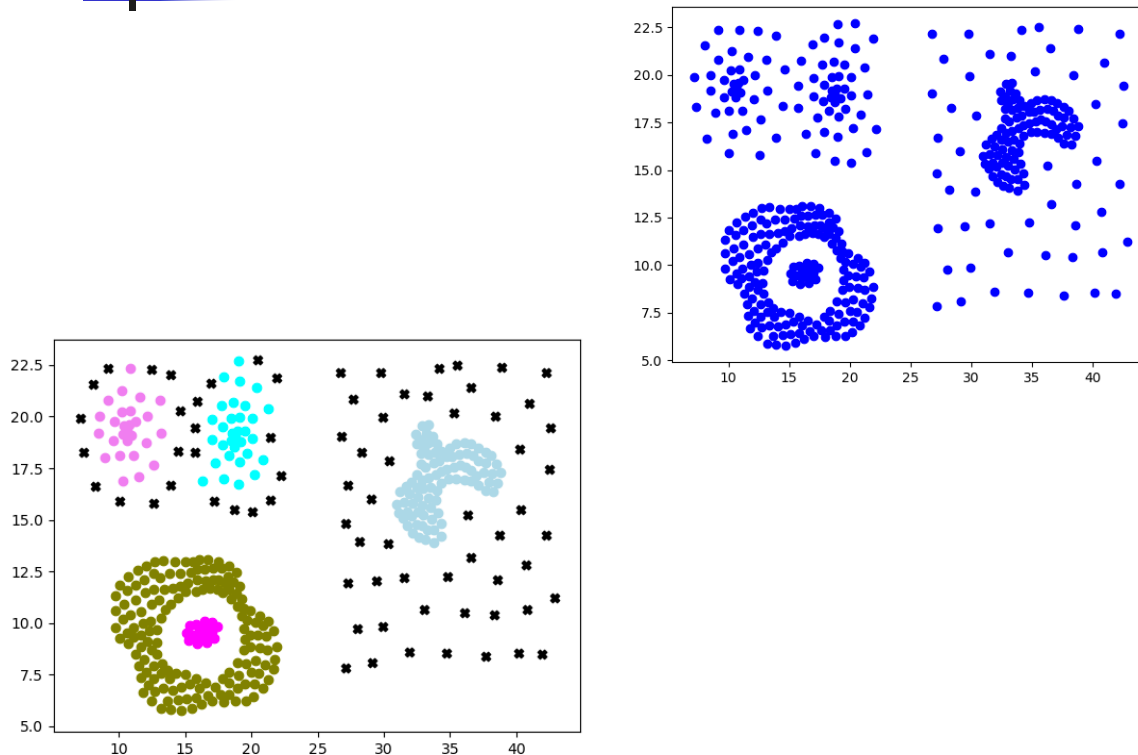
$\epsilon=2, \text{min_pts}=5$



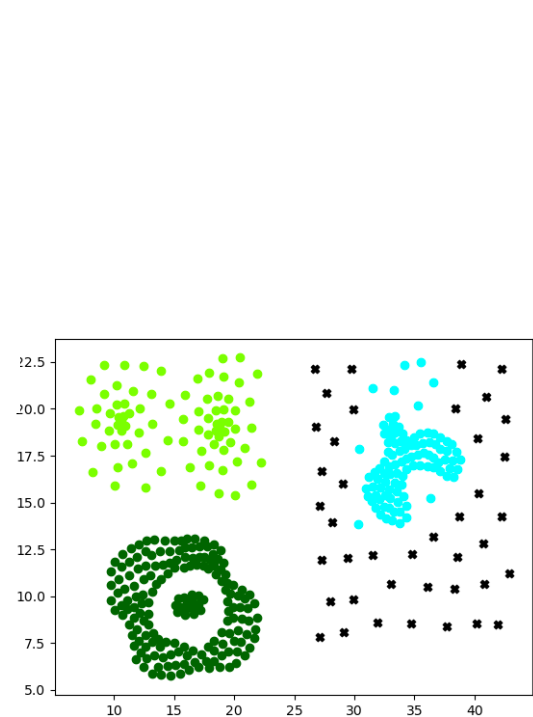
$\epsilon=2, \text{min_pts}=15$

Courtesy: Jayasree Saha

DBSCAN: Effect of varying parameters (Another example)



$\epsilon = 1.3, \text{min_pts} = 5$



$\epsilon = 1.8, \text{min_pts} = 3$

Courtesy: Jayasree Saha



Summary

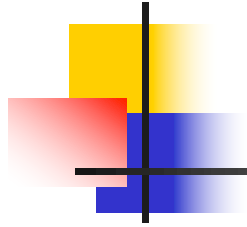
- Clustering techniques
 - Semi-parametric approaches.
 - K-means algorithm
 - No explicit parameter estimation
 - Expectation Maximization
 - Mixture of Gaussian
 - Hierarchical clustering method
 - Builds a tree hierarchy following a bottom-up approach.
 - Uses distance matrix instead of explicit feature representation.



Summary

- Graph based approaches.

- A clique: A graph with every vertex connected to every other vertex.
- A clique graph: Each component is a clique.
- Distance graphs: Feature Vectors represented as vertices and edges if distance less than a threshold.
 - Corrupted clique problem
 - Smallest number of additions and removals of edges to transform a graph into a clique graph.
 - PCC and CAST
- Without explicit computation of distance graph
 - DBSCAN



Thank you!