

Introduction to Data Mining

Chapter 11. Cluster Analysis: Advanced Methods
Jiawei Han, Computer Science, Univ. Illinois at
Urbana-Champaign, 2017

1

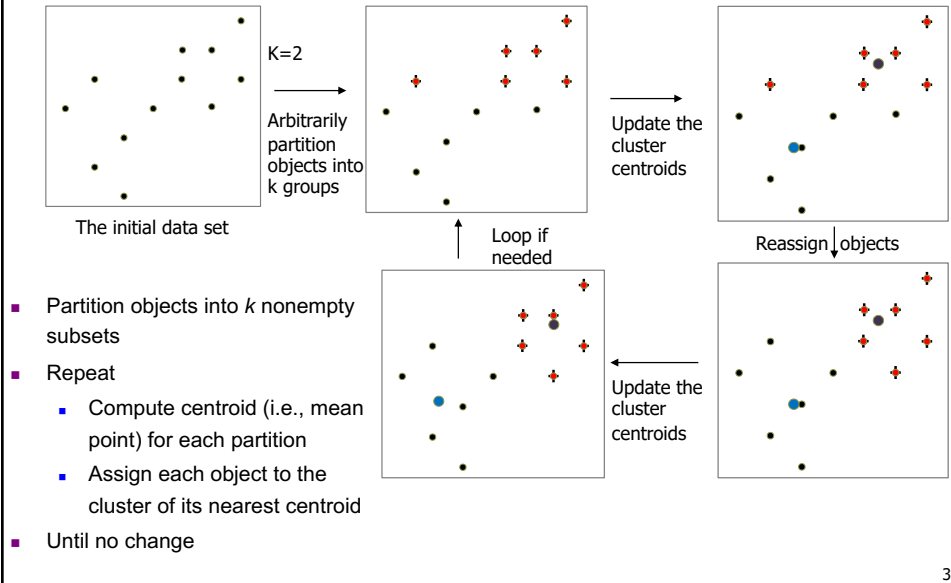
Review: Basic Cluster Analysis Methods (Chap. 10)

- Cluster Analysis: Basic Concepts
 - Group data so that object similarity is high within clusters but low across clusters
- Partitioning Methods
 - K-means and k-medoids algorithms and their refinements
- Hierarchical Methods
 - Agglomerative and divisive method
- Density-Based Methods
 - DBScan
- Grid-Based Methods
 - STING
- Evaluation of Clustering
 - Assess clustering tendency, determine # of clusters, and measure clustering quality

2

2

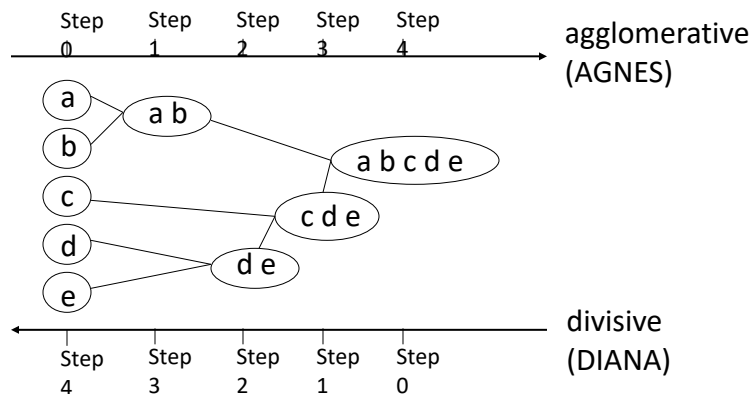
K-Means Clustering



3

Hierarchical Clustering

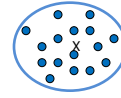
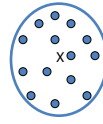
- Use distance matrix as clustering criteria. This method does not require the number of clusters k as an input, but needs a termination condition



4

4

Distance between Clusters



- **Single link:** smallest distance between an element in one cluster and an element in the other, i.e., $\text{dist}(K_i, K_j) = \min(t_{ip}, t_{jq})$
- **Complete link:** largest distance between an element in one cluster and an element in the other, i.e., $\text{dist}(K_i, K_j) = \max(t_{ip}, t_{jq})$
- **Average:** avg distance between an element in one cluster and an element in the other, i.e., $\text{dist}(K_i, K_j) = \text{avg}(t_{ip}, t_{jq})$
- **Centroid:** distance between the centroids of two clusters, i.e., $\text{dist}(K_i, K_j) = \text{dist}(C_i, C_j)$
- **Medoid:** distance between the medoids of two clusters, i.e., $\text{dist}(K_i, K_j) = \text{dist}(M_i, M_j)$
 - Medoid: a chosen, centrally located object in the cluster

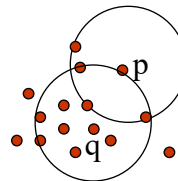
5

5

Density-Based Clustering: DBSCAN

- Two parameters:
 - **Eps:** Maximum radius of the neighbourhood
 - **MinPts:** Minimum number of points in an Eps-neighbourhood of that point
- $N_{Eps}(p)$: $\{q \text{ belongs to } D \mid \text{dist}(p, q) \leq \text{Eps}\}$
- **Directly density-reachable:** A point p is directly density-reachable from a point q w.r.t. Eps , $MinPts$ if
 - p belongs to $N_{Eps}(q)$
 - core point condition:

$$|N_{Eps}(q)| \geq MinPts$$



MinPts = 5

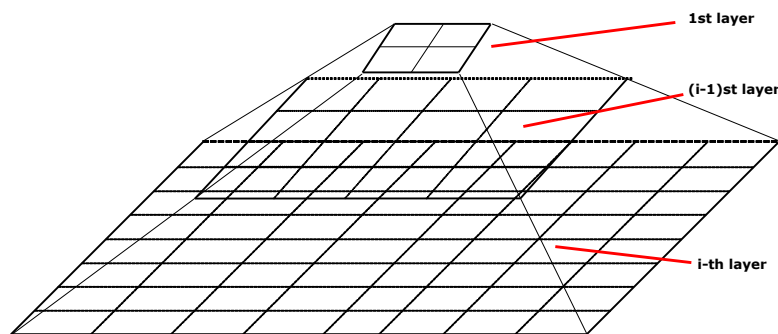
Eps = 1 cm

6

6

STING: A Statistical Information Grid Approach

- Wang, Yang and Muntz (VLDB'97)
- The spatial area is divided into rectangular cells
- There are several levels of cells corresponding to different levels of resolution



7

7

Evaluation of Clustering Quality

- Assessing Clustering Tendency
 - Assess if non-random structure exists in the data by measuring the probability that the data is generated by a uniform data distribution
- Determine the Number of Clusters
 - Empirical method: # of clusters $\approx \sqrt{n}/2$
 - Elbow method: Use the turning point in the curve of sum of within cluster variance w.r.t # of clusters
 - Cross validation method
- Measuring Clustering Quality
 - Extrinsic: supervised
 - Compare a clustering against the ground truth using certain clustering quality measure
 - Intrinsic: unsupervised
 - Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are

8

8


Advanced Clustering Analysis

- Probability Model-Based Clustering
 - Each object may take a probability to belong to a cluster
- Clustering High-Dimensional Data
 - Curse of dimensionality: Difficulty of distance measure in high-D space
- Clustering Graphs and Network Data
 - Similarity measurement and clustering methods for graph and networks
- Clustering with Constraints
 - Cluster analysis under different kinds of constraints, e.g., that raised from background knowledge or spatial distribution of the objects

9

9

Chapter 11. Cluster Analysis: Advanced Methods

- Probability Model-Based Clustering 
- Clustering High-Dimensional Data
- Clustering Graphs and Network Data
- Summary

10

10

10

Fuzzy Set and Fuzzy Cluster

- Clustering methods discussed so far
 - Every data object is assigned to exactly one cluster
- Some applications may need for fuzzy or soft cluster assignment
 - Ex. An e-game could belong to both entertainment and software
- Methods: fuzzy clusters and probabilistic model-based clusters
- Fuzzy cluster: A fuzzy set $S: F_S: X \rightarrow [0, 1]$ (value between 0 and 1)
- Example: Popularity of cameras is defined as a fuzzy mapping

Camera	Sales (units)
A	50
B	1320
C	860
D	270

$$\text{Pop}(o) = \begin{cases} 1 & \text{if 1,000 or more units of } o \text{ are sold} \\ \frac{i}{1000} & \text{if } i \text{ (} i < 1000 \text{) units of } o \text{ are sold} \end{cases}$$

- Then, $A(0.05), B(1), C(0.86), D(0.27)$

11

11

Fuzzy (Soft) Clustering

- Example: Let cluster features be
 - C_1 : “digital camera” and “lens”
 - C_2 : “computer”

Review-id	Keywords
R_1	digital camera, lens
R_2	digital camera
R_3	lens
R_4	digital camera, lens, computer
R_5	computer, CPU
R_6	computer, computer game

$$M = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \frac{2}{3} & \frac{1}{3} \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

- Fuzzy clustering
 - k fuzzy clusters C_1, \dots, C_k , represented as a partition matrix $M = [w_{ij}]$
 - P1: for each object o_i and cluster C_j , $0 \leq w_{ij} \leq 1$ (fuzzy set)
 - P2: for each object o_i , $\sum_{j=1}^k w_{ij} = 1$ equal participation in the clustering
 - P3: for each cluster C_j , $0 < \sum_{i=1}^n w_{ij} < n$ ensures there is no empty cluster

- Let c_1, \dots, c_k as the center of the k clusters

- For an object o_i , sum of the squared error (SSE), p is a parameter:

- For a cluster C_j , SSE:
$$\text{SSE}(C_j) = \sum_{i=1}^n w_{ij}^p \text{dist}(o_i, c_j)^2 \quad \text{SSE}(o_i) = \sum_{j=1}^k w_{ij}^p \text{dist}(o_i, c_j)^2$$

- Measure how well a clustering fits the data:

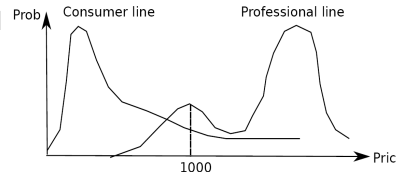
$$\text{SSE}(C) = \sum_{i=1}^n \sum_{j=1}^k w_{ij}^p \text{dist}(o_i, c_j)^2$$

12

12

Probabilistic Model-Based Clustering

- Cluster analysis is to find hidden categories. A hidden category (i.e., *probabilistic cluster*) is a distribution over the data space, which can be mathematically represented using a probability density function (or distribution function).
- Ex. 2 categories for digital cameras sold
 - consumer line vs. professional line
 - density functions f_1, f_2 for C_1, C_2
 - obtained by probabilistic clustering
- A **mixture model** assumes that a set of observed objects D is a mixture of instances from multiple probabilistic clusters, and conceptually each observed object is generated independently
- Task:** infer a set of k probabilistic clusters that is mostly likely to generate D using the above data generation process



13

13

Model-Based Clustering

- A set C of k probabilistic clusters C_1, \dots, C_k with probability density functions f_1, \dots, f_k , respectively, and their probabilities $\omega_1, \dots, \omega_k$.
- Probability of an object o generated by cluster C_j is $P(o|C_j) = \omega_j f_j(o)$
- Probability of o generated by the set of cluster C is $P(o|C) = \sum_{j=1}^k \omega_j f_j(o)$
- Since objects are assumed to be generated independently, for a data set $D = \{o_1, \dots, o_n\}$, we have,

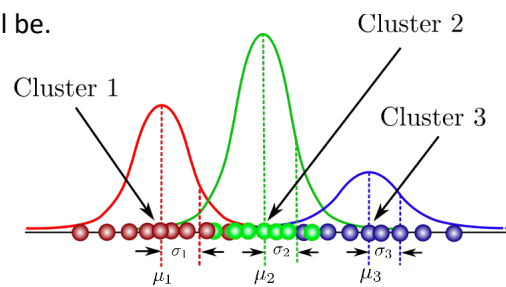
$$P(D|C) = \prod_{i=1}^n P(o_i|C) = \prod_{i=1}^n \sum_{j=1}^k \omega_j f_j(o_i)$$
- Task: Find a set C of k probabilistic clusters s.t. $P(D|C)$ is maximized
- However, maximizing $P(D|C)$ is often intractable since the probability density function of a cluster can take an arbitrarily complicated form

14

14

Gaussian Mixture Model (GMM)

- A *Gaussian Mixture* is a function that is comprised of several Gaussians, each identified by $k \in \{1, \dots, K\}$, where K is the number of clusters of our dataset. Each Gaussian k in the mixture is comprised of the following parameters:
 - A mean μ that defines its centre.
 - A covariance Σ that defines its width.
 - A mixing probability π that defines how big or small the Gaussian function will be.



15

15

Univariate Gaussian Mixture Model

- $O = \{o_1, \dots, o_n\}$ (n observed objects), $\Theta = \{\theta_1, \dots, \theta_k\}$ (parameters of the k distributions), and $P_j(o_i | \theta_j)$ is the probability that o_i is generated from the j -th distribution using parameter θ_j , we have

$$P(o_i | \Theta) = \sum_{j=1}^k \omega_j P_j(o_i | \theta_j) \quad P(O | \Theta) = \prod_{i=1}^n \sum_{j=1}^k \omega_j P_j(o_i | \theta_j)$$

- Univariate Gaussian mixture model
 - Assume the probability density function of each cluster follows a 1-d Gaussian distribution. Suppose that there are k clusters.
 - The probability density function of each cluster are centered at μ_j with standard deviation σ_j , $\theta_j = (\mu_j, \sigma_j)$, we have

$$P(o_i | \theta_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_i - \mu_j)^2}{2\sigma_j^2}} \quad P(o_i | \Theta) = \sum_{j=1}^k \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_i - \mu_j)^2}{2\sigma_j^2}}$$

$$P(O | \Theta) = \prod_{i=1}^n \sum_{j=1}^k \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_i - \mu_j)^2}{2\sigma_j^2}}$$

16

16

The EM (Expectation Maximization) Algorithm

- The k-means algorithm has two steps at each iteration:
 - Expectation Step** (E-step): Given the current cluster centers, each object is assigned to the cluster whose center is closest to the object: An object is *expected to belong to the closest cluster*
 - Maximization Step** (M-step): Given the cluster assignment, for each cluster, the algorithm *adjusts the center* so that *the sum of distance* from the objects assigned to this cluster and the new center is minimized
- The (EM) algorithm:** A framework to approach maximum likelihood or maximum a posteriori estimates of parameters in statistical models.
 - E-step** assigns objects to clusters according to the current fuzzy clustering or parameters of probabilistic clusters
 - M-step** finds the new clustering or parameters that maximize the sum of squared error (SSE) or the expected likelihood

17

17

Computing Mixture Models with EM

- Given n objects $O = \{o_1, \dots, o_n\}$, we want to mine a set of parameters $\Theta = \{\theta_1, \dots, \theta_k\}$ s.t., $P(O|\Theta)$ is maximized, where $\theta_j = (\mu_j, \sigma_j)$ are the mean and standard deviation of the j -th univariate Gaussian distribution
- We initially assign random values to parameters θ_j , then iteratively conduct the E- and M- steps until converge or sufficiently small change
- At the E-step, for each object o_i , calculate the probability that o_i belongs to each distribution,

$$P(\theta_j|o_i, \Theta) = \frac{P(o_i|\theta_j)}{\sum_{l=1}^k P(o_i|\theta_l)}$$

- At the M-step, adjust the parameters $\theta_j = (\mu_j, \sigma_j)$ so that the expected likelihood $P(O|\Theta)$ is maximized

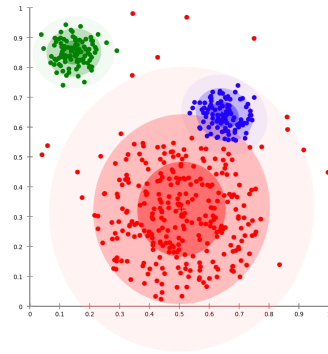
$$\mu_j = \frac{\sum_{i=1}^n o_i \frac{P(\theta_j|o_i, \Theta)}{\sum_{l=1}^k P(\theta_l|o_i, \Theta)}}{\sum_{i=1}^n \frac{P(\theta_j|o_i, \Theta)}{\sum_{l=1}^k P(\theta_l|o_i, \Theta)}} \quad \sigma_j = \sqrt{\frac{\sum_{i=1}^n \frac{P(\theta_j|o_i, \Theta)}{\sum_{l=1}^k P(\theta_l|o_i, \Theta)} (o_i - \mu_j)^2}{\sum_{i=1}^n \frac{P(\theta_j|o_i, \Theta)}{\sum_{l=1}^k P(\theta_l|o_i, \Theta)}}}$$

18

18

Advantages and Disadvantages of Mixture Models

- Strength
 - Mixture models are more general than partitioning and fuzzy clustering
 - Clusters can be characterized by a small number of parameters
 - The results may satisfy the statistical assumptions of the generative models
- Weakness
 - Converge to local optimal (overcome: run multi-times w. random initialization)
 - Computationally expensive if the number of distributions is large, or the data set contains very few observed data points
 - Need large data sets
 - Hard to estimate the number of clusters




<https://commons.wikimedia.org/wiki/File:EM-Gaussian-data.svg>

19

19

Chapter 11. Cluster Analysis: Advanced Methods

- Probability Model-Based Clustering
- Clustering High-Dimensional Data 
- Clustering Graphs and Network Data
- Summary

20

20

Clustering High-Dimensional Data

- Clustering high-dimensional data (How high is high-D in clustering?)
 - Many applications: text documents, DNA micro-array data
 - Major challenges:
 - Many irrelevant dimensions may mask clusters
 - Distance measure becomes meaningless—due to equidistance
 - Clusters may exist only in some subspaces
- Methods
 - **Subspace-clustering:** Search for clusters existing in subspaces of the given high dimensional data space
 - CLIQUE, ProClus, and bi-clustering approaches
 - **Dimensionality reduction approaches:** Construct a much lower dimensional space and search for clusters there (may construct new dimensions by combining some dimensions in the original data)
 - Dimensionality reduction methods and spectral clustering

21

21

Traditional Distance Measures May Not Be Effective on High-D Data

- Traditional distance measure could be dominated by noises in many dimensions
- Ex. Which pairs of customers are more similar?

Customer	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
Ada	1	0	0	0	0	0	0	0	0	0
Bob	0	0	0	0	0	0	0	0	0	1
Cathy	1	0	0	0	1	0	0	0	0	1

- By Euclidean distance, we get,

$$\text{dist}(\text{Ada}, \text{Bob}) = \text{dist}(\text{Bob}, \text{Cathy}) = \text{dist}(\text{Ada}, \text{Cathy}) = \sqrt{2}$$
 - despite Ada and Cathy look more similar
- Clustering should not only consider dimensions but also attributes (features)
 - **Feature transformation:** effective if most dimensions are relevant (PCA & SVD useful when features are highly correlated/redundant)
 - **Feature selection:** useful to find a subspace where the data have nice clusters

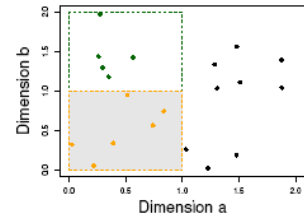
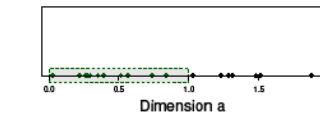
22

22

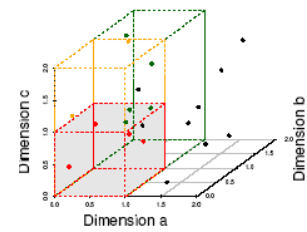
The Curse of Dimensionality

(graphs adapted from Parsons et al. KDD Explorations 2004)

- Data in only one dimension is relatively packed
- Adding a dimension “stretch” the points across that dimension, making them further apart
- Adding more dimensions will make the points further apart—high dimensional data is extremely sparse
- Distance measure becomes meaningless—due to equidistance



(b) 6 Objects in One Unit Bin



(c) 4 Objects in One Unit Bin

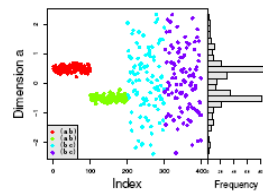
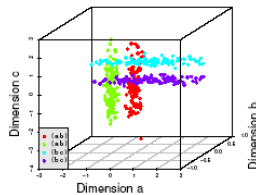
23

23

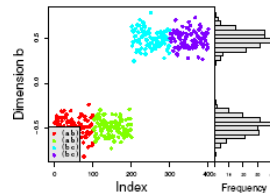
Why Subspace Clustering?

(adapted from Parsons et al. SIGKDD Explorations 2004)

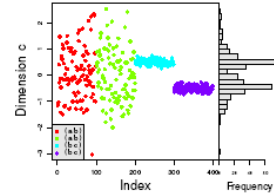
- Clusters may exist only in some subspaces
- Subspace-clustering: find clusters in all the subspaces



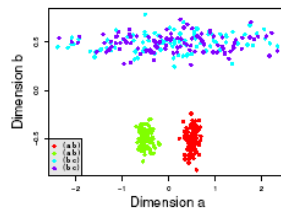
(a) Dimension a



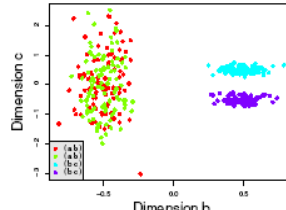
(b) Dimension b



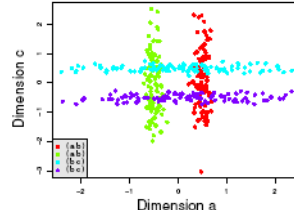
(c) Dimension c



(a) Dims a & b



(b) Dims b & c



(c) Dims a & c

24

Subspace Clustering Methods

- Subspace search methods: Search various subspaces to find clusters
 - Bottom-up approaches
 - Top-down approaches
- Correlation-based clustering methods
 - E.g., PCA based approaches
- Bi-clustering methods
 - Optimization-based methods
 - Enumeration methods

25

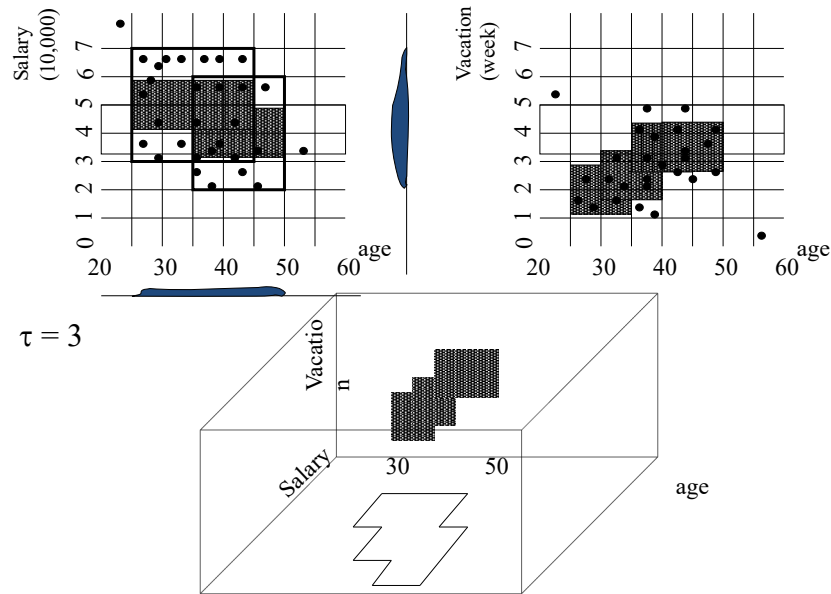
Subspace Clustering Method (I): Subspace Search Methods

- Search various subspaces to find clusters
- *Bottom-up approaches*
 - Start from low-D subspaces and search higher-D subspaces only when there may be clusters in such subspaces
 - Various pruning techniques to reduce the number of higher-D subspaces to be searched
 - Ex. CLIQUE (Agrawal et al. 1998)
- *Top-down approaches*
 - Start from full space and search smaller subspaces recursively
 - Effective only if the *locality assumption* holds: restricts that the subspace of a cluster can be determined by the local neighborhood
 - Ex. PROCLUS (Aggarwal et al. 1999): a *k*-medoid-like method

26

26

CLIQUE: SubSpace Clustering with Aprori Pruning



27

Subspace Clustering Method (II): Correlation-Based Methods

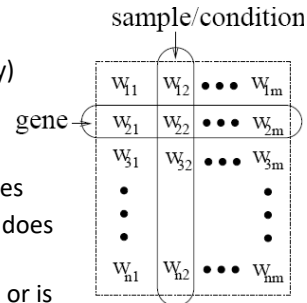
- Subspace search method: similarity based on distance or density
- Correlation-based method: based on advanced correlation models
- Ex. PCA-based approach:
 - Apply PCA (for Principal Component Analysis) to derive a set of new, uncorrelated dimensions,
 - then mine clusters in the new space or its subspaces

28

28

Subspace Clustering Method (III): Bi-Clustering Methods

- Bi-clustering: Cluster both objects and attributes simultaneously (treat objs and attrs in symmetric way)
- Four requirements:
 - Only a small set of objects participate in a cluster
 - A cluster only involves a small number of attributes
 - An object may participate in multiple clusters, or does not participate in any cluster at all
 - An attribute may be involved in multiple clusters, or is not involved in any cluster at all



- Ex 1. *Gene expression or microarray data*: a gene sample/condition matrix.

- Each element in the matrix, a real number, records the expression level of a gene under a specific condition

		products			
	w_{11}	w_{12}	\dots	w_{1m}	
	w_{21}	w_{22}	\dots	w_{2m}	
	\dots	\dots	\dots	\dots	
	w_{n1}	w_{n2}	\dots	w_{nm}	
customers					

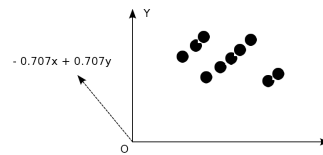
- Ex. 2. Clustering customers and products

- Another bi-clustering problem

29

Dimensionality-Reduction Methods


- Dimensionality reduction: In some situations, it is more effective to construct a new space instead of using some subspaces of the original data
- Ex. To cluster the points in the right figure, any subspace of the original one, X and Y, cannot help, since all the three clusters will be projected into the overlapping areas in X and Y axes.
 - Construct a new dimension as the dashed one, the three clusters become apparent when the points projected into the new dimension
- Dimensionality reduction methods
 - Feature selection and extraction: But may not focus on clustering structure finding
 - Spectral clustering: Combining feature extraction and clustering (i.e., use the *spectrum* of the similarity matrix of the data to perform dimensionality reduction for clustering in fewer dimensions)
 - Normalized Cuts (Shi and Malik, CVPR'97 or PAMI'2000)
 - The Ng-Jordan-Weiss algorithm (NIPS'01)



30

30

Chapter 11. Cluster Analysis: Advanced Methods

- Probability Model-Based Clustering
- Clustering High-Dimensional Data
- Clustering Graphs and Network Data 
- Summary

31

31

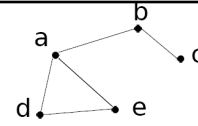
Clustering Graphs and Network Data

- Applications
 - Bi-partite graphs, e.g., customers and products, authors and conferences
 - Web search engines, e.g., click through graphs and Web graphs
 - Social networks, friendship/coauthor graphs
- Similarity measures
 - Geodesic distances
 - Distance based on random walk (SimRank)
- Graph clustering methods
 - Minimum cuts: FastModularity (Clauset, Newman & Moore, 2004)
 - Density-based clustering: SCAN (Xu et al., KDD'2007)

32

32

Geodesic Distance



- It is a Similarity Measure for nodes. Geodesic distance (A, B) is the length (i.e., # of edges) of the *shortest path* between A and B (if not connected, defined as infinite)
- Eccentricity** of v , $\text{eccen}(v)$: The largest geodesic distance between v and any other vertex $u \in V - \{v\}$.
 - E.g., $\text{eccen}(a) = \text{eccen}(b) = 2$; $\text{eccen}(c) = \text{eccen}(d) = \text{eccen}(e) = 3$
- Radius** of graph G : The minimum eccentricity of all vertices, i.e., the distance between the “most central point” and the “farthest border”
 - $r = \min_{v \in V} \text{eccen}(v)$
 - E.g., $\text{radius}(g) = 2$
- Diameter** of graph G : The maximum eccentricity of all vertices, i.e., the largest distance between any pair of vertices in G
 - $d = \max_{v \in V} \text{eccen}(v)$
 - E.g., $\text{diameter}(g) = 3$
- A **peripheral vertex** is a vertex that achieves the diameter.
 - E.g., Vertices c , d , and e are peripheral vertices

33

33

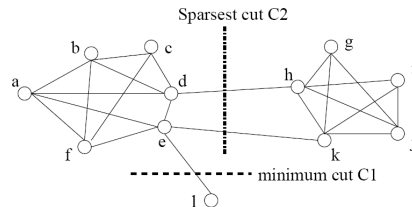
Graph Clustering: Sparsest Cut

- $G = (V, E)$. The *cut set* of a cut is the set of edges $\{(u, v) \in E \mid u \in S, v \in T\}$ and S and T are in two partitions
- Size of the cut: # of edges in the cut set
- Min-cut (e.g., C_1) is not a good partition
- A better measure: **Sparsity**: $\Phi = \frac{\text{the size of the cut}}{\min\{|S|, |T|\}}$
 - A cut is **sparsest** if its sparsity is not greater than that of any other cut
 - Ex. Cut $C_2 = (\{a, b, c, d, e, f, l\}, \{g, h, i, j, k\})$ is the sparsest cut
 - For k clusters, the **modularity** of a clustering assesses the quality of the clustering:

$$Q = \sum_{i=1}^k \left(\frac{l_i}{|E|} - \left(\frac{d_i}{2|E|} \right)^2 \right)$$

l_i : # edges between vertices in the i -th cluster

d_i : the sum of the degrees of the vertices in the i -th cluster
 - The *modularity* of a clustering of a graph is the difference between the fraction of all edges that fall into individual clusters and the fraction that would do so if the graph vertices were randomly connected
 - The optimal clustering of graphs maximizes the modularity



34

34

Graph Clustering: Challenges of Finding Good Cuts

- High computational cost
 - Many graph cut problems are computationally expensive
 - The sparsest cut problem is NP-hard
 - Need to tradeoff between efficiency/scalability and quality
- Sophisticated graphs
 - May involve weights and/or cycles.
- High dimensionality
 - A graph can have many vertices. In a similarity matrix, a vertex is represented as a vector (a row in the matrix) whose dimensionality is the number of vertices in the graph
- Sparsity
 - A large graph is often sparse, meaning each vertex on average connects to only a small number of other vertices
 - A similarity matrix from a large sparse graph can also be sparse

35

35

Two Approaches for Graph Clustering

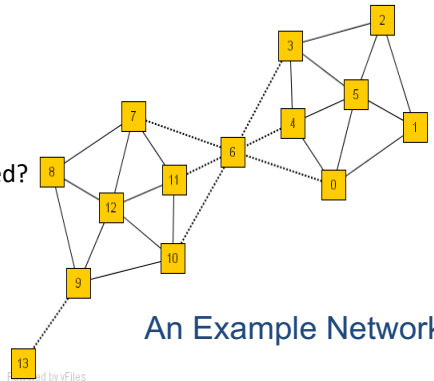
- Two approaches for clustering graph data
 - Use *generic clustering methods* for high-dimensional data
 - *Designed specifically for clustering graphs*
- Using clustering methods for high-dimensional data
 - Extract a similarity matrix from a graph using a similarity measure
 - A generic clustering method can then be applied on the similarity matrix to discover clusters
 - Ex. Spectral clustering: approximate optimal graph cut solutions
- Methods specific to graphs
 - Search the graph to find well-connected components as clusters
 - Ex. SCAN (Structural Clustering Algorithm for Networks)
 - X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger, "SCAN: A Structural Clustering Algorithm for Networks", KDD'07

36

36

SCAN: Density-Based Clustering of Networks

- How many clusters?
- What size should they be?
- What is the best partitioning?
- Should some points be segregated?



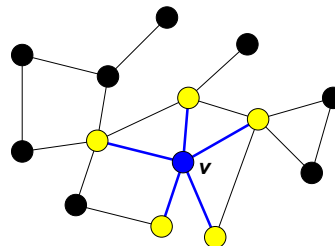
- Application: Given simply information of who associates with whom, could one identify clusters of individuals with common interests or special relationships (families, cliques, terrorist cells)?

37

37

A Social Network Model

- Cliques, hubs and outliers
 - Individuals in a tight social group, or **clique**, know many of the same people, regardless of the size of the group
 - Individuals who are **hubs** know many people in different groups but belong to no single group. Politicians, for example bridge multiple groups
 - Individuals who are **outliers** reside at the margins of society. Hermits, for example, know few people and belong to no group
- The Neighborhood of a Vertex
 - Define $\Gamma(v)$ as the immediate neighborhood of a vertex (i.e. the set of people that an individual knows)



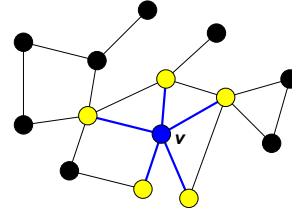
38

38

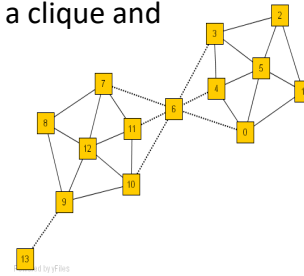
Structure Similarity

- The desired features tend to be captured by a measure we call Structural Similarity

$$\sigma(v, w) = \frac{|\Gamma(v) \cap \Gamma(w)|}{\sqrt{|\Gamma(v)| |\Gamma(w)|}}$$



- Structural similarity is large for members of a clique and small for hubs and outliers



39

39

Structural Connectivity [1]

- \mathcal{E} -Neighborhood: $N_{\mathcal{E}}(v) = \{w \in \Gamma(v) \mid \sigma(v, w) \geq \mathcal{E}\}$

- Core: $CORE_{\mathcal{E}, \mu}(v) \Leftrightarrow |N_{\mathcal{E}}(v)| \geq \mu$

- Direct structure reachable:

$$DirRECH_{\mathcal{E}, \mu}(v, w) \Leftrightarrow CORE_{\mathcal{E}, \mu}(v) \wedge w \in N_{\mathcal{E}}(v)$$

- Structure reachable: transitive closure of direct structure reachability

- Structure connected:

$$CONNECT_{\mathcal{E}, \mu}(v, w) \Leftrightarrow \exists u \in V : RECH_{\mathcal{E}, \mu}(u, v) \wedge RECH_{\mathcal{E}, \mu}(u, w)$$

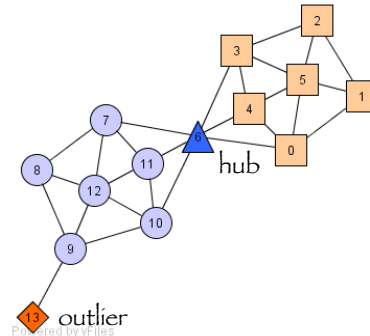
[1] M. Ester, H. P. Kriegel, J. Sander, & X. Xu (KDD'96) "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases"

40

40

Structure-Connected Clusters

- Structure-connected cluster C
 - Connectivity: $\forall v, w \in C : \text{CONNECT}_{\varepsilon, \mu}(v, w)$
 - Maximality: $\forall v, w \in V : v \in C \wedge \text{REACH}_{\varepsilon, \mu}(v, w) \Rightarrow w \in C$
- Hubs:
 - Not belong to any cluster
 - Bridge to many clusters
- Outliers:
 - Not belong to any cluster
 - Connect to less clusters



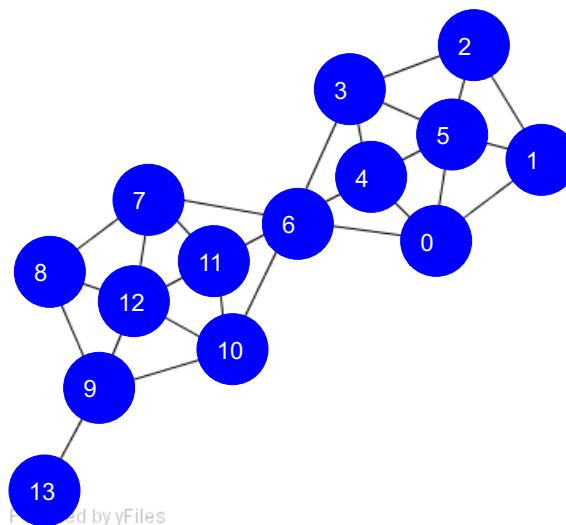
41

41

Algorithm

$$\mu = 2$$

$$\varepsilon = 0.7$$



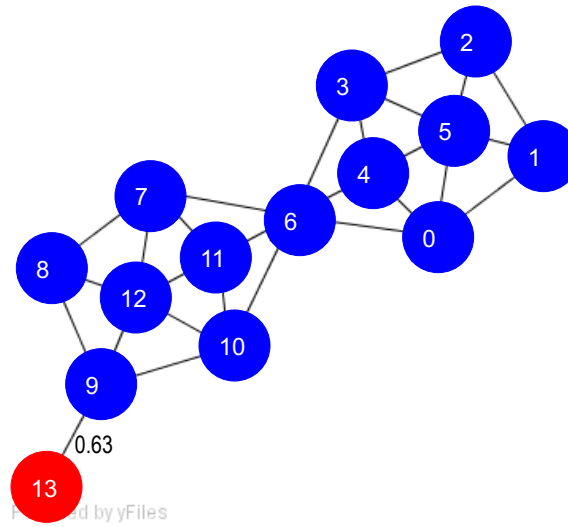
42

42

Algorithm

$$\mu = 2$$

$$\varepsilon = 0.7$$



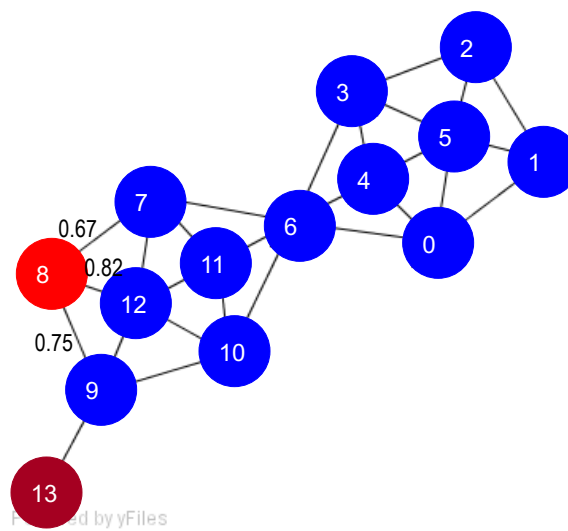
43

43

Algorithm

$$\mu = 2$$

$$\varepsilon = 0.7$$



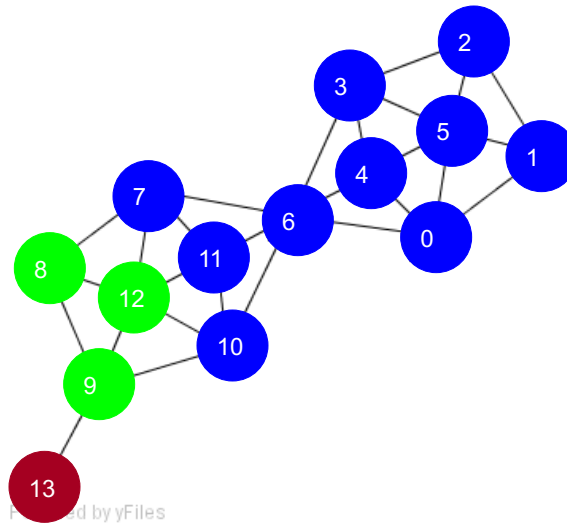
44

44

Algorithm

$$\mu = 2$$

$$\varepsilon = 0.7$$



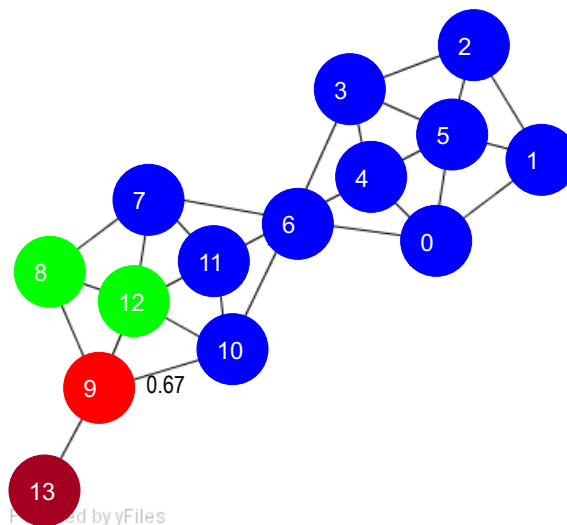
45

45

Algorithm

$$\mu = 2$$

$$\varepsilon = 0.7$$



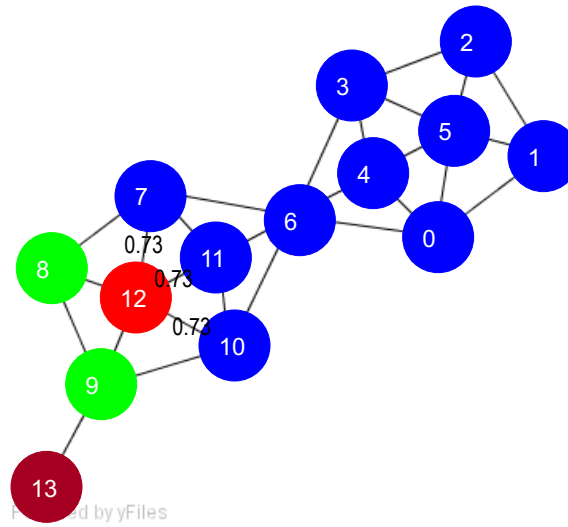
46

46

Algorithm

$$\mu = 2$$

$$\varepsilon = 0.7$$



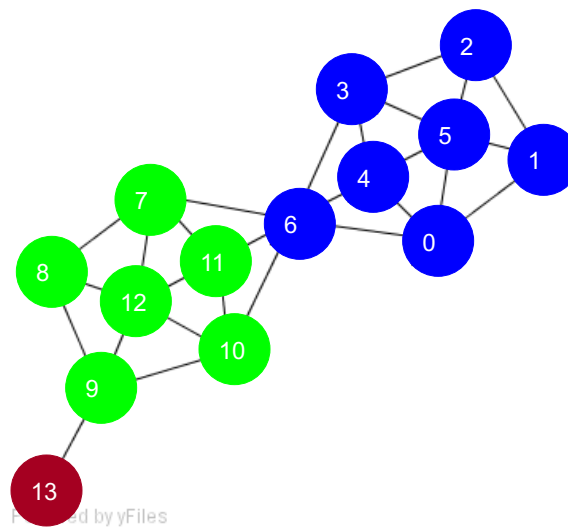
47

47

Algorithm

$$\mu = 2$$

$$\varepsilon = 0.7$$



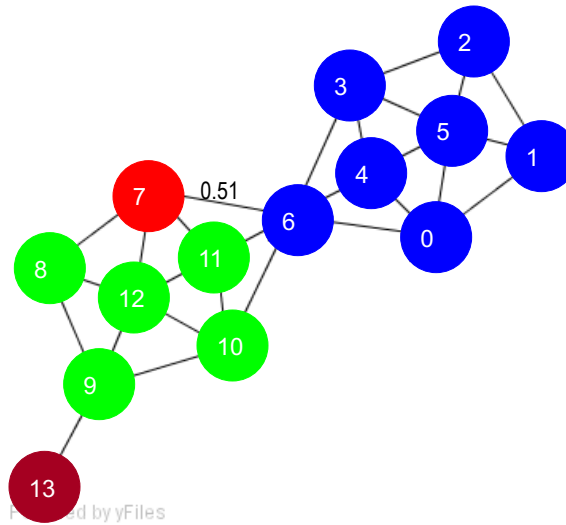
48

48

Algorithm

$$\mu = 2$$

$$\varepsilon = 0.7$$



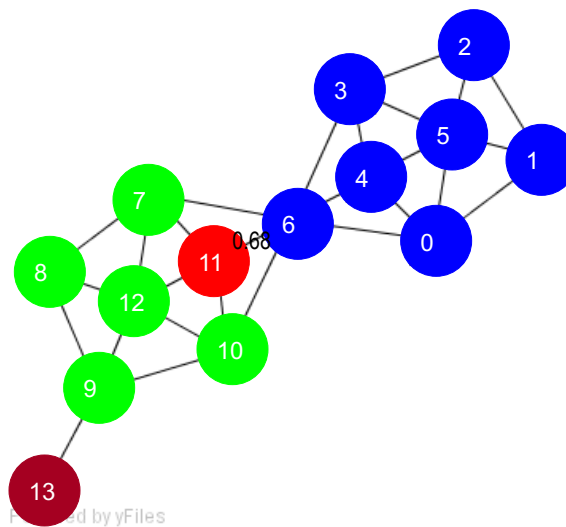
49

49

Algorithm

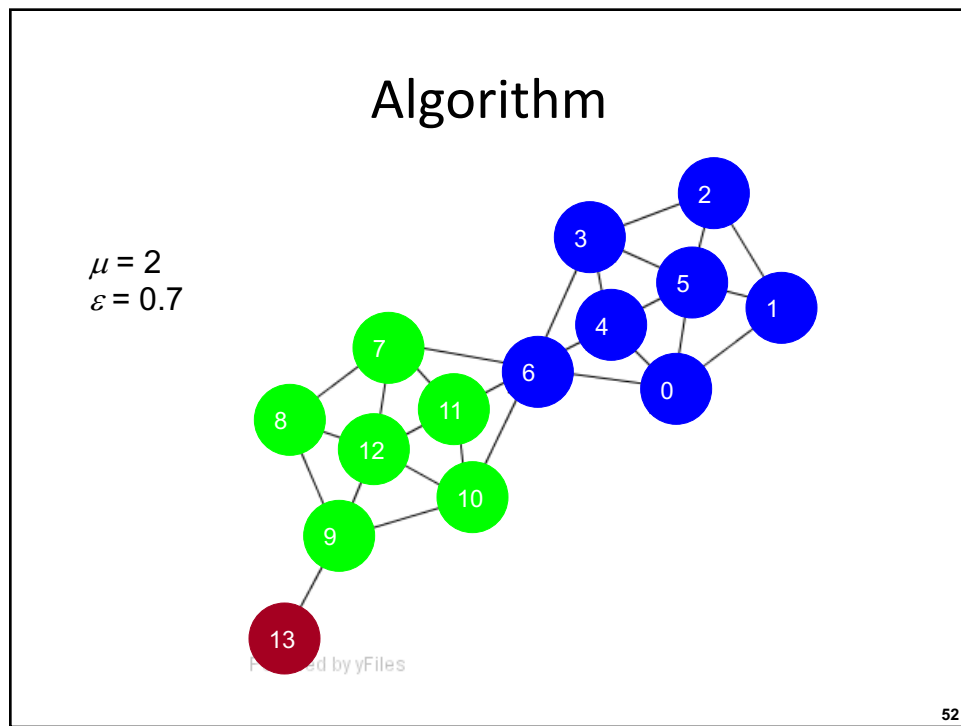
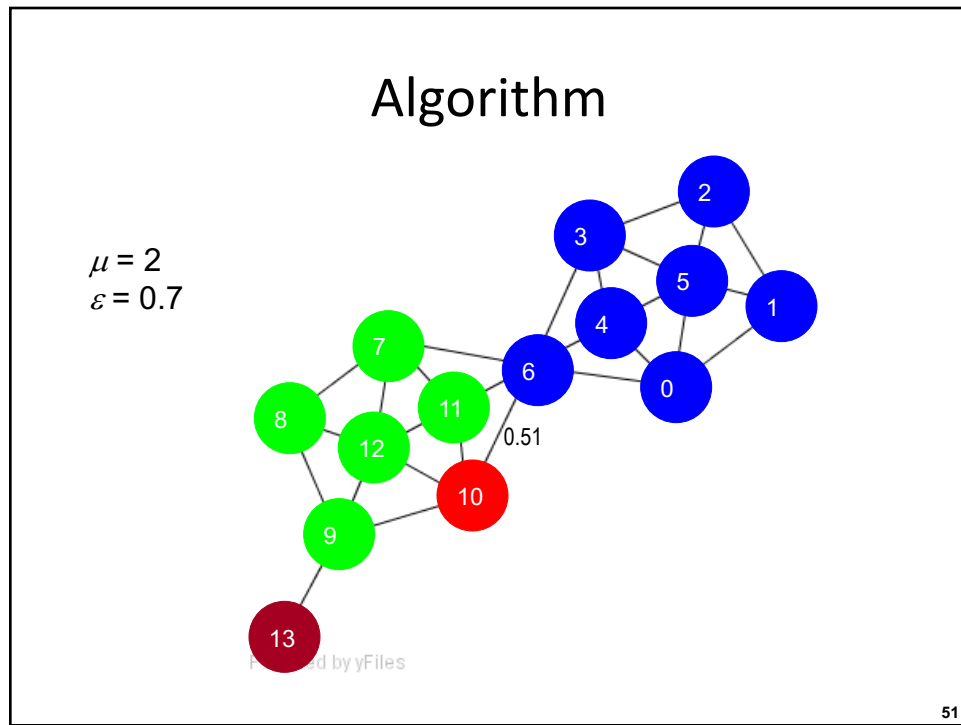
$$\mu = 2$$

$$\varepsilon = 0.7$$



50

50

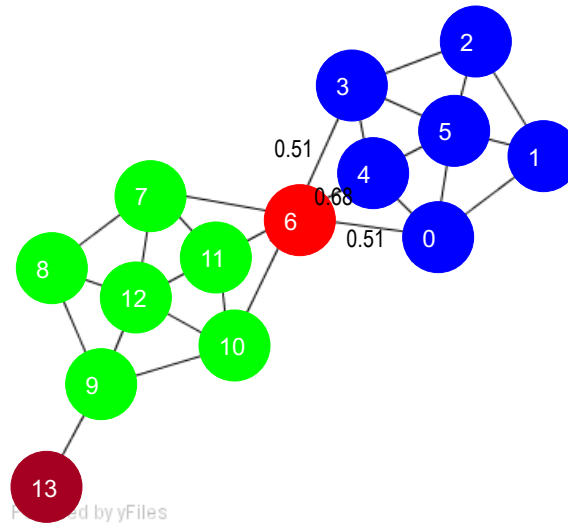


52

Algorithm

$$\mu = 2$$

$$\varepsilon = 0.7$$



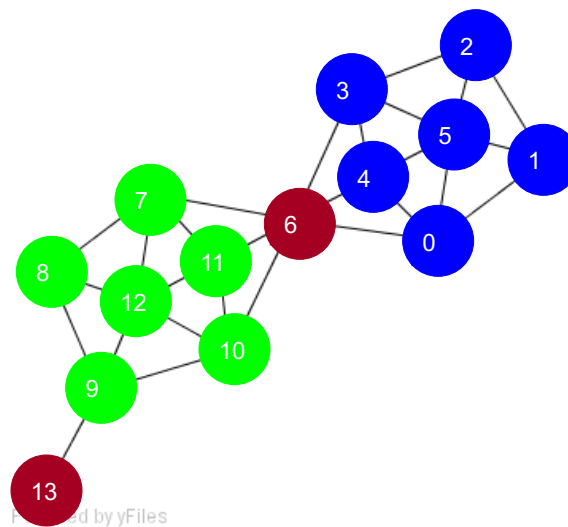
53

53

Algorithm

$$\mu = 2$$

$$\varepsilon = 0.7$$



54

54

Summary

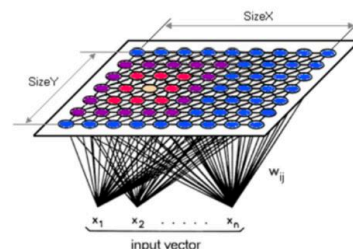
- Probability Model-Based Clustering
 - Fuzzy clustering
 - Probability-model-based clustering
 - The EM algorithm
- Clustering High-Dimensional Data
 - Subspace clustering: bi-clustering methods
 - Dimensionality reduction: Spectral clustering
- Clustering Graphs and Network Data
 - Graph clustering: min-cut vs. sparsest cut
 - High-dimensional clustering methods
 - Graph-specific clustering methods, e.g., SCAN

55

55

Neural Network Approach to Clustering

- Neural network approaches represent each cluster as an exemplar, acting as a “prototype” of the cluster. New objects are distributed to the cluster whose exemplar is the most similar according to some distance measure.
- A typical method is SOM (Soft-Organizing feature Map). SOM is actually a constrained K-Means! Fixing the centers and we push the weights towards the data.



56

56

Self-Organizing Feature Map (SOM)

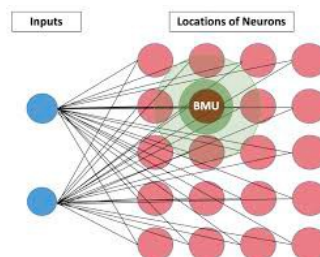
- SOMs, also called topological ordered maps, or Kohonen Self-Organizing Feature Map (KSOMs), the model self-organizes based on learning rules.
- It maps all the points in a high-dimensional source space into a 2 to 3-d target space, s.t., the distance and proximity relationship (i.e., topology) are preserved as much as possible
- Similar to k-means: cluster centers tend to lie in a low-dimensional manifold in the feature space
- Clustering is performed by having *several units competing* for the current object
 - The unit whose weight vector is closest to the current object wins
 - The winner and its neighbors learn by having their weights adjusted
- SOMs are believed to resemble processing that can occur in the brain
- Useful for visualizing high-dimensional data in 2- or 3-D space

57

57

Self-Organizing Feature Map (SOM)

- SOMs are trained using unsupervised learning to produce a low-dimensional (typically two-dimensional) representation of the input, called a map.
- SOMs differ from other artificial neural networks as they apply competitive learning as opposed to error-correction learning (such as backpropagation with gradient descent), and in the sense that they use a neighborhood function to preserve the topological properties of the input space.

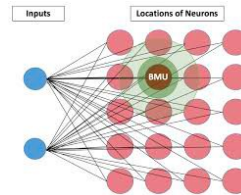


58

58

SOM Algorithm

1. Each node's weights are initialized.
2. A vector is chosen at random from the set of training data.
3. Every node is examined to calculate which one's weights are closest to the input vector. The winning node is the **Best Matching Unit (BMU)**.
4. Then the neighbourhood of the BMU is calculated. The amount of neighbors decreases over time.
5. The winning weight is rewarded with becoming more like the sample vector. The neighbors also become more like the sample vector. The closer a node is to the BMU, the more its weights get altered and the farther away the neighbor is from the BMU, the less it learns.
6. Repeat step 2 for N iterations.

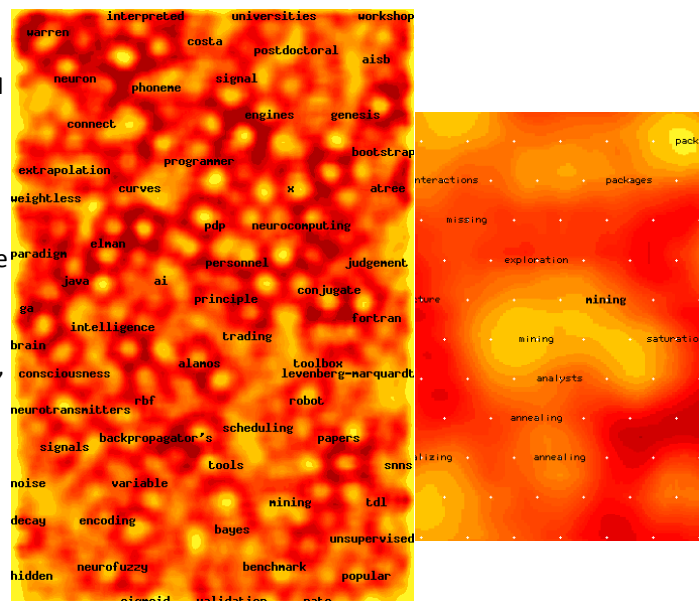


59

59

Web Document Clustering Using SOM

- The result of SOM clustering of 12088 Web articles
- The picture on the right: drilling down on the keyword “mining”
- Based on websom.hut.fi Web page



60

60

References (I)

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD'98*
- C. C. Aggarwal, C. Procopiuc, J. Wolf, P. S. Yu, and J.-S. Park. Fast algorithms for projected clustering. *SIGMOD'99*
- S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56:5:1–5:37, 2009.
- J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, 1981.
- K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is “nearest neighbor” meaningful? *ICDT'99*
- Y. Cheng and G. Church. Biclustering of expression data. *ISMB'00*
- I. Davidson and S. S. Ravi. Clustering with constraints: Feasibility issues and the k-means algorithm. *SDM'05*
- I. Davidson, K. L. Wagstaff, and S. Basu. Measuring constraint-set utility for partitional clustering algorithms. *PKDD'06*
- C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *J. American Stat. Assoc.*, 97:611–631, 2002.
- F. Hoppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. Wiley, 1999.
- G. Jeh and J. Widom. SimRank: a measure of structural-context similarity. *KDD'02*
- H.-P. Kriegel, P. Kroeger, and A. Zimek. Clustering high dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. Knowledge Discovery from Data (TKDD)*, 3, 2009.
- U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007

61

61

References (II)

- G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. John Wiley & Sons, 1988.
- B. Mirkin. Mathematical classification and clustering. *J. of Global Optimization*, 12:105–108, 1998.
- S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 1, 2004.
- A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *NIPS'01*
- J. Pei, X. Zhang, M. Cho, H. Wang, and P. S. Yu. Maple: A fast algorithm for maximal pattern-based clustering. *ICDM'03*
- M. Radovanović, A. Nanopoulos, and M. Ivanović. Nearest neighbors in high-dimensional data: the emergence and influence of hubs. *ICML'09*
- S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1:27–64, 2007.
- A. K. H. Tung, J. Hou, and J. Han. Spatial clustering in the presence of obstacles. *ICDE'01*
- A. K. H. Tung, J. Han, L. V. S. Lakshmanan, and R. T. Ng. Constraint-based clustering in large databases. *ICDT'01*
- A. Tanay, R. Sharan, and R. Shamir. Biclustering algorithms: A survey. In *Handbook of Computational Molecular Biology*, Chapman & Hall, 2004.
- K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. *ICML'01*
- H. Wang, W. Wang, J. Yang, and P. S. Yu. Clustering by pattern similarity in large data sets. *SIGMOD'02*
- X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger. SCAN: A structural clustering algorithm for networks. *KDD'07*
- X. Yin, J. Han, and P. S. Yu. “Cross-Relational Clustering with User’s Guidance”, *KDD'05*

62