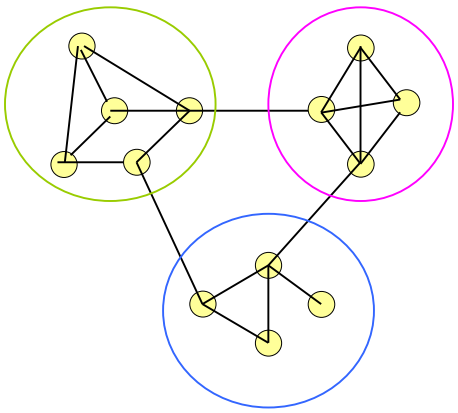
Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
  - Graph clustering (community finding)
- Evaluation of Clustering
- Summary

Graph clustering (community finding)

- Community structure:
  - Groups of vertices within which connections are dense but between which they are sparser.
  - Within-group(intra-group) edges.
    - High density
  - Between-group(inter-group) edges.
    - Low density.

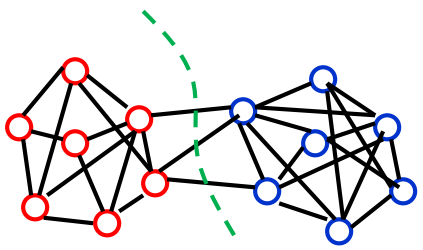
Community Structure



102

Community finding vs. other approaches

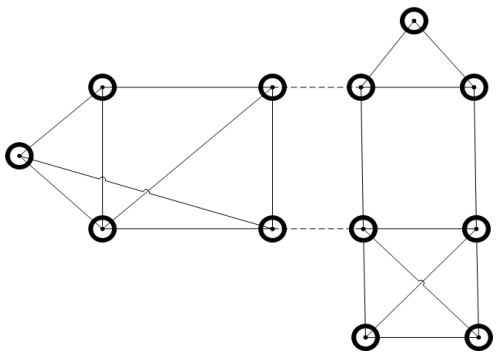
- Social and other networks have a natural community structure
- We want to discover this structure rather than impose a certain size of community or fix the number of communities



104

Detecting Community Structure (Clustering)

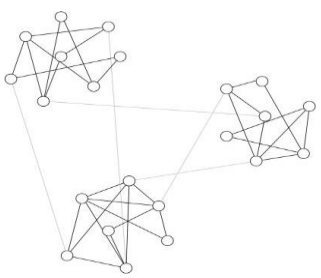
- Cluster analysis seeks grouping of elements into subsets based on similarity between pairs of elements.



105

Edge betweenness

- Number of shortest paths between pairs of vertices that run along it
- The edges connecting communities will have high edge betweenness
- Separate communities by removing these edges

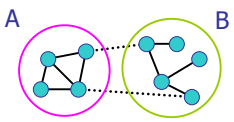


106

Girvan and Newman(GN) Algorithm

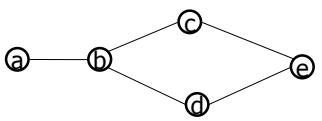
1. Calculate the betweenness for all edges in the network.
2. Remove the edge with the highest betweenness.
3. Recalculate betweenness for all edges affected by the removal.
4. Repeat from step 2 until no edges remain.
5. cross cut the dendrogram of components.

- By removing these edges, we separate groups from one another as components.



GN Algorithm- Example

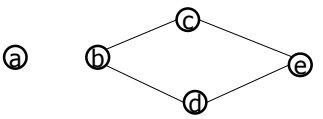
- 1. Calculate the betweenness for all edges in the network.



ab	**4**
bc	**3**
bd	**3**
ce	**3**
de	**3**

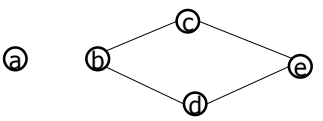
GN Algorithm- Example(cont.)

- 2. Remove the edge with the highest betweenness.



GN Algorithm- Example(cont.)

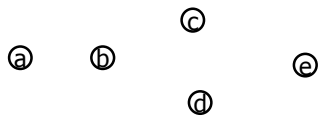
- 3. Recalculate betweennesses for all edges affected by the removal.



bc	**2**
bd	**2**
ce	**2**
de	**2**

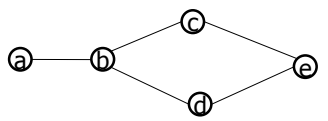
GN Algorithm- Example(cont.)

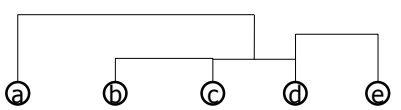
- 4. Repeat from step 2 until no edges remain.



111

GN Algorithm- Example(cont.)





112

Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

Determine the Number of Clusters

- Empirical method
  - # of clusters: k ≈√n/2 for a dataset of n points, e.g., n = 200, k = 10
- Cross validation method
  - Divide a given data set into *m* parts
  - Use *m* – 1 parts to obtain a clustering model
  - Use the remaining part to test the quality of the clustering
    - E.g., For each point in the test set, find the closest centroid, and use the sum of squared distance between all points in the test set and the closest centroids to measure how well the model fits the test set
  - For any k > 0, repeat it *m* times, compare the overall quality measure w.r.t. different *k's*, and find # of clusters that fits the data the best

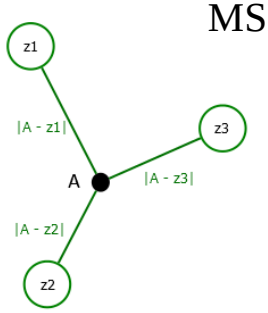
Cluster Evaluation and assessment

- Internal evaluation: Unsupervised, criteria derived from data itself
  - Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are
  - Methods: Dunn index, Davies–Bouldin, Silhouette coefficient
- External evaluation: supervised, employ criteria not inherent to the dataset)
  - Compare a clustering against prior or expert-specified knowledge (i.e., the ground truth) using certain clustering quality measure
  - Methods: Rand measure, F-measure, Jaccard index, Fowlkes–Mallows index, Confusion matrix
- Relative: directly compare different clusterings, usually those obtained via different parameter settings for the same algorithm

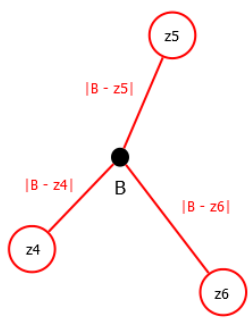
Clustering Error

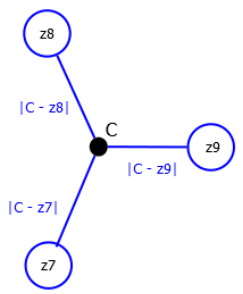
$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (d(p, c_i))^2$$

$$MSE = \frac{1}{|k|} \Sigma_{i=1}^{k} \Sigma_{j \in C_i} (j - c_i)^2$$



MSE= [|A - z1|^2 + |A - z2|^2 + |A - z3|^2 +
|B - z4|^2 + |B - z5|^2 + |B - z6|^2 +
|C - z7|^2 + |C - z8|^2 + |C - z9|^2] / |Z|





Davies-Bouldin index (DB ↓)

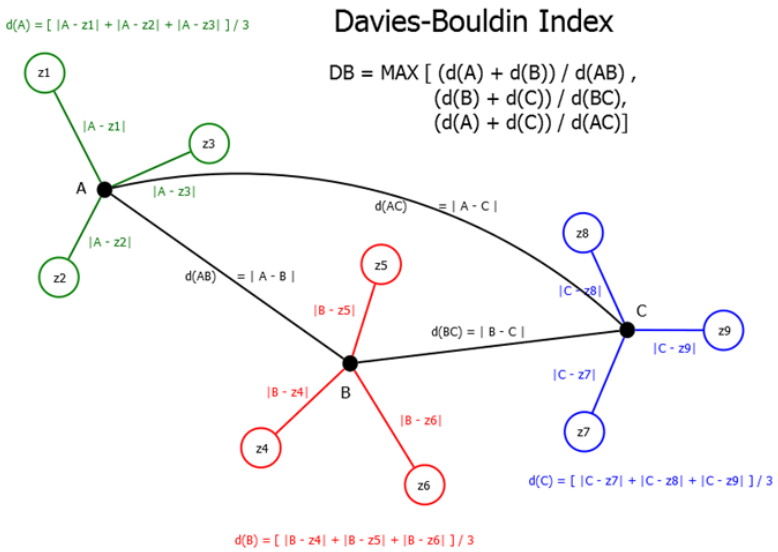
- A function of the ratio of the sum of within-cluster (i.e. intra-cluster) scatter to between cluster (i.e. inter-cluster) separation
- Let $C=\{C_1,\ldots,C_k\}$ be a clustering of a set of N objects:

$$DB = \frac{1}{k}.\sum_{i=1}^{k} R_i$$

$$R_i = \max_{j=1,..k, i\neq j} R_{ij} \qquad R_{ij} = \frac{\mathrm{var}(C_i)+\mathrm{var}(C_j)}{\| c_i - c_j \|}$$

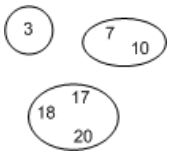
C_i is the i^{th} cluster
c_i is the centroid for cluster i

Davies-Bouldin index (DB ↓)



Davies-Bouldin index example

- Consider the shown clusters (ine one dimension)

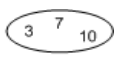


- Compute $R_{\substack{ij \\ i \neq j}} = \frac{\text{var}(C_i) + \text{var}(C_j)}{\| c_i - c_j \|}$
- *var*(C_1)=0, *var*(C_2)=4.5, *var*(C_3)=2.33
- Centroid is simply the mean here, so c_1=3, c_2=8.5, c_3=18.33
- *So,* R_{12}=1, R_{13}=0.152, R_{23}=0.797
- Now, compute $R_i = \max_{j=1,..k, i \neq j} R_{ij}$
- R_1=1 (max of R_{12} and R_{13}); R_2=1 (max of R_{21} and R_{23}); R_3=0.797 (max of R_{31} and R_{32})
- Finally, compute $DB = \frac{1}{k} . \sum_{i=1}^{k} R_i$
- **DB=0.932**

119

Davies-Bouldin index example (ctd)

- Consider the shown clusters: for the clusters sh



- Compute $R_{\substack{ij \\ i \neq j}} = \frac{\text{var}(C_i) + \text{var}(C_j)}{\| c_i - c_j \|}$
- Only 2 clusters here
- *var*(C_1)=12.33 while *var*(C_2)=2.33; c_1=6.67 while c_2=18.33
- R_{12}=1.26
- Now compute
- *Since we have only 2 clusters here,* R_1=R_{12}=1.26; R_2=R_{21}=*1.26*
- Finally, compute $R_i = \max_{j=1,..k, i \neq j} R_{ij}$
- DB=1.26

$$DB = \frac{1}{k} . \sum_{i=1}^{k} R_i$$

120

Dunn index (D ↑)

- The Dunn index aims to identify **dense and well-separated clusters**. It is defined as the ratio between the minimal inter-cluster distance to maximal intra-cluster distance. For each cluster partition, the Dunn index can be calculated by the following formula

$$D = \frac{d_{\min}}{d_{\max}}$$

Min: Distance between 2 data in inter-cluster

Max: Distance between 2 data among intra-cluster

0<D<∞

$$D = \min_{i=1\ldots n_c} \left\{ \min_{j=i+1\ldots n_c} \left(\frac{d(c_i, c_j)}{\max_{k=1\ldots n_c}(diam(c_k))} \right) \right\}$$

$$D(C) = \frac{\min_{c_k \in C}\{\min_{c_l \in C, c_k}\{\delta(c_k, c_l)\}\}}{\max_{c_k \in C}\{\Delta(c_k)\}},$$

where

$$\delta(c_k, c_l) = \min_{x_i \in c_k} \min_{x_j \in c_l}\{d_e(x_i, x_j)\},$$

$$\Delta(c_k) = \max_{x_i, x_j \in c_k}\{d_e(x_i, x_j)\}.$$

$$diam(c_i) = \max_{x,y \in c_i}\{d(x,y)\} \qquad d(c_i, c_j) = \min_{x \in c_i, y \in c_j}\{d(x, y)\}$$

121

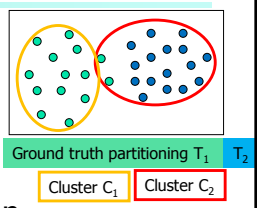
Measuring Clustering Quality: External Methods

- Clustering quality measure: *Q(C, T),* for a clustering *C* given the ground truth *T*
- *Q* is good if it satisfies the following **4** essential criteria
  - Cluster homogeneity: the purer, the better
  - Cluster completeness: should assign objects belong to the same category in the ground truth to the same cluster
  - Rag bag: putting a heterogeneous object into a pure cluster should be penalized more than putting it into a *rag bag* (i.e., "miscellaneous" or "other" category)
  - Small cluster preservation: splitting a small category into pieces is more harmful than splitting a large category into pieces

124

Some Commonly Used External Measures

- Matching-based measures
  - Purity, maximum matching, F-measure
- Entropy-Based Measures
  - Conditional entropy, normalized mutual information (NMI), variation of information
- Pair-wise measures
  - Four possibilities: True positive (TP), FN, FP, TN
  - Jaccard coefficient, Rand statistic, Fowlkes-Mallow measure
- Correlation measures
  - Discretized Huber static, normalized discretized Huber static



External evaluation

- Purity
- Rand measure
- F-measure
- Jaccard index
- Fowlkes–Mallows index
- Confusion matrix

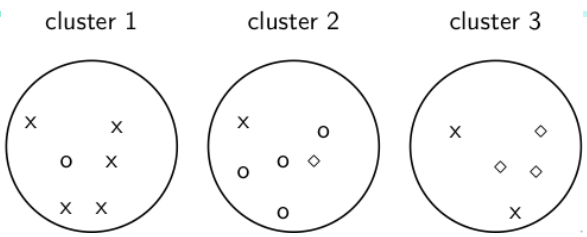
Purity

$$\text{purity}(\Omega, \mathbb{C}) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

- $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ is the set of clusters and
- $C = \{c_1, c_2, \ldots, c_J\}$ is the set of classes.
- For each cluster ω_k : find class c_j with most members n_{kj} in ω_k
- Sum all n_{kj} and divide by total number of points

127

Purity



(class x, cluster 1) $\rightarrow \max_j |\omega_1 \cap c_j| = 5$

(class o, cluster 2) $\rightarrow \max_j |\omega_2 \cap c_j| = 4$

(class ◇, cluster 3) $\rightarrow \max_j |\omega_3 \cap c_j| = 3$

Purity is

$(1/17) \times (5 + 4 + 3) \approx 0.71$

F-measure

- **Precision**: exactness – what % of tuples that the classifier labeled as positive are actually positive

$$precision = \frac{TP}{TP + FP}$$

- **Recall:** completeness – what % of positive tuples did the classifier label as positive?
- Perfect score is 1.0
- Inverse relationship between precision & recall

$$recall = \frac{TP}{TP + FN}$$

- ***F* measure (*F_1* or *F*-score)**: harmonic mean of precision and recall,

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

- ***F_β*:** weighted measure of precision and recall
  - assigns ß times as much weight to recall as to precision

$$F_\beta = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$$

129

Evaluation Metrics: Confusion Matrix

Confusion Matrix:

Actual class\Predicted class	C_1	¬ C_1
C_1	**True Positives (TP)**	**False Negatives (FN)**
¬ C_1	**False Positives (FP)**	**True Negatives (TN)**

Example of Confusion Matrix:

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	**6954**	**46**	7000
buy_computer = no	**412**	**2588**	3000
Total	7366	2634	10000

- Given *m* classes, an entry, ***$CM_{i,j}$*** in a **confusion matrix** indicates # of tuples in class *i* that were labeled by the classifier as class *j*
- May have extra rows/columns to provide totals

130

Evaluation Metrics:
Accuracy, Error Rate, Sensitivity and Specificity

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

- **Classifier Accuracy,** or recognition rate: percentage of test set tuples that are correctly classified

  Accuracy = (TP + TN)/All
- **Error rate:** *1 – accuracy*, or

  Error rate = (FP + FN)/All

- **Class Imbalance Problem**:
  - One class may be *rare*, e.g. fraud, or HIV-positive
  - Significant *majority of the negative class* and minority of the positive class
  - **Sensitivity**: True Positive recognition rate
    - **Sensitivity = TP/P**
  - **Specificity**: True Negative recognition rate
    - **Specificity = TN/N**

131

Confusion matrix: Example

Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	**90**	**210**	300	30.00 (*sensitivity*
cancer = no	**140**	**9560**	9700	98.56 (*specificity)*
Total	230	9770	10000	96.40 (*accuracy*)

- *Precision* = 90/230 = 39.13%
- *Recall* = 90/300 = 30.00%

132

Rand index

- Definition: $RI = \frac{TP+TN}{TP+FP+FN+TN}$

- TP is the number of true positives
- TN is the number of true negatives
- FP is the number of false positives
- FN is the number of false negatives

- TP+FN+FP+TN is the total number of pairs.

133

Jaccard index

- The Jaccard index is used to quantify the similarity between two datasets. The Jaccard index takes on a value between 0 and 1. An index of 1 means that the two dataset are identical, and an index of 0 indicates that the datasets have no common elements. The Jaccard index is defined by the following formula:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN}$$

- This is simply the number of unique elements common to both sets divided by the total number of unique elements in both sets.

134

Fowlkes–Mallows index

- The Fowlkes-Mallows index computes the similarity between the clusters returned by the clustering algorithm and the benchmark classifications. The higher the value of the Fowlkes-Mallows index the more similar the clusters and the benchmark classifications are. It can be computed using the following formula:

$$FM = \sqrt{\frac{TP}{TP + FP} \cdot \frac{TP}{TP + FN}}$$

- The index is the geometric mean of the precision and recall and , while the F-measure is their harmonic mean

135

Measures for Graph: Ratio Cut (↓) & Normalized Cut (↓)

$$\text{Ratio Cut}(\pi) = \frac{1}{k} \sum_{i=1}^{k} \frac{cut(C_i, \bar{C}_i)}{|C_i|},$$

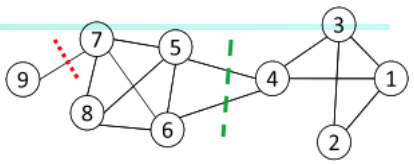
$$\text{Normalized Cut}(\pi) = \frac{1}{k} \sum_{i=1}^{k} \frac{cut(C_i, \bar{C}_i)}{vol(C_i)}$$

C_i : i^{th} community
$|C_i|$: number of nodes in C_i (size of community)
$vol(C_i)$: sum of degrees in C_i (volume of community)

- A good partitioning should minimize ratio cut and normalized cut

136

Ratio Cut & Normalized Cut Example



For partition in red: π_1

$$\text{Ratio Cut}(\pi_1) = \frac{1}{2}\left(\frac{1}{1} + \frac{1}{8}\right) = 9/16 = 0.56$$

$$\text{Normalized Cut}(\pi_1) = \frac{1}{2}\left(\frac{1}{1} + \frac{1}{27}\right) = 14/27 = 0.52$$

For partition in green: π_2

$$\text{Ratio Cut}(\pi_2) = \frac{1}{2}\left(\frac{2}{4} + \frac{2}{5}\right) = 9/20 = 0.45 < \text{Ratio Cut}(\pi_1)$$

$$\text{Normalized Cut}(\pi_2) = \frac{1}{2}\left(\frac{2}{12} + \frac{2}{16}\right) = 7/48 = 0.15 < \text{Normalized Cut}(\pi_1)$$

137

Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

138

Summary

- Cluster analysis groups objects based on their similarity and has wide applications
- Measure of similarity can be computed for various types of data
- Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods
- K-means and K-medoids algorithms are popular partitioning-based clustering algorithms
- Birch and Chameleon are interesting hierarchical clustering algorithms, and there are also probabilistic hierarchical clustering algorithms
- DBSCAN, OPTICS, and DENCLU are interesting density-based algorithms
- STING and CLIQUE are grid-based methods, where CLIQUE is also a subspace clustering algorithm
- Quality of clustering results can be evaluated in various ways

139

References (1)

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD'98
- M. R. Anderberg. Cluster Analysis for Applications. Academic Press, 1973.
- M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure, SIGMOD'99.
- Beil F., Ester M., Xu X.: "Frequent Term-Based Text Clustering", KDD'02
- M. M. Breunig, H.-P. Kriegel, R. Ng, J. Sander. LOF: Identifying Density-Based Local Outliers. SIGMOD 2000.
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. KDD'96.
- M. Ester, H.-P. Kriegel, and X. Xu. Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification. SSD'95.
- D. Fisher. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2:139-172, 1987.
- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. VLDB'98.
- V. Ganti, J. Gehrke, R. Ramakrishan. CACTUS Clustering Categorical Data Using Summaries. KDD'99.

141

References (2)

- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. In Proc. VLDB'98.
- S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. SIGMOD'98.
- S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. In *ICDE'99*, pp. 512-521, Sydney, Australia, March 1999.
- A. Hinneburg, D.l A. Keim: An Efficient Approach to Clustering in Large Multimedia Databases with Noise. KDD'98.
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Printice Hall, 1988.
- G. Karypis, E.-H. Han, and V. Kumar. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. *COMPUTER*, 32(8): 68-75, 1999.
- L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.
- E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. VLDB'98.

References (3)

- G. J. McLachlan and K.E. Bkasford. Mixture Models: Inference and Applications to Clustering. John Wiley and Sons, 1988.
- R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. VLDB'94.
- L. Parsons, E. Haque and H. Liu, Subspace Clustering for High Dimensional Data: A Review, SIGKDD Explorations, 6(1), June 2004
- E. Schikuta. Grid clustering: An efficient hierarchical clustering method for very large data sets. Proc. 1996 Int. Conf. on Pattern Recognition,.
- G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. VLDB'98.
- A. K. H. Tung, J. Han, L. V. S. Lakshmanan, and R. T. Ng. Constraint-Based Clustering in Large Databases, ICDT'01.
- A. K. H. Tung, J. Hou, and J. Han. Spatial Clustering in the Presence of Obstacles, ICDE'01
- H. Wang, W. Wang, J. Yang, and P.S. Yu. Clustering by pattern similarity in large data sets, SIGMOD' 02.
- W. Wang, Yang, R. Muntz, STING: A Statistical Information grid Approach to Spatial Data Mining, VLDB'97.
- T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH : An efficient data clustering method for very large databases. SIGMOD'96.
- Xiaoxin Yin, Jiawei Han, and Philip Yu, "LinkClus: Efficient Clustering via Heterogeneous Semantic Links", in Proc. 2006 Int. Conf. on Very Large Data Bases (VLDB'06), Seoul, Korea, Sept. 2006.