Clustering

Jay Urbain, PhD

Credits: Tom Mitchell, Jiawei Han, Micheline Kamber, and Jian Pei

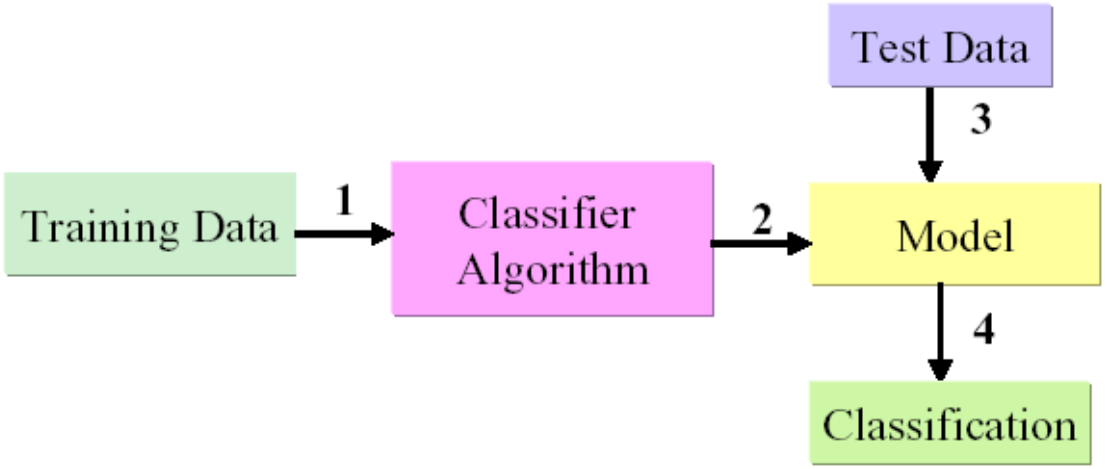
Cluster Analysis: Basic Concepts and Methods

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

Supervised Learning

Supervised Learning

- Learn by example from training data with a class label
- Create model by running algorithm on training data
- Identify a class label for the incoming new data



Supervised Learning Algorithms

Supervised Learning Algorithms

- Linear Regression
- Logistic Regression
- Naive Bayes
- Bayes Nets
- Neural Networks
- Decision Trees
- Support Vector Machine

Unsupervised Learning Algorithms

There are many machine learning situations in which class labels are not available, so *unsupervised* methods are needed.

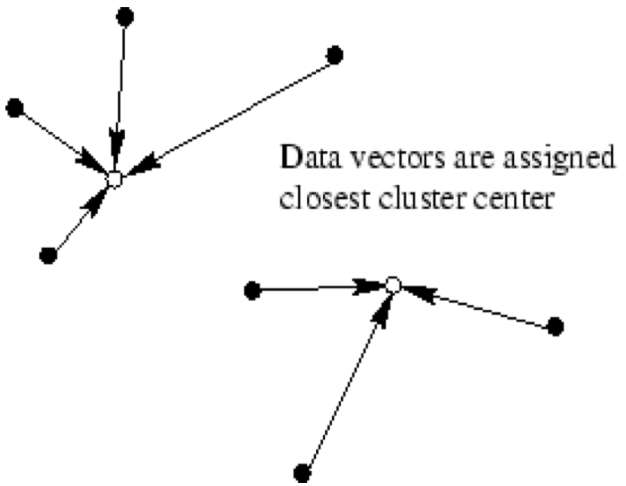
Unsupervised Learning Algorithms

- Clustering – *many*
- Association Rules - *Apriori*
- PCA – Principal Component Analysis
- Topic Models
- Collaborative Filtering

What is Cluster Analysis?

- **Cluster: A collection of data objects**
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups

- **Cluster analysis (or *clustering, data segmentation, …*)**
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters



What is Cluster Analysis?

- Unsupervised learning: no predefined classes (i.e., *learning by observations* vs. learning by examples: supervised)
- Typical applications
  - As a stand-alone tool to get insight into data distribution, data exploration
  - As a preprocessing step for other algorithms

Applications of Cluster Analysis

- Data reduction:
  - Summarization: Preprocessing for regression, PCA, classification, and association analysis
  - Compression: Image processing: vector quantization
- Hypothesis generation and testing
- Prediction based on groups:
  - Cluster & find characteristics/patterns for each group – define as class
- Finding K-nearest neighbors:
  - Localizing search to one or a small number of clusters
- Outlier detection: Outliers are often viewed as those "far away" from any cluster

Clustering: Application Examples

- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults
- Climate: understanding earth climate, find patterns of atmospheric and ocean
- Economic Science: market research

Basic Steps to Develop a Clustering Task

1. Feature selection
   - Select info concerning the task of interest
   - Minimal information redundancy
2. Proximity measure
   - Similarity of two feature vectors, or feature distributions
3. Clustering criterion
   - Expressed via a cost function, maximum likelihood estimate, or rules
4. Clustering algorithms
   - Choice of algorithms
5. Validation of the results
   - Validation test (also, *clustering tendency* test)
6. Interpretation of the results
   - Integration with applications, visualization

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters:
  - high *intra-class* similarity: cohesive within clusters
  - low *inter-class* similarity: distinctive between clusters

- The quality of a clustering method depends on:
  - the similarity measure used by the method
  - its implementation, and
  - its ability to discover some or all of the hidden patterns

How to improve the quality of clusters?

Measure the Quality of Clustering

- Dissimilarity/Similarity metric
  - Similarity is expressed in terms of a distance function, typically metric: $d(i, j)$
  - The definitions of distance functions are usually rather different for Boolean, categorical, ordinal, discrete, & continuous variables
  - Weights should be associated with different variables based on applications and data semantics

- Quality of clustering:
  - There is usually a separate "quality" function that measures the "goodness" of a cluster.
  - Its hard to define "similar enough" or "good enough"
    - The answer is typically highly subjective
    - Open research question

Considerations for Cluster Analysis

- Partitioning criteria
  - Single level vs. hierarchical partitioning (often, multi-level hierarchical partitioning is desirable)
- Separation of clusters
  - Exclusive/hard (one customer belongs to only one region, e.g., k-means) vs. non-exclusive/soft (one document may belong to more than one class: e.g., Expectation Maximization clustering)
- Similarity measure
  - Distance-based (e.g., Manhattan (L1), Euclidian (L2), Cosine); connectivity-based (e.g., density or contiguity, network); and maximum likelihood.
- Clustering space
  - Full space (often when low dimensional) vs. subspaces (often in high-dimensional clustering)

Requirements and Challenges

- Scalability
  - Clustering all the data vs. samples
- Ability to deal with different types of attributes
  - Numerical, binary, categorical, ordinal, linked, and combinations
- Constraint-based clustering
  - User may give inputs on constraints
  - Use domain knowledge to determine input parameters
- Interpretability/visualization and usability
- Others
  - Discovery of clusters with arbitrary shape
  - Ability to deal with noisy data
  - Incremental clustering and insensitivity to input order
  - High dimensionality

Major Clustering Approaches (I)

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of squared errors
  - Typical (hard) methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue
- Grid-based approach:
  - Based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE

Major Clustering Approaches (II)

- Model-based:
  - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
  - Typical methods: EM, Topic, SOM, COBWEB
- Frequent pattern-based:
  - Based on the analysis of frequent patterns
  - Typical methods: p-Cluster
- User-guided or constraint-based:
  - Clustering by considering user-specified or application-specific constraints
  - Typical methods: COD (obstacles), constrained clustering
- Link-based (network) clustering:
  - Objects are often linked together in various ways
  - Massive links can be used to cluster objects: SimRank, LinkClus

Cluster Analysis: Basic Concepts and Methods

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

Partitioning Algorithms: Basic Concept

- Partitioning method: Partitioning a database ***D*** of ***n*** objects into a set of ***k*** clusters, such that the *sum of squared distances is minimized* (where c_i is the of cluster C_i .) – represented by *centroid* or *medoid*, etc.

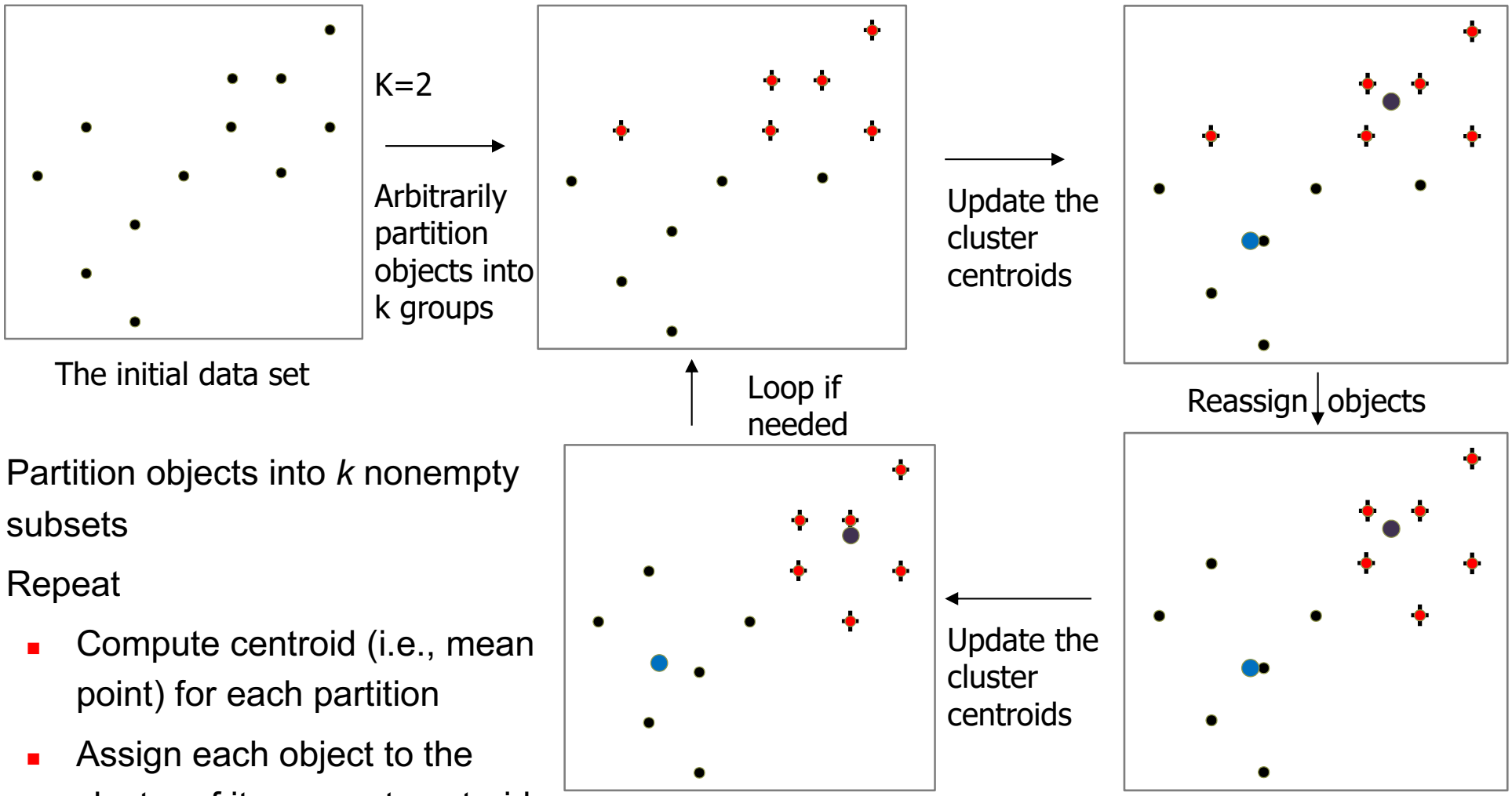
$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (d(p, c_i))^2$$

- Given ***k***, find a partition of ***k*** *clusters* that optimizes the chosen partitioning criterion:
  - Global optimal: exhaustively enumerate all partitions – *how large?*
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67, Lloyd'57/'82): Each cluster is represented by the center of the cluster (centroid, average; weak, strong linkage)
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

The *K-Means* Clustering Method

- Given *k*, the *k-means* algorithm is implemented in four steps:
  1. Partition objects into *k* non-empty subsets (randomly)
  2. Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., *mean point*, of the cluster)
  3. Assign each object to the cluster with the nearest seed point
  4. Go back to Step 2, stop when the assignment does not change (convergence criteria)

An Example of *K-Means* Clustering



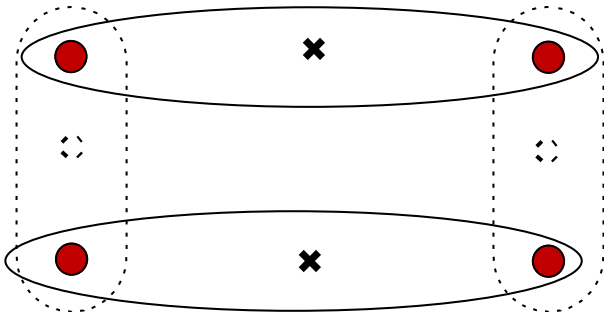
- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

Comments on the *K-Means* Method

- Strength: *Efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t << n$.
- Comment: Often terminates at a *local optimum*
- Weakness
  - Applicable only to objects in a continuous (convex) n-dimensional space
    - Use the *k-modes* method for categorical data or convert categorical data types to numerical
    - In comparison, *k-medoids* can be applied to a wide range of data
  - Need to specify *k,* the *number* of clusters, in advance (there are ways to automatically determine the best k (see Hastie et al., 2009)
  - Sensitive to noisy data and *outliers*
  - Sub-optimal clusters with *non-convex shapes (stuck in local minima)*

Variations of the *K-Means* Method

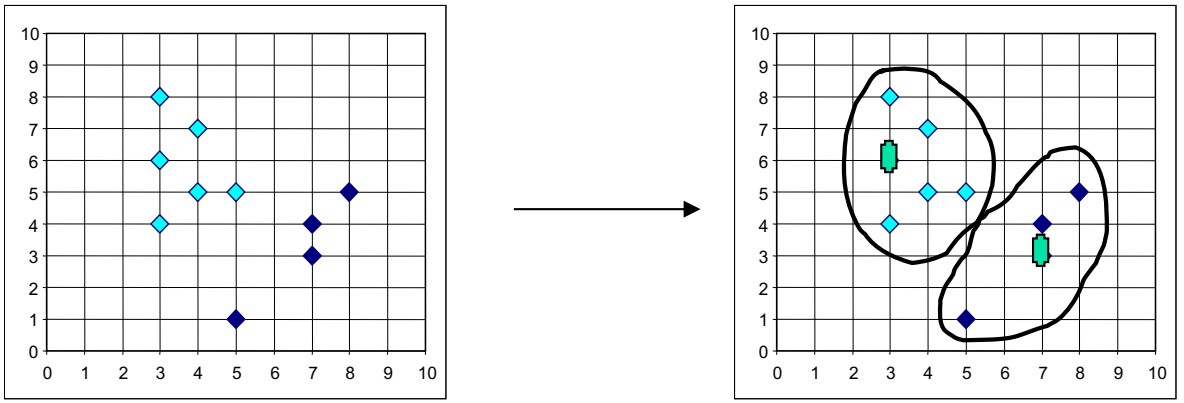
- Most of the variants of the *k-means* differ in:
  - Selection of the initial ***k***
  - Dissimilarity calculations
  - Strategies to calculate cluster means



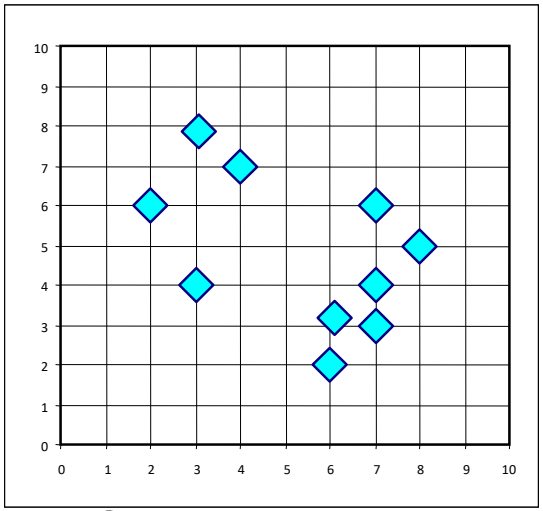
- Handling categorical data: *k-modes*
  - Replacing means of clusters with modes (categorical feature counts)
  - Use new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method

What Is the Problem of the K-Means Method?

- The k-means algorithm is sensitive to outliers!
  - Since an object with an extremely large value may substantially distort the distribution of the data
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** (*most representative*) object in a cluster

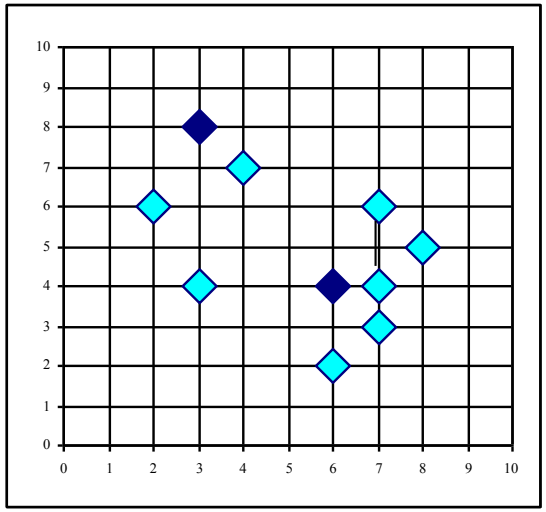


PAM: A Typical K-Medoids Algorithm
CLARA: sample



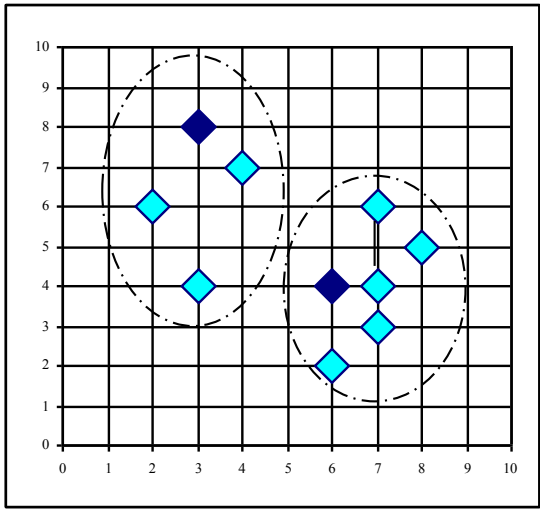
K=2

Arbitrarily choose k objects as initial *medoids*

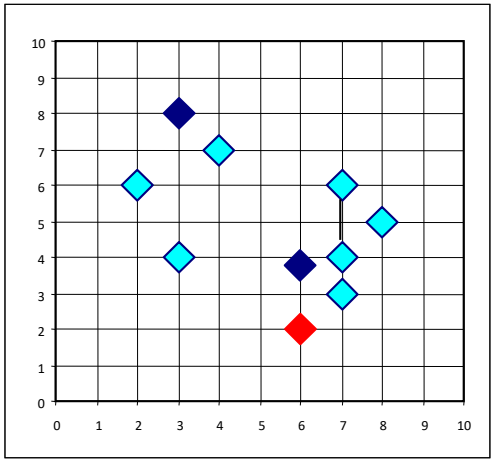


Assign each remaining object to nearest medoid

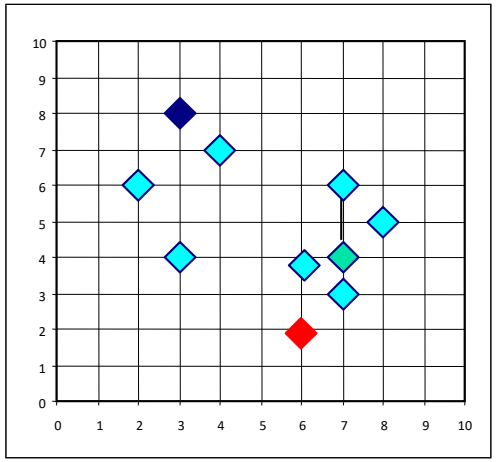
Total Cost = 20



Select/sample a nonmedoid object, O_{ramdom}



Compute total cost of swapping (using nonmedoid as new medoid)



Total Cost = 26

Swapping O and O_{ramdom}

If quality is improved.

Do loop

Until no change

The K-Medoid Clustering Method

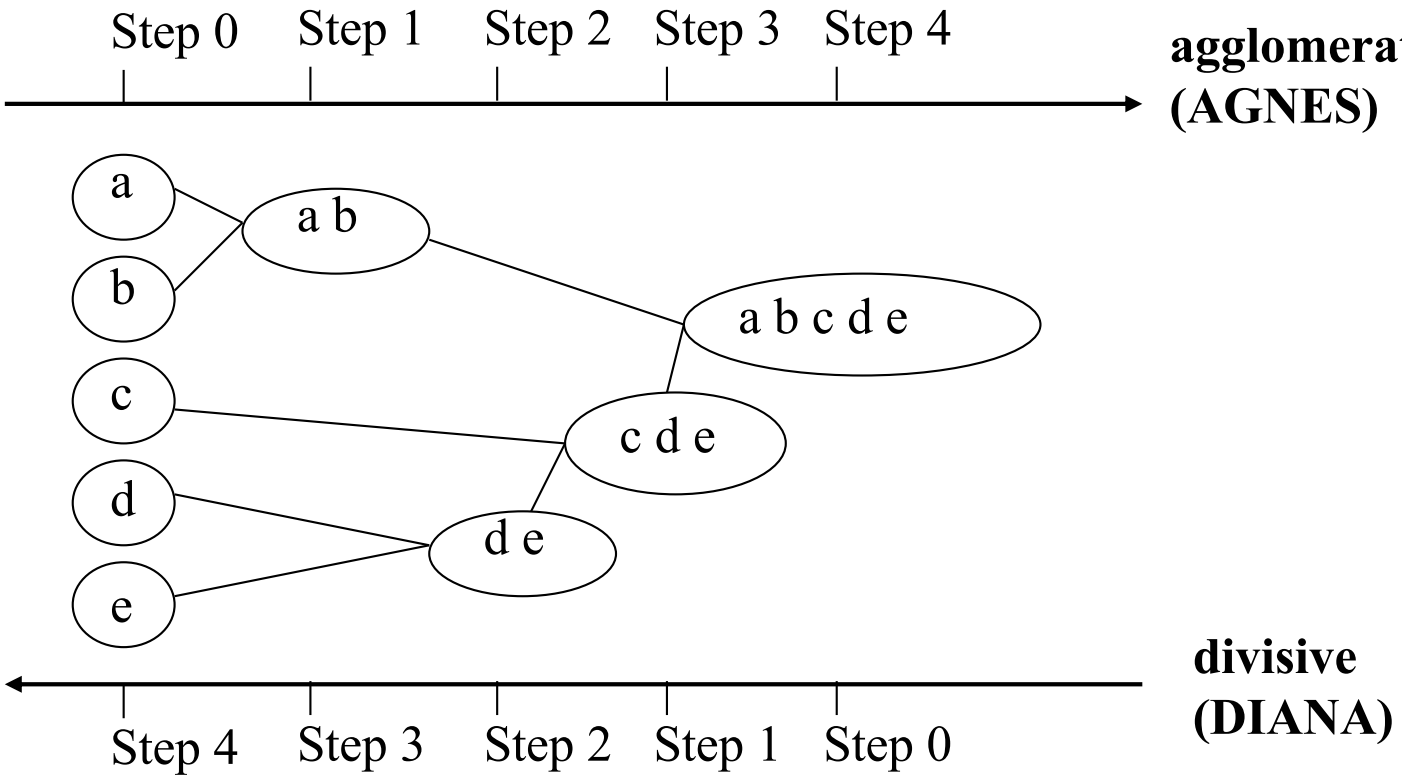
- *K-Medoids* Clustering: Find *representative* objects (medoids) in clusters
  - *PAM* (Partitioning Around Medoids, Kaufmann & Rousseeuw 1987)
    - Starts from an initial set of *medoids* and iteratively replaces one of the *medoids* by one of the non-medoids if it improves the total distance of the resulting clustering
    - *PAM* works effectively for small data sets, but does not scale well for large data sets (due to the computational complexity)
- Efficiency improvement on PAM
  - *CLARA* (Kaufmann & Rousseeuw, 1990): PAM on samples
  - *CLARANS* (Ng & Han, 1994): Randomized re-sampling

Cluster Analysis: Basic Concepts and Methods

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

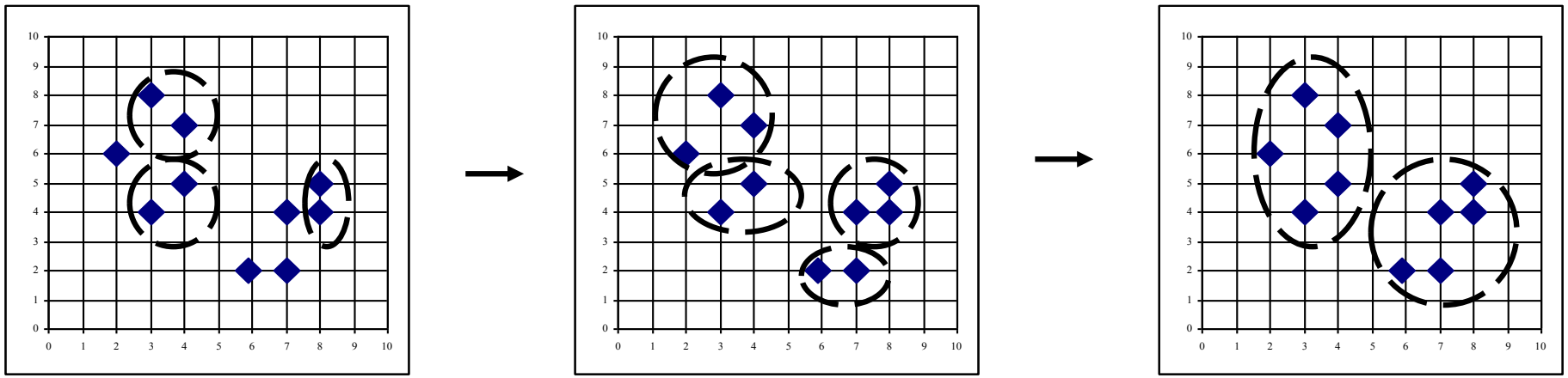
Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input.



AGNES (Agglomerative Nesting)

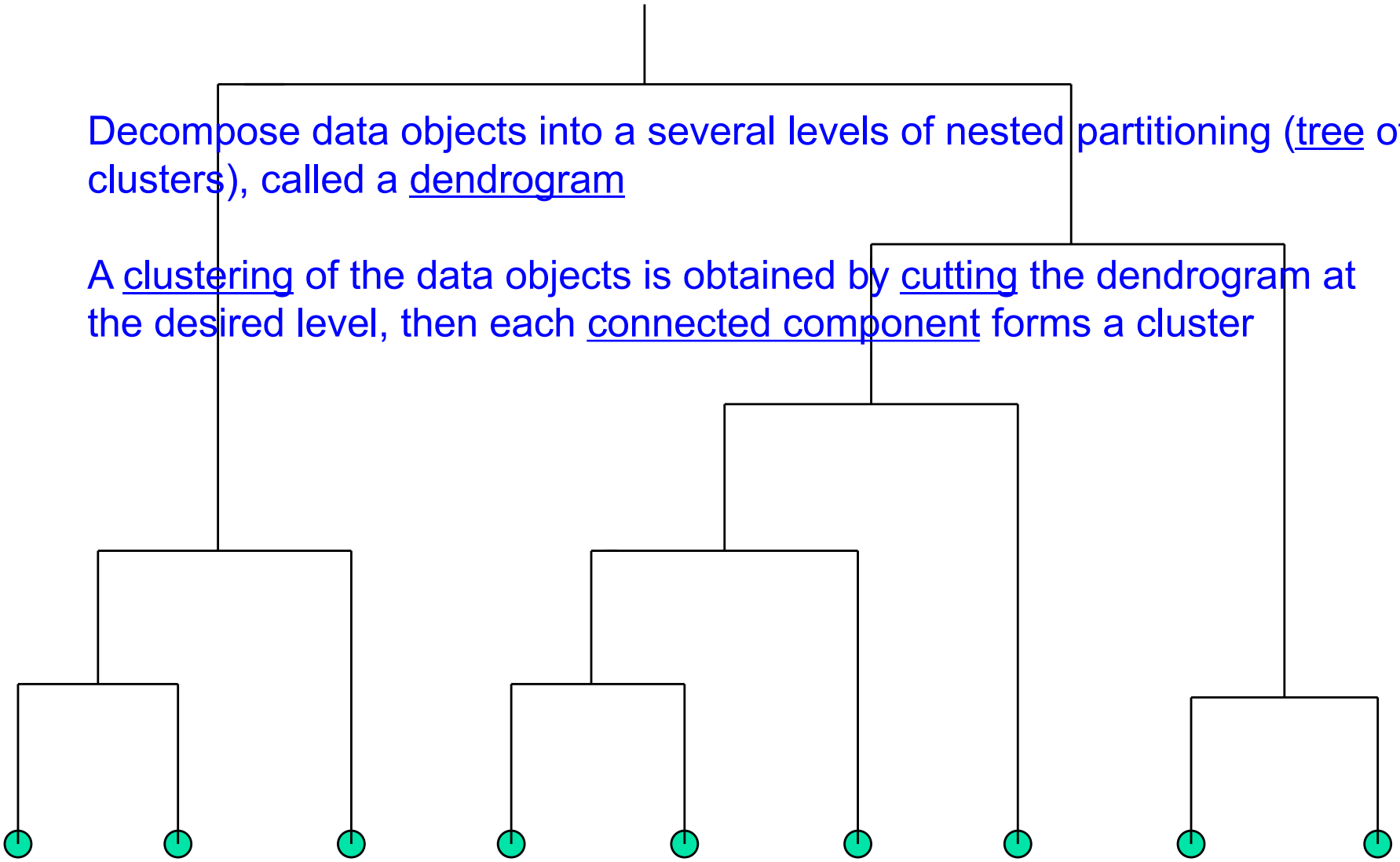
- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical packages, e.g., Splus
- Uses dissimilarity matrix
- Merge nodes that have the *least dissimilarity/max similarity*
- Eventually all nodes belong to the same cluster



Dendrogram: Shows How Clusters are Merged

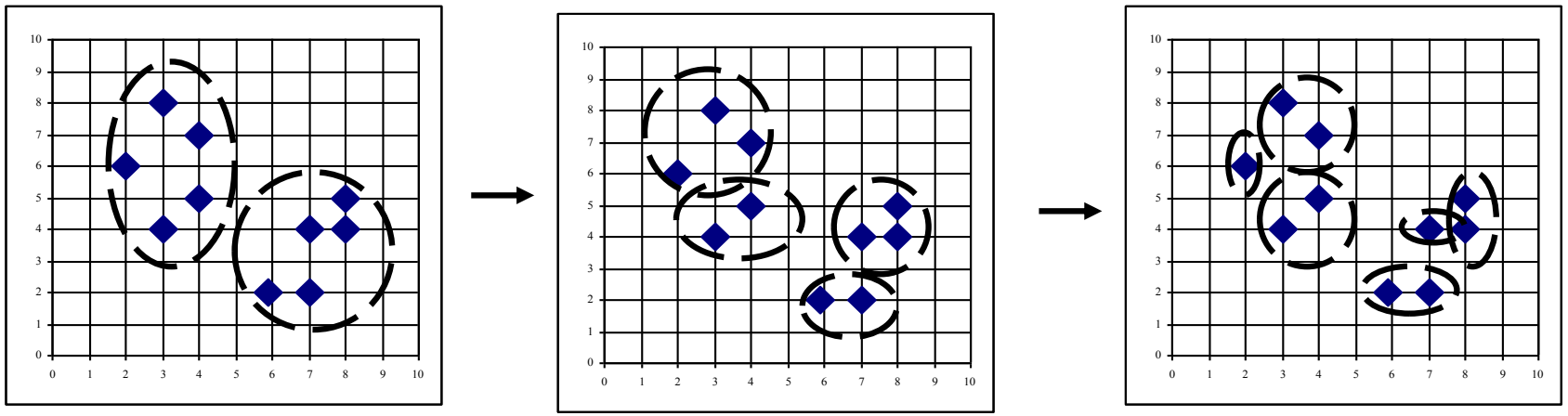
Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster

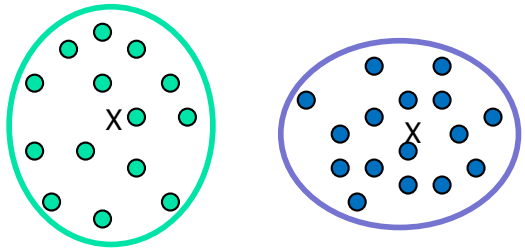


DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



Distance between Clusters



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

Centroid, Radius and Diameter of a Cluster (for numerical data sets)

- Centroid: the "middle" of a cluster

$$C_m = \frac{\sum_{i=1}^{N} (t_{ip})}{N}$$

- Radius: square root of average distance from any point of the cluster to its centroid

$$R_m = \sqrt{\frac{\sum_{i=1}^{N} (t_{ip} - c_m)^2}{N}}$$

- Diameter: square root of average mean squared distance between all pairs of points in the cluster

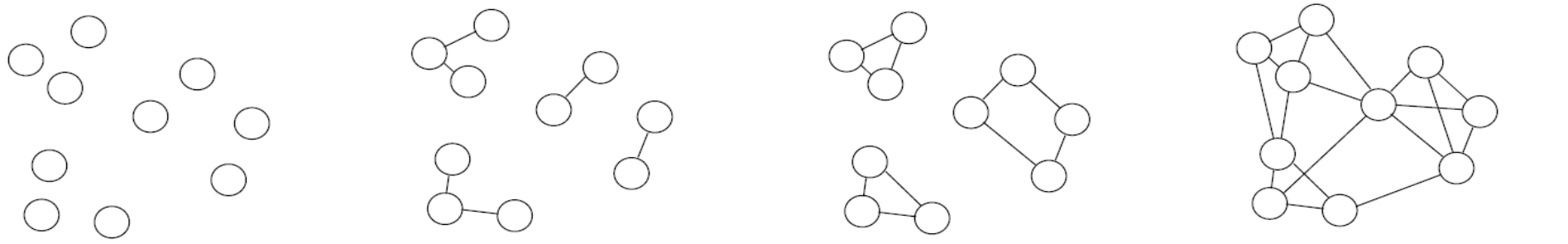
$$D_m = \sqrt{\frac{\sum_{i=1}^{N} \sum_{i=1}^{N} (t_{ip} - t_{iq})^2}{N(N-1)}}$$

Extensions to Hierarchical Clustering

- Major weakness of agglomerative clustering methods
  - Can never undo what was done previously
  - Does not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
- Integration of hierarchical & distance-based clustering
  - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
  - CHAMELEON (1999): hierarchical clustering using dynamic modeling

KNN Graphs & Interconnectivity

- k-nearest graphs from an original data in 2D:



(a) Original Data in 2D (b) 1-nearest neighbor graph (c) 2-nearest neighbor graph (d) 3-nearest neighbor graph

- $EC_{\{Ci,Cj\}}$:The absolute inter-connectivity between C_i *and* C_j*: the sum* of the weight of the edges that connect vertices in C_i *to vertices in* C_j
- Internal inter-connectivity of a cluster C_i : *the size of its min-cut bisector* EC_{Ci} *(i.e., the weighted sum* of edges that partition the graph into two roughly equal parts)
- Relative Inter-connectivity (RI):

$$RI(C_i, C_j) = \frac{|EC_{\{C_i,C_j\}}|}{\frac{|EC_{C_i}|+|EC_{C_j}|}{2}}$$

Relative Closeness & Merge of Sub-Clusters

- **Relative closeness** between a pair of clusters C_i *and* C_j *: the absolute closeness* between C_i *and* C_j *normalized w.r.t. the internal closeness of the two clusters* C_i *and* C_j

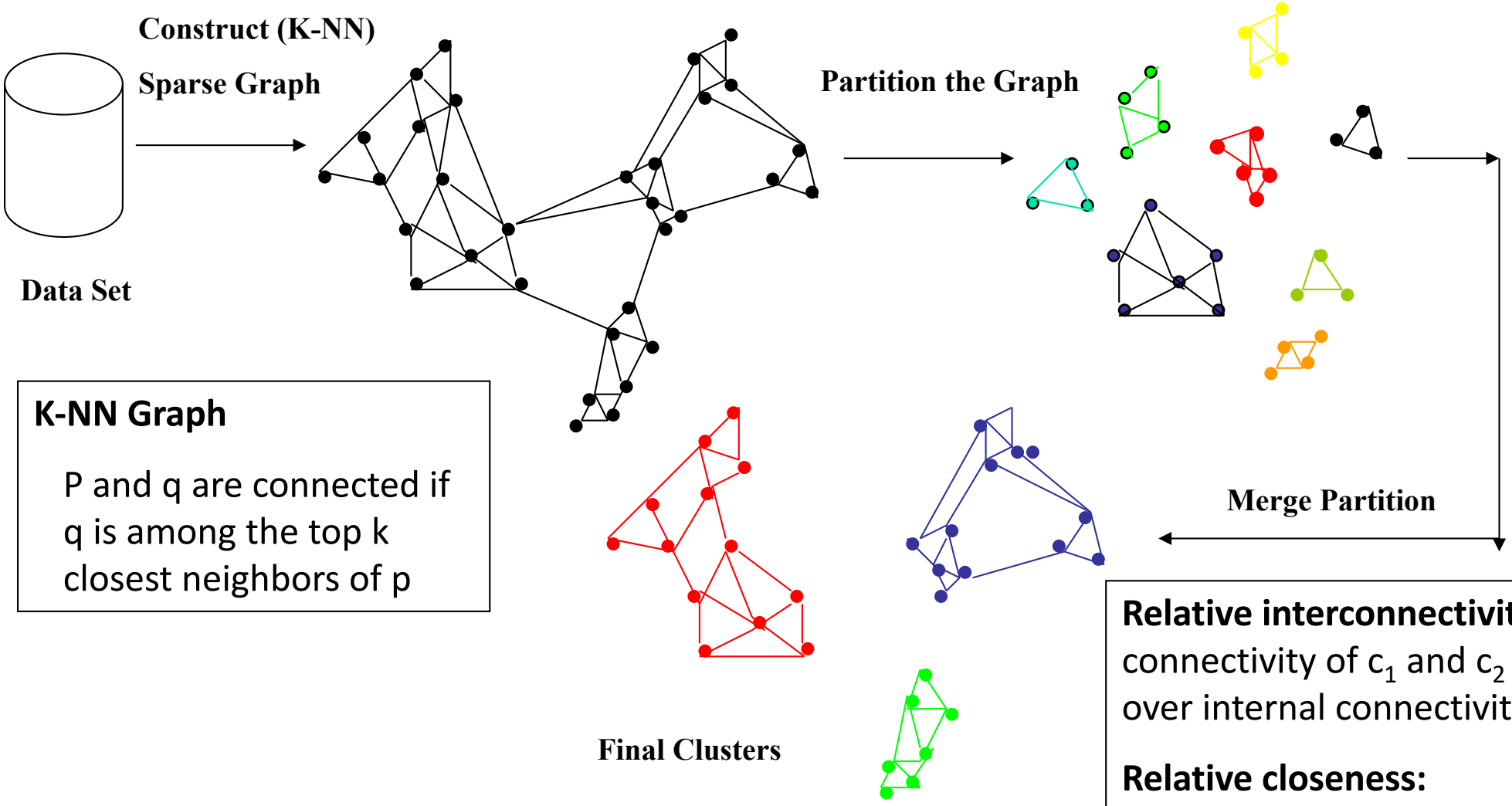
$$RC(C_i, C_j) = \frac{\overline{S}_{EC_{\{C_i, C_j\}}}}{\frac{|C_i|}{|C_i|+|C_j|}\overline{S}_{EC_{C_i}} + \frac{|C_j|}{|C_i|+|C_j|}\overline{S}_{EC_{C_j}}}$$

  - $\overline{S}_{EC_{C_i}}$ and $\overline{S}_{EC_{C_j}}$ are the average weights of the edges that belong in the min-cut bisector of clusters C_i *and* C_j *,* respectively, and $\overline{S}_{EC_{\{C_i, C_j\}}}$ is the average weight of the edges that connect vertices in C_i *to vertices in* C_j

- **Merge Sub-Clusters:**
  - Merges only those pairs of clusters whose RI and RC are both above some user-specified thresholds
  - Merge those maximizing the function that combines RI and RC

C

Overall Framework of CHAMELEON



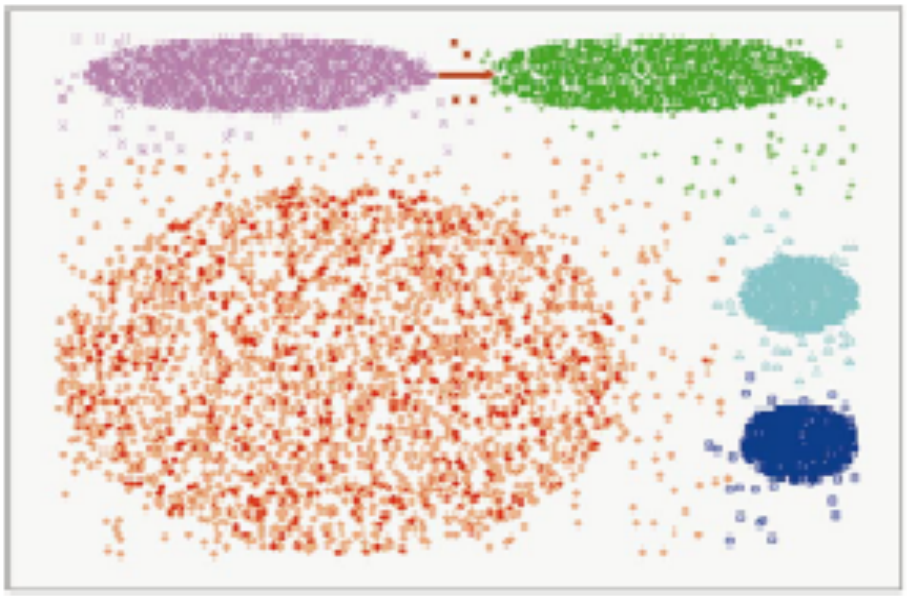
K-NN Graph

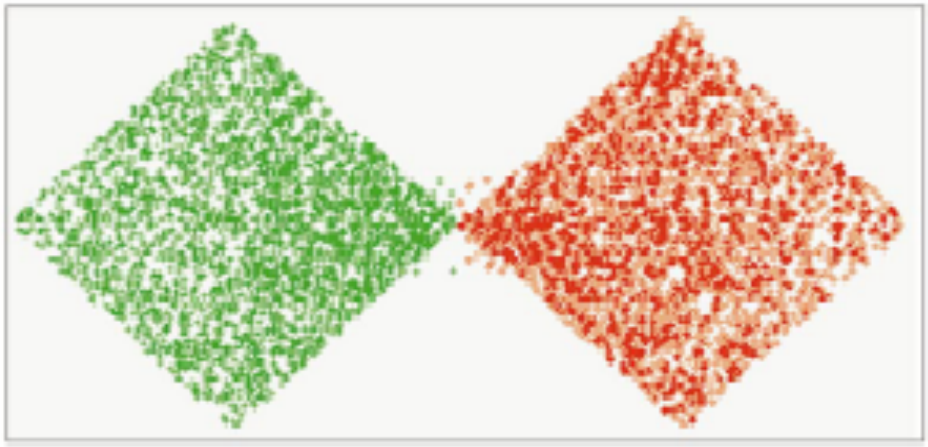
P and q are connected if q is among the top k closest neighbors of p

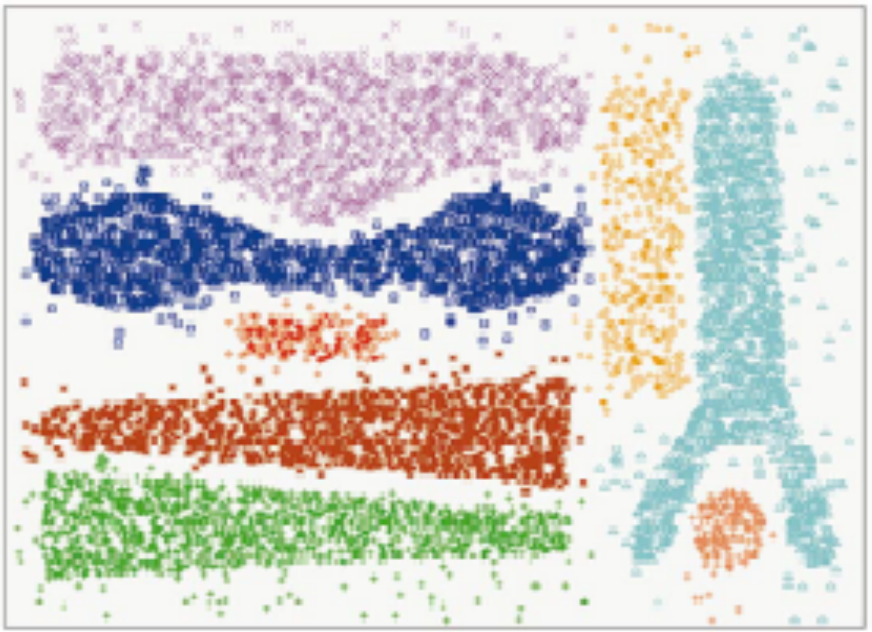
Relative interconnectivity: connectivity of c_1 and c_2 over internal connectivity

Relative closeness: closeness of c_1 and c_2 over internal closeness

CHAMELEON (Clustering Complex Objects)







Concept-Concept Co-occurence

```
SELECT ci.term, ci1.term, count(*) n
FROM conceptinvertedindex ci, conceptpostinglist cp,
conceptinvertedindex ci1, conceptpostinglist cp1
where ci.termid=cp.termid
and ci.term = 'United States'
and ci1.termid=cp1.termid
and cp.docid=cp1.docid
and cp.parid=cp1.parid
and cp.sentid=cp1.sentid
group by ci.term, ci1.term
order by n desc;
```

term	term	n
United States	United States	110
United States	North Korea	47
United States	Iran	32
United States	China	31
United States	US	28
United States	Thai	22
United States	Middle East	16
United States	Pyongyang	12
United States	PRC	11
United States	Thailand	9
United States	Obama	8
United States	DPRK	8
United States	Israel	8
United States	State	8
United States	Beijing	7
United States	Japan	5
United States	Bosworth	5
United States	Bangkok	5
United States	IAEA	5
United States	UNSC	5
United States	Russia	5
United States	South Asia	4

Concept-Term Co-occurence

```
SELECT ci.term, ci1.term, count(*) n
FROM conceptinvertedindex ci, conceptpostinglist cp,
invertedindex ci1, postinglist cp1
where ci.termid=cp.termid
and ci.term = 'United States'
and ci1.termid=cp1.termid
and cp.docid=cp1.docid
and cp.parid=cp1.parid
and cp.sentid=cp1.sentid
group by ci.term, ci1.term
order by n desc;
```

term	term	n
United States	state	126
United States	unit	112
United States	korea	57
United States	north	53
United States	nuclear	47
United States	weapon	40
United States	thai	37
United States	china	31
United States	iran	27
United States	tip	24
United States	author	23
United States	said	23
United States	illicit	21
United States	concern	20

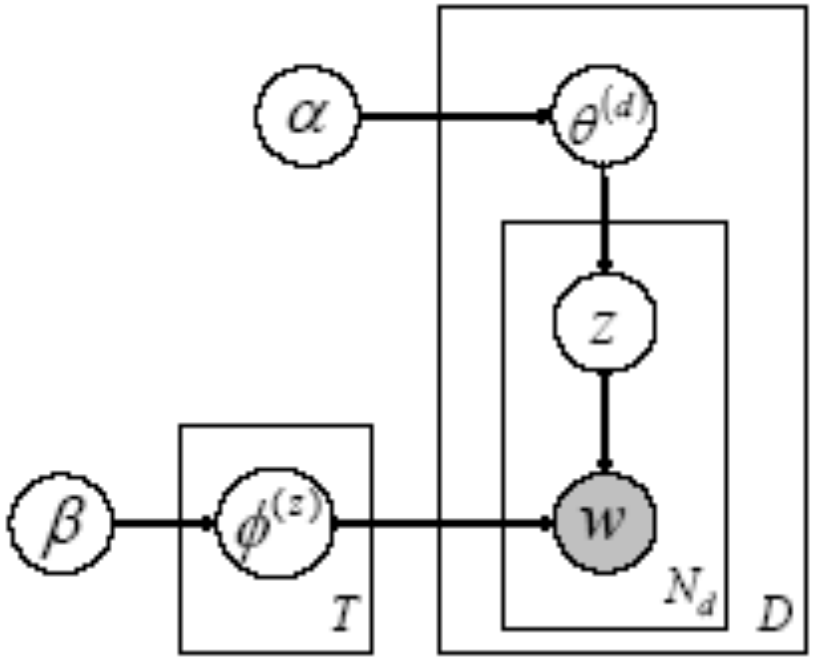
Topic Model

$f(Q, D, \theta_d)$

- Generative method - specifies how a document is generated.

1. Define document distribution over topics.

2. Select each word in the document by sampling a word from this topic distribution at random.



$$P(w_i) = \sum_{j=1}^{T} P(w_i | z_i = j) P(z_i = j)$$

Topic Model

The Gibbs sampling -

1. Consider each word token in the text collection in turn
2. Estimate probability of assigning the current word token to each topic, conditioned on the topic assignments to all other word tokens.
3. From this conditional distribution, a topic is sampled and stored as the new topic assignment for this word token.

$$P(z_i = j \mid z_{-i}, w_i, d_i, .) \approx \frac{C^{WT}_{w_i,j} + \beta}{\sum_{w=1}^{W} C^{WT}_{w,j} + W\beta} \frac{C^{DT}_{d,j} + \alpha}{\sum_{t=1}^{T} C^{DT}_{d,t} + T\alpha}$$

Topic Model

- Estimate sampling a new word instance w_i from topic j:

$$\phi^{(j)} = \frac{C_{wi,j}^{WT} + \beta}{\sum_{w=1}^{W} C_{w,j}^{WT} + W\beta}$$

- Sampling a new word in document d from topic j:

$$\theta^{(d)} = \frac{C_{d,j}^{DT} + \alpha}{\sum_{t=1}^{T} C_{d,t}^{DT} + T\alpha}$$

Topic Models Related to Events

Identify topics related to event ‘attack’

- Note on topic model parameterization: 50 topics, alpha=5 (large), phi=0.5 (large), iterations=100 (very small, should be higher for convergence – just a test), burn-in=10, sample lag=2, thin interval=1.

// Identify topics highly likely to generate terms affiliated with ‘*attack*’

```
SELECT i.term, i.termid, p.topicid, p.phi
FROM _phi p, invertedindex i, _nw n
where p.termid=i.termid
and n.termid=i.termid
and i.term='attack'
and p.topicid=n.topicid
order by p.phi desc
limit 20;
```

Topic Models Related to Attack

term	termid	topicid	phi
attack	**71**	**18**	**0.0135349**
attack	71	33	0.0026009
attack	71	15	0.0023404
attack	71	20	0.0023344
attack	71	19	0.0011068
attack	71	40	0.0003311
attack	71	42	0.0002832
attack	71	22	0.0002763
attack	71	43	0.0002721
attack	71	26	0.0002527
attack	71	7	0.0002282
attack	71	2	0.0001683
attack	71	13	0.0001455
attack	71	31	0.0001453
attack	71	49	0.0001335
attack	71	14	0.0001308
attack	71	29	0.0001068
attack	71	28	0.0001026
attack	71	38	0.0001
attack	71	10	0.0000952

Topic Models Related to Attack

Clearly, topic 18 has a very high likelihood generating the word '*attack*'.

What does topic 18 look like?

// list the probabilities of words being generated by *topic 18* in descending order:

```
SELECT i.term, p.phi
FROM _phi p, invertedindex i
where p.termid=i.termid
and p.topicid=18
order by p.phi desc
limit 20;
```

Topic Models Related to Attack

These words compare favorably with the Serif type=conflict, subtype=attack instance words including: attack, operation, bombing, straddling, violent, fought, activities, war, terrorism, projects, attacking.

Other words that are highly likely in topic 18 above are included in other entity types, e.g., bin laden and american are *persons* affiliated with an *attack* event, al qaeda is a *non-governmental organizations* associated with an *attack* event.

term	phi
in	0.067238
al	0.028876
bin	0.019418
attack	0.013535
laden	0.012561
bomb	0.010009
terrorist	0.009661
said	0.009242
qaeda	0.008696
offici	0.008154
oper	0.007893
intellig	0.006949
group	0.006148
arrest	0.005611
state	0.005553
american	0.005158
after	0.005094
not	0.005001
secur	0.004873
unit	0.00466

Topic Modeling Parameters

Parameter	Value
T – number of topics	100
α- Dirichlet hyperparameter for θ	0.5
β- Dirichlet hyperparameter for ϕ	0.02
# of sampling iterations (assume convergence)	200
Sample lag	20
Sample thinning (prevent autocorrelation)	2 (save every other sample)

Topics extracted from TREC 2005 Genomics MEDLINE collection

Topic 24: ***Cancer***	**Topic 1:** ***Cell Death***	**Topic 158:** ***Antibody***	**Topic 103:** ***Transcription***
tumor (0.0697)	apoptosi (0.0596)	antibodi (0.07521)	transcript (0.0705)
neoplasm (0.0402)	protein (0.0553)	m (0.0462)	bind (0.0656)
carcinoma(0.0367)	cell (0.0474)	IG (0.0442)	protein (0.0511)
pathology (0.0234)	oncogen (0.0303)	g (0.0440)	factor (0.0486)
NM (0.0222)	BCL (0.0256)	immunolog (0.0334)	nuclear (0.0373)
NM23 (0.0197)	proto (0.0225)	immunoglobulin (0.0317)	promot (0.0342)
cancer (0.0180)	protooncogen(0.0202)	anti (0.0266)	activ (0.0243)
express (0.0142)	BCL2 (0.01990)	monoclon (0.01942)	metabol (0.0240)
malign (0.0125)	death (0.0194)	Ig-m (0.0182)	DNA (0.0212)
tumour (0.0121)	express (0.0138)	assai (0.0147)	DNA-bind (0.0210)
H (0.0113)	wild (0.0104)	Ig-g (0.0144)	regul (0.0147)
surviv (0.0101)	apoptot (0.0104)	antigen (0.0139)	site (0.0145)
stage (0.0095)	mediat (0.0102)	link (0.0119)	genet (0.0141)
metastasi (0.0087)	caspas (0.0090)	epitop (0.0076)	element (0.0125)7
progress (0.0082)	Bax (0.0084)	immunosorb (0.0075)	respons (0.0111)
suppressor(0.0080)	induct (0.0083)	elisa (0.0064)	gene (0.0103)
23H (0.0080)	myc (0.0082)	serum (0.0062)	HNF (0.0097)1
NM23H (0.0080)	regul (0.0079)	blood (0.0061)	transactiv (0.0084)
correl (0.0079)	tumor (0.0073)	sera (0.0059)	regulatori (0.0079)
primary (0.0074)	inhibitor (0.0071)	AB (0.0051)	HFN4 (0.0070)

Expanding Topic Models

- Topic models help with broader view of collection content.
- Most useful for unexplored collections.
- Idea behind topic models can be expanded to include structured and unstructured information including entities and terms at multiple levels of aggregation.
- Scalability is limited.
- Run apriori.

Probabilistic Hierarchical Clustering

- Algorithmic hierarchical clustering
  - Nontrivial to choose a good distance measure
  - Hard to handle missing attribute values
  - Optimization goal not clear: heuristic, local search
- Probabilistic hierarchical clustering
  - Use probabilistic models to measure distances between clusters
  - *Generative model: Regard the set of data objects to be clustered as a sample of the underlying data generation mechanism to be analyzed*
  - Easy to understand, same efficiency as algorithmic agglomerative clustering method, can handle partially observed data
- In practice, assume the generative models adopt common distributions functions, e.g., Gaussian distribution or Bernoulli distribution, governed by parameters

Generative Model

- Given a set of 1-D points $X = \{x_1, \ldots, x_n\}$ for clustering analysis & assuming they are generated by a Gaussian distribution:

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- The probability that a point $x_i \in X$ is generated by the model

$$P(x_i|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

- The likelihood that X is generated by the model:

$$L(\mathcal{N}(\mu, \sigma^2) : X) = P(X|\mu, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

- The task of learning the generative model: find the parameters μ and σ^2 such that

$$\mathcal{N}(\mu_0, \sigma_0^2) = \arg\max\{L(\mathcal{N}(\mu, \sigma^2) : X)\}$$

the maximum likelihood

A Probabilistic Hierarchical Clustering Algorithm

- For a set of objects partitioned into *m* clusters $C_1, \ldots, C_m$, the quality can be measured by,

$$Q(\{C_1, \ldots, C_m\}) = \prod_{i=1}^{m} P(C_i)$$

  where *P*() is the maximum likelihood

- If we merge two clusters C_{j1} and C_{j2} into a cluster $C_{j1} \cup C_{j2}$, then, the change in quality of the overall clustering is

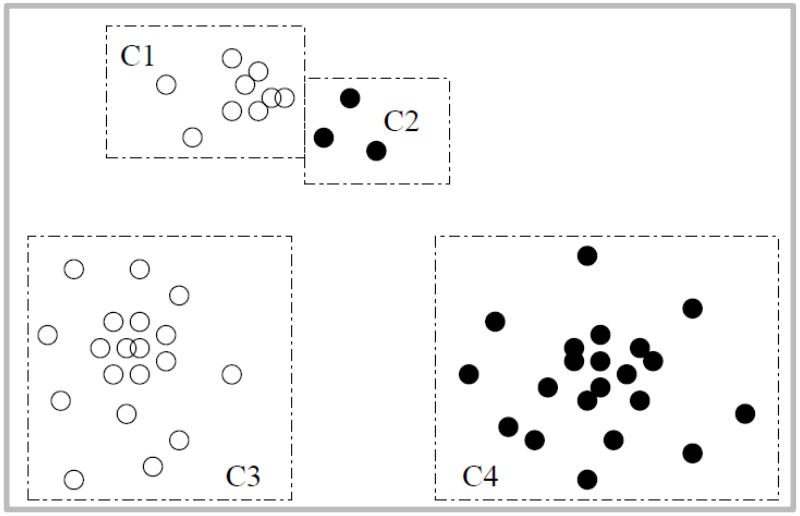
$$Q((\{C_1, \ldots, C_m\} - \{C_{j_1}, C_{j_2}\}) \cup \{C_{j_1} \cup C_{j_2}\}) - Q(\{C_1, \ldots, C_m\})$$

$$= \frac{\prod_{i=1}^{m} P(C_i) \cdot P(C_{j_1} \cup C_{j_2})}{P(C_{j_1}) P(C_{j_2})} - \prod_{i=1}^{m} P(C_i)$$

$$= \prod_{i=1}^{m} P(C_i) \left(\frac{P(C_{j_1} \cup C_{j_2})}{P(C_{j_1}) P(C_{j_2})} - 1 \right)$$

- Distance between clusters C_1 and C_2:

$$dist(C_i, C_j) = -\log \frac{P(C_1 \cup C_2)}{P(C_1) P(C_2)}$$



Cluster Analysis: Basic Concepts and Methods

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

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
- Test spatial randomness by statistic test: Hopkins Static
  - Given a dataset D regarded as a sample of a random variable o, determine how far away o is from being uniformly distributed in the data space
  - Sample n points, $p_1, \ldots, p_n$, uniformly from D. For each p_i, find its nearest neighbor in D: $x_i = min\{dist(p_i, v)\}$ where v in D
  - Sample n points, $q_1, \ldots, q_n$, uniformly from D. For each q_i, find its nearest neighbor in D – $\{q_i\}$: $y_i = min\{dist(q_i, v)\}$ where v in D and v ≠ q_i
  - Calculate the Hopkins Statistic: $$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$
  - If D is uniformly distributed, $\sum x_i$ and $\sum y_i$ will be close to each other and H is close to 0.5. If D is clustered, H is close to 1

Determine the Number of Clusters

- Empirical method
  - # of clusters ≈√n/2 for a dataset of n points
- Elbow method
  - Use the turning point in the curve of sum of within cluster variance w.r.t the # of clusters
- 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

Measuring Clustering Quality

- Two methods: extrinsic vs. intrinsic
- Extrinsic: supervised, i.e., the ground truth is available
  - Compare a clustering against the ground truth using certain clustering quality measure
  - Ex. BCubed precision and recall metrics
- Intrinsic: unsupervised, i.e., the ground truth is unavailable
  - Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are
  - Ex. Silhouette coefficient

Measuring Clustering Quality: Extrinsic Methods

- Clustering quality measure: $Q(C, C_g)$, for a clustering C given the ground truth C_g.
- 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

Cluster Analysis: Basic Concepts and Methods

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

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

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.

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.

Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
  - What Is Cluster Analysis?
  - What is Good Clustering? Measuring the Quality of Clustering
  - Major categories of clustering methods
- Clustering structures
  - Calculating Distance between Clusters
- Partitioning Methods
  - *k*-Means: A Classical Partitioning Method
  - Alternative Methods: *k*-Medoids, *k*-Median, and its Variations
- Hierarchical Methods
  - Agglomerative and Divisive Hierarchical Clustering
  - BIRCH: A Hierarchical, Micro-Clustering Approach
  - Chameleon: A Hierarchical Clustering Algorithm Using Dynamic Modeling
- Density-Based Methods
  - DBSCAN and OPTICS: Density-Based Clustering Based on Connected Regions
  - DENCLUE: Clustering Based on Density Distribution Functions
- Link-Based Cluster Analysis
  - SimRank: Exploring Links in Cluster Analysis
  - LinkClus: Scalability in Link-Based Cluster Analysis
- Grid-Based Methods
  - STING: STatistical INformation Grid
  - WaveCluster: Clustering Using Wavelet Transformation
  - CLIQUE: A Dimension-Growth Subspace Clustering Method
- Summary

65

STOP Clustering #1

BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies)

- Zhang, Ramakrishnan & Livny, SIGMOD'96
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
  - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
  - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree
- *Scales linearly*: finds a good clustering with a single scan and improves the quality with a few additional scans
- *Weakness:* handles only numeric data, and sensitive to the order of the data record

Clustering Feature Vector in BIRCH

Clustering Feature (CF): ***CF = (N, LS, SS)***

N: **Number of data points**

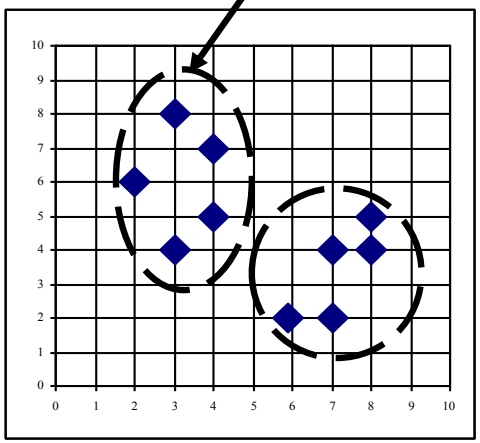
LS: linear sum of N points:

$$\sum_{i=1}^{N} X_i$$

SS: square sum of N points

$$\sum_{i=1}^{N} X_i^2$$

CF = (5, (16,30),(54,190))

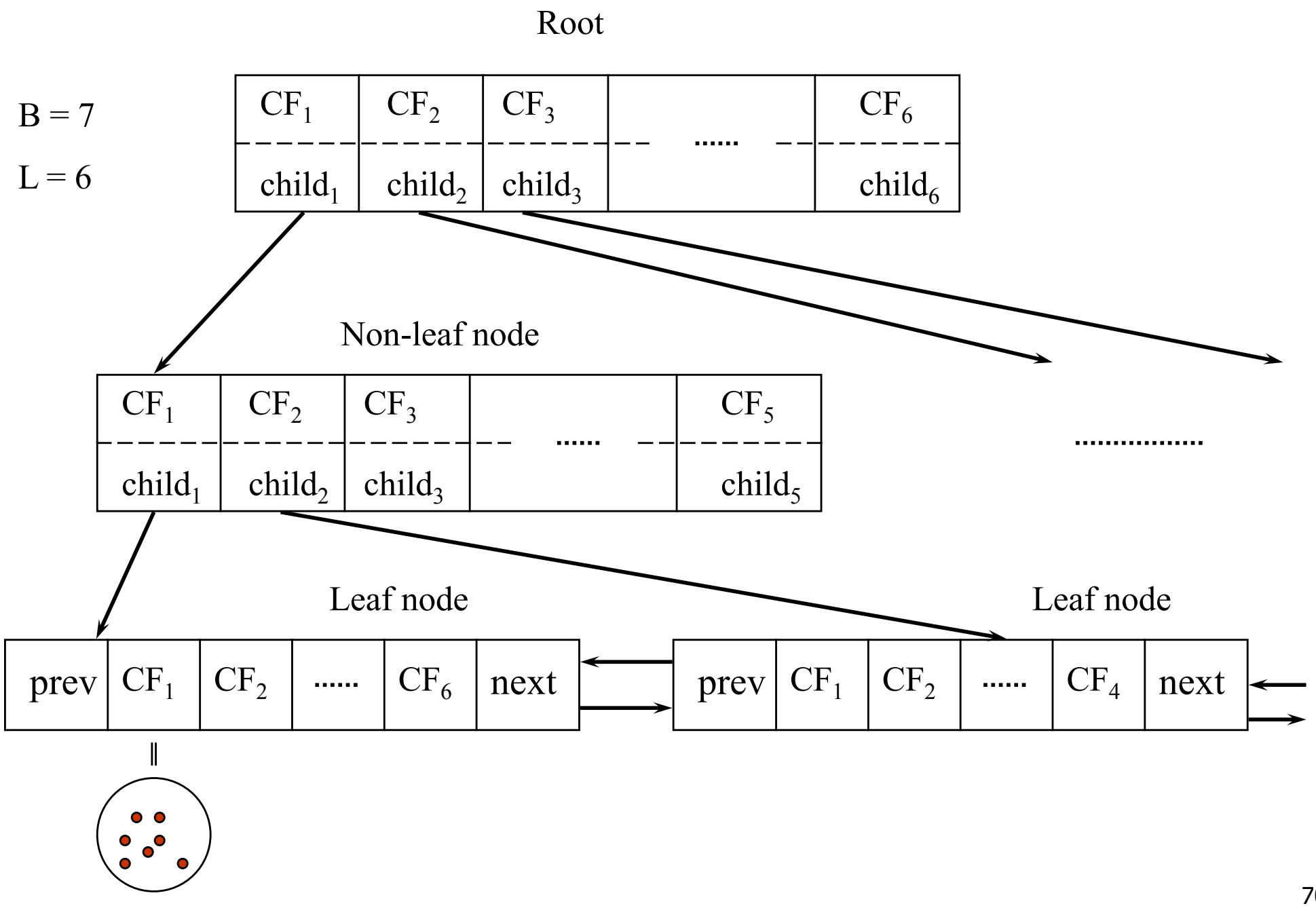


(3,4)
(2,6)
(4,5)
(4,7)
(3,8)

CF-Tree in BIRCH

- Clustering feature:
  - Summary of the statistics for a given subcluster: the 0-th, 1st, and 2nd moments of the subcluster from the statistical point of view
  - Registers crucial measurements for computing cluster and utilizes storage efficiently
- A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering
  - A nonleaf node in a tree has descendants or “children”
  - The nonleaf nodes store sums of the CFs of their children
- A CF tree has two parameters
  - Branching factor: max # of children
  - Threshold: max diameter of sub-clusters stored at the leaf nodes

The CF Tree Structure



The Birch Algorithm

- Cluster Diameter $$\sqrt{\frac{1}{n(n-1)}\sum(x_i - x_j)^2}$$
- For each point in the input
  - Find closest leaf entry
  - Add point to leaf entry and update CF
  - If entry diameter > max_diameter, then split leaf, and possibly parents
- Algorithm is O(n)
- Concerns
  - Sensitive to insertion order of data points
  - Since we fix the size of leaf nodes, so clusters may not be so natural
  - Clusters tend to be spherical given the radius and diameter measures

CHAMELEON: Hierarchical Clustering Using Dynamic Modeling (1999)

- CHAMELEON: G. Karypis, E. H. Han, and V. Kumar, 1999
- Measures the similarity based on a dynamic model
  - Two clusters are merged only if the *interconnectivity* and *closeness (proximity)* between two clusters are high *relative to* the internal interconnectivity of the clusters and closeness of items within the clusters
- Graph-based, and a two-phase algorithm
  1. Use a graph-partitioning algorithm: cluster objects into a large number of relatively small sub-clusters
  2. Use an agglomerative hierarchical clustering algorithm: find the genuine clusters by repeatedly combining these sub-clusters

Cluster Analysis: Basic Concepts and Methods

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

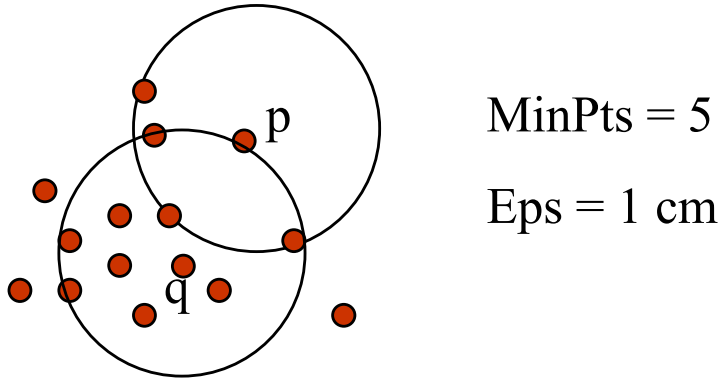
Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)

Density-Based Clustering: Basic Concepts

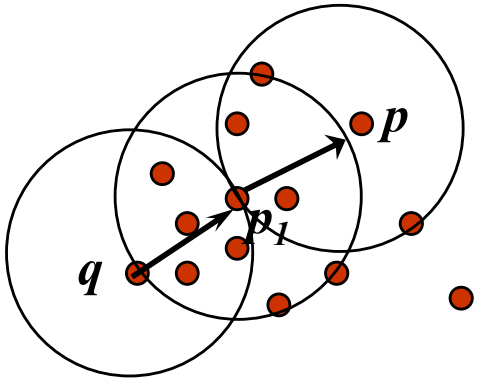
- Two parameters:
  - *Eps*: Maximum radius of the neighbourhood
  - *MinPts*: Minimum number of points in an Eps-neighbourhood of that point
- $N_{Eps}(q)$: {p belongs to D | dist(p,q) ≤ Eps}
- Directly density-reachable: A point *p* is directly density-reachable from a point *q* w.r.t. *Eps*, *MinPts*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

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

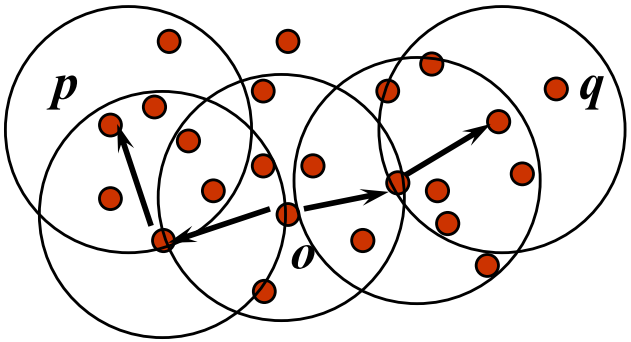


Density-Reachable and Density-Connected

- Density-reachable:
  - A point *p* is density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i

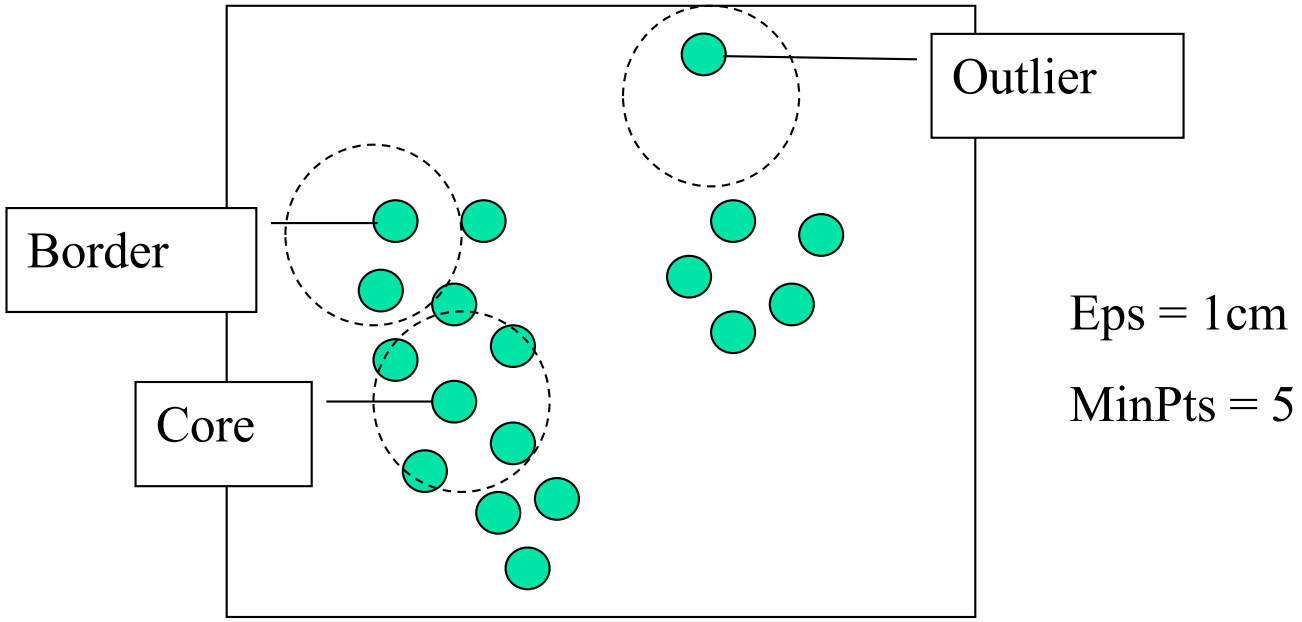


- Density-connected
  - A point *p* is density-connected to a point *q* w.r.t. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* w.r.t. *Eps* and *MinPts*



DBSCAN: Density-Based Spatial Clustering of Applications with Noise

- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise



DBSCAN: The Algorithm

- Arbitrary select a point *p*
- Retrieve all points density-reachable from *p* w.r.t. *Eps* and *MinPts*
- If *p* is a core point, a cluster is formed
- If *p* is a border point, no points are density-reachable from *p* and DBSCAN visits the next point of the database
- Continue the process until all of the points have been processed
- *If a spatial index is used, the computational complexity of DBSCAN is O(nlogn), where n is the number of database objects. Otherwise, the complexity is* $O(n^2)$

DBSCAN: Sensitive to Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

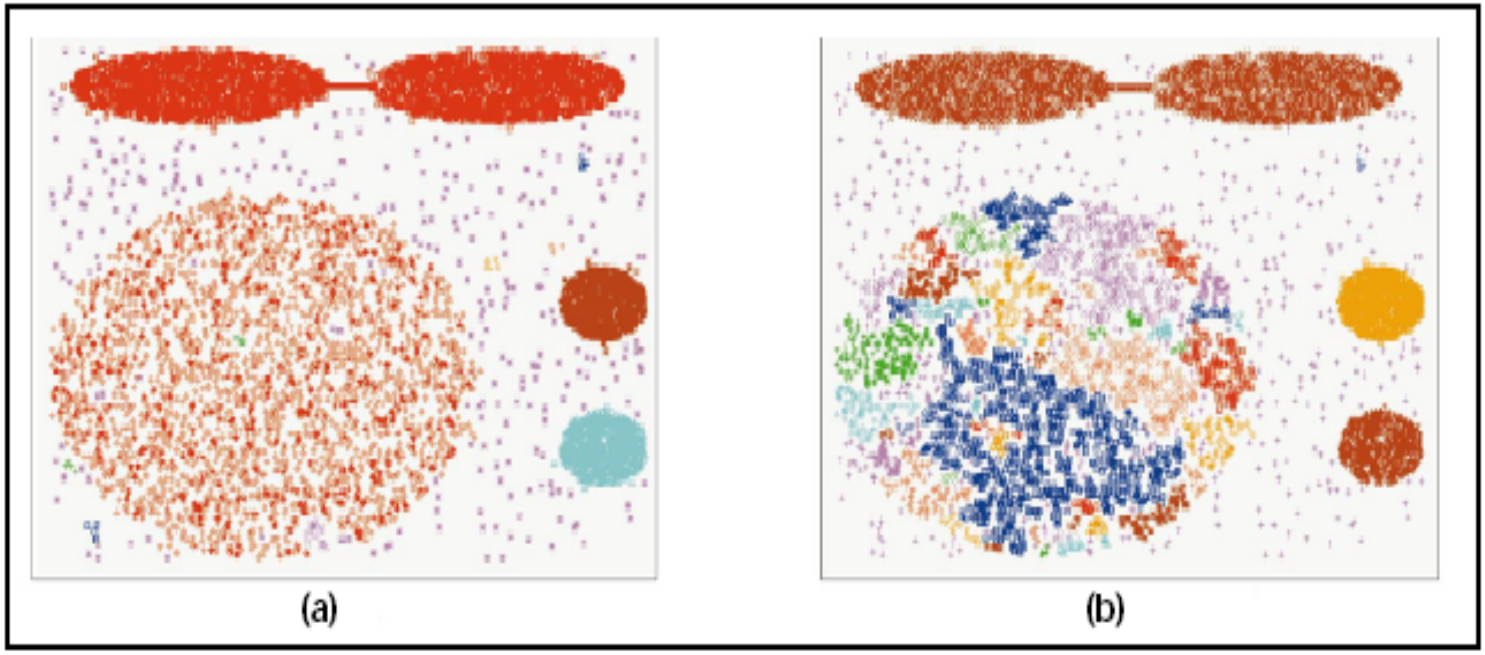
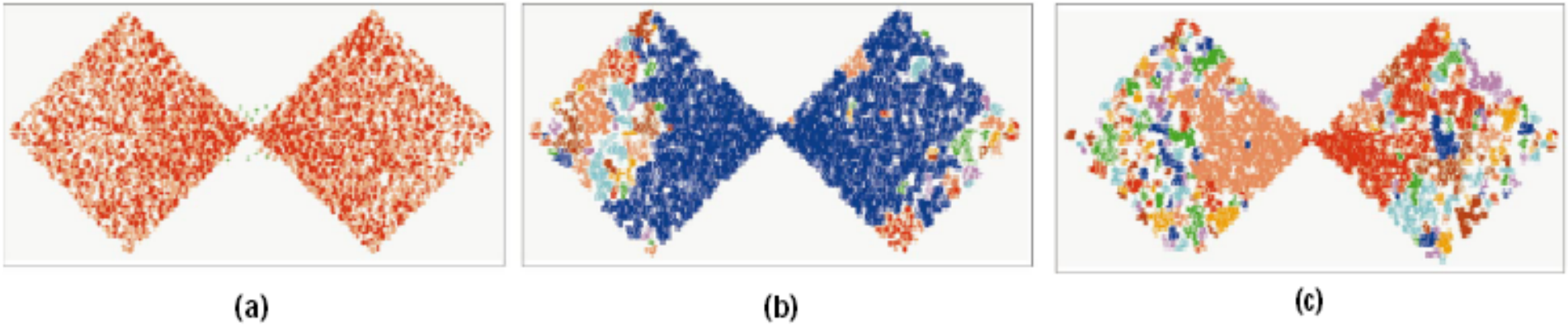


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



DBSCAN online Demo:

http://webdocs.cs.ualberta.ca/~yaling/Cluster/Applet/Code/Cluster.html

OPTICS: A Cluster-Ordering Method (1999)

- OPTICS: Ordering Points To Identify the Clustering Structure
  - Ankerst, Breunig, Kriegel, and Sander (SIGMOD'99)
  - Produces a special order of the database wrt its density-based clustering structure
  - This cluster-ordering contains info equiv to the density-based clusterings corresponding to a broad range of parameter settings
  - Good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure
  - Can be represented graphically or using visualization techniques

OPTICS: Some Extension from DBSCAN

- Index-based: k = # of dimensions, N: # of points
  - Complexity: O(N*logN)
- Core Distance of an object p: the smallest value ε such that the ε-neighborhood of p has at least MinPts objects

  Let $N_\varepsilon(p)$: ε-neighborhood of p, ε is a distance value

  $\text{Core-distance}_{\varepsilon, \text{MinPts}}(p)$ = Undefined if $\text{card}(N_\varepsilon(p)) < \text{MinPts}$

  MinPts-distance(p), otherwise
- Reachability Distance of object p from core object q is the min radius value that makes p density-reachable from q

  $\text{Reachability-distance}_{\varepsilon, \text{MinPts}}(p, q)$ =

  Undefined if q is not a core object

  max(core-distance(q), distance (q, p)), otherwise

Core Distance & Reachability Distance

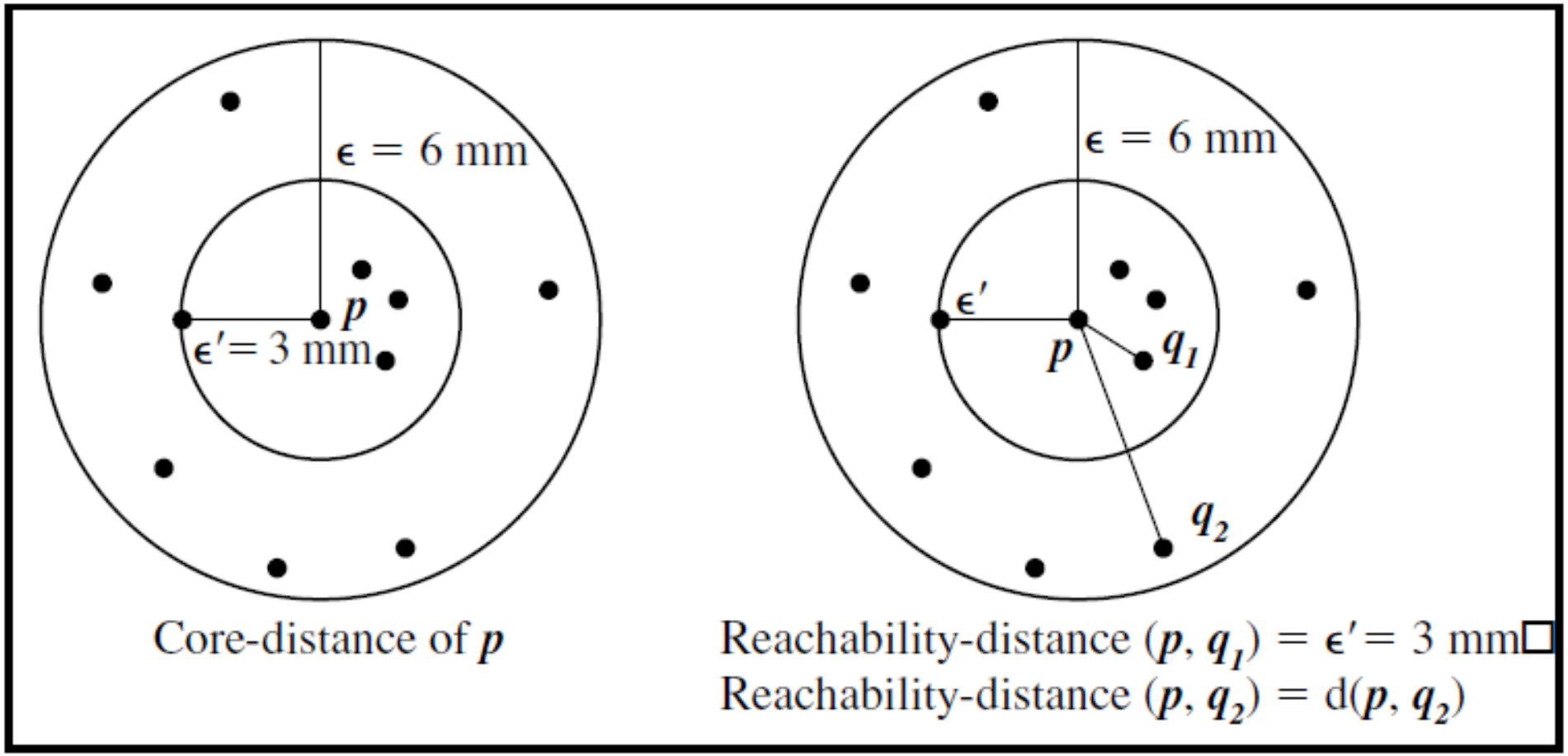
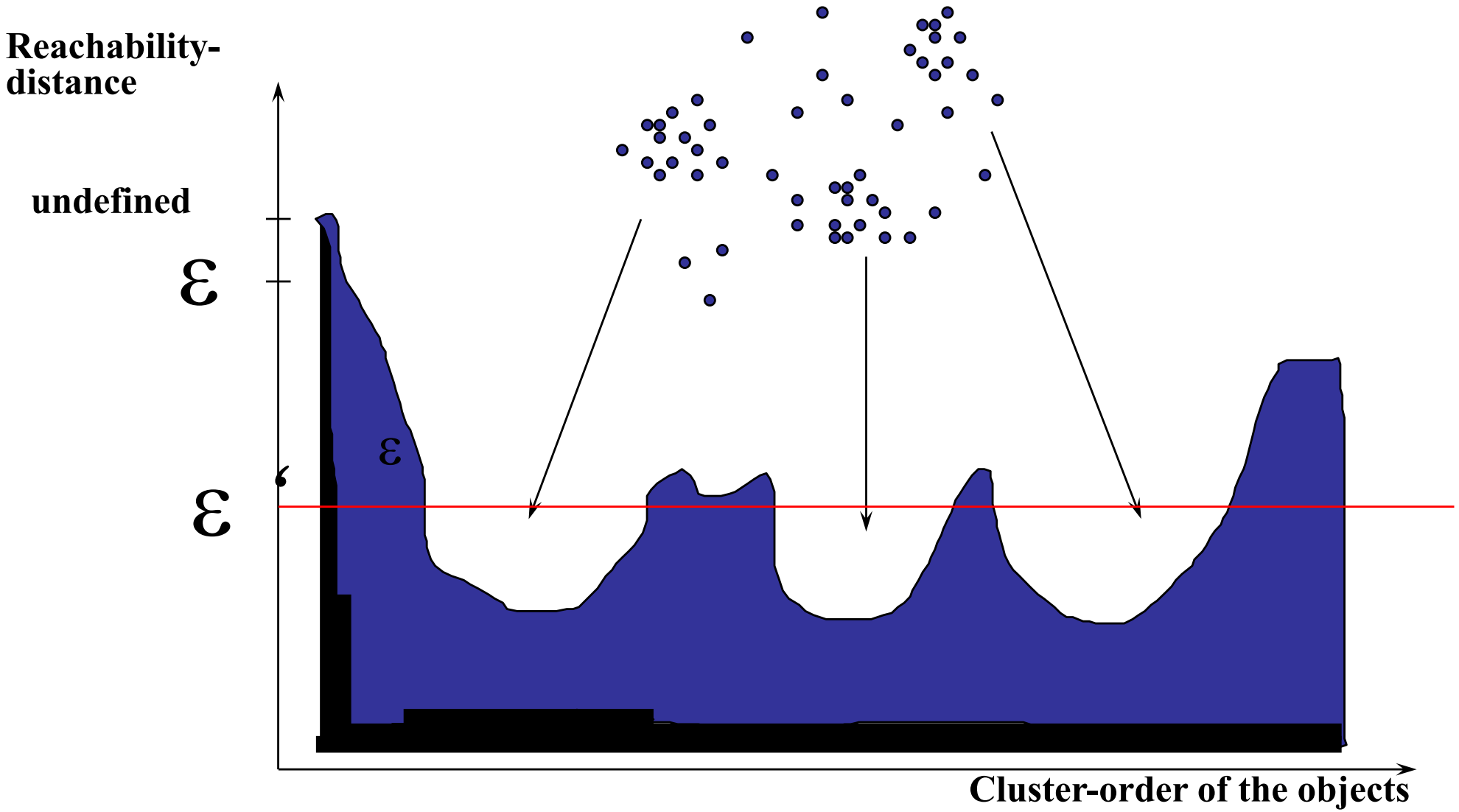
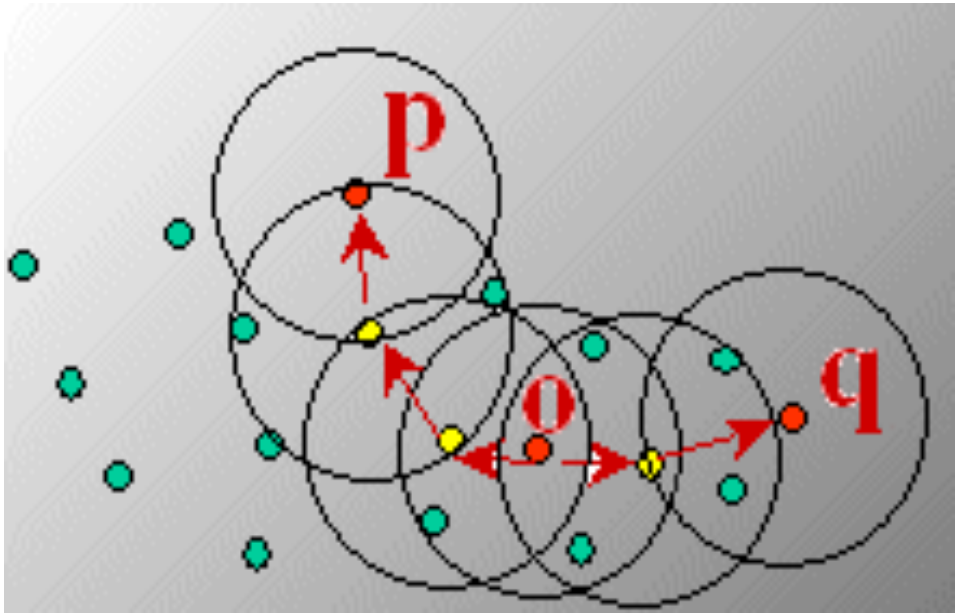


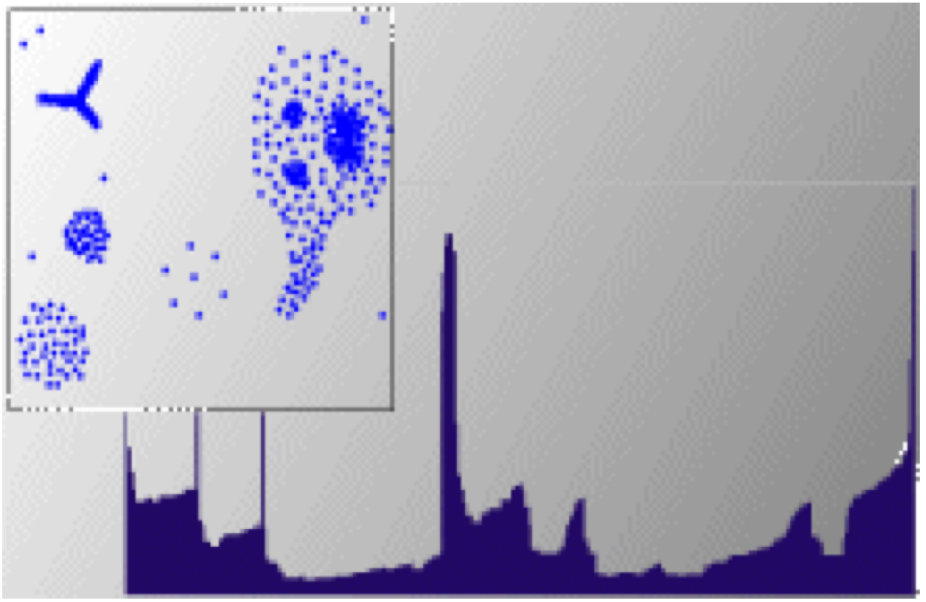
Figure 10.16: OPTICS terminology. Based on [ABKS99].

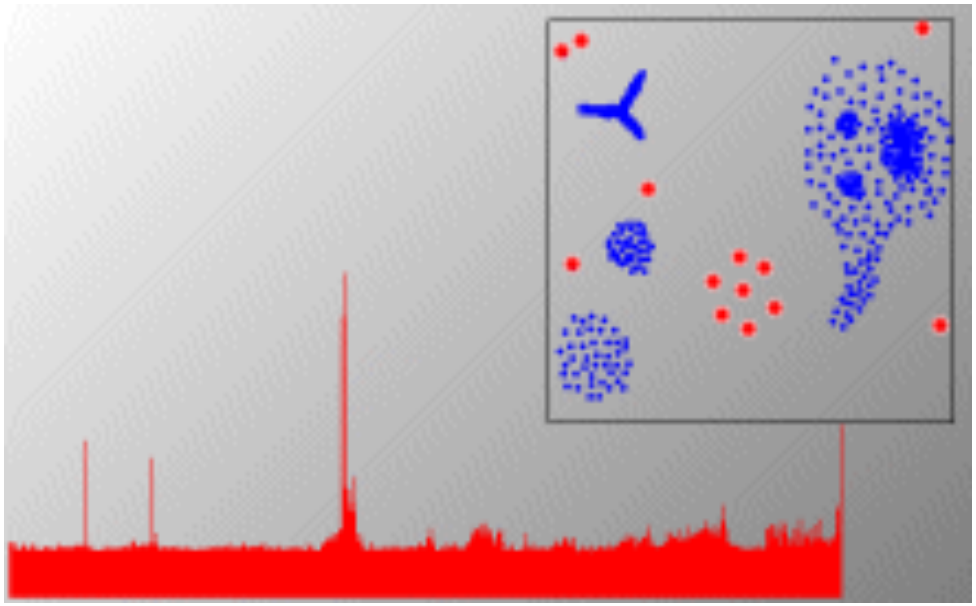

Reachability-distance
undefined
ε
ε'
ε
Cluster-order of the objects

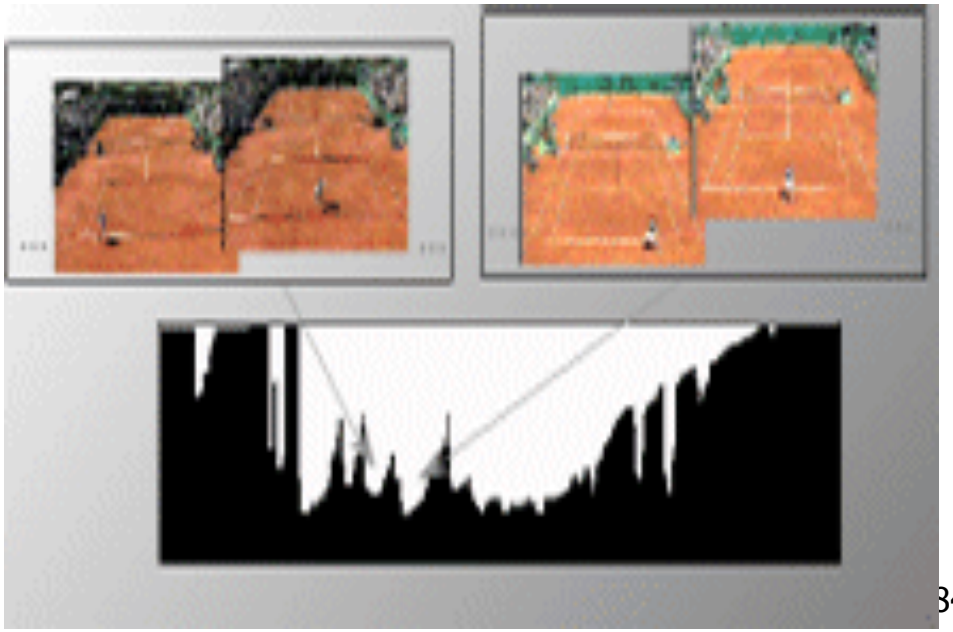
Density-Based Clustering: OPTICS & Applications

demo: http://www.dbs.informatik.uni-muenchen.de/Forschung/KDD/Clustering/OPTICS/Demo









DENCLUE: Using Statistical Density Functions

- DENsity-based CLUstEring by Hinneburg & Keim (KDD'98)
- Using statistical density functions:

$$f_{Gaussian}(x,y) = e^{-\frac{d(x,y)^2}{2\sigma^2}}$$

influence of y on x

$$f^D_{Gaussian}(x) = \sum_{i=1}^{N} e^{-\frac{d(x,x_i)^2}{2\sigma^2}}$$

total influence on x

$$\nabla f^D_{Gaussian}(x,x_i) = \sum_{i=1}^{N} (x_i - x) \cdot e^{-\frac{d(x,x_i)^2}{2\sigma^2}}$$

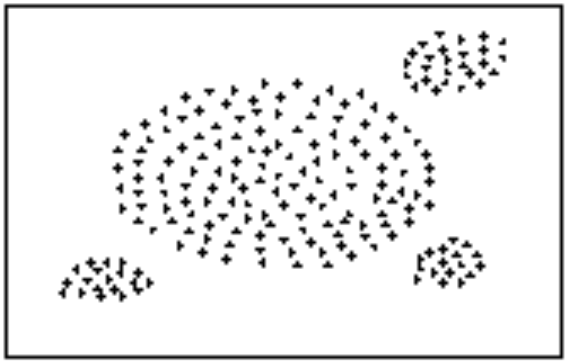
gradient of x in the direction of x_i

- Major features
  - Solid mathematical foundation
  - Good for data sets with large amounts of noise
  - Allows a compact mathematical description of arbitrarily shaped clusters in high-dimensional data sets
  - Significant faster than existing algorithm (e.g., DBSCAN)
  - But needs a large number of parameters

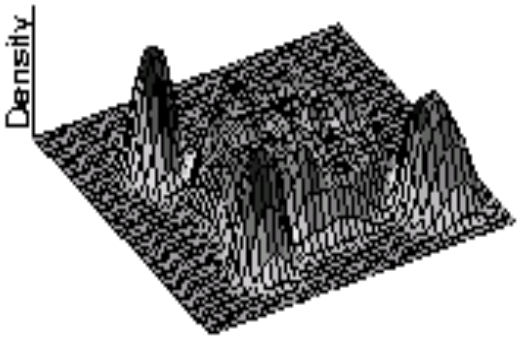
Denclue: Technical Essence

- Uses grid cells but only keeps information about grid cells that do actually contain data points and manages these cells in a tree-based access structure
- Influence function: describes the impact of a data point within its neighborhood
- Overall density of the data space can be calculated as the sum of the influence function of all data points
- Clusters can be determined mathematically by identifying density attractors
- Density attractors are local maximal of the overall density function
- Center defined clusters: assign to each density attractor the points density attracted to it
- Arbitrary shaped cluster: merge density attractors that are connected through paths of high density (> threshold)

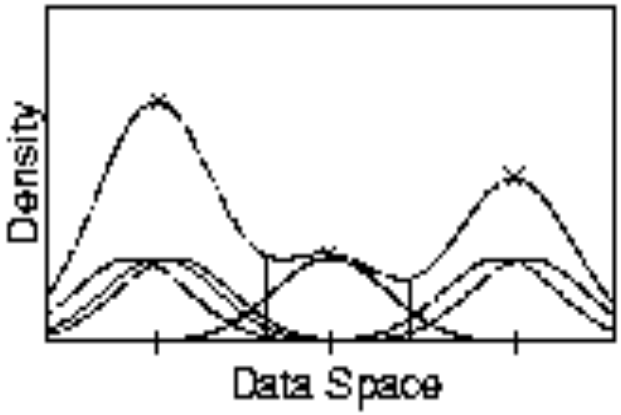
Density Attractor



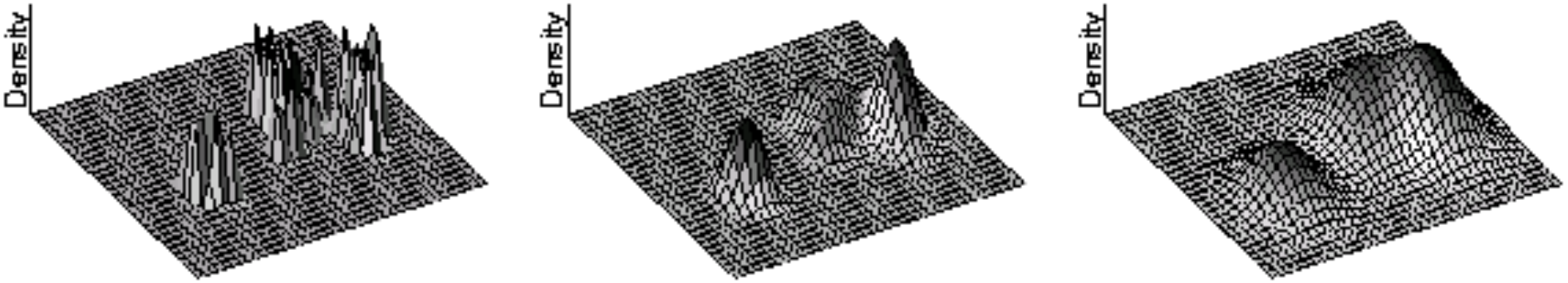
(a) Data Set



(c) Gaussian

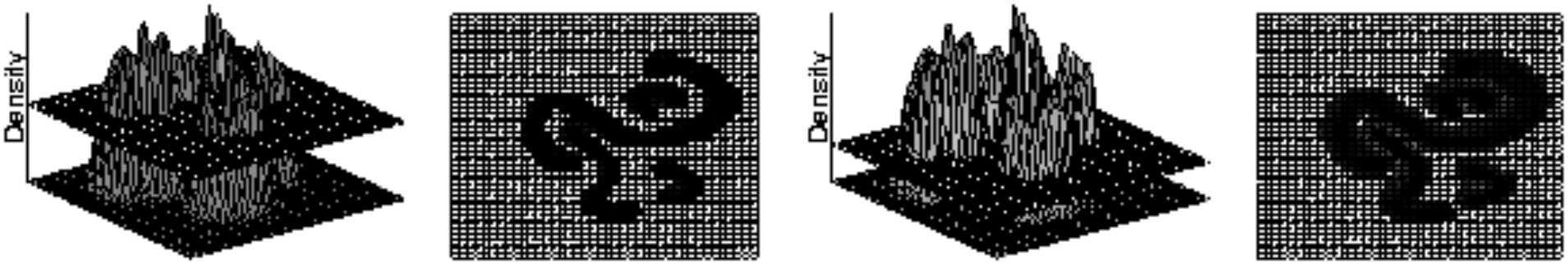


Center-Defined and Arbitrary



(a) $\sigma = 0.2$ (b) $\sigma = 0.6$ (d) $\sigma = 1.5$

Figure 3: Example of Center-Defined Clusters for different σ



(a) $\xi = 2$ (b) $\xi = 2$ (c) $\xi = 1$ (d) $\xi = 1$

Figure 4: Example of Arbitray-Shape Clusters for different ξ

Cluster Analysis: Basic Concepts and Methods

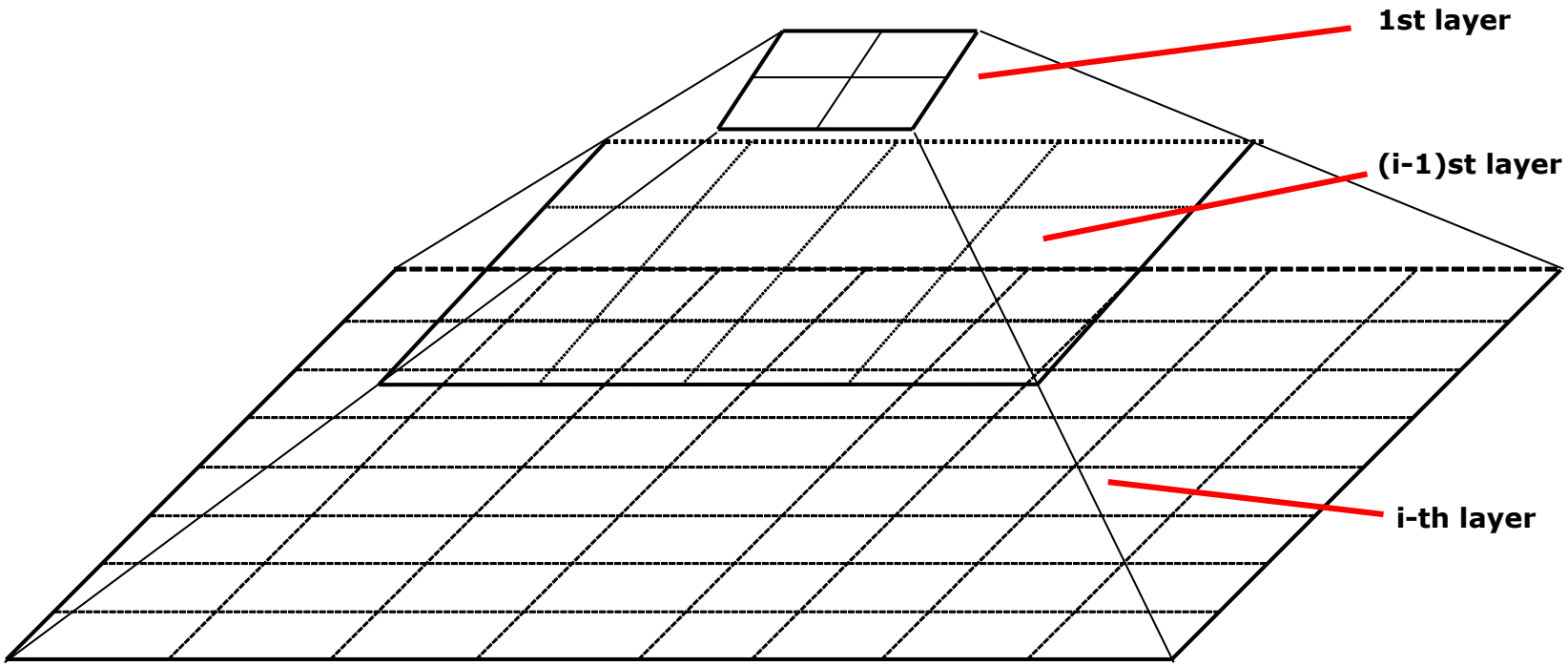
- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods 
- Evaluation of Clustering
- Summary

Grid-Based Clustering Method

- Using multi-resolution grid data structure
- Several interesting methods
  - STING (a STatistical INformation Grid approach) by Wang, Yang and Muntz (1997)
  - WaveCluster by Sheikholeslami, Chatterjee, and Zhang (VLDB'98)
    - A multi-resolution clustering approach using wavelet method
  - CLIQUE: Agrawal, et al. (SIGMOD'98)
    - Both grid-based and subspace clustering

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



The STING Clustering Method

- Each cell at a high level is partitioned into a number of smaller cells in the next lower level
- Statistical info of each cell is calculated and stored beforehand and is used to answer queries
- Parameters of higher level cells can be easily calculated from parameters of lower level cell
  - *count*, *mean*, *s*, *min*, *max*
  - type of distribution—*normal*, *uniform*, etc.
- Use a top-down approach to answer spatial data queries
- Start from a pre-selected layer—typically with a small number of cells
- For each cell in the current level compute the confidence interval

STING Algorithm and Its Analysis

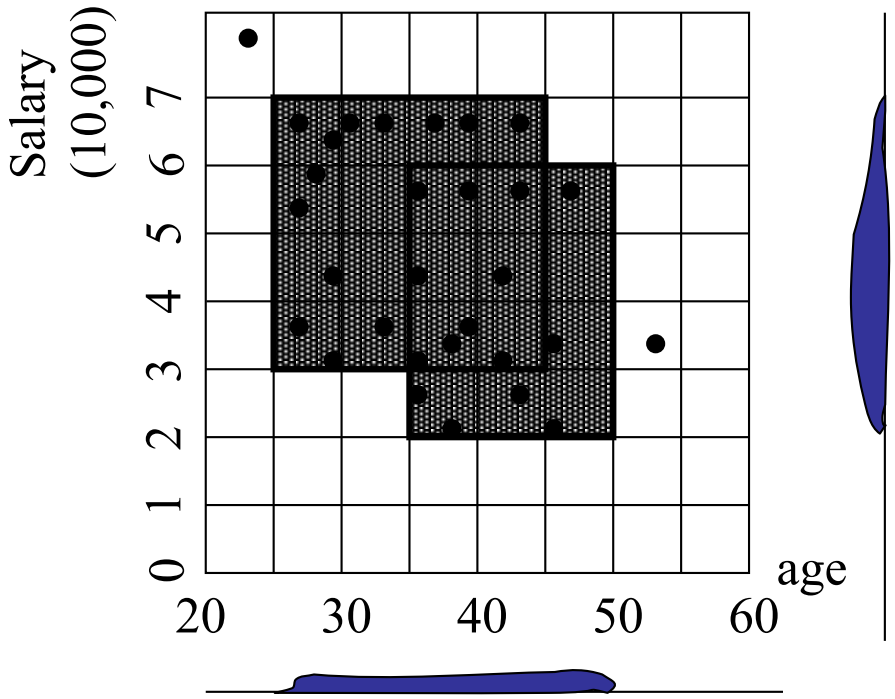
- Remove the irrelevant cells from further consideration
- When finish examining the current layer, proceed to the next lower level
- Repeat this process until the bottom layer is reached
- Advantages:
  - Query-independent, easy to parallelize, incremental update
  - *O(K),* where *K* is the number of grid cells at the lowest level
- Disadvantages:
  - All the cluster boundaries are either horizontal or vertical, and no diagonal boundary is detected

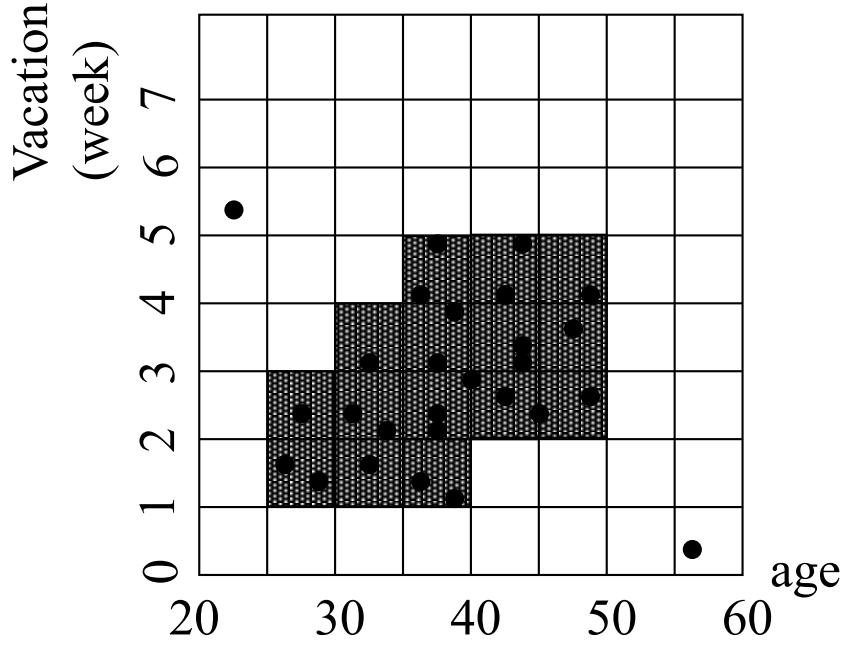
CLIQUE (Clustering In QUEst)

- Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD'98)
- Automatically identifying subspaces of a high dimensional data space that allow better clustering than original space
- CLIQUE can be considered as both density-based and grid-based
  - It partitions each dimension into the same number of equal length interval
  - It partitions an m-dimensional data space into non-overlapping rectangular units
  - A unit is dense if the fraction of total data points contained in the unit exceeds the input model parameter
  - A cluster is a maximal set of connected dense units within a subspace

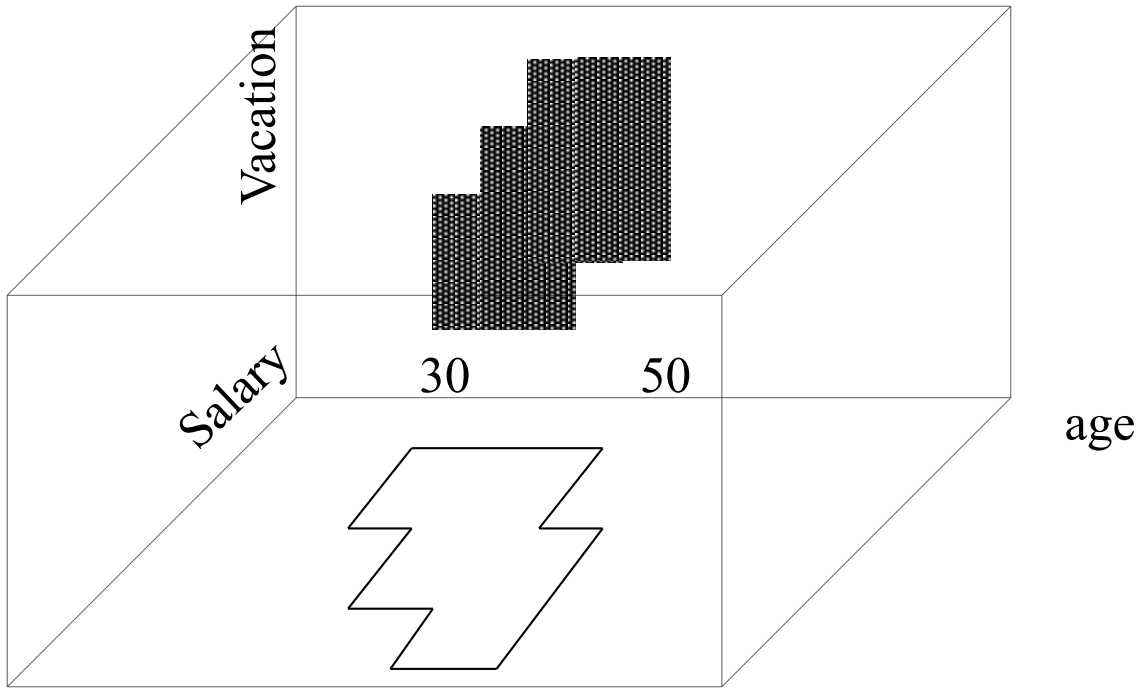
CLIQUE: The Major Steps

- Partition the data space and find the number of points that lie inside each cell of the partition.
- Identify the subspaces that contain clusters using the Apriori principle
- Identify clusters
  - Determine dense units in all subspaces of interests
  - Determine connected dense units in all subspaces of interests.
- Generate minimal description for the clusters
  - Determine maximal regions that cover a cluster of connected dense units for each cluster
  - Determination of minimal cover for each cluster


Salary
(10,000)
0 1 2 3 4 5 6 7
20 30 40 50 60
age


Vacation
(week)
0 1 2 3 4 5 6 7
20 30 40 50 60
age

$\tau = 3$

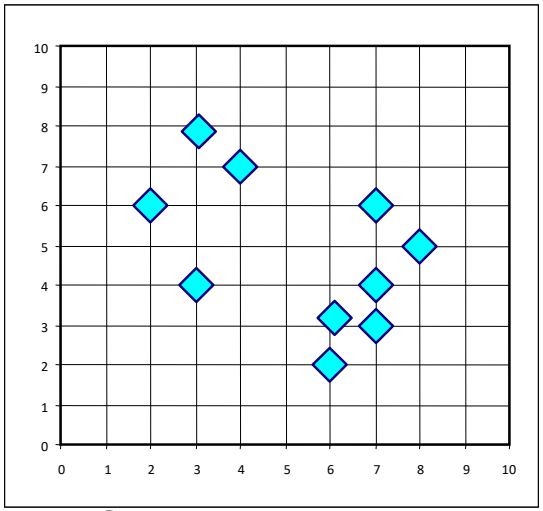

Vacation
Salary
30
50
age

Strength and Weakness of *CLIQUE*

- Strength
  - *automatically* finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces
  - *insensitive* to the order of records in input and does not presume some canonical data distribution
  - scales *linearly* with the size of input and has good scalability as the number of dimensions in the data increases
- Weakness
  - The accuracy of the clustering result may be degraded at the expense of simplicity of the method

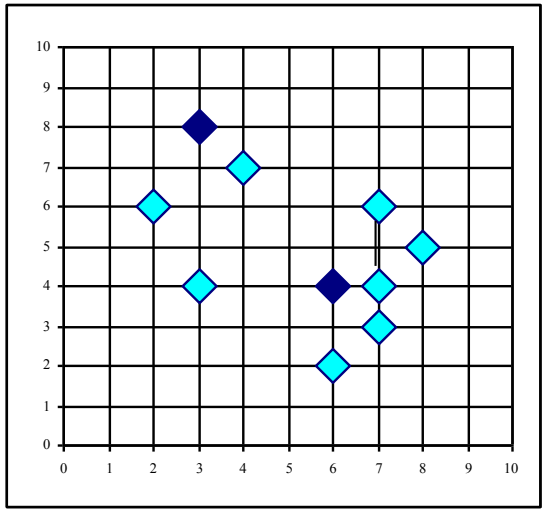
Extra slides unused in class

A Typical K-Medoids Algorithm (PAM)



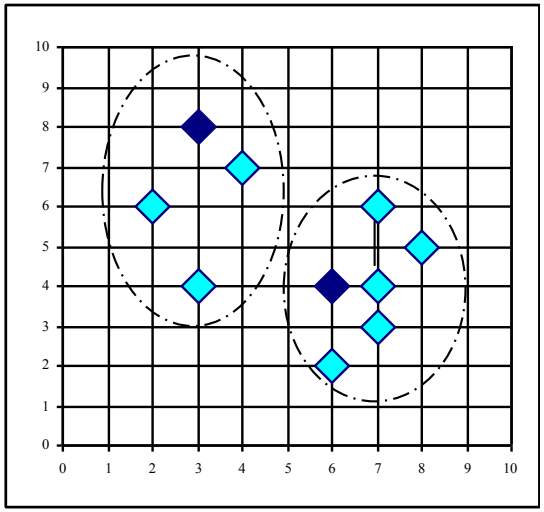
K=2

Arbitrary choose k object as initial medoids

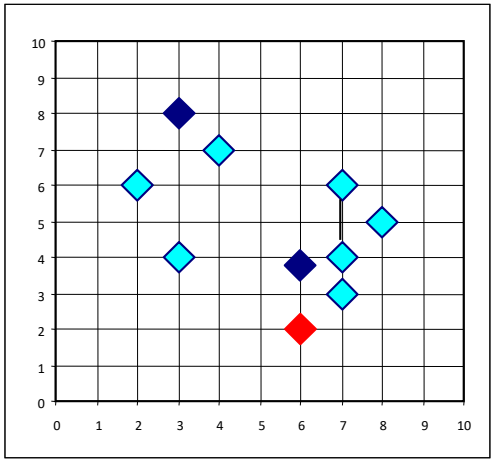


Assign each remaining object to nearest medoids

Total Cost = 20

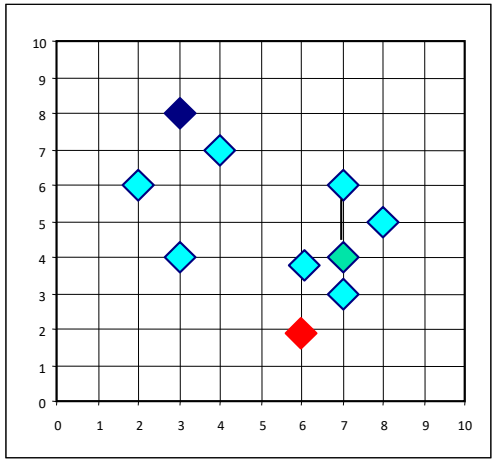


Randomly select a nonmedoid object, O_{ramdom}



Compute total cost of swapping

Total Cost = 26



Swapping O and O_{ramdom}

If quality is improved.

Do loop

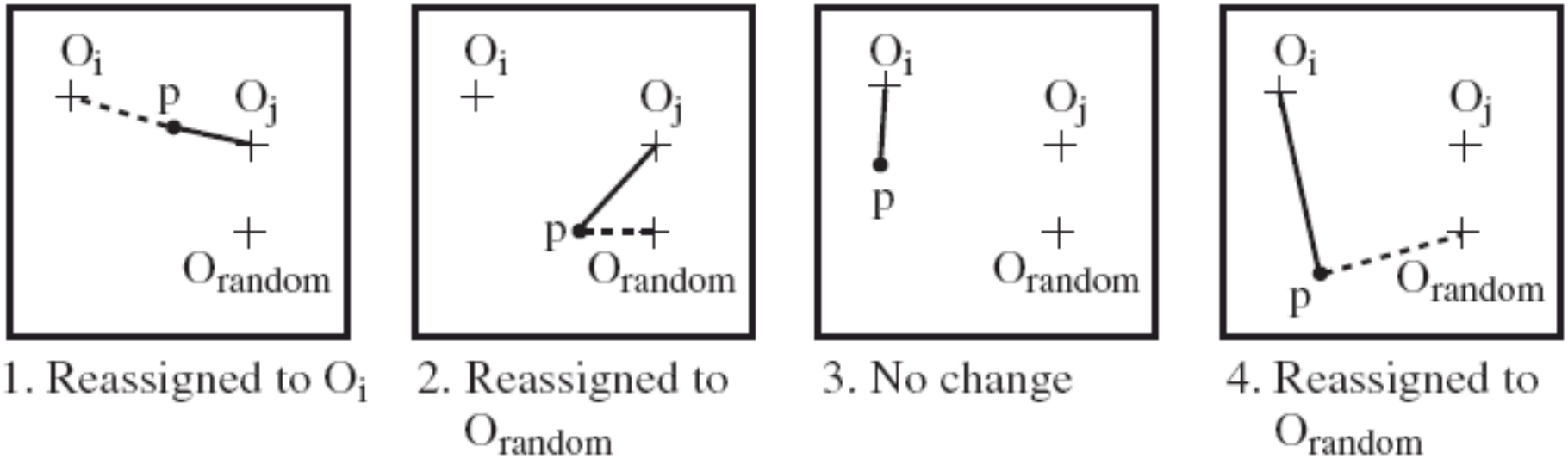
Until no change

PAM (Partitioning Around Medoids) (1987)

- PAM (Kaufman and Rousseeuw, 1987), built in Splus
- Use real object to represent the cluster
  - Select ***k*** representative objects arbitrarily
  - For each pair of non-selected object ***h*** and selected object ***i***, calculate the total swapping cost TC_{ih}
  - For each pair of ***i*** and ***h***,
    - If $TC_{ih} < 0$, ***i*** is replaced by ***h***
    - Then assign each non-selected object to the most similar representative object
  - repeat steps 2-3 until there is no change

PAM Clustering: Finding the Best Cluster Center

- Case 1: p currently belongs to o_j. If o_j is replaced by o_{random} as a representative object and p is the closest to one of the other representative object o_i, then p is reassigned to o_i



What Is the Problem with PAM?

- Pam is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean
- Pam works efficiently for small data sets but does not **scale well** for large data sets.
  - $O(k(n-k)^2)$ for each iteration

    where n is # of data,k is # of clusters

➔Sampling-based method

CLARA(Clustering LARge Applications)

CLARA (Clustering Large Applications) (1990)

- *CLARA* (Kaufmann and Rousseeuw in 1990)
  - Built in statistical analysis packages, such as SPlus
  - It draws *multiple samples* of the data set, applies *PAM* on each sample, and gives the best clustering as the output
- Strength: deals with larger data sets than *PAM*
- Weakness:
  - Efficiency depends on the sample size
  - A good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased

CLARANS ("Randomized" CLARA) *(1994)*

- *CLARANS* (A Clustering Algorithm based on Randomized Search) (Ng and Han'94)
  - Draws sample of neighbors dynamically
  - The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of *k* medoids
  - If the local optimum is found, *it* starts with new randomly selected node in search for a new local optimum
- Advantages: More efficient and scalable than both *PAM* and *CLARA*
- Further improvement: Focusing techniques and spatial access structures (Ester et al.'95)

ROCK: Clustering Categorical Data

- ROCK: RObust Clustering using linKs
  - S. Guha, R. Rastogi & K. Shim, ICDE'99
- Major ideas
  - Use links to measure similarity/proximity
  - Not distance-based
- Algorithm: sampling-based clustering
  - Draw random sample
  - Cluster with links
  - Label data in disk
- Experiments
  - Congressional voting, mushroom data

Similarity Measure in ROCK

- Traditional measures for categorical data may not work well, e.g., Jaccard coefficient
- Example: Two groups (clusters) of transactions
  - C_1. <a, b, c, d, e>: {a, b, c}, {a, b, d}, {a, b, e}, {a, c, d}, {a, c, e}, {a, d, e}, {b, c, d}, {b, c, e}, {b, d, e}, {c, d, e}
  - C_2. <a, b, f, g>: {a, b, f}, {a, b, g}, {a, f, g}, {b, f, g}
- Jaccard co-efficient may lead to wrong clustering result
  - C_1: 0.2 ({a, b, c}, {b, d, e}} to 0.5 ({a, b, c}, {a, b, d})
  - C_1 & C_2: could be as high as 0.5 ({a, b, c}, {a, b, f})
- Jaccard co-efficient-based similarity function:

$$Sim(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$$

  - Ex. Let T_1 = {a, b, c}, T_2 = {c, d, e}

$$Sim(T_1, T_2) = \frac{|\{c\}|}{|\{a, b, c, d, e\}|} = \frac{1}{5} = 0.2$$

Link Measure in ROCK

- Clusters
  - C_1:<a, b, c, d, e>: {a, b, c}, {a, b, d}, {a, b, e}, {a, c, d}, {a, c, e}, {a, d, e}, {b, c, d}, {b, c, e}, {b, d, e}, {c, d, e}
  - C_2: <a, b, f, g>: {a, b, f}, {a, b, g}, {a, f, g}, {b, f, g}
- Neighbors
  - Two transactions are neighbors if sim(T_1,T_2) > threshold
  - Let T_1 = {a, b, c}, T_2 = {c, d, e}, T_3 = {a, b, f}
    - T_1 connected to: {a,b,d}, {a,b,e}, {a,c,d}, {a,c,e}, {b,c,d}, {b,c,e}, {a,b,f}, {a,b,g}
    - T_2 connected to: {a,c,d}, {a,c,e}, {a,d,e}, {b,c,e}, {b,d,e}, {b,c,d}
    - T_3 connected to: {a,b,c}, {a,b,d}, {a,b,e}, {a,b,g}, {a,f,g}, {b,f,g}
- Link Similarity
  - Link similarity between two transactions is the # of common neighbors
  - link(T_1, T_2) = 4, *since they have 4 common neighbors*
    - {a, c, d}, {a, c, e}, {b, c, d}, {b, c, e}
  - link(T_1, T_3) = 3, *since they have 3 common neighbors*
    - {a, b, d}, {a, b, e}, {a, b, g}