第五章：

文本自动聚类技术

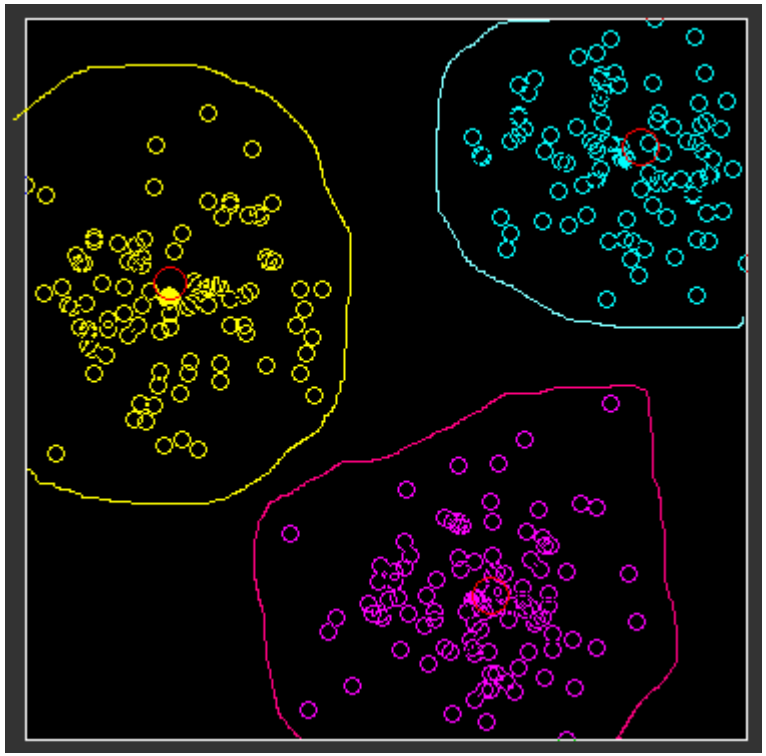
杨建武

北京大学计算机科学技术研究所

Email:yangjianwu@icst.pku.edu.cn

什么是聚类?

- 聚类(簇Cluster)：数据对象的集合
  - 在同一个簇中，数据对象是相似的
  - 不同簇之间的对象是不相似的

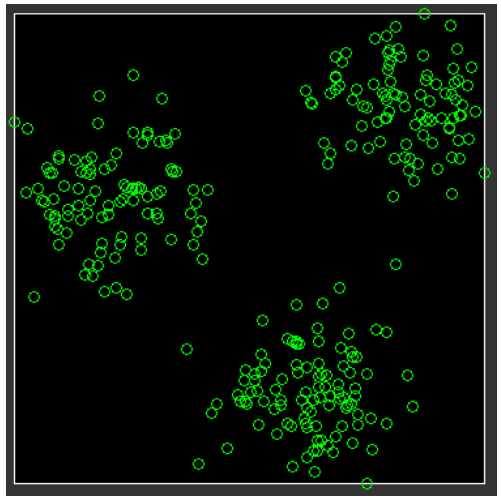


什么是聚类分析?

聚类分析就是按照一定的规律和要求对事物进行区分和分类的过程，在这一过程中没有任何关于类分的先验知识，没有指导，仅靠事物间的相似性作为类属划分的准则。

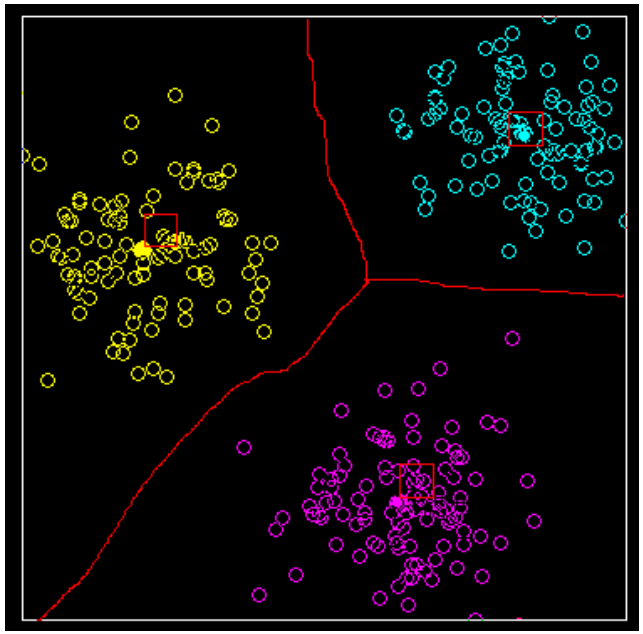
- 一个数据集合分组成几个簇
- 聚类分析是一种无监督分类:没有预定义的类

聚类分析：数据集的划分

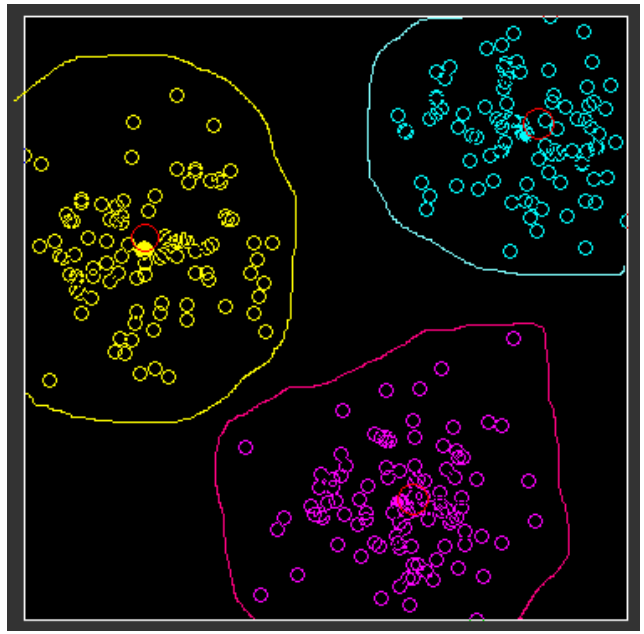


无标记的
样本集

空间划分



空间覆盖



聚类分析的数学描述

- 聚类分析(Clustering)：给定一数据样本集$\boldsymbol{x}$ $\{X_1, X_2, \cdots, X_n\}$，根据数据点间的相似程度将数据集合分成k簇$\{C_1, C_2, \cdots, C_k\}$的过程称为聚类分析。
- 簇记为$C_i = \{X_{j1}^i, X_{j2}^i, \cdots, X_{jni}^i\}$
- C_i（$i=1, \cdots, k$）是$\boldsymbol{x}$的子集，且满足：
  - $C_1 \cup C_2 \cup \cdots \cup C_k = \boldsymbol{x}$
  - $C_i \cup C_j = \Phi$，$i \neq j$。
- 相似样本在同一簇中，相异样本在不同簇中。

聚类分析的典型应用

- 典型应用
  - 作为一个独立的工具透视数据分布
  - 可以作为其他算法的预处理步骤

应用聚类分析的例子

- 市场销售：帮助市场人员发现客户中的不同群体，然后用这些知识来开展一个目标明确的市场计划；
- 保险：对购买了汽车保险的客户，标识那些有较高平均赔偿成本的客户；
- 城市规划：根据类型、价格、地理位置等来划分不同类型的住宅；
- 地震研究：根据地质断层的特点把已观察到的地震中心分成不同的类；

文本聚类

- Document Clustering (DC) is partitioning a set of documents into groups or clusters
- Clusters should be computed to
  - Contain similar documents
  - Separate as much as possible different documents
- For instance, if similarity between documents is defined to capture semantic relatedness, documents in a cluster should deal with the same topics, and topics in each cluster should be different

文本聚类的应用

- Guiding the organization of a document collection (e.g. [Sahami 98])
  - Progressively clustering groups of documents allow to build a topic hierarchy
- Supporting browsing and interactive retrieval
  - Grouping retrieved documents to allow a faster relevant documents selection process
  - Some search engines (Vivisimo)
- Pre-processing for other tasks, e.g.
  - To detect the main semantic dimensions of the text collection and to support a kind of concept indexing [Karypis & Han 00]
  - For text summarization

web news images wikipedia blogs jobs more »

Clusty

china

Search

advanced preferences

clusters sources sites

remix

All Results (217)

- Tibet, Beijing (14)
- History (15)
- Investing (11)
- Photos (12)
- Tours, China Travel (9)
- Asia, Rénmín Gònghéguó (5)
- Supplier (10)
- China. Manufacturer (12)
- Died (6)
- Maps (7)

more | all clusters

find in clusters:

Top **214** results of at least **31,785,288** retrieved for the query **china** (definition) (details)

Current stock trading for CDC CORPORATION CL-A (CHINA)

Symbol	Last	Change	Open	Prev Close
CHINA	3.95	⬆0.01 (0.25%)	3.88	3.94

Search Results

1. **China**

This article is about the Chinese civilization. For the modern-day state commonly referred to as **China**, *see People's Republic of China. For other meanings, see China (disambiguation).*
en.wikipedia.org/wiki/China - [cache] - Wikipedia, Gigablast, Live

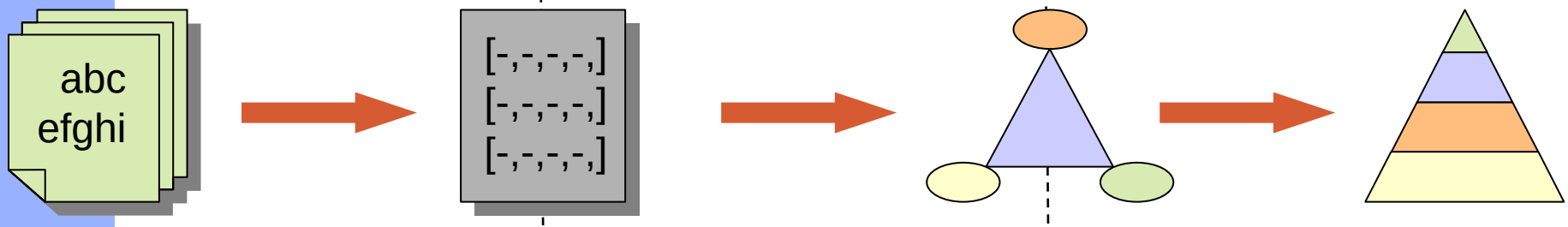
2. **People's Republic of China**

The **People's Republic of China** (**PRC**; Pinyin: Zhōnghuá Rénmín Gònghéguó ; Traditional: 中華人民共和國 Simplified: 中华人民共和国), commonly referred to as **China**, and sometimes **Red China**, is a country in East Asia.
en.wikipedia.org/wiki/People's_Republic_of_China - [cache] - Wikipedia, Gigablast, Live

文本聚类基本步骤

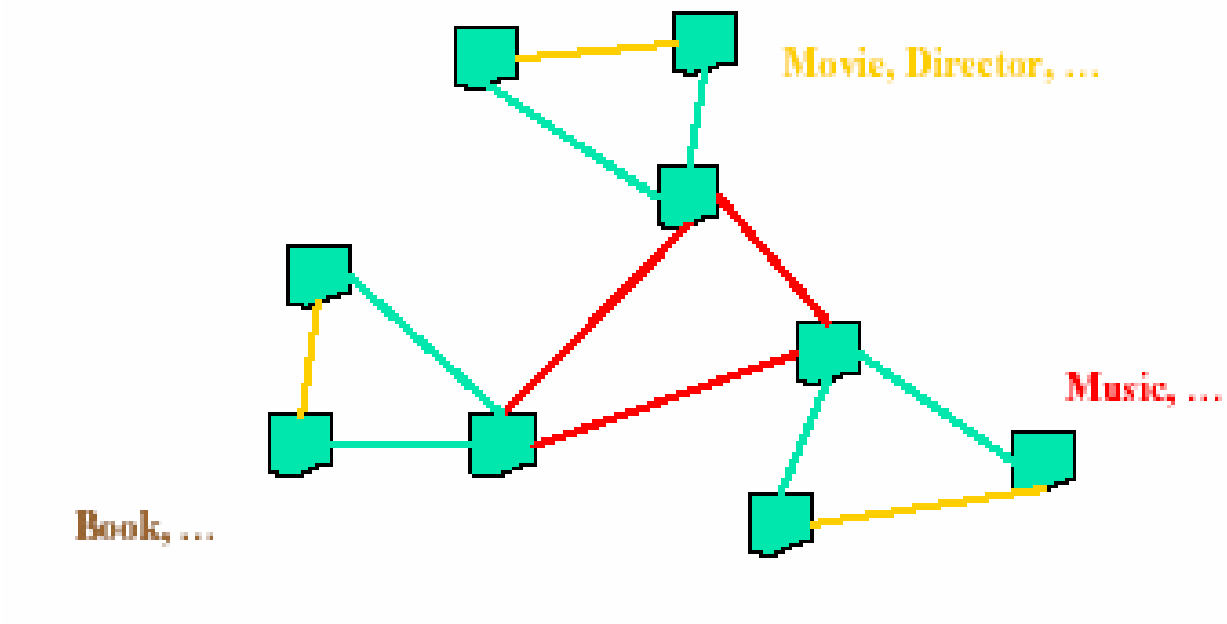
- As other text processing tasks, DC has several steps
  - Document representation
  - Dimensionality reduction
  - Applying a clustering algorithm
  - Evaluating the effectiveness of the process

- 文档表示
- 聚类算法
- 可视化



文本聚类基本步骤

- 1. 用户指定用于聚类的数据集合
- 2. 特征选取
- 3. 聚合：将每个文档分配到相应的类中
- 4. 标注：给每个聚类选择关键词



评价指标

聚类结果的评价

- 「准确率」（P，precision）
- 「召回率」（R，recall）
- F—Measure

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}}$$

$$F_1 = \frac{2PR}{P+R}$$

		Human	
		True	False
Classifier	Yes	a	b
	No	c	d

准确率 P = a/(a+b)

召回率 R = a/(a+c)

聚类结果的评价

- 所有类的总体评价

$$F_i = \frac{1}{\alpha \frac{1}{P_i} + (1-\alpha)\frac{1}{R_i}}$$

- 宏平均 Macro

$$F_{1i} = \frac{2P_iR_i}{P_i + R_i}$$

$$Macro - F = \frac{1}{m}\sum_{i=1}^{m} F_i$$

- 微平均 Micro

$$Micro - F = \sum_{i=1}^{m}(n_i \cdot F_i) \Big/ \sum_{i=1}^{m} n_i$$

什么是一个好的聚类方法?

- 聚类方法的好坏：该方法是否能发现某些或所有的隐含模式；
- 一个好的聚类方法要能产生高质量的聚类结果——簇，这些簇要具备以下两个特点：
  - 高的簇内相似性
  - 低的簇间相似性
- 聚类结果的好坏取决于：
  - 聚类方法采用的相似性评估方法
  - 该方法的具体实现；

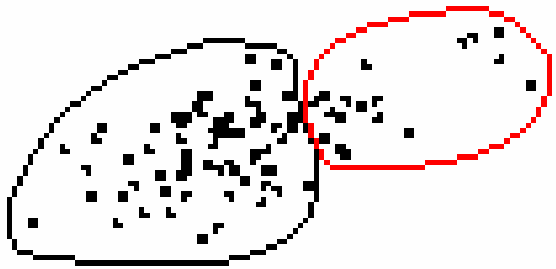
聚类的准则函数

- 误差平方和准则
- **(sum-of-squared-error criterion）:**

$$J_e = \sum_{i=1}^{k} \sum_{X \in C_i} |X - m_i|^2$$

其中$X \in C_i$，m_i是C_i的质心

J_e = large

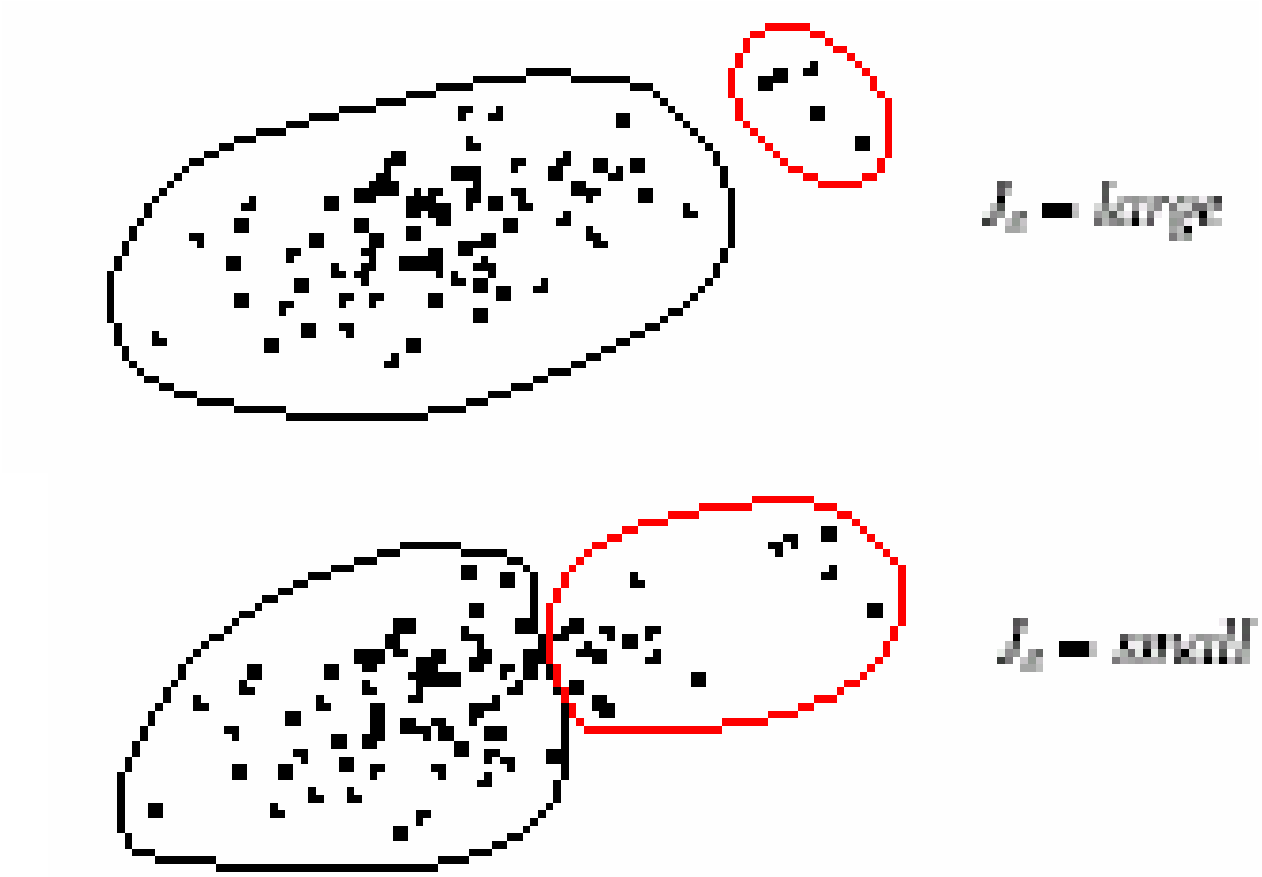


J_e = small

聚类的准则函数

- 误差平方和准则

$$J_e = \sum_{i=1}^{k} \sum_{X \in C_i} | X - m_i |^2$$



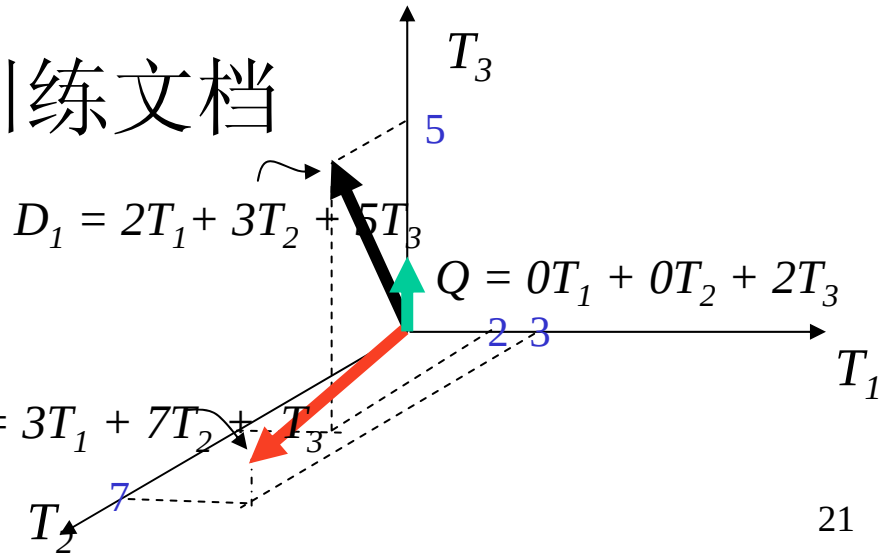
聚类算法的评价

- 可伸缩性
- 能够处理不同类型的属性
- 能发现任意形状的簇
- 在决定输入参数的时候，尽量不需要特定的领域知识；
- 能够处理噪声和异常
- 对输入数据对象的顺序不敏感
- 能处理高维数据
- 能产生一个好的、能满足用户指定约束的聚类结果
- 结果是可解释的、可理解的和可用的

聚类算法

文档间距离

- 向量空间模型(Vector Space Model)
  - M个无序标引项t_i（特征），词根/词/短语/其他
  - 每个文档d_j可以用标引项向量来表示
    - $(a_{1j}, a_{2j}, \cdots, a_{Mj})$
  - 权重计算，N个训练文档
    - $A_{M*N} = (a_{ij})$
  - 相似度计算
    - Cosine计算
    - 内积计算



类间距离

类G_p与类G_q之间的距离D_{pq}

($d(x_i, x_j)$表示点$x_i \in G_p$和$x_j \in G_q$之间的距离)

最短距离法:

$$D_{pq} = \min d(x_i, x_j)$$

最长距离法:

$$D_{pq} = \max d(x_i, x_j)$$

重心法:

$$D_{pq} = \min d(\overline{x}_p, \overline{x}_q)$$

类平均法:

$$D_{pq} = \frac{1}{n_1 n_2} \sum_{x_i \in G_p} \sum_{x_j \in G_q} d(x_i, x_j)$$

离差平方和:(Wald)

$$D_1 = \sum_{x_i \in G_p} (x_i - \overline{x}_p)'(x_i - \overline{x}_p), D_2 = \sum_{x_j \in G_q} (x_j - \overline{x}_q)'(x_j - \overline{x}_q),$$

$$D_{1+2} = \sum_{x_k \in G_p \cup G_q} (x_k - \overline{x})'(x_i - \overline{x}) \Rightarrow D_{pq} = D_{1+2} - D_1 - D_2$$

聚类方法

- 划分的方法
- 层次的方法
- 基于密度的方法
- 基于网格的方法
- 在线聚类

划分方法

划分方法
（partitioning method）

- 划分方法的基本思想是，给定一个n个样本的数据集，划分方法将数据划分为k个簇（k<=n），满足：
  - a. 每个簇至少包含一个样本；
  - b. 每个样本必须属于且仅属于一个簇。

划分方法

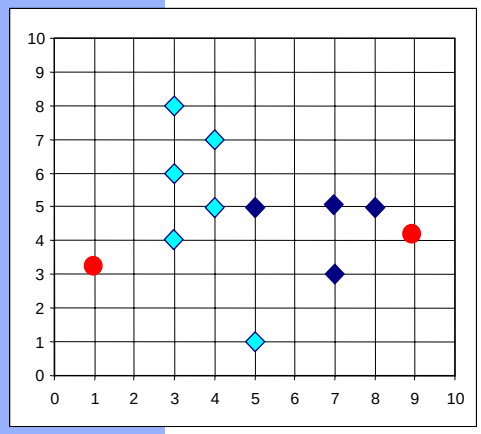
- 将文档集D=$\{d_1, \ldots, d_i, \ldots, d_n\}$ 分割为的若干类，具体过程：
  1. 确定要生成的类的数目k；
  2. 按照某种原则生成k个聚类中心作为聚类的种子S=$\{s_1, \ldots, s_j, \ldots, s_k\}$；
  3. 对D中的每一个文档d_i，依次计算它与各个种子s_j的相似度sim(d_i, s_j)；
  4. 选取具有最大的相似度的种子arg max sim(d_i, s_j)，将d_i归入以s_j为聚类中心的类C_j，从而得到D的一个聚类C=$\{c_1, \ldots, c_k\}$；
  5. 重复步骤2~4若干次，以得到较为稳定的聚类结果。

该方法速度快，但k要预先确定，种子选取难

k-means（k-均值）(MacQueen'67)

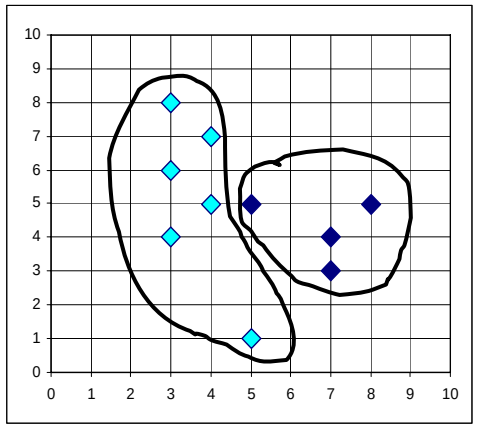
- 1. 选择一个含有随机选择样本的k个簇的初始划分，计算这些簇的质心。
- 2. 根据欧氏距离把剩余的每个样本分配到距离它最近的簇质心的一个划分。
- 3. 计算被分配到每个簇的样本的均值向量，作为新的簇的质心。
- 4. 重复2,3直到k个簇的质心点不再发生变化或准则函数收敛。

k-means算法示例

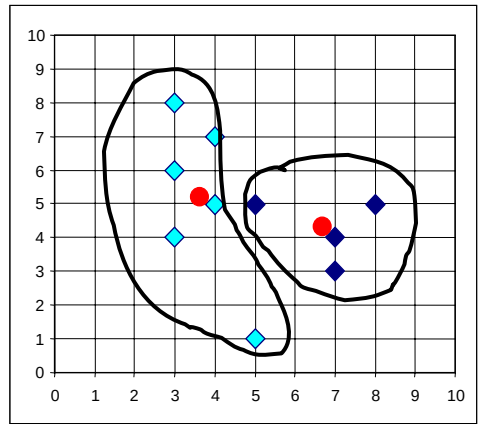


K=2

Arbitrarily choose K object as initial cluster center

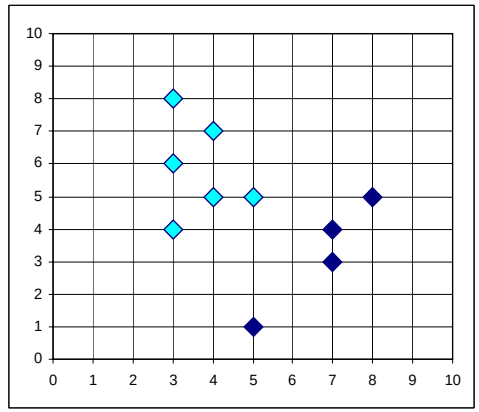


Assign each objects to most similar center

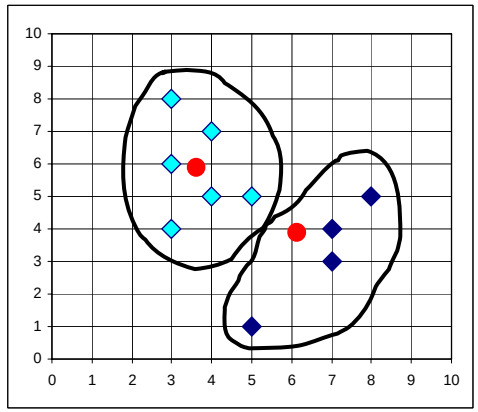


Update the cluster means

reassign



Update the cluster means



reassign

k-means算法示例

- 坐标表示5个点{X_1, X_2, X_3, X_4, X_5}作为一个聚类分析的二维样本：

  X_1＝（0,2），　X_2＝（0,0），

  X_3＝（1.5,0），X_4＝（5,0），

  X_5＝（5,2）。

- 假设要求的簇的数量k=2。

k-means算法示例

- 第1步：由样本的随机分布形成两个簇：

  $C_1 = \{X_1, X_2, X_4\}$ 和 $C_2 = \{X_3, X_5\}$ 。

- 这两个簇的质心 M_1 和 M_2 是：

  $M_1 = \{ (0+0+5)/3, (2+0+0)/3 \} = \{1.66, 0.66\}$；

  $M_2 = \{ (1.5+5)/2, (0+2)/2 \} = \{3.25, 1.00\}$；

k-means算法示例

- 样本初始随机分布之后，方差是：

$e_1^2 = [(0-1.66)^2+(2-0.66)^2]$

$+ [(0-1.66)^2+(0-0.66)^2]$

$+ [(5-1.66)^2+(0-0.66)^2]$

$=19.36$；

$e_2^2=8.12$；

$$J_e = \sum_{i=1}^{k} \sum_{X \in C_i} |X - m_i|^2$$

- 总体平方误差是：$E^2 = e_1^2+e_2^2 = 19.36+8.12 = 27.48$（公式）

k-means算法示例

- 第2步：按与质心（M_1或M_2）间距离关系，选择<u>最小</u>距离分配所有样本，簇内样本的重新分布如下：

  $d(M_1, X_1)=(1.66^2+1.34^2)^{1/2}=2.14$

  $d(M_2, X_1)=3.40 \Rightarrow X_1 \in C_1$;

  $d(M_1, X_2)=1.79$ 和 $d(M_2, X_2)=3.40 \Rightarrow X_2 \in C_1$

  $d(M_1, X_3)=0.83$ 和 $d(M_2, X_3)=2.01 \Rightarrow X_3 \in C_1$

  $d(M_1, X_4)=3.41$ 和 $d(M_2, X_4)=2.01 \Rightarrow X_4 \in C_2$

  $d(M_1, X_5)=3.60$ 和 $d(M_2, X_5)=2.01 \Rightarrow X_5 \in C_2$

- 新簇$C_1=\{X_1, X_2, X_3\}$ 和$C_2=\{X_4, X_5\}$

k-means算法示例

- 第3步：计算新的质心：

  M_1＝｛0.5，0.67｝；　　M_2＝｛5.0，1.0｝。

- 相应的方差及总体平方误差分别是：

  e_1^2＝4.17；　　e_2^2＝2.00；　　E＝6.17；

- 可以看出第一次迭代后，总体误差显著减小（从值27.48到6.17）。

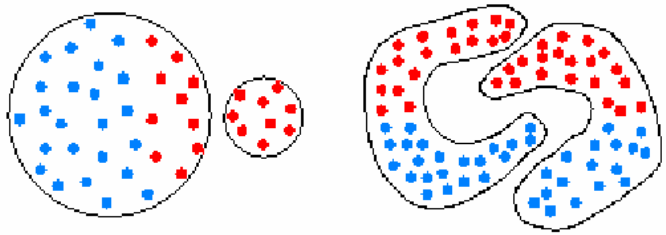
- 在这个简单的例子中，第一次迭代同时也是最后一次迭代，因为如果继续分析新中心和样本间的距离，样本将会全部分给同样的簇，不将重新分配，算法停止。

k-means算法分析

- **Strength:** *Relatively efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, k, $t \ll n$.
  - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- **Comment:** Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*

k-means的缺陷

- 要求用户必须事先给出要生成的簇的数目，选择初始划分的最佳方向、更新和停止准则
- 难以处理大小很不相同的簇或具有凹状的簇。



- 算法只有在簇的平均值被定义的情况下才能使用，这不适涉及有分类属性的数据。
  - k－prototypes算法对数值与离散两种混合的数据聚类， k－modes方法对离散属性
- 对噪音和异常点非常敏感

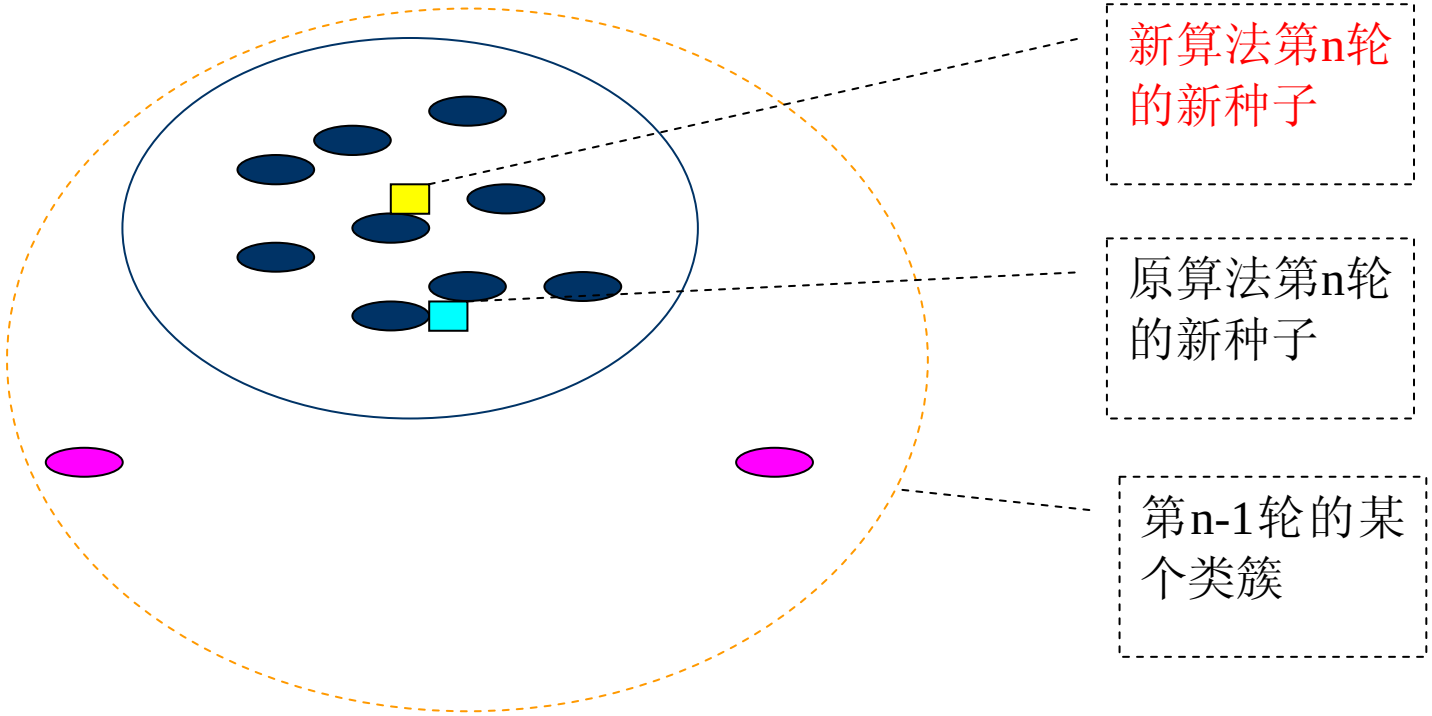
对k-means算法的改进

- **改进目的**

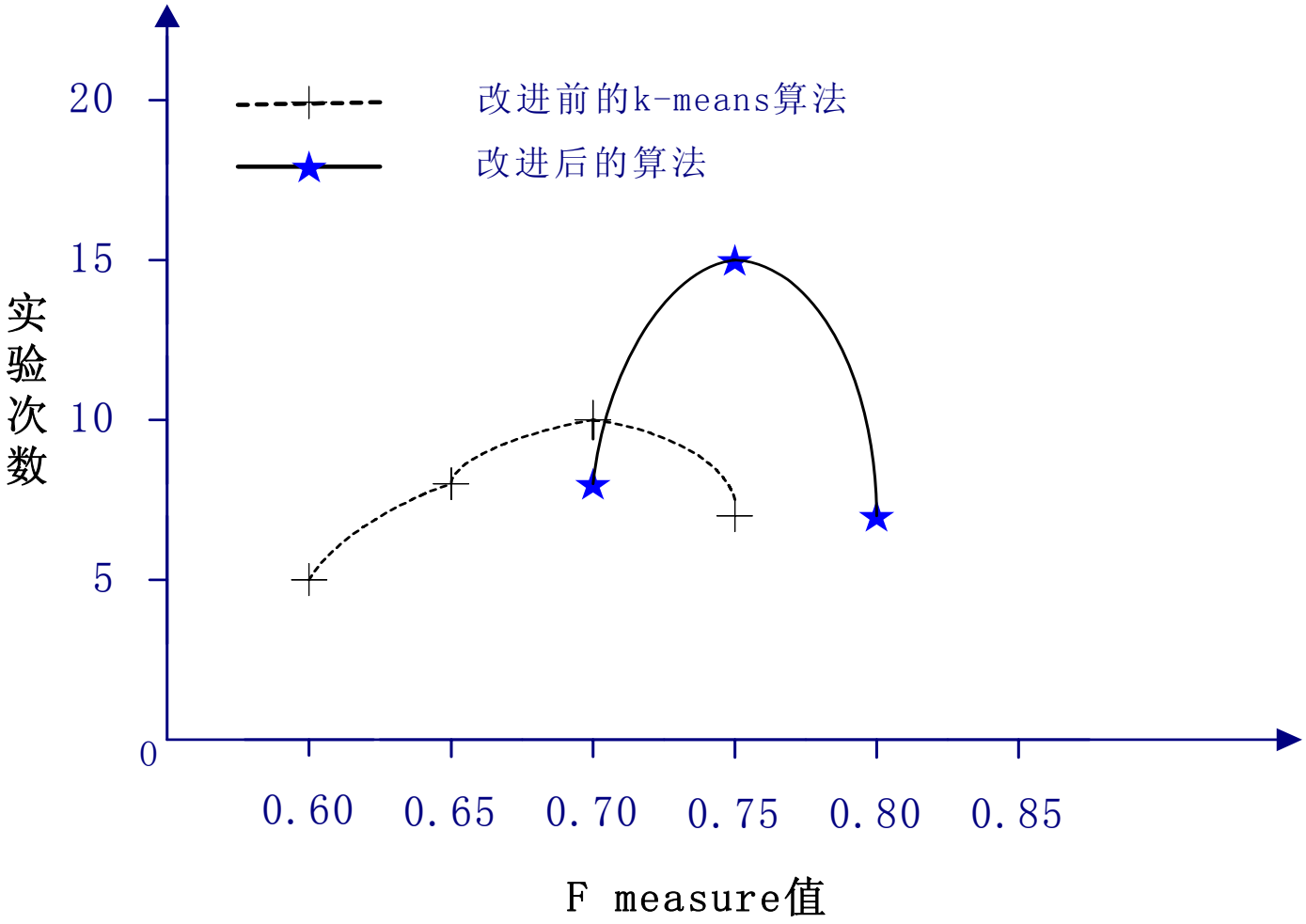
  避免孤立点的影响，提高稳定性和效果。

- **改进思想**

  将聚类均值点与聚类种子相分离。



对比实验结果


20
15
10
5
0
实验次数
改进前的k-means算法
改进后的算法
0.60
0.65
0.70
0.75
0.80
0.85
F measure值

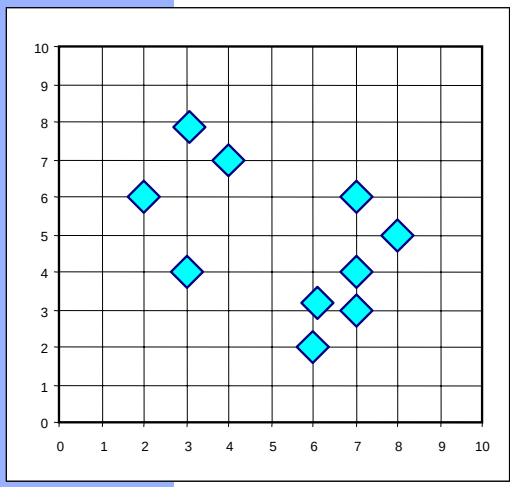
k-mediods(k-中心点)方法

- 基本策略：不采用簇中样本的平均值作为参照点，选用簇中位置最中心的对象——中心点作为参照点。
- PAM(Partitioning Around Medoids围绕中心点划分)
  - 最早提出的k-中心点算法之一(1987)；
  - 基本思想：最初随机选择k个中心点后，反复尝试找更好的中心点

PAM算法(1987)

- 1.随机选择k个代表对象作为初始的中心点。
- 2.repeat
- 3.指派每个剩余对象给离它最近的中心点所代表的簇；
- 4.随机的选择一个非中心点对象O_{random}
- 5.计算用O_{random}代替O_j的总代价
- 6.if s为负，则O_{random}代替O_j，形成新的k个中心点的集合
- 7.Until不发生变化

Typical k-medoids algorithm (PAM)

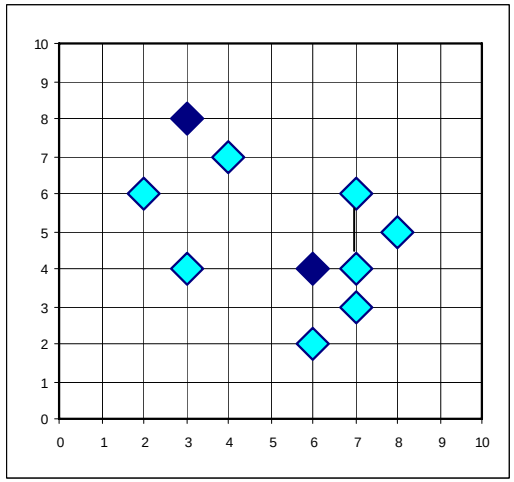


K=2

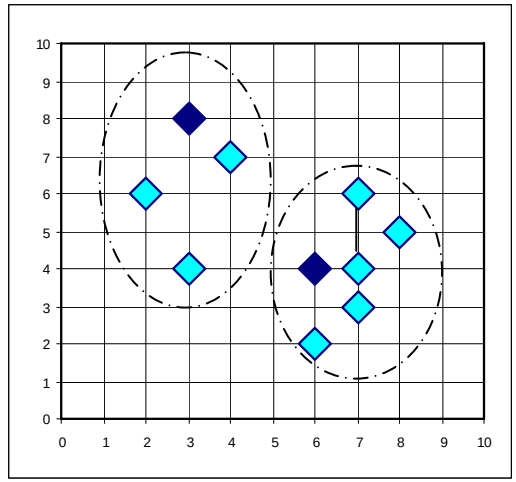
Do loop

Until no change

Arbitrary choose k object as initial medoids

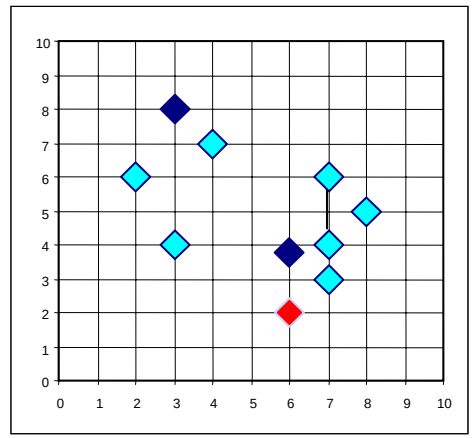


Assign each remaining object to nearest medoids

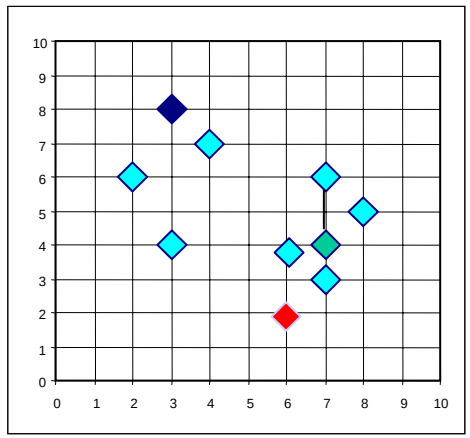


Total Cost = 20

Randomly select a nonmedoid object, O_{ramdom}



Compute total cost of swapping



Total Cost = 26

Swapping O and O_{ramdom}

If quality is improved.

PAM算法分析

- 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 (1990)

- CLARA (Clustering Large Applications)
- (Kaufmann and Rousseeuw in 1990)
- 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 (1994)

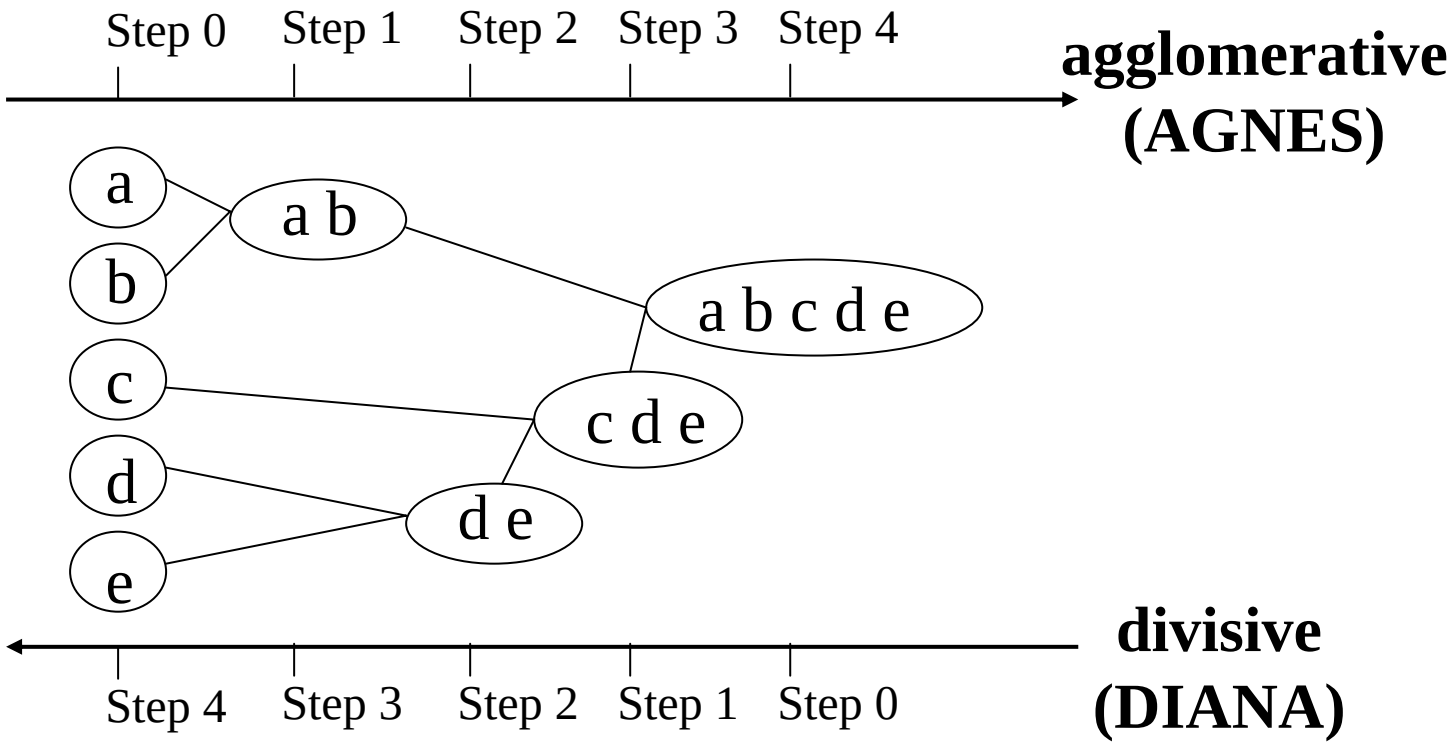
- *CLARANS* (A Clustering Algorithm based on Randomized Search) (Ng and Han'94)
- CLARANS draws sample of neighbors dynamically
- The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of *k* medoids
- If the local optimum is found, *CLARANS* starts with new randomly selected node in search for a new local optimum
- It is more efficient and scalable than both *PAM* and *CLARA*

层次聚类

层次聚类

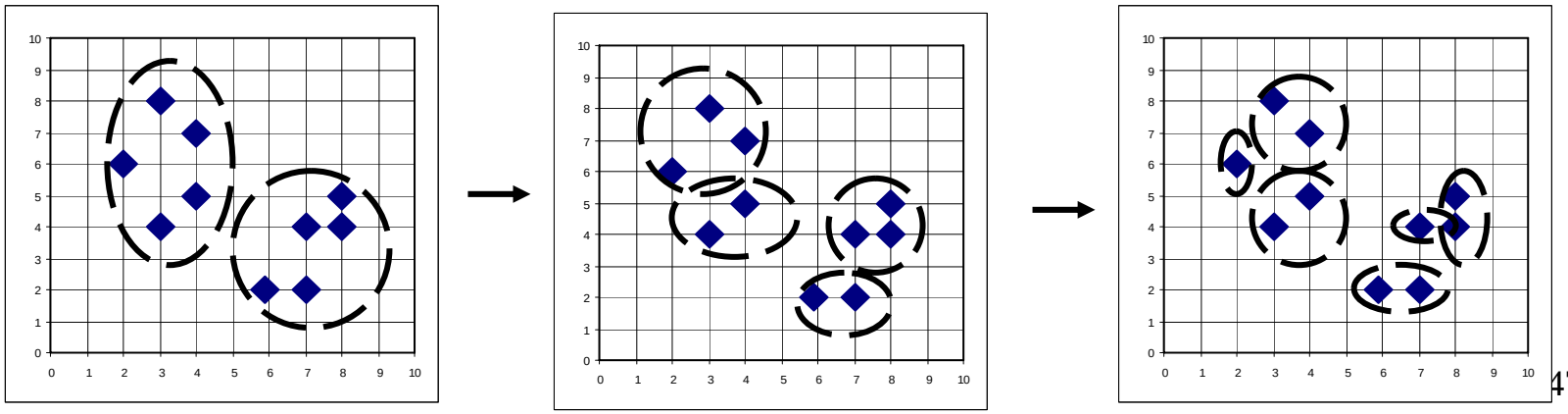
- 自底向上的聚类（凝聚）
  - 每一项自成一类
  - 迭代，将最近的两类合为一类
- 自顶向下的聚类（分裂）
  - 将所有项看作一类
  - 找出最不相似的项分裂出去成为两类

层次聚类


Step 0
Step 1
Step 2
Step 3
Step 4
agglomerative
(AGNES)
a
b
c
d
e
a b
c d e
d e
a b c d e
divisive
(DIANA)
Step 4
Step 3
Step 2
Step 1
Step 0

DIANA (1990)

- DIANA (Divisive Analysis)
- Kaufmann and Rousseeuw (1990)
- Inverse order of AGNES
- Eventually each node forms a cluster on its own

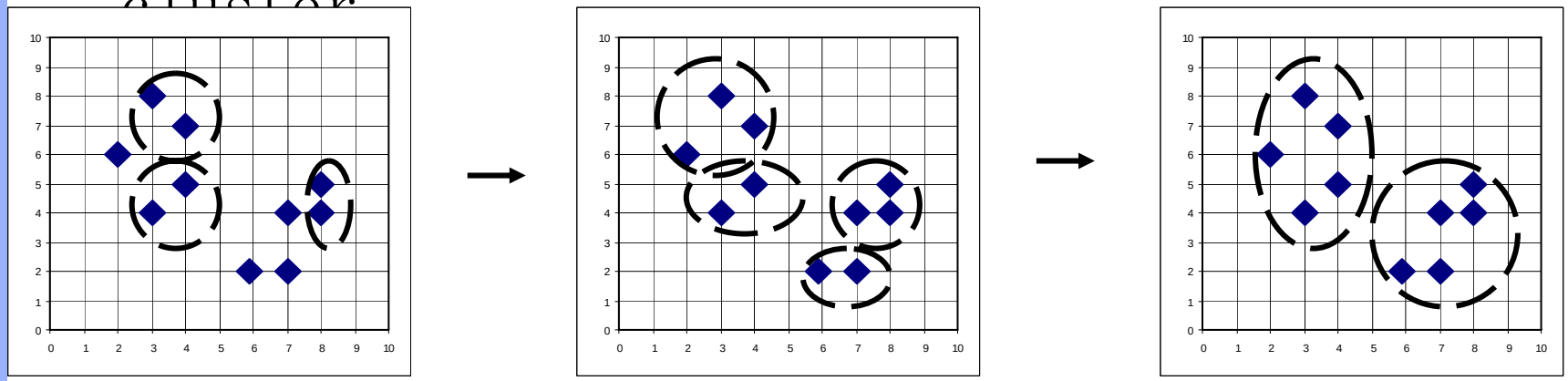


凝聚层次聚类

- 将文档集D=$\{d_1, \ldots, d_i, \ldots, d_n\}$ 中的每一个文档d_i看作是一个具有单个成员的类$C_i=\{d_i\}$，这些类构成了D的一个聚类C=$\{c_1, \ldots, c_i, \ldots, c_n\}$；
- 计算C中每对类（ c_i， c_j ）之间的相似度sim(c_i， c_j)；
- 选取具有最大相似度的类对arg max sim(c_i， c_j)，并将c_i 和c_j合并为一个新的类$c_k=c_i \cup c_j$ ，从而构成D的一个新的类C=$\{c_1, \ldots, c_{n-1}\}$；
- 重复上述步骤，直到C中只剩下一个类为止。

AGNES (1990)

- AGNES (Agglomerative Nesting)
- Kaufmann and Rousseeuw (1990)
- Use the Single-Link method and the dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



AGNES (1990)

- 一．单连接算法(single-linkage)（最近邻(Nearest Neighbor)）：
  - 基本思想：两个簇之间的距离用从两个簇中抽取的每对样本的最小距离

$$D_{\min}(C_i, C_j)$$

  - 作为距离度量，一旦最近的两个类的距离超过某个任意给定的阈值，算法就自动结束。
- 二．全连接算法
- 三．平均连接算法

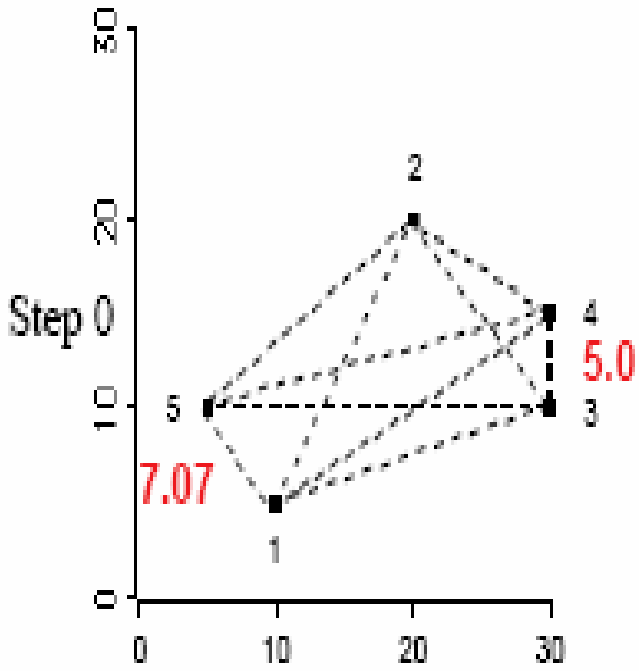
单连接算法

- 先将五个样本都分别看成是一个簇，最靠近的两个簇是3和4，因为他们具有最小的簇间距离
- D（3，4）=5.0。

第一步：合并簇3和4，得到新簇集合1,2,（34）,5

	x1	x2
1	10	5
2	20	20
3	30	10
4	30	15
5	5	10

	1	2	3	4	5
1	0.00				
2	18.0	0.00			
3	20.6	14.1	0.00		
4	22.4	11.2	5.00	0.00	
5	7.07	18.0	25.0	25.5	0.00



单连接算法

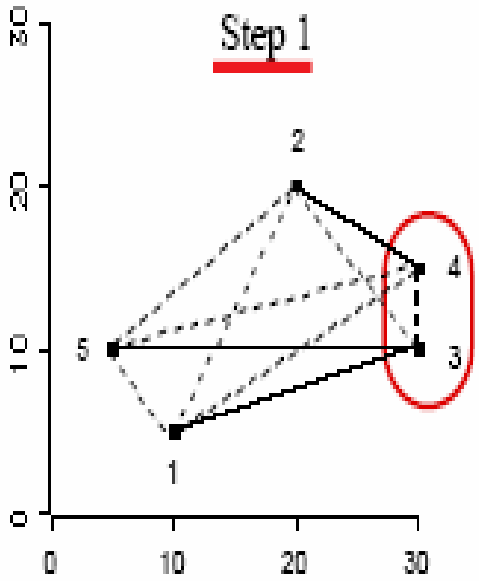
- 更新距离矩阵：
- D(1,(34)) = min(D(1,3),D(1,4)) = min(20.6, 22.4) = 20.6;
- D(2,(34)) = min(D(2,3),D(2,4)) = min(14.1, 11.2) = 11.2;
- D(5,(34)) = min(D(3,5),D(4,5)) = min(25.0, 25.5) = 25.0.
- 原有簇1,2,5间的距离不变，修改后的距离矩阵如图所示，在四个簇1,2,(34),5中，最靠近的两个簇是1和5，它们具有最小簇间距离D（1,5）＝7.07。

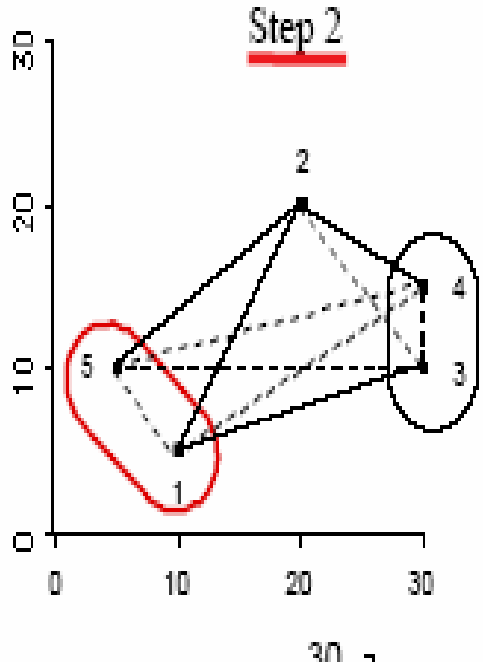
单连接算法

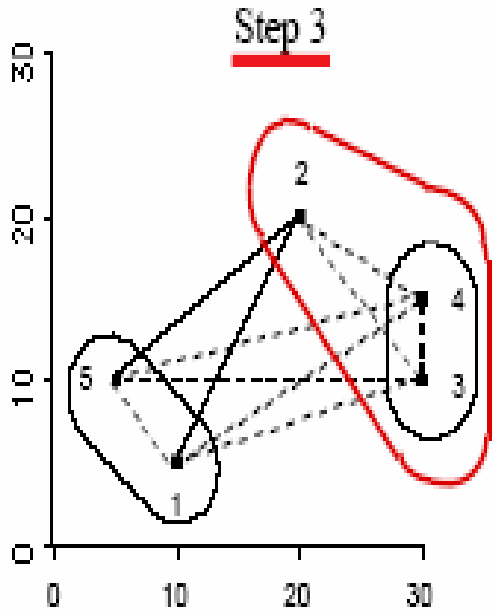
	1	2	5	(34)
1	0.00			
2	18.0	0.00		
5	7.07	18.0	0.00	
(34)	20.6	11.2	25.0	0.00

	2	(34)	(15)
2	0.00		
(34)	11.2	0.00	
(15)	18.0	20.6	0.00

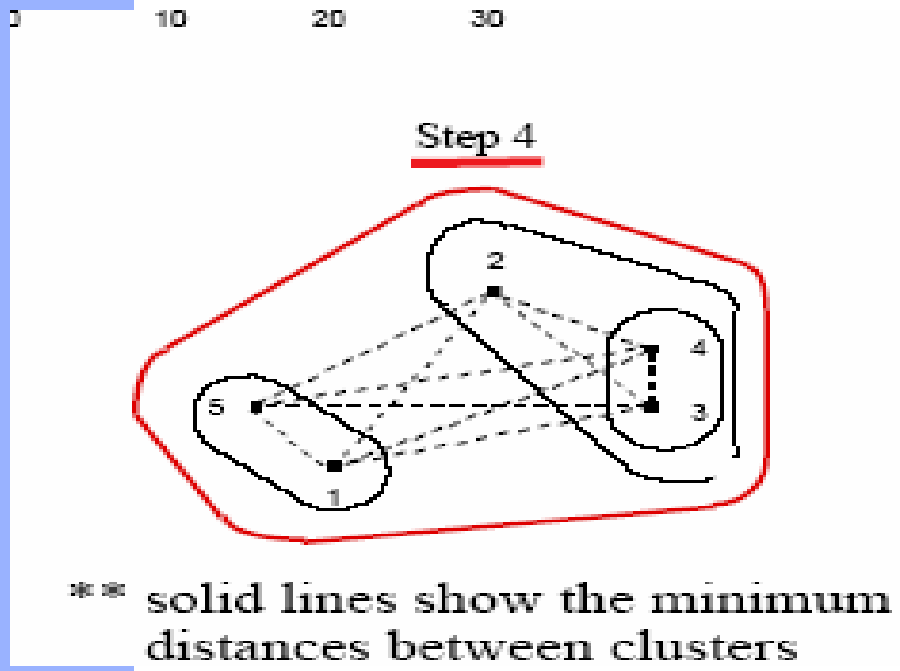
	(15)	(234)
(15)	0.00	
(234)	18.0	0.00

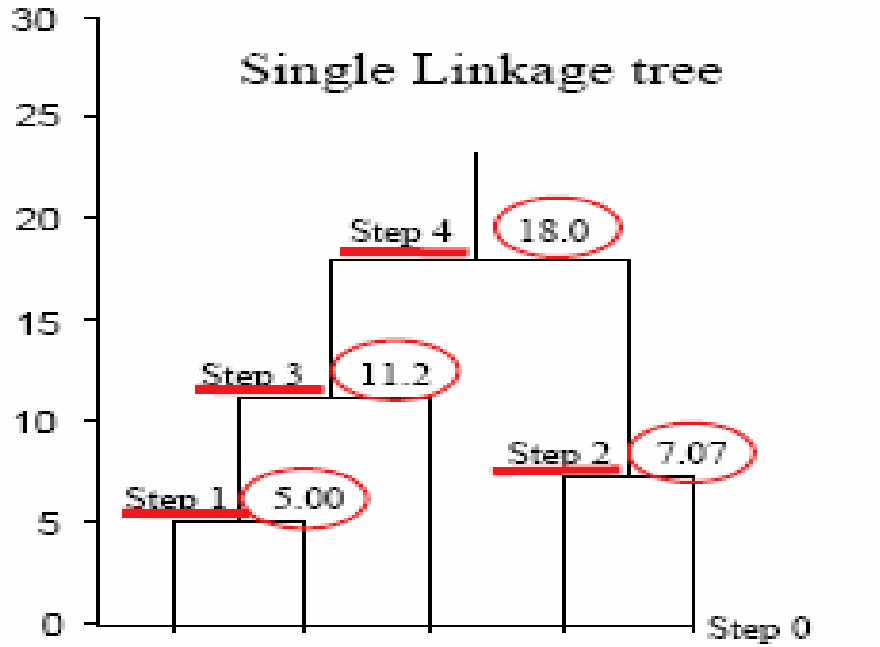






单连接算法

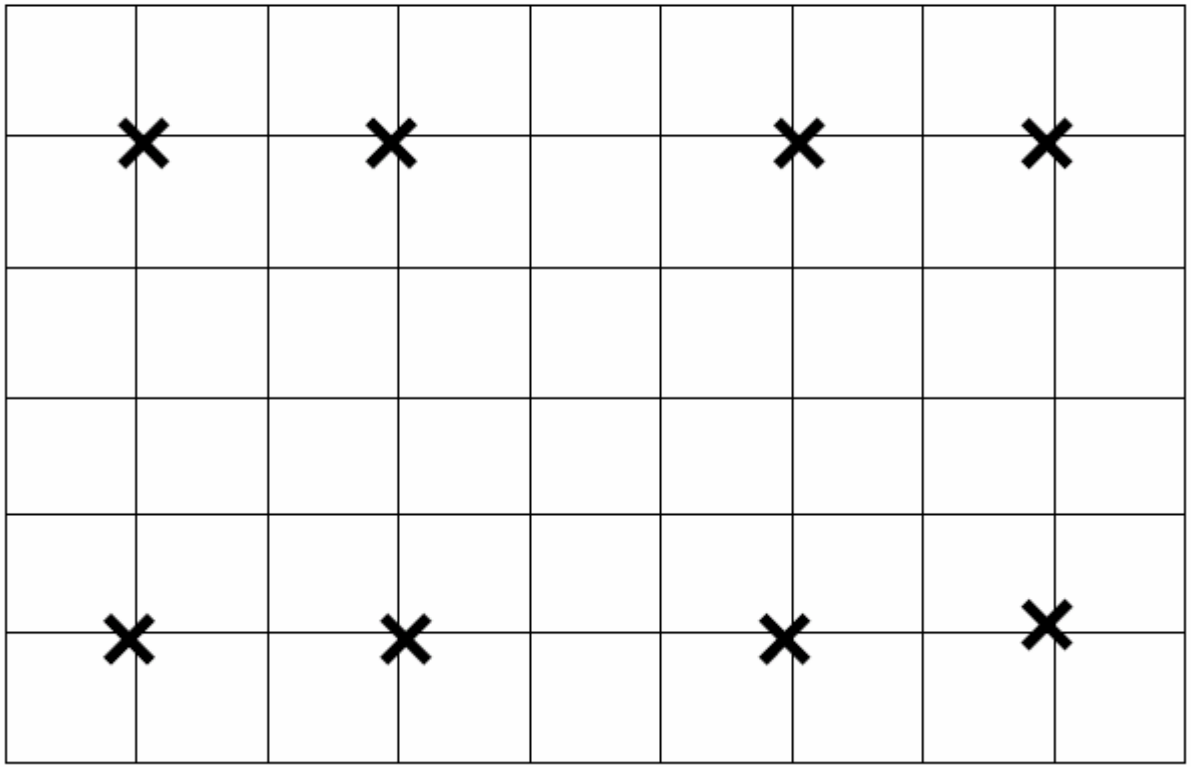

0
10
20
30
Step 4
2
4
5
3
1
** solid lines show the minimum distances between clusters


Single Linkage tree
30
25
20
15
10
5
0
Step 4
18.0
Step 3
11.2
Step 2
7.07
Step 1
5.00
Step 0

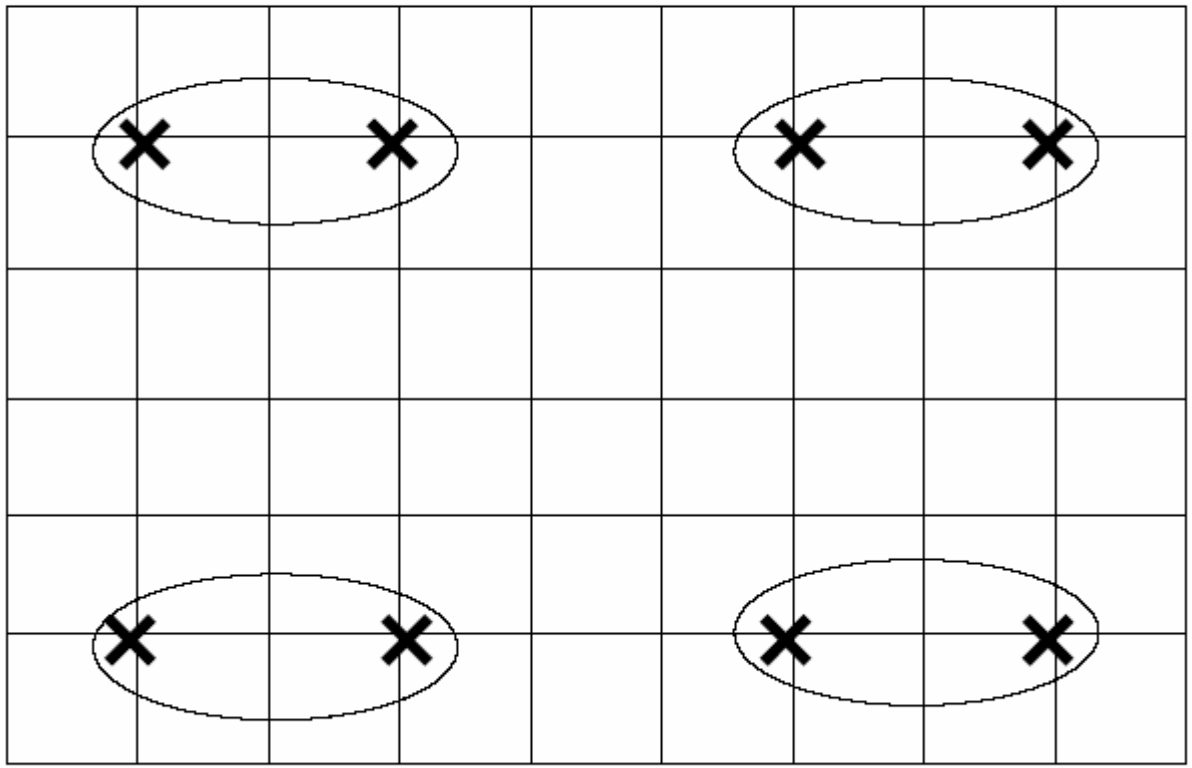
类的相似度度量

- 类与类之间的相似度三种方法：
  - 单连接方法
  - 全连接方法
  - 组平均方法

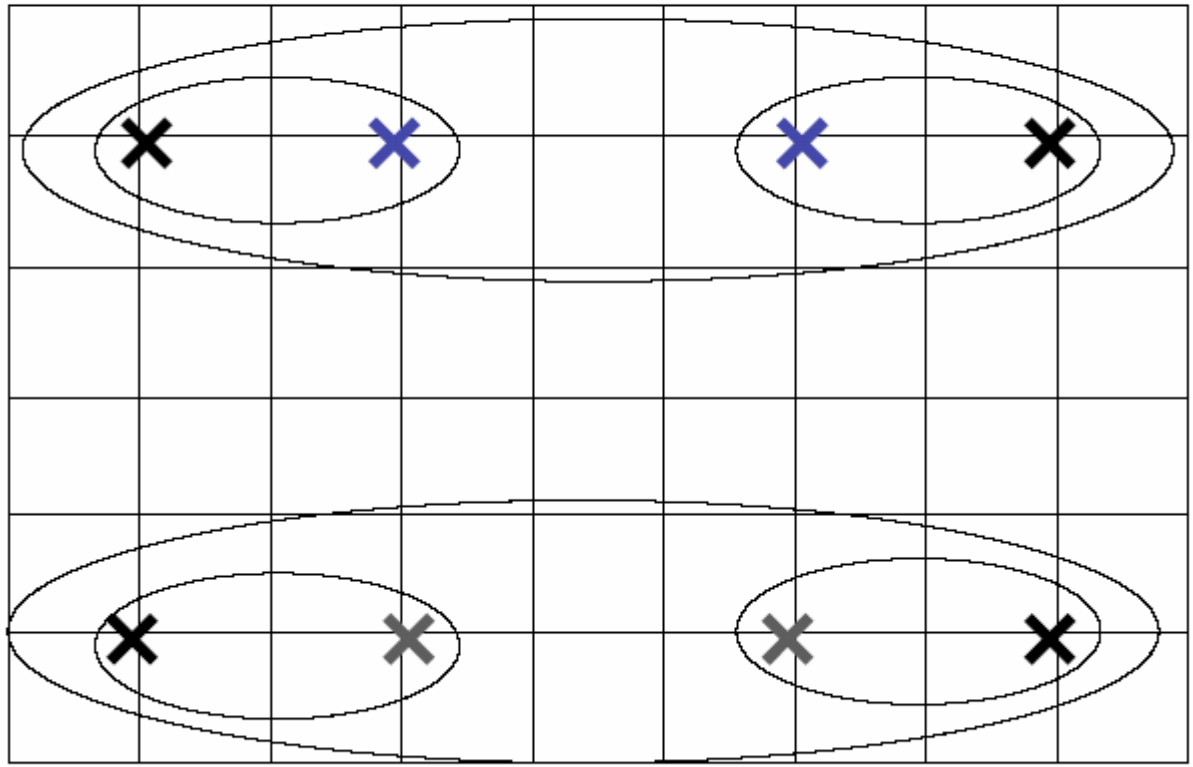
Single vs. Complete link



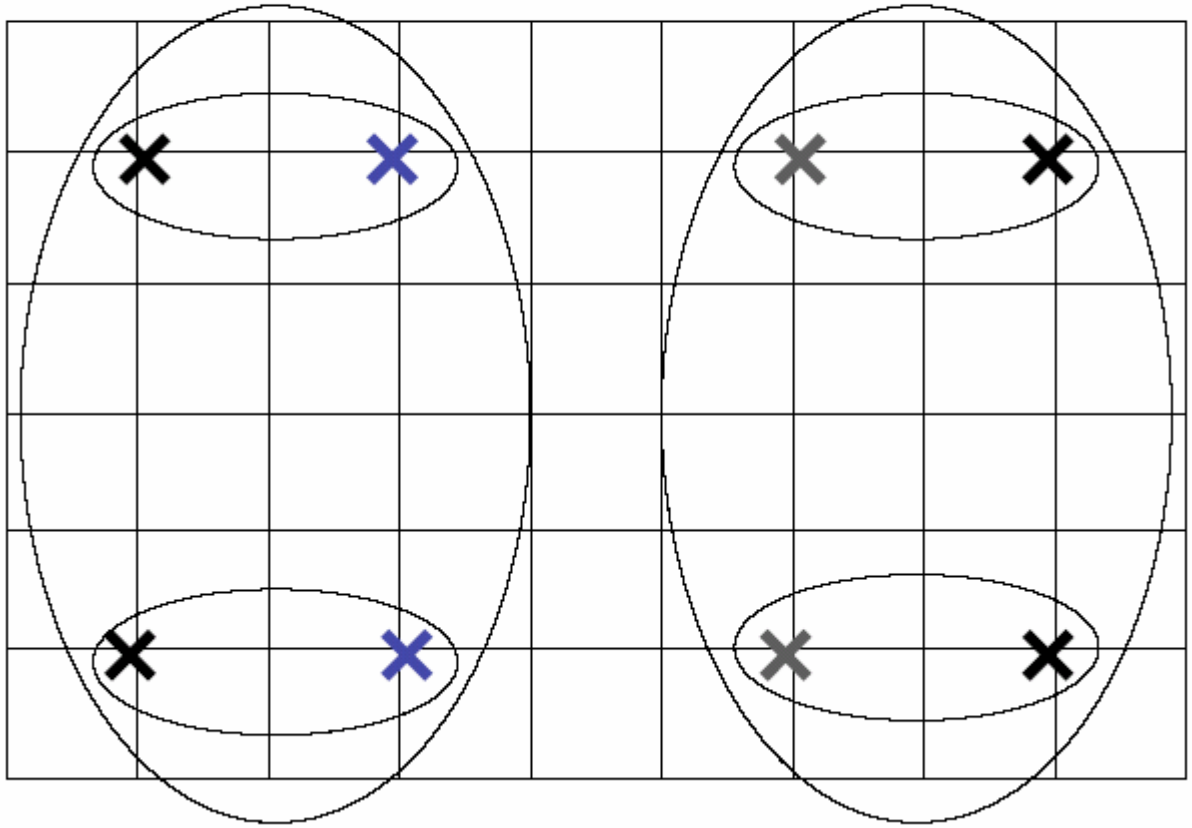
Single vs. Complete link



Single link



Complete link



AGNES算法分析

- Major weakness of agglomerative clustering methods
  - do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
  - can never undo what was done previously

改进的层次聚类

- BIRCH(1996)是一个综合的层次聚类方法，它引入了两个概念：聚类特征和聚类特征树（CF树）
- CURE(1998)采用了一种新颖的层次聚类算法，该算法选择基于质心和基于代表对象方法之间的中间策略。
- ROCK方法是一个适用于分类属性层次聚类算法
- Chameleon（变色龙）(1999)是一个在层次聚类中采用动态模型的层次聚类算法

BIRCH (1996)

- Birch: Balanced Iterative Reducing and Clustering using Hierarchies
- by 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) （用树状结构对对象进行层次划分）
  - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree （其它算法对叶节点进行聚类）

CF-Tree in BIRCH

- CF is a compact storage for data on points in a cluster
- Has enough information to calculate the intra-cluster distances
  - summary of the statistics for a given subcluster.
- Additivity theorem allows us to merge sub-clusters

Clustering Feature Vector

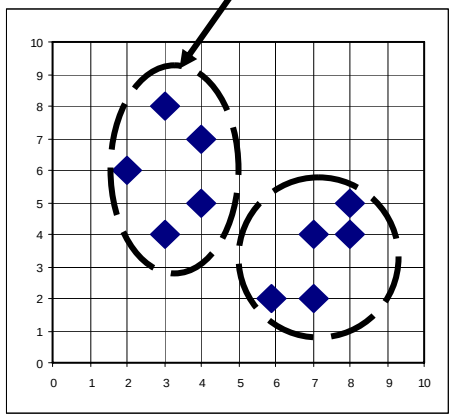
Clustering Feature: $CF = (N, \overrightarrow{LS}, SS)$

N: **Number of data points**

LS: $\sum_{i=1}^{N} \overrightarrow{X_i}$

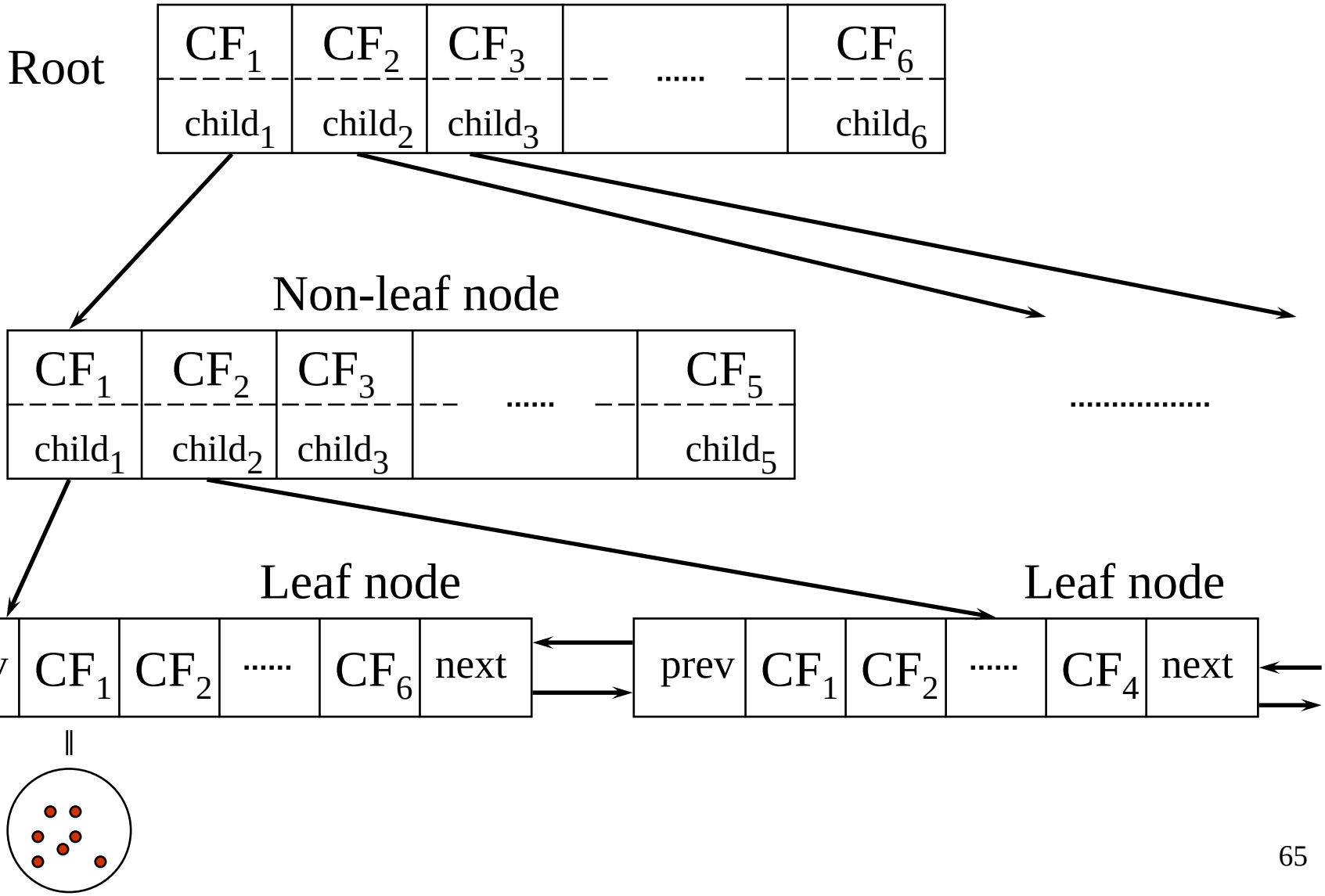
SS: $\sum_{i=1}^{N} \overrightarrow{X_i}^2$

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



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

CF Tree


Root
CF1 CF2 CF3 ..... CF6
child1 child2 child3 child6
Non-leaf node
CF1 CF2 CF3 ..... CF5
child1 child2 child3 child5
Leaf node
Leaf node
prev CF1 CF2 ..... CF6 next
prev CF1 CF2 ..... CF4 next

CF-Tree in BIRCH

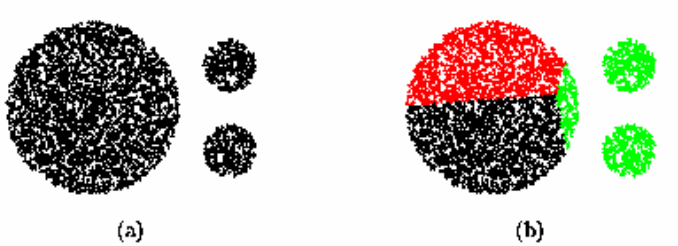
- A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering
  - A nonleaf node in a tree has descendants or "children"
  - The nonleaf nodes store sums of the CFs of their children
- A CF tree has two parameters
  - Branching factor: specify the maximum number of children.
  - threshold: max diameter of sub-clusters stored at the leaf nodes

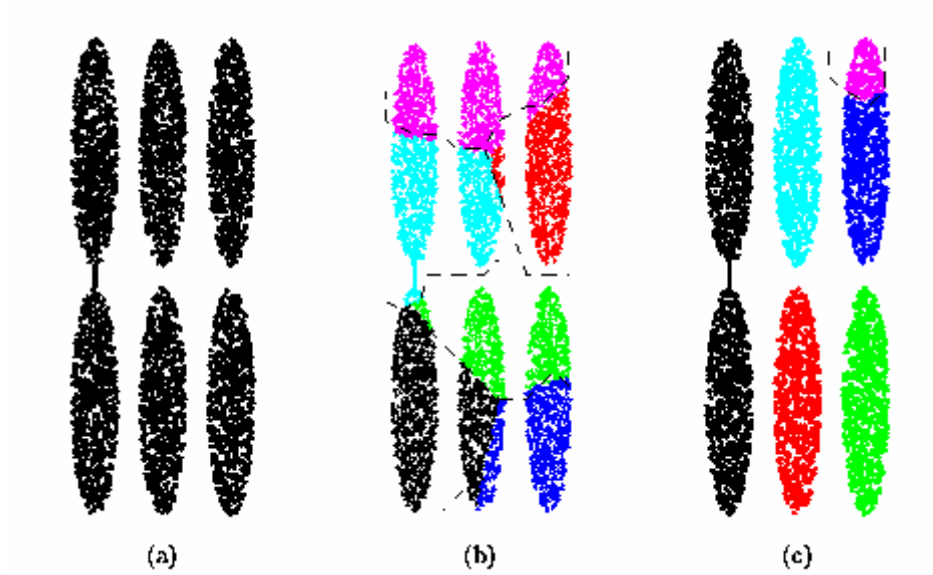
BIRCH优缺点

- *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.

基于距离计算方法的缺陷

- Consider only one point as representative of a cluster
- Good only for convex shaped, similar size and density, and if k can be reasonably estimated





CURE (1998)

- CURE (Clustering Using REpresentatives)
- proposed by Guha, Rastogi & Shim, 1998
- Shrink the multiple representative points towards the gravity center by a fraction of α.
- Multiple representatives capture the shape of the cluster

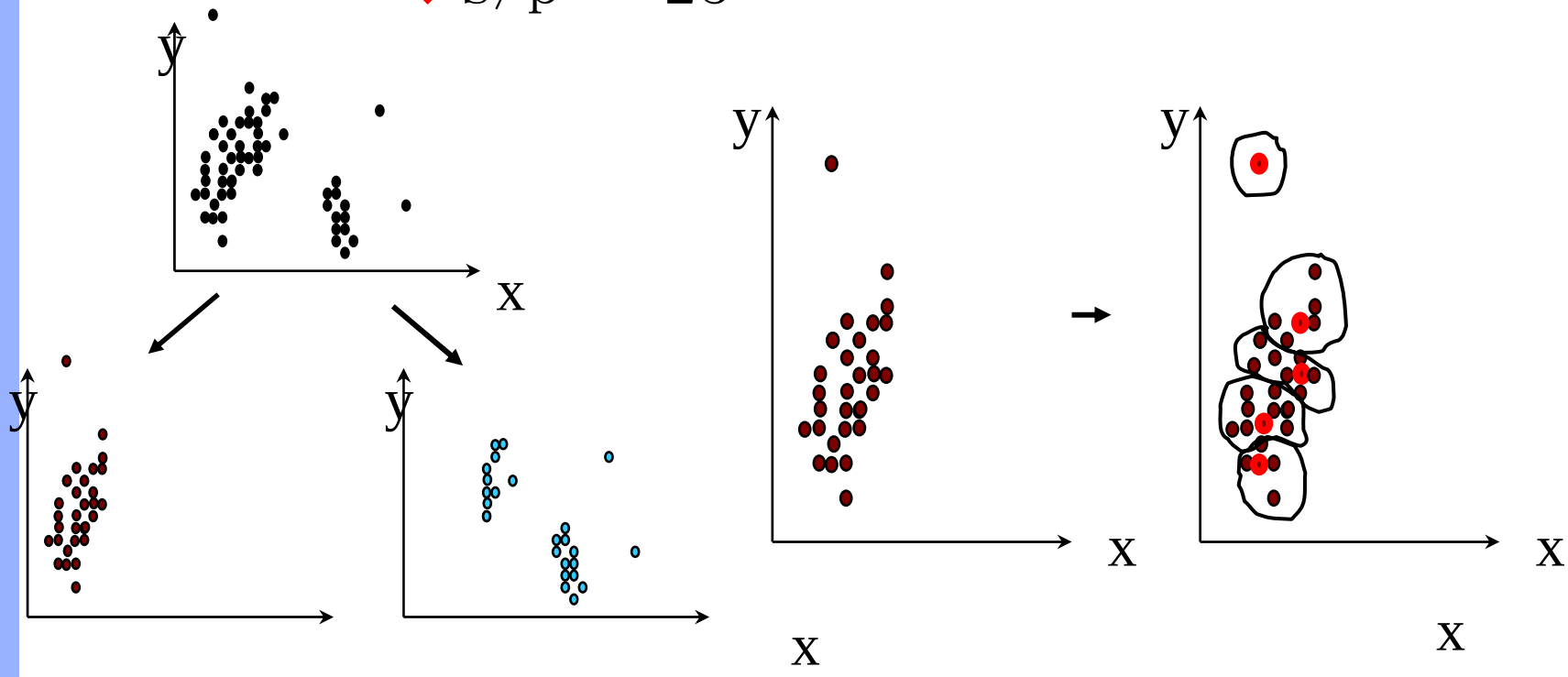
CURE 算法

- 从源数据对象中抽取一个随机样本S
- 将样本分割为一组划分p（size s/p）
- 对每个划分局部地聚成 s/pq个 clusters
- 随机取样剔除孤立点，去掉增长较慢的类
- 对局部的簇进行聚类
  - 落在每个新形成的簇中的代表点根据用户定义的收缩因子a收缩或向中心移动，这些点代表或捕捉到簇的形状
- 用相应的簇标签来标记数据

CURE 算法示例

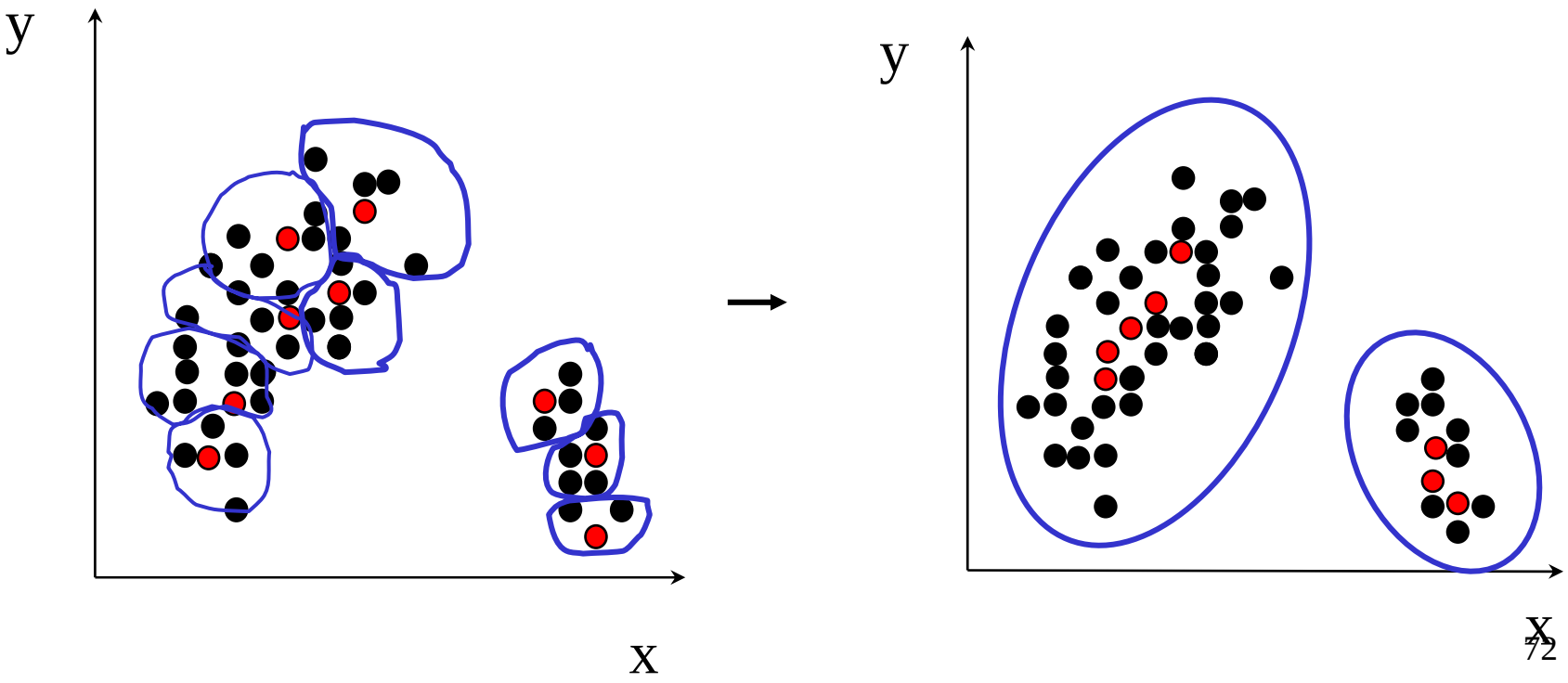
- s = 50
- p = 2
- s/p = 25

- s/pq = 5



CURE收缩代表点

- Shrink the multiple representative points towards the gravity center by a fraction of α.
- Multiple representatives capture the shape of the cluster



基于密度的方法

基于密度的方法

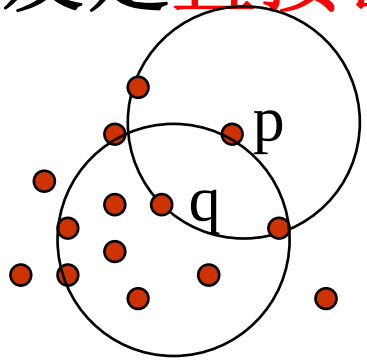
- 基于样本之间的距离的聚类方法只能发现球状的簇；
- 基于密度的方法可用来过滤“噪声”孤立点数据，以发现任意形状的簇。
- 主要思想：只要临近区域的密度（样本的数目）超过某个阈值则继续聚类。即对于给定簇中的每个样本，在一个给定范围的区域中必须至少包含某个数目的样本。

基于密度的方法

- 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)

基于密度聚类的相关定义

- a. 给定对象半径 ε 内的区域称为该对象的 ε —邻域。
- b. 如果一个对象的 ε —邻域至少包含最小数目MinPts个对象，则称该对象为核心对象。
- c. 给定一个对象集合D，如果p是在q的 ε —邻域内，而q是一个核心对象，则称对象p从对象q出发是直接密度可达的。

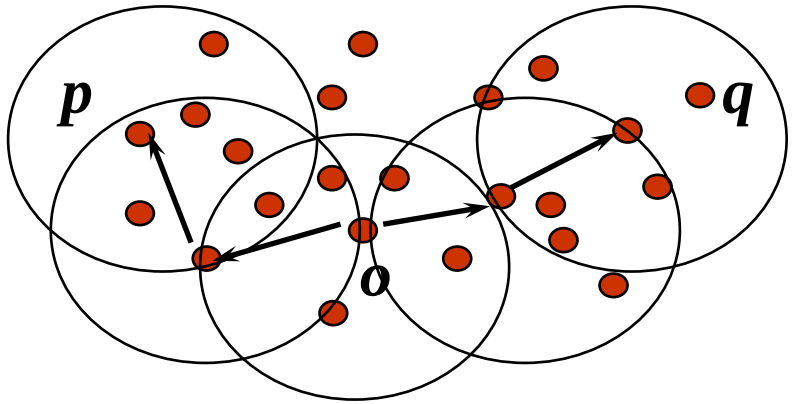


MinPts = 5

Eps = 1 cm

基于密度聚类的相关定义

- d. 如果存在一个对象链$p_1, p_2, \cdots, p_n$，$p_1 = q$，$p_n = p$，对$p_i \in D$（$1 <= i <= n$），p_{i+1}是从p_i关于 ε 和MinPts直接密度可达的，则对象p是从对象q关于 ε 和MinPts密度可达的。
- e. 如果对象集合D中存在一个对象o，使得对象p和q是从o关于 ε 和MinPts密度可达的，那么对象p和q是关于 ε 和MinPts密度相连的。

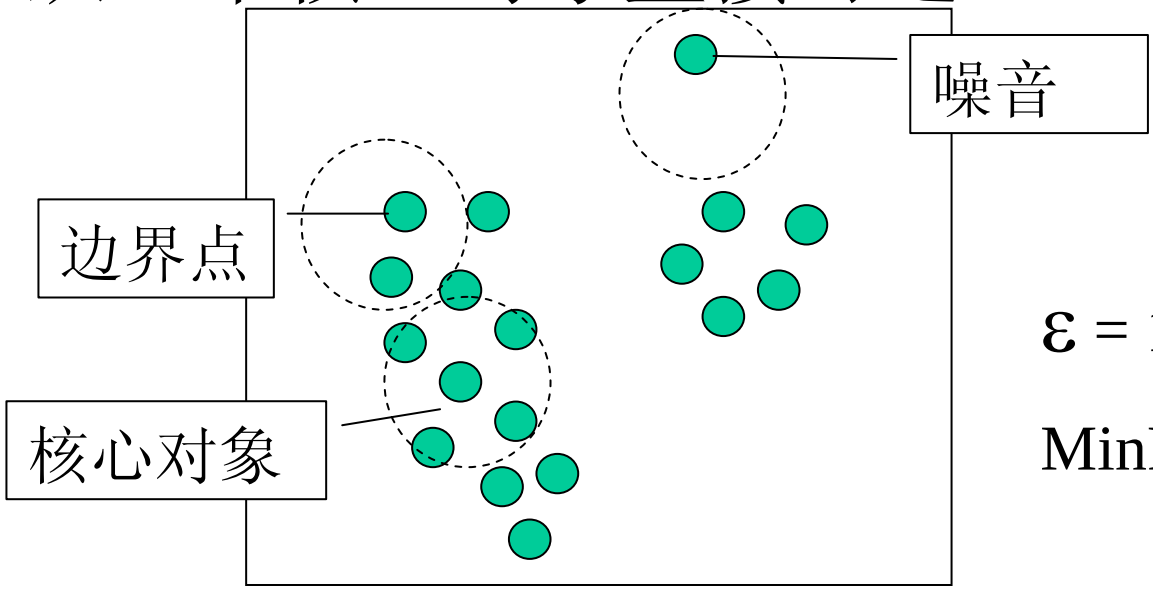


DBSCAN (KDD-96)

- DBSCAN （Density-Based Spatial Clustering of Applications with Noise）
- 基于高密度连接区域的密度聚类方法
- Martin Ester, KDD-96

DBSCAN基本思想

- 簇：基于密度可达性的最大的密度相连对象的集合
- 噪音：不在任何簇中的对象
- 边界对象：不是核心对象，但在簇中，即至少从一个核心对象直接可达

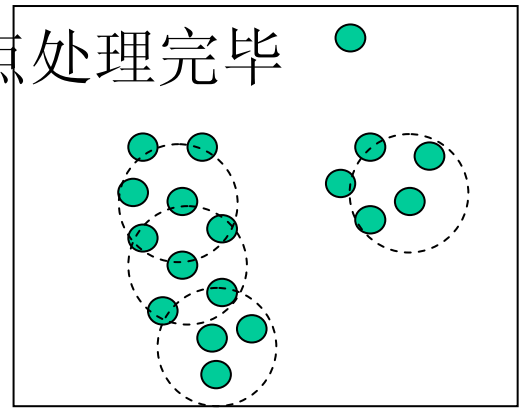


DBSCAN算法

- 1）任意选择没有加簇标签的点 p
- 2）找到从p关于ε and $MinPts$ 密度可达的所有点
- 3）如果$|N\varepsilon(q)| \geq MinPts$ ，则p是核心对象，形成一个新的簇，给簇内所有的对象点加簇标签
- 4）如果p 是边界点，则处理数据库的下一点
- 5）重复上述过程，直到所有的点处理完毕

ε = 1cm

MinPts = 5



DBSCAN算法的不足和改进

- 只能发现密度相仿的簇
- 对用户定义的参数（ ε and MinPts ）敏感
- 计算复杂度为O(n^2)

- 采用R-树等空间索引技术，计算复杂度为o(nlogn)

OPTICS (SIGMOD'99)

- OPTICS:Ordering Points To Identify the Clustering Structure（通过对象排序识别聚类结构）
- Mihael Ankerst .ACM SIGMOD'99 Int.Conf,1999
- 对DBSCAN的改进
  - 对输入参数不敏感
  - 可以发现不同密度的簇
  - 用图表等可视化的方式来表示
  - 按可达距离排序
  - 可自动挖掘，也可与用户交互

OPTICS 引入两个新概念

- P为对象，数据集D，ε为距离值，MinPts，Nε(q)为邻域对象数
- P的核心距离：
  - 使得P成为核心对象的最小ε
  - 若|(Nε(q)|< MinPts，即P不是核心对象，则无定义，即无穷大
- P关于对象q的可达距离：p的核心距离和p,q的欧氏距离之间的较大值
  - Max（核心距离，|（p,q）|）
  - 若|Nε(q)|< MinPts，即P不是核心对象，则无定义

OPTICS 概念图示

- 核心距离
- 可达距离

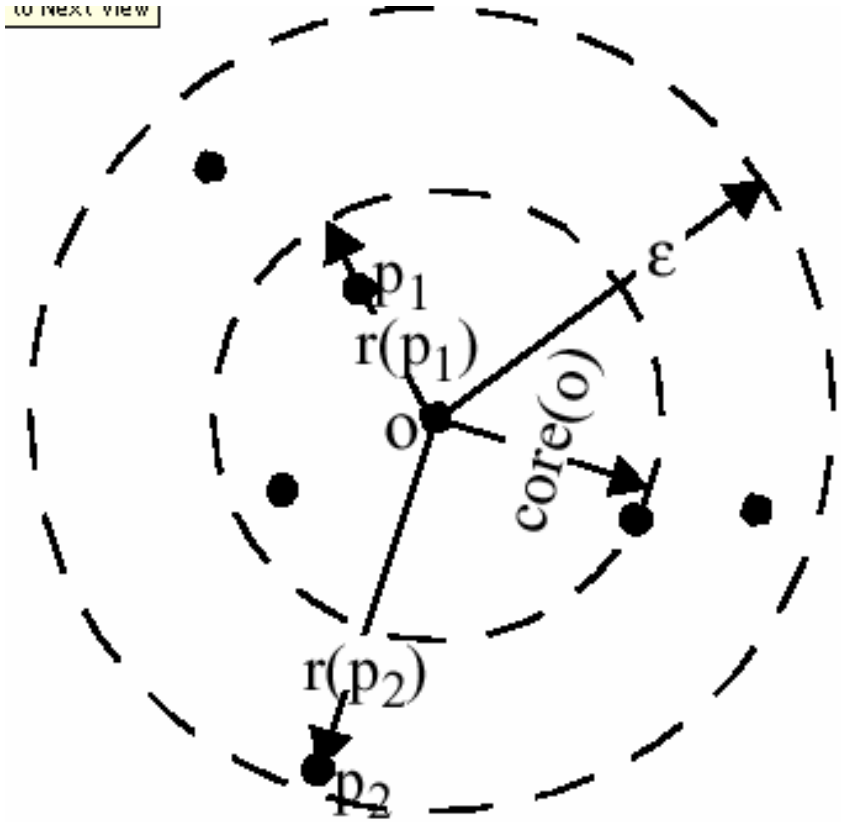


Figure 4. Core-distance(*o*), reachability-distances r(p_1,*o*), r(p_2,*o*) for *MinPts*=4

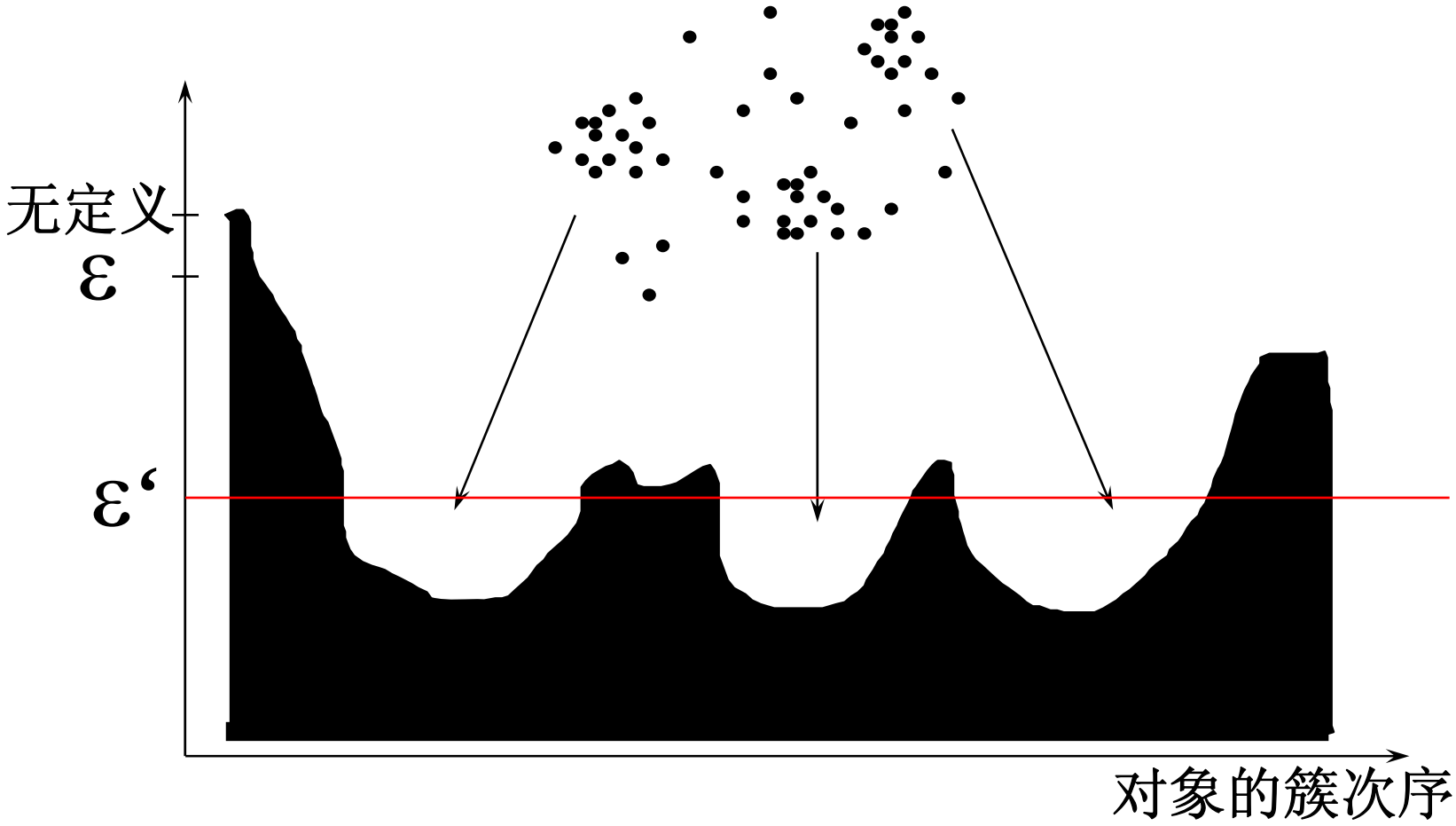
OPTICS算法

- 1. 计算数据点p的核心距离和可达距离
- 2. 如果p为核心对象，找到所有它的关于ε和MinPts的直接密度可达点，按可达距离排序并插入队列。
- 3. 处理下一个数据点

Complexity: $O(kN^2)$

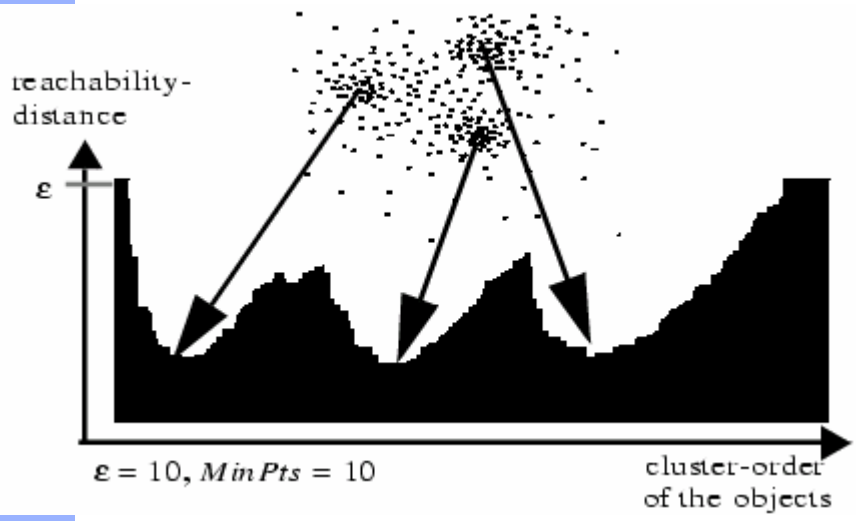
OPTICS可达距离

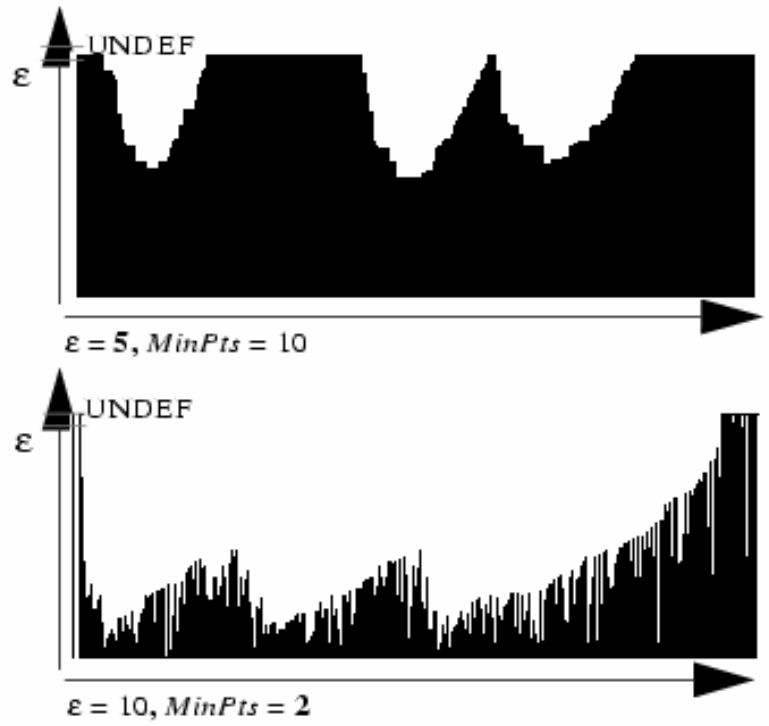
可达距离



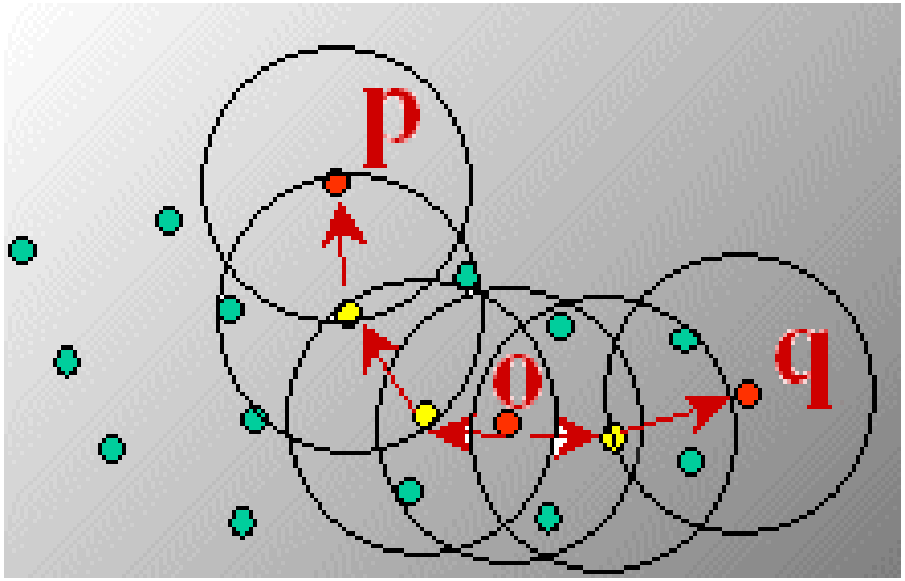
参数的影响

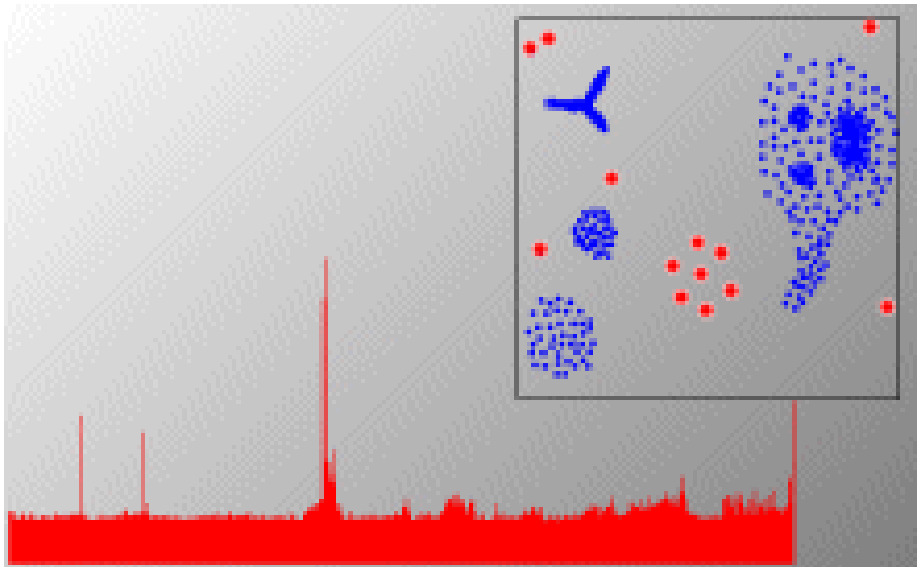
- ε减小，则可达距离为无穷大的点增多；
- MinPts减小，核心对象增多，图象更尖锐





密度聚类的应用



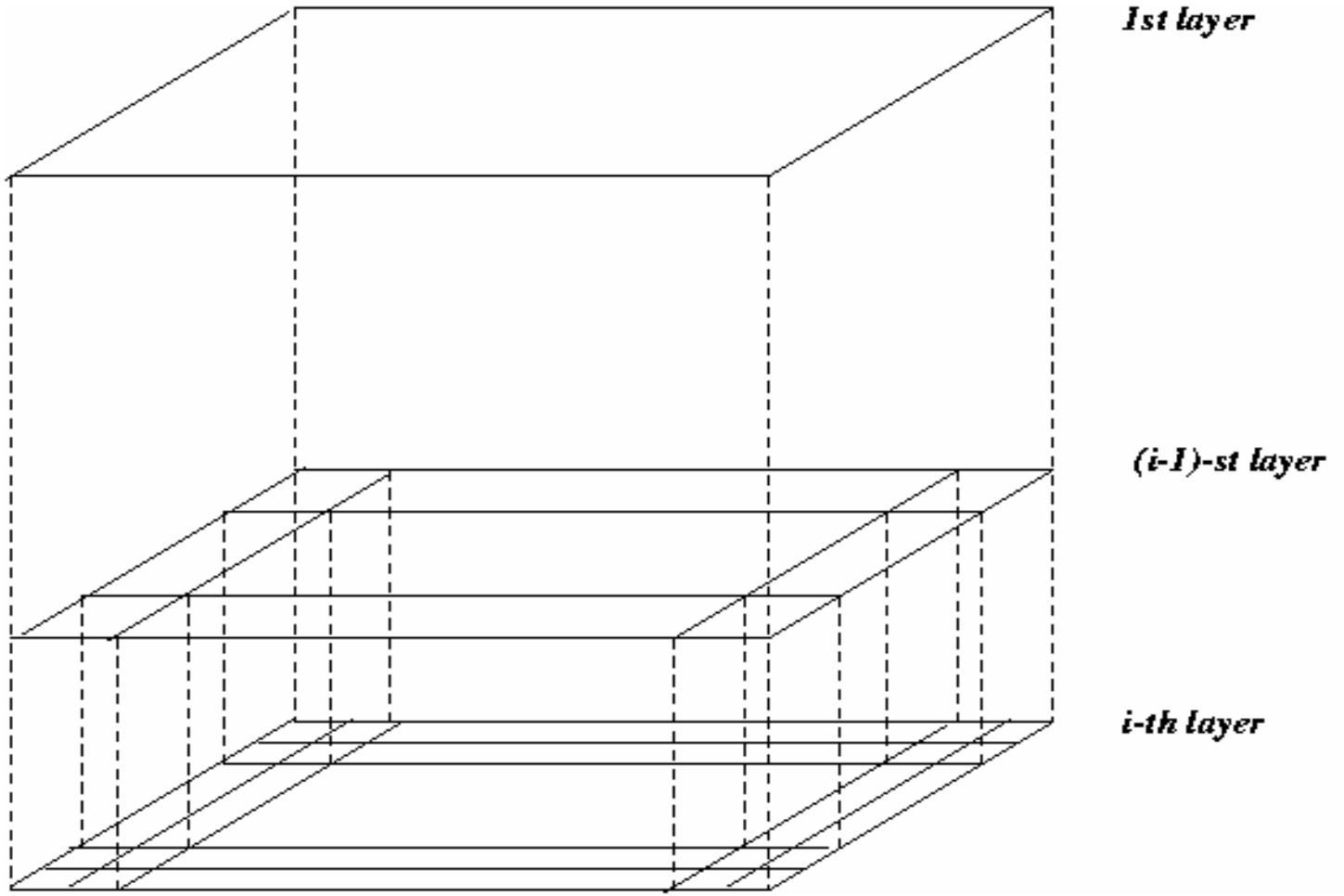


基于网格的方法

主要思想

- 数据空间区域被划分为矩形单元
- 对应于不同级别的分辨率，存在着不同级别的矩形单元：高层的每个单元被分为多个低一层的单元。
- 每个网格单元的统计信息被预先计算和存储，以供处理查询之用

主要思想

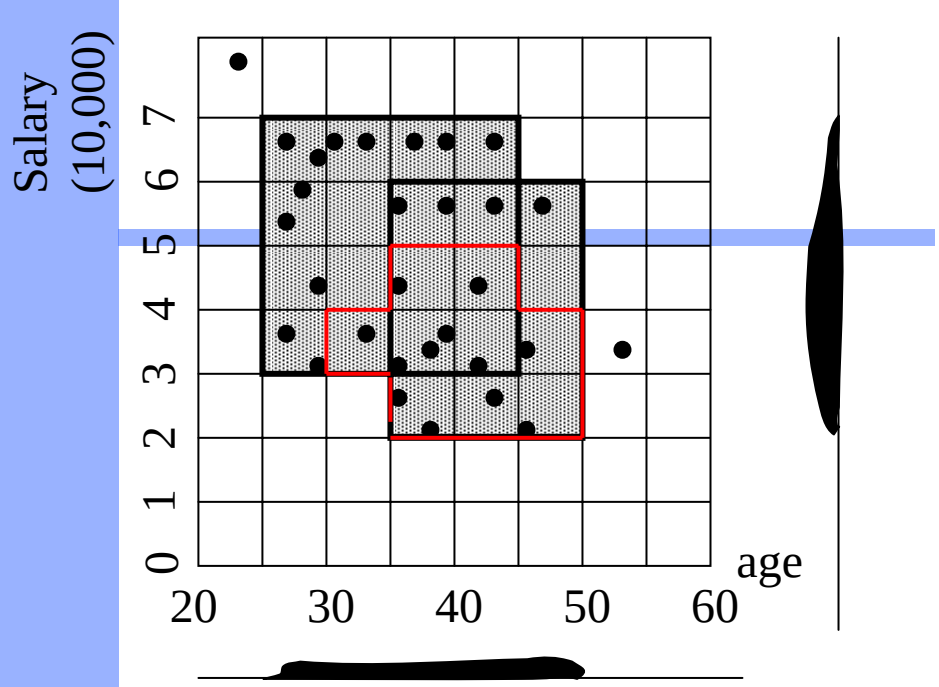


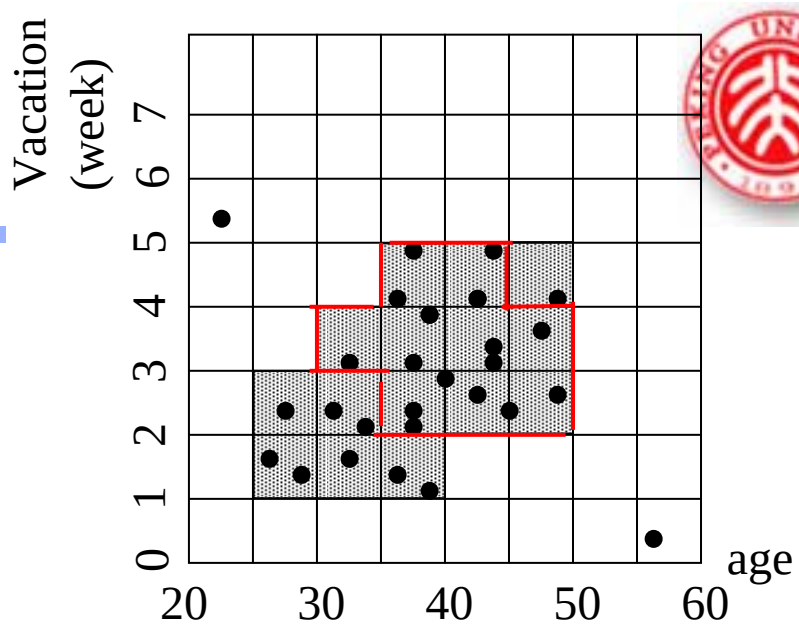
CLIQUE(1998)

- CLIQUE:Clustering In QUEst,1998
- 给定多维数据集合，数据点在数据空间中不是均衡分布的。
- 如果一个单元中的包含数据点超过了某个输入模型参数，则该单元是密集的。
- 簇：相连的密集单元的最大集合

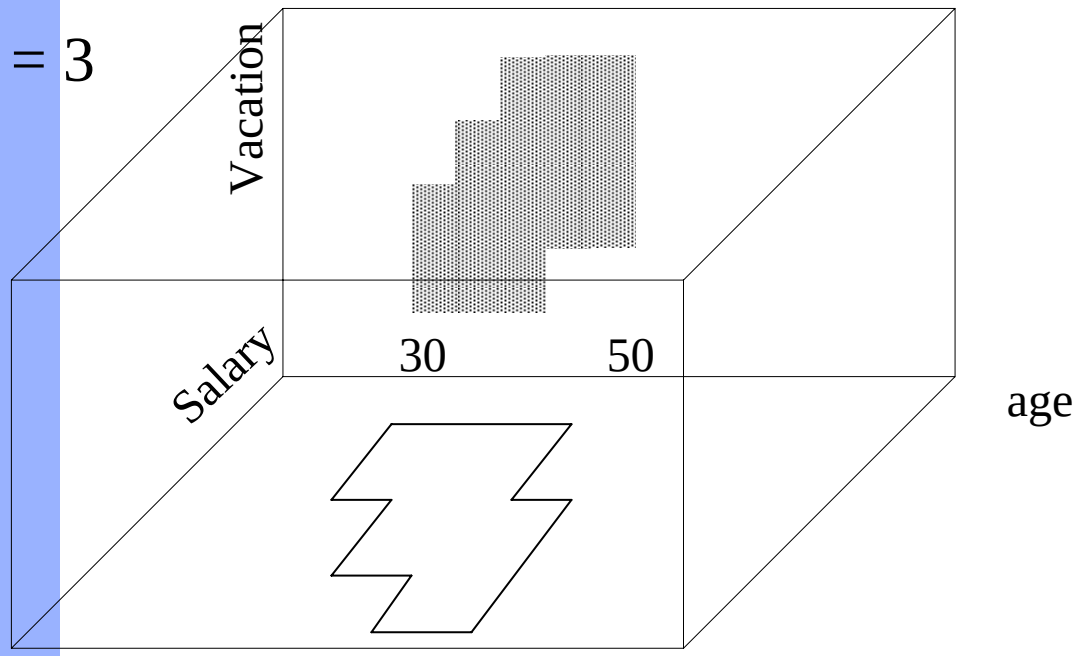
CLIQUE主要步骤

- 1. 将数据空间划分为互不相交的长方形单元，记录每个单元里的对象数
- 2. 用先验性质识别包含簇的子空间
- 3. 识别簇：
  - 在符合兴趣度的子空间中找出密集单元
  - 在符合兴趣度的子空间中找出相连的密集单元
- 4. 为每个簇生成最小化的描述
  - 先验性质：如果一个K维单元是密集的，那么它在k-1空间上的投影也是密集的。
  - 即给定一个k维的侯选密集单元，如果检查它的k-1维投影空间，发现任何一个不是密集的，那么知道第k维的单元也不可能是密集的。





$\tau = 3$



关于age对salary和vocation维的密集单元，这些密集单元相交形成更高维度密集单元的一个侯选搜索空间

CLIQUE有效性和缺点

- 自动地发现最高维的子空间，高密度聚类存在于这些子空间中。
- 对元组的输入顺序不敏感，无需假设任何规范的数据分布
- 随输入数据的大小线形地扩展，当数据的维数增加时具有良好的可伸缩性

- 聚类结果的准确度较低

孤立点分析

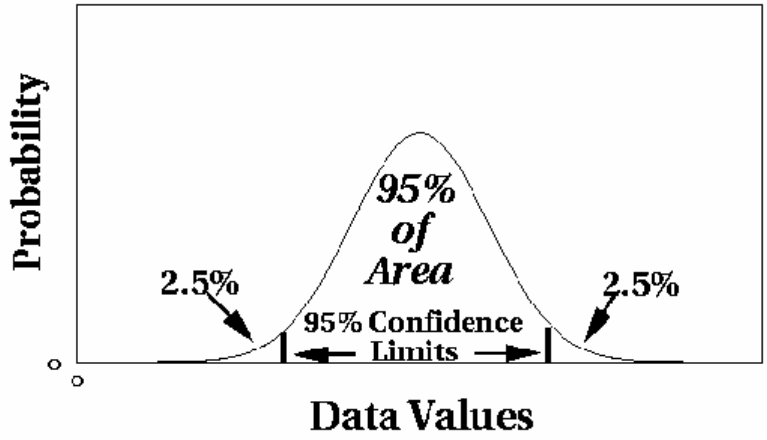
孤立点分析

- 孤立点：与数据的其他部分不同的数据对象
- 信用卡欺诈探测、收入极高或极低的客户分区、医疗分析
- 孤立点挖掘
  - 在给定的数据集合中定义什么样的数据为不一致的
  - 找到一个有效的方法来挖掘孤立点

基于统计的孤立点检测

- 工作假设（working hypothesis）

  N个数据对象的整个数据集合来自一个初始的分布模型F
- 不一致检测
  - 数据分布
  - 分布参数（平均值、变量
  - 孤立点数目的期望值
- 缺点
  - 只能在单属性上作检测
  - 大部分情况下，数据分布未知



基于距离的孤立点检测

- 解决了统计方法的主要缺陷
  - 在未知数据分布状态下做多维数据分析
- 基于距离的DB（p, d）孤立点：数据集合T中至少有p部分与对象o的距离大于d
- 主要算法
  - 基于索引的算法
  - 嵌套—循环算法
  - 基于单元的算法

基于偏离的孤立点检测

- 检查组中对象的主要特征
- 偏离主要特征的对象被认为是孤立点
  - 序列异常技术（sequential exception technique ）
    - 模仿了人类从一系列推测类似的对象中识别异常对象的方式
  - OLAP数据立方体方法（ OLAP data cube technique）
    - 在大型多维数据中使用数据立方体来确定反常区域

On-line clustering

On-line clustering

- Web Search Results: multiple sub-topics are mixed together for the given query.
- Organizing Web search results into clusters facilitates users' quick browsing.

On-line clustering

- Zamir and Etzioni'95:
  - clustering algorithm should take the document snippets instead of the whole documents as input, since the downloading of original documents is time-consuming;
  - the clustering algorithm should be fast enough for online calculation
  - the generated clusters should have readable descriptions for quick browsing by users.
- Zamir and Etzioni presented a Suffix Tree Clustering (STC) ''95 (Web Document Clustering: A Feasibility Demonstration)
  - first identifies sets of documents that share common phrases,
  - then create clusters according to these phrases

Suffix Tree Clustering (STC)

- STC is a linear time clustering algorithm that is based on a suffix tree which efficiently identifies sets of documents that share common phrases.
- STC satisfies the key requirements:
  - STC treats a document as a string, making use of proximity information between words.
  - STC is novel, incremental, and O(n) time algorithm.
  - STC succinctly summarizes clusters' contents for users.
  - Quick because of working on smaller set of documents, incremantality
  - …

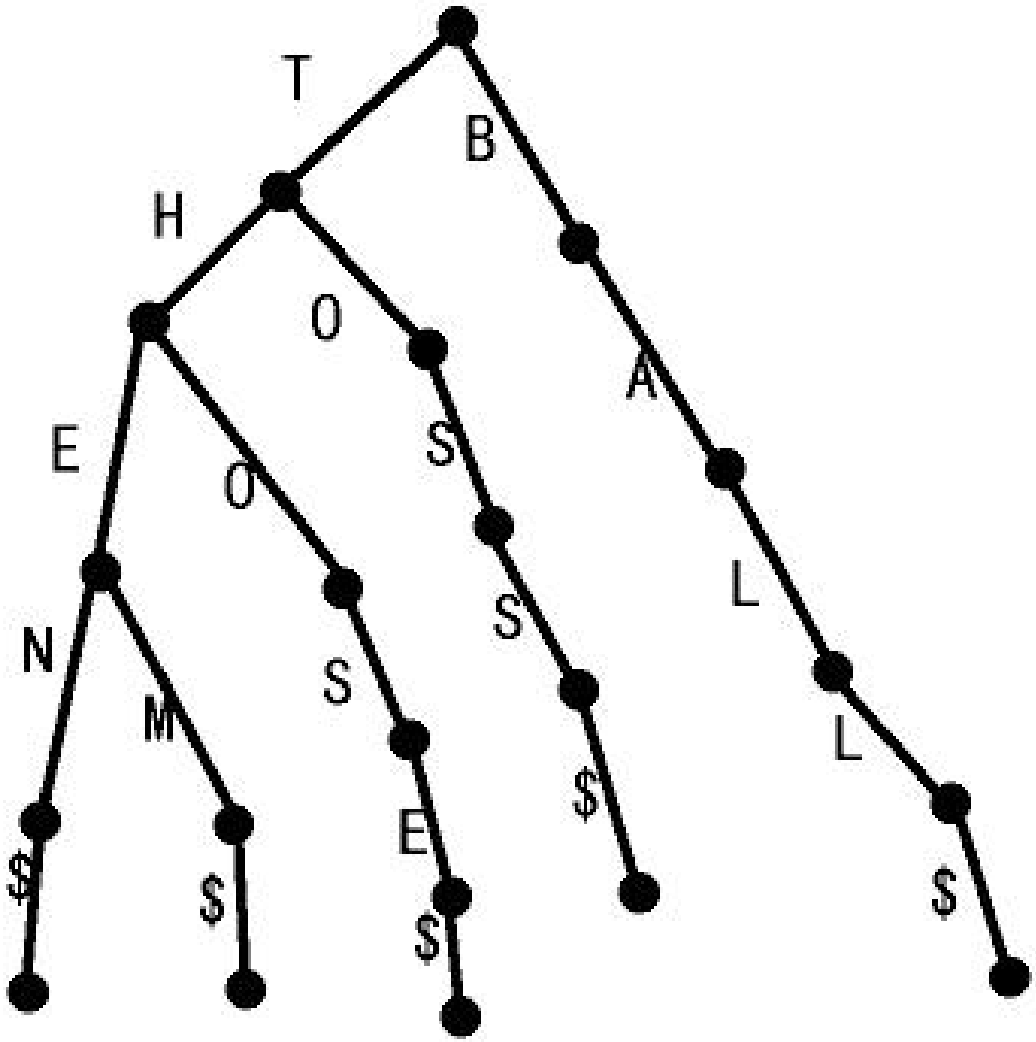
后缀树

- 一棵后缀树包含了一个或者多个字符串的所有后缀。空字符串也算其中一个后缀。
- 对于字符串banana，其所有后缀为：banana anana nana ana na a 空。
- 通常为了更清楚地表示出后缀，我们在字符串末尾添加一个特殊字符作为结束标记($)。

后缀树

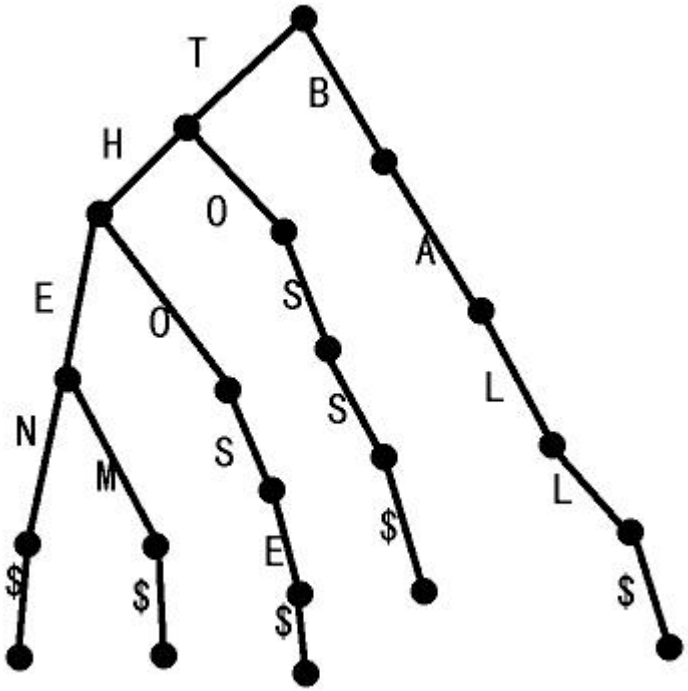
- A suffix tree is a rooted, directed tree.
- Each internal node has 2+ children.
- Each edge is labeled with a non-empty sub-string of S. The label of a node is defined to be the concatenation of the edge-labels on the path from the root to that node
- No two edges out of the same node can have edge-labels that begin with the same word——compact.

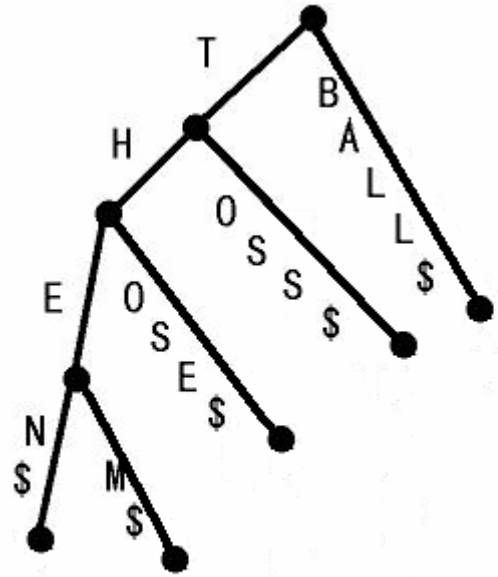
Trie树示例



压缩后的Trie树

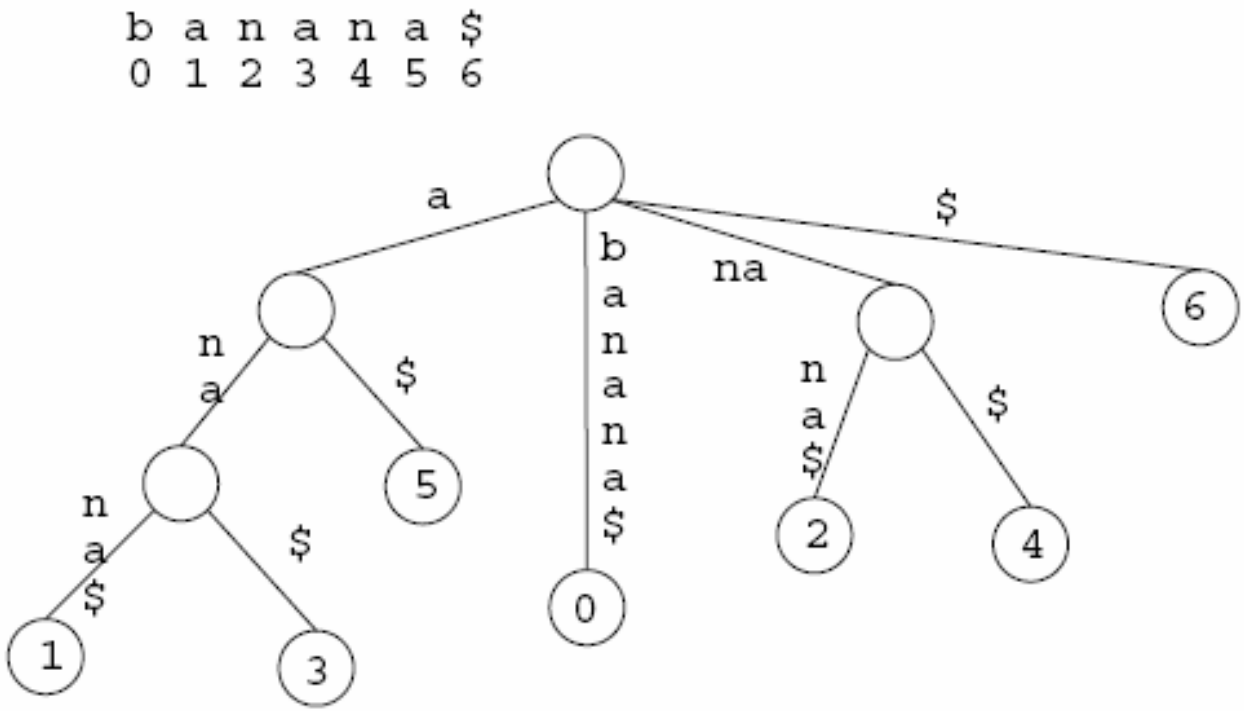
- 我们可以对Trie进行压缩，对只有一个儿子的节点进行合并：





后缀树

- banana所对应的后缀树如下：



后缀树的构造

- 后缀树可以用O(n)的算法构造出来。

STC算法

- Step1: Document “cleaning”
  - Html -> plain text
  - Words stemming
  - Mark sentence boundaries
  - Remove non-word tokens
- Step 2: Identifying Base Clusters
- Step3: Combining Base Clusters

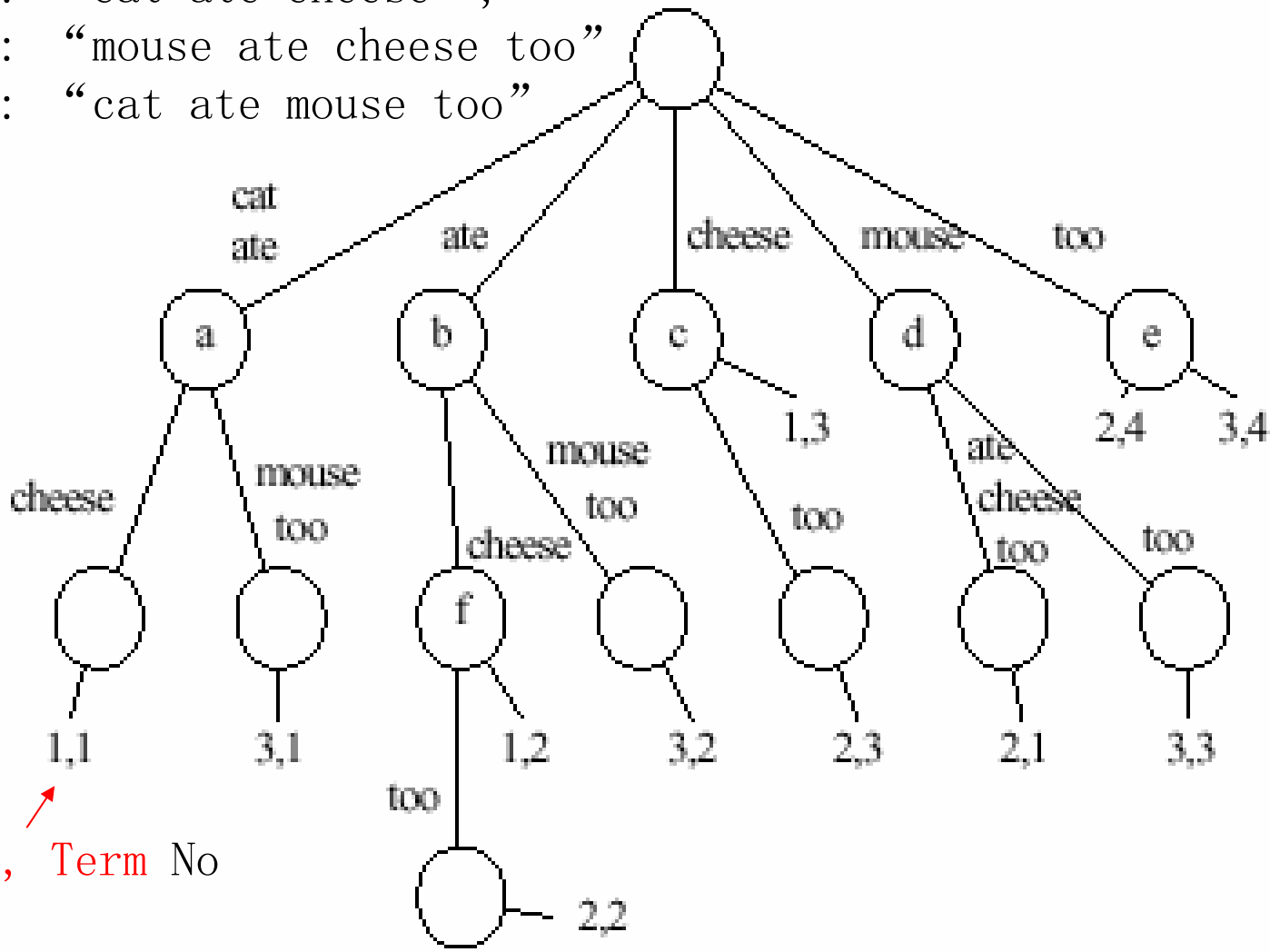
Step2: Identifying base Clusters—*Suffix Tree*

* *STC treats a document as a set of strings*...

- Suffix tree of string S: a compact tree containing all the suffixes of S
  - Suffix of a word: lovely
  - Suffix of a string: "Friends" is a lovely show.

Ex. A *Suffix Tree of Strings*

- String1: “cat ate cheese”,
- String2: “mouse ate cheese too”
- String3: “cat ate mouse too”



String No, Term No

Base clusters

Node	Phrase	Documents
a	cat ate	1,3
b	ate	1,2,3
c	cheese	1,2
d	mouse	2,3
e	too	2,3
f	ate cheese	1,2

Base clusters corresponding to the suffix tree nodes

Cluster score

- s(B) = |B| * f(|P|)
  - |B| is the number of documents in base cluster B
  - |P| is the number of words in P that have a non-zero score
    - zero score words: stopwords, too few(<3) or too many(>40%)

Step 3: Combining Base Clusters

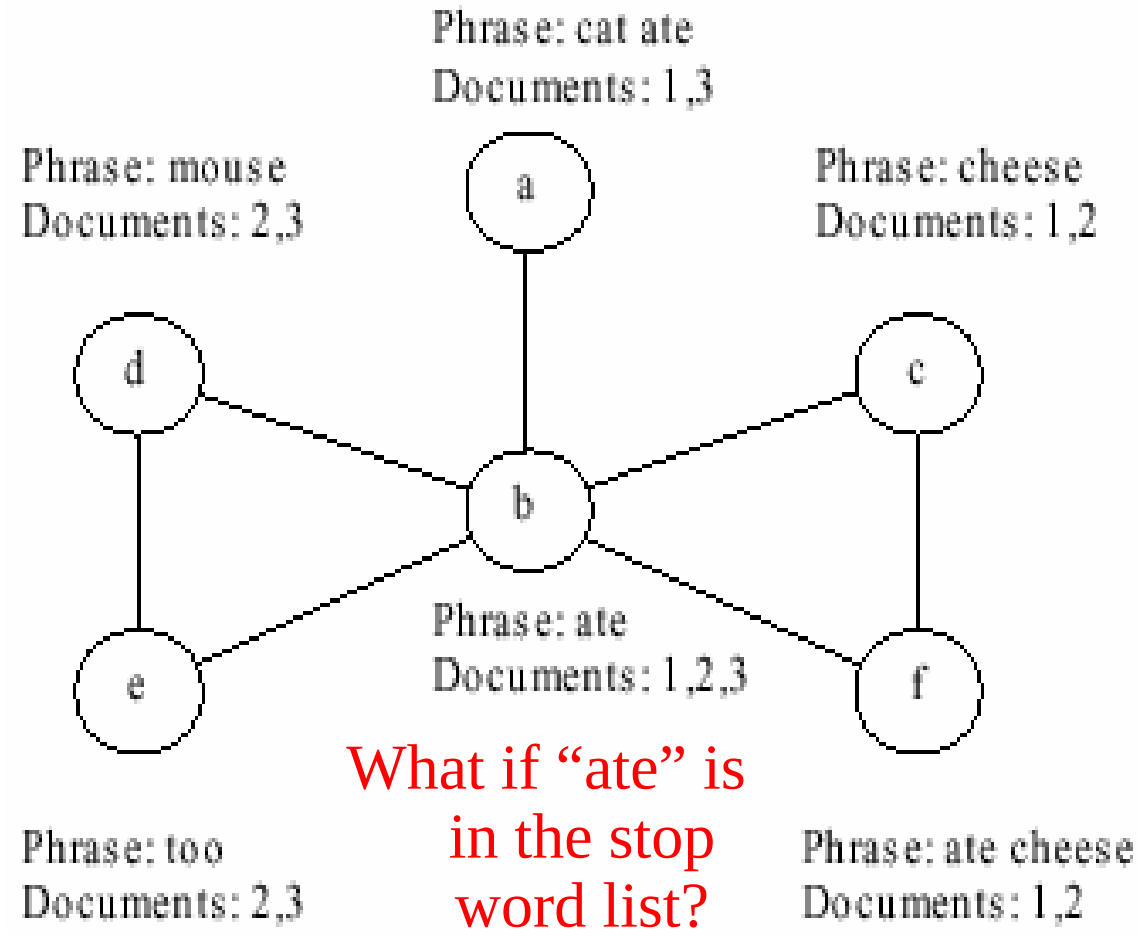
- Merge base clusters with a high overlap in their document sets
  - documents may share multiple phrases.
- Similarity of B_m and B_n (0.5 is paramter)

$$= \begin{cases} 1 & \text{iff } |B_m \cap B_n| / |B_m| > 0.5 \\ & \text{and } |B_m \cap B_n| / |B_n| > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Base Cluster Graph

Node: cluster

Edge: similarity between two clusters > 1



STC is Incremental

- As each document arrives from the web, we
  - "clean" it (linear with collection size)
  - Add it to the suffix tree. Each node that is updated/created as a result of this is tagged(linear)
  - Update the relevant base clusters and recalculate the similarity of these base clusters to the rest of k highest scoring base clusters(linear)
  - Check any changes to the final clusters (linear)
  - Score and sort the final clusters, choose top 10...(linear)

STC allows cluster overlap

- Why overlap is reasonable?

  a document often has 1+ topics

- STC allows a document to appear in 1+ clusters, since documents may share 1+ phrases with other documents
- But not too similar to be merged into one cluster..

Evaluation–Precision

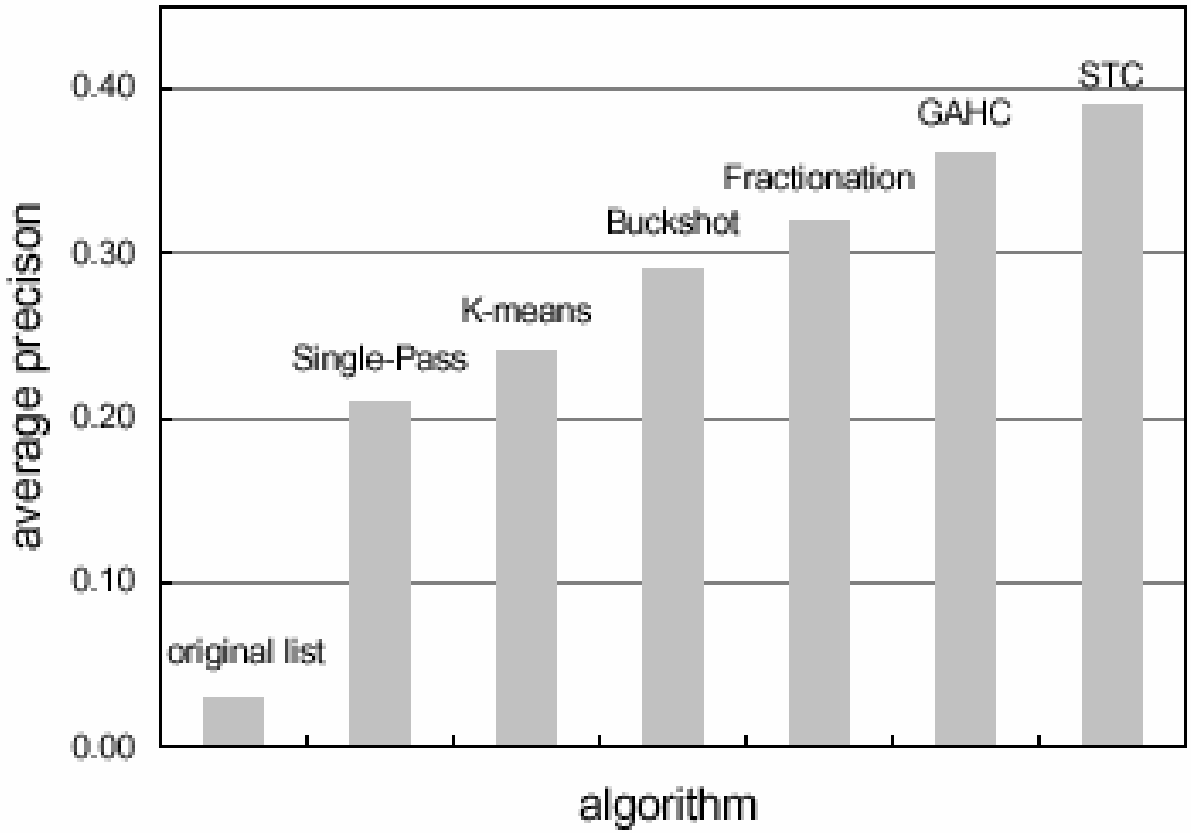


Figure 5: The average precision of the clustering algorithms and of the original ranked list returned by the search engine, averaged over the 10 original document collections.

Snippets versus Whole Document

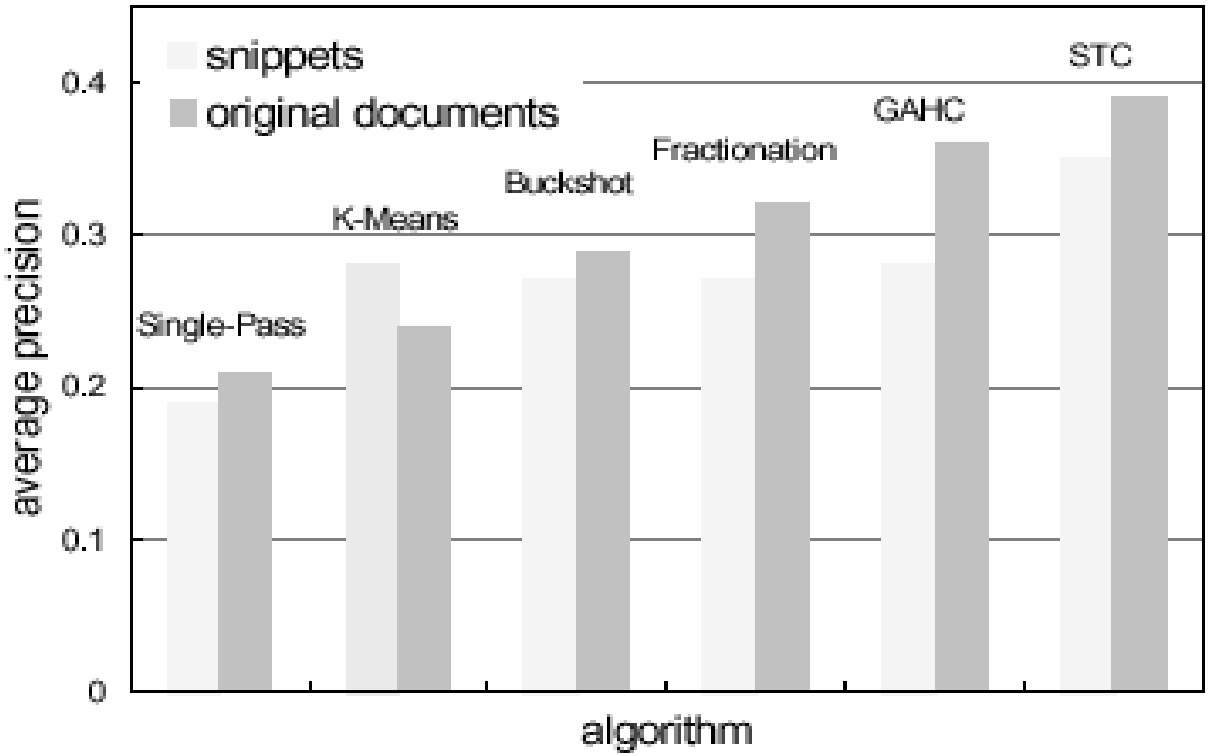
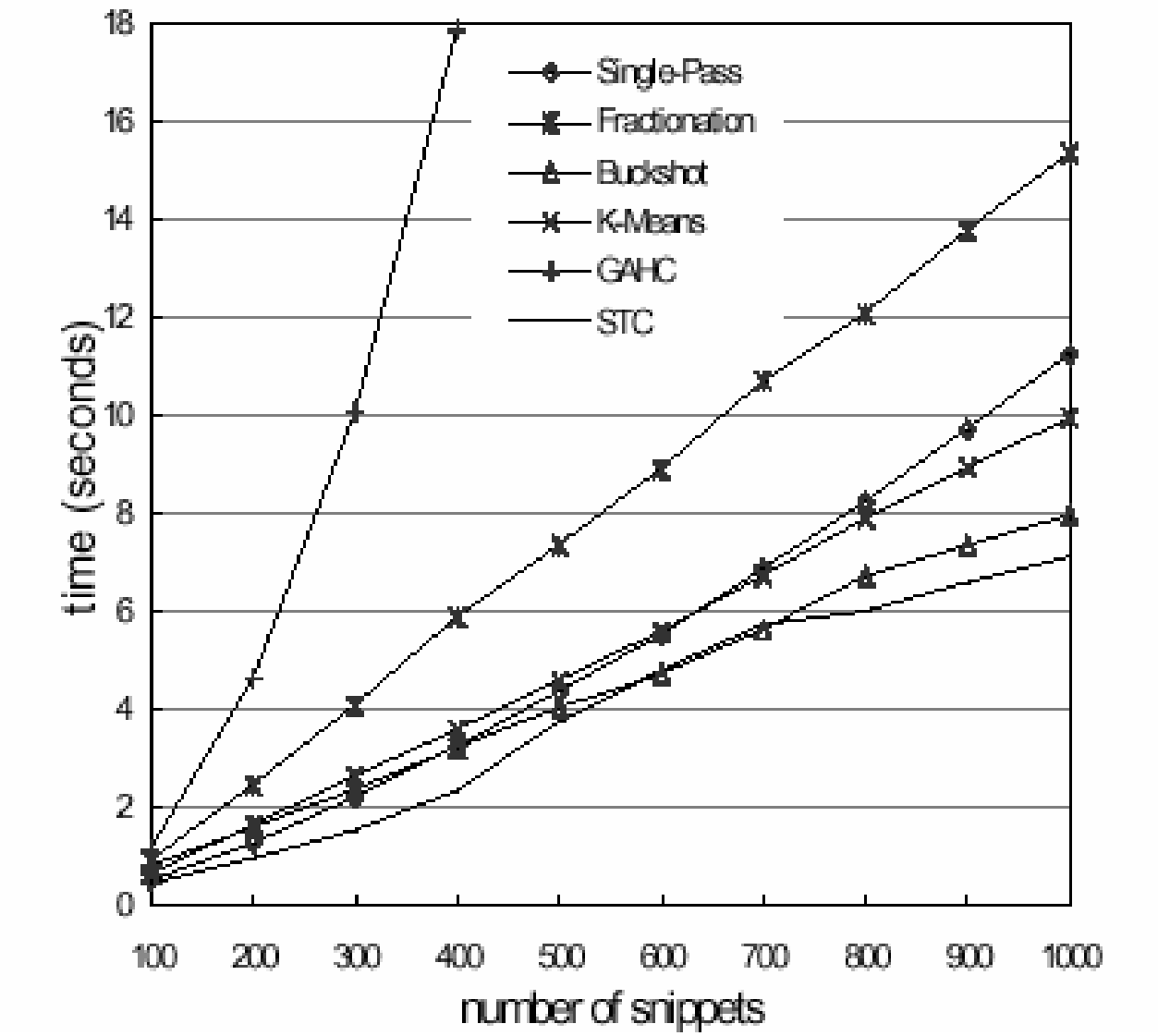


Figure 10: The average precision of clustering algorithms on the snippet collections compared with the average precision on the original Web documents collections.

Execution time



STC

- STC – an incremental, o(n) time clustering algorithm
- A clustering algorithms on Web search engine results

A paper: Cluster Web Search Results, 2004

- Hua-Jun Zeng, etc. Learning to Cluster Web Search Results, SIGIR'04, 2004. (Microsoft research, Asia)
- These traditional clustering techniques generate clusters with poor readable names.

A paper: Cluster Web Search Results, 2004

- Zeng's method reformalize the clustering problem as a salient phrase ranking problem.
  - first extracts and ranks salient phrases as candidate cluster names, based on a regression model learned from human labeled training data.
  - The documents are assigned to relevant salient phrases to form candidate clusters.
  - the final clusters are generated by merging these candidate clusters.

SALIENT PHRASES EXTRACTION

- denote the current phrase (an *n*-gram) as *w*, and the set of documents that contains *w* as *D*(*w*).
- five properties which are calculated during the document parsing.
  - **Phrase Frequency / Inverted Document Frequency**
  - **Phrase Length**
  - **Intra-Cluster Similarity**
  - **Phrase Independence**

Phrase Frequency / Inverted Document Frequency and Phrase Length

- **Phrase Frequency / Inverted Document Frequency**

$$TFIDF = f(w) \cdot \log \frac{N}{|D(w)|}$$

- Phrase Length: the count of words in a phrase.

  a longer name is preferred for users‘ browsing.

$$LEN = n$$

Cluster Entropy

- Cluster Entropy (CE) to represent the distinctness of a phrase.

$$CE = -\sum_{t} \frac{|D(w) \cap D(t)|}{|D(w)|} \log \frac{|D(w) \cap D(t)|}{|D(w)|}$$

Phrase Independence

- a phrase is independent when the entropy of its context is high (i.e. the left and right contexts are random enough).

$$IND_l = -\sum_{t=l(W)} \frac{f(t)}{TF} \log \frac{f(t)}{TF}$$

$$IND = \frac{IND_l + IND_r}{2}$$

Experimental Results ***---Property Comparison***

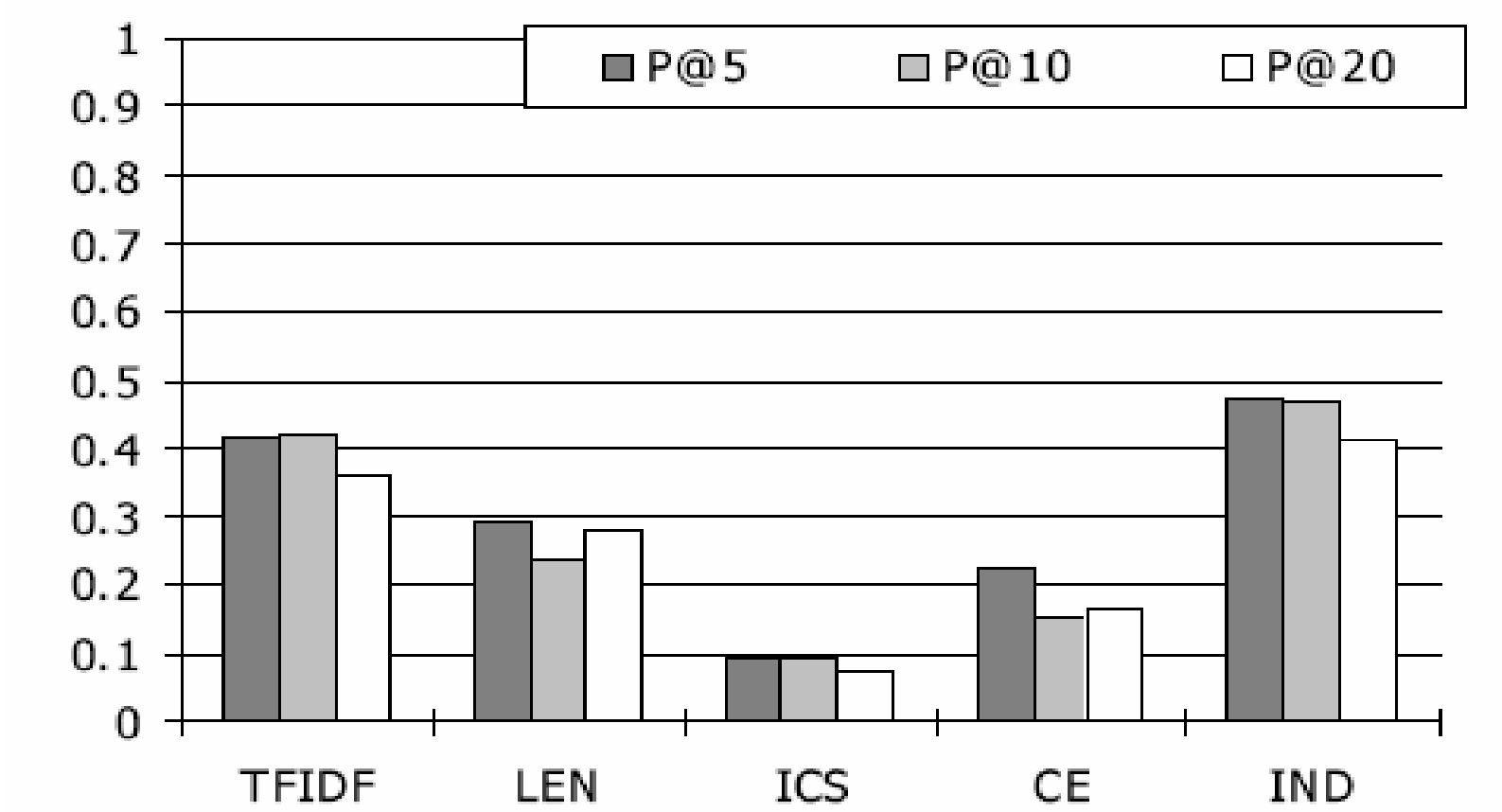


Figure 2. Performance for each single property

Experimental Results --- *Learning Methods Comparison*

- The coefficients of one of the linear regression models, as follows

$$y = -0.427 + 0.146 \times TFIDF + 0.241 \times LEN - 0.022 \times ICS + 0.065 \times CE + 0.266 \times IND$$

Clustering results for "iraq" 伊拉克

1.	war (32)	1. AlterNet: War on Iraq 2. War in Iraq - Christianity Today Magazine 3. End The War 4. Iraq Aftermath: The Human Face of War: AFSC 5. NOLA.com: War on Iraq	6.	country (19)	1. Library of Congress / Federal Research Division / Country ... 2. U.S. Department of State: Iraq Country Information 3. Iraq Country Analysis Brief 4. ArabBay.com: Arab Countries/Iraq 5. Countries: Iraq: Arabic Search Engine: Directory of arabic ...
2.	middle east (31)	1. Middle East Studies: Iraq 2. Amnesty International Report 2002 - Middle East and North ... 3. Human Rights Watch: Middle East and Northern Africa : ... 4. Columbus World Travel Guide - Middle East - Iraq - Overview 5. Iraq/Middle East	7.	special report (13)	1. Guardian Unlimited | Special reports | Special report: Iraq 2. Operation Iraqi Freedom - A White House Special Report 3. RFE/RL Iraq Report 4. Amnesty International Report 2002 - Middle East and North ... 5. Ethnologue report for Iraq
3.	map (18)	1. UT Library Online - Perry-Casta?eda Map Collection - Iraq ... 2. ABC Maps of Iraq; Flag, Map, Economy, Geography, ... 3. Flags of Iraq - geography; Flags, Map, Economy, Geography, ... 4. Lonely Planet - Iraq Map 5. Map of Iraq	8.	guide (13)	1. Lonely Planet World Guide | Destination Iraq | Introduction 2. Columbus World Travel Guide - Middle East - Iraq - Overview 3. Herald.com - Your Miami Everything Guide 4. Kansas.com - Your Kansas Everything Guide 5. Kansascity.com - Your Kansas City Everything Guide
4.	saddam hussein (13)	1. Iraq Resource Information Site - News History Culture People ... 2. U.S. Department of State - Saddam Hussein's Iraq 3. Iraq Crisis - Global Policy Forum - UN Security Council 4. New Scientist | Conflict in Iraq 5. Almuajaha - The Iraqi Witness: home	9.	united nations (11)	1. Mission of Iraq to the United Nations 2. united nations 3. United for Peace and Justice 4. U.S. Department of State: Iraq Country Information 5. Iraq Crisis - Global Policy Forum - UN Security Council
5.	human rights (11)	1. Human Rights Watch: Middle East and Northern Africa : Iraq ... 2. Iraq: Amnesty International's Human Rights Concerns 3. Human Rights Watch: Background on the Crisis in Iraq 4. Iraq:Amnesty International's Human Rights Concerns for 5. Iraq Aftermath: The Human Face of War: AFSC	10.	travel, business (16)	1. Iraq: Complete travel information to Iraq, travel facts, ... 2. EIN news - Iraq - Political, Business and Breaking ... 3. Iraq - Travel Warning 4. Columbus World Travel Guide - Middle East - Iraq - ... 5. Iraq Visa Application - Tourist Visas, Business Visas, ...

小结

- 聚类概述
- 聚类算法
  - 划分方法
  - 层次方法
  - 密度方法
  - 网格方法
  - 在线聚类

Any Question?