Lecture 4 : Clustering

楊立偉教授

wyang@ntu.edu.tw

本投影片修改自Introduction to Information Retrieval一書之投影片
Ch 16 & 17

Clustering : Introduction

Clustering: Definition

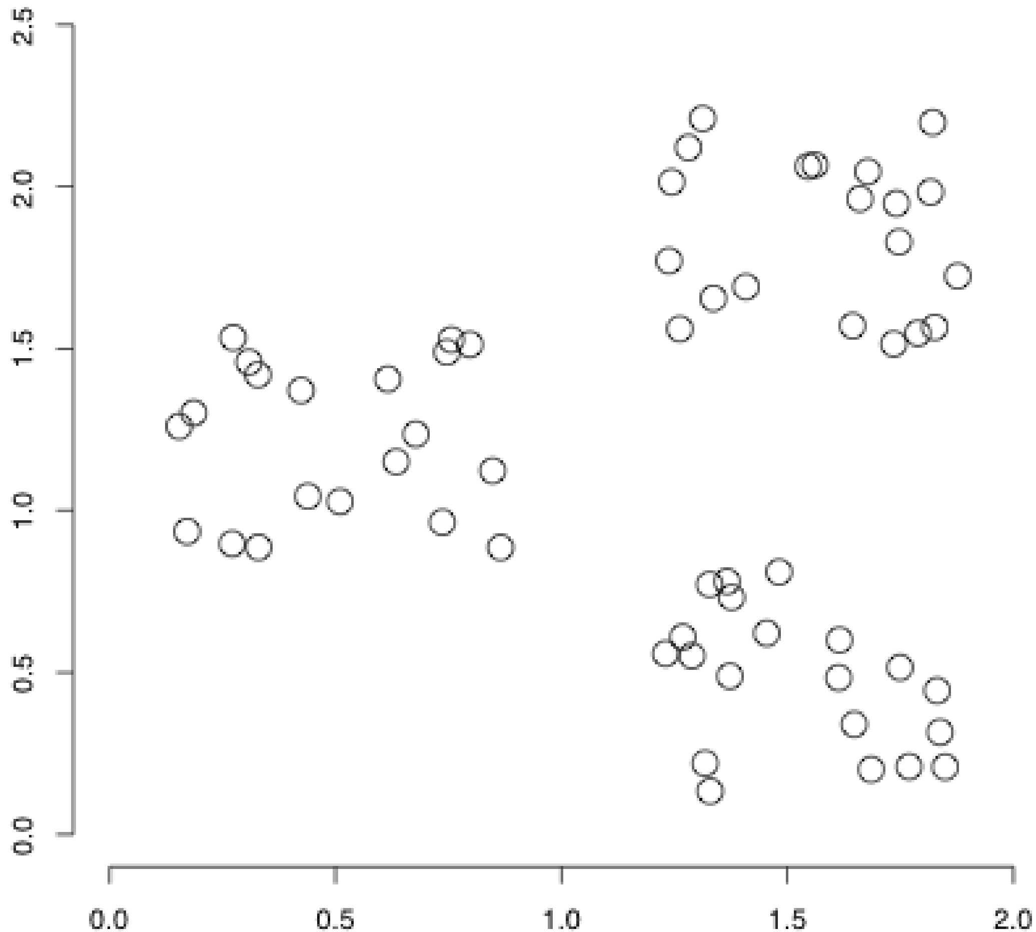
- (Document) clustering is the process of grouping a set of documents into clusters of similar documents.
  - Documents within a cluster should be similar.

  群內盡量相似

  - Documents from different clusters should be dissimilar.

  群間盡量相異
- Clustering is the most common form of unsupervised learning.
  - Unsupervised = there are no labeled or annotated data.

Data set with clear cluster structure



Propose algorithm for finding the cluster structure in this example

Classification vs. Clustering

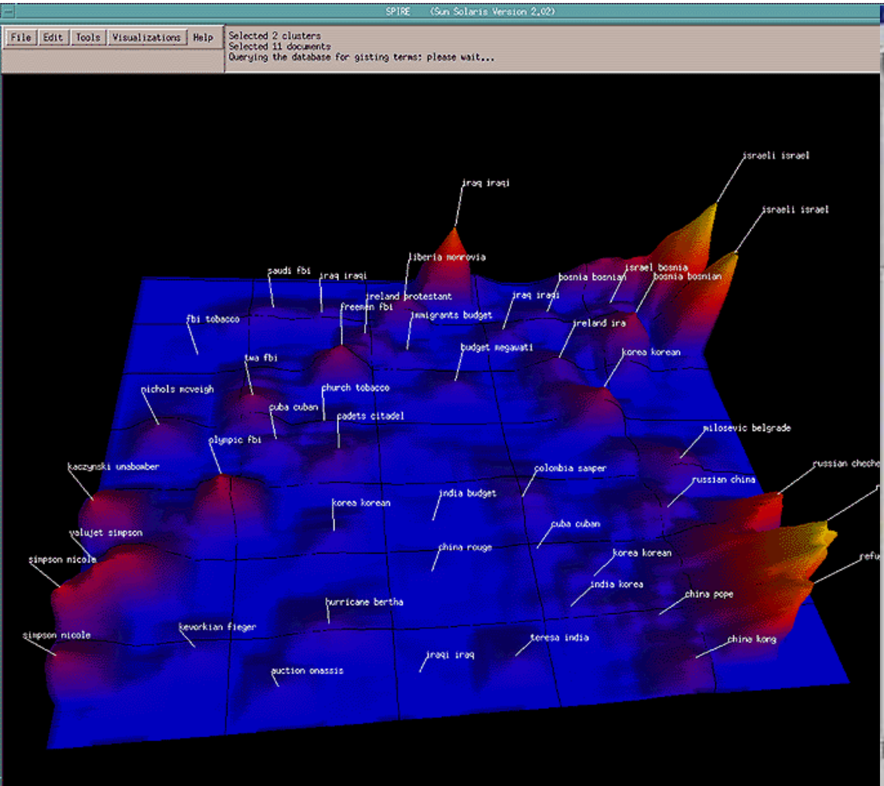
- Classification
  - Supervised learning
  - Classes are human-defined and part of the input to the learning algorithm.
- Clustering
  - Unsupervised learning
  - Clusters are inferred from the data without human input.

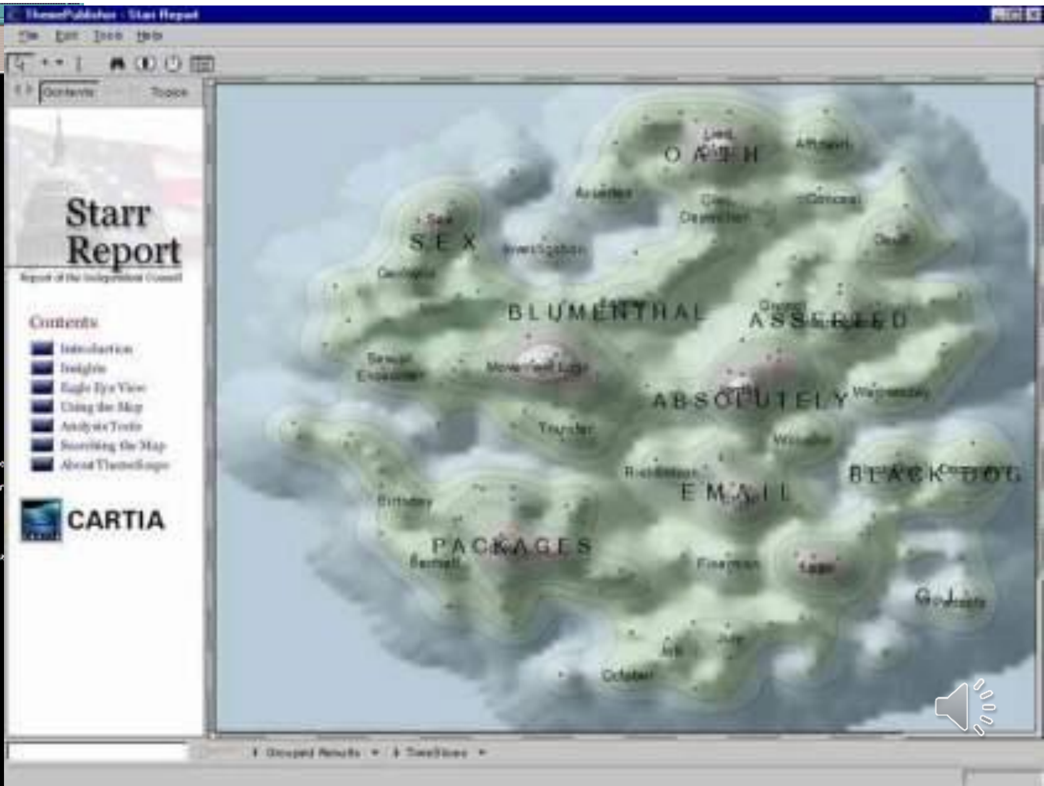
Why cluster documents?

- Whole corpus analysis/navigation
  - Better user interface 提供文件(資料)集合的分析與導覽
- For improving recall in search applications
  - Better search results 提供完整的搜尋結果(相似的也找出)
- For better navigation of search results
  - Effective "user recall" will be higher 搜尋結果導覽
- For speeding up vector space retrieval
  - Faster search 加快搜尋速度(因為限縮了範圍)

For visualizing a document collection

- Wise et al, "Visualizing the non-visual" PNNL
- ThemeScapes, Cartia
  - [Mountain height = cluster size]





For improving search recall

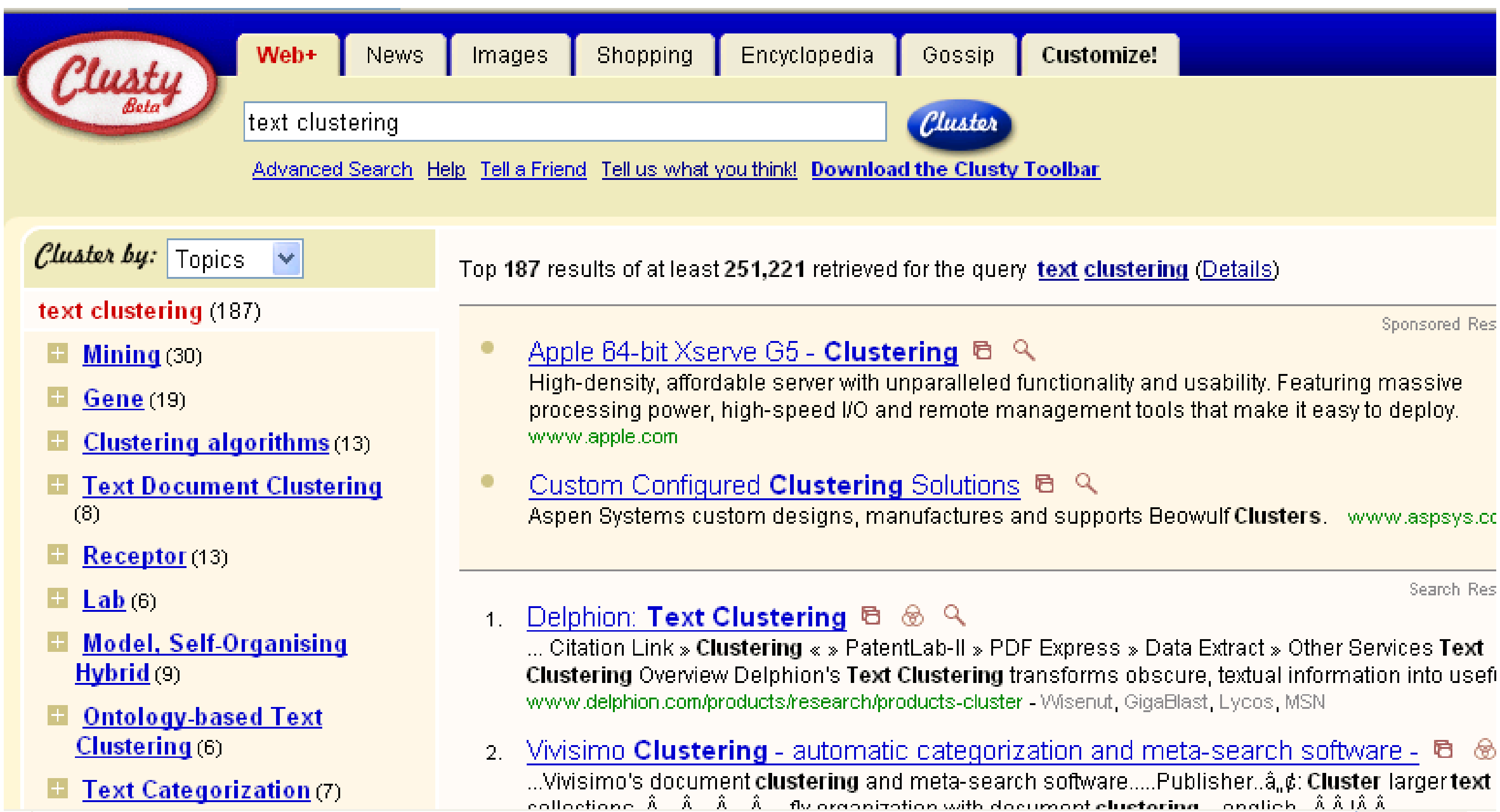
- *Cluster hypothesis - "closely associated documents tend to be relevant to the same requests".*
- Therefore, to improve search recall:
  - Cluster docs in corpus 先將文件做分群
  - When a query matches a doc *D*, also return other docs in the cluster containing *D* 也建議符合的整群
- Hope if we do this: The query "car" will also return docs containing *automobile*
  - Because clustering grouped together docs containing *car* with those containing *automobile.*

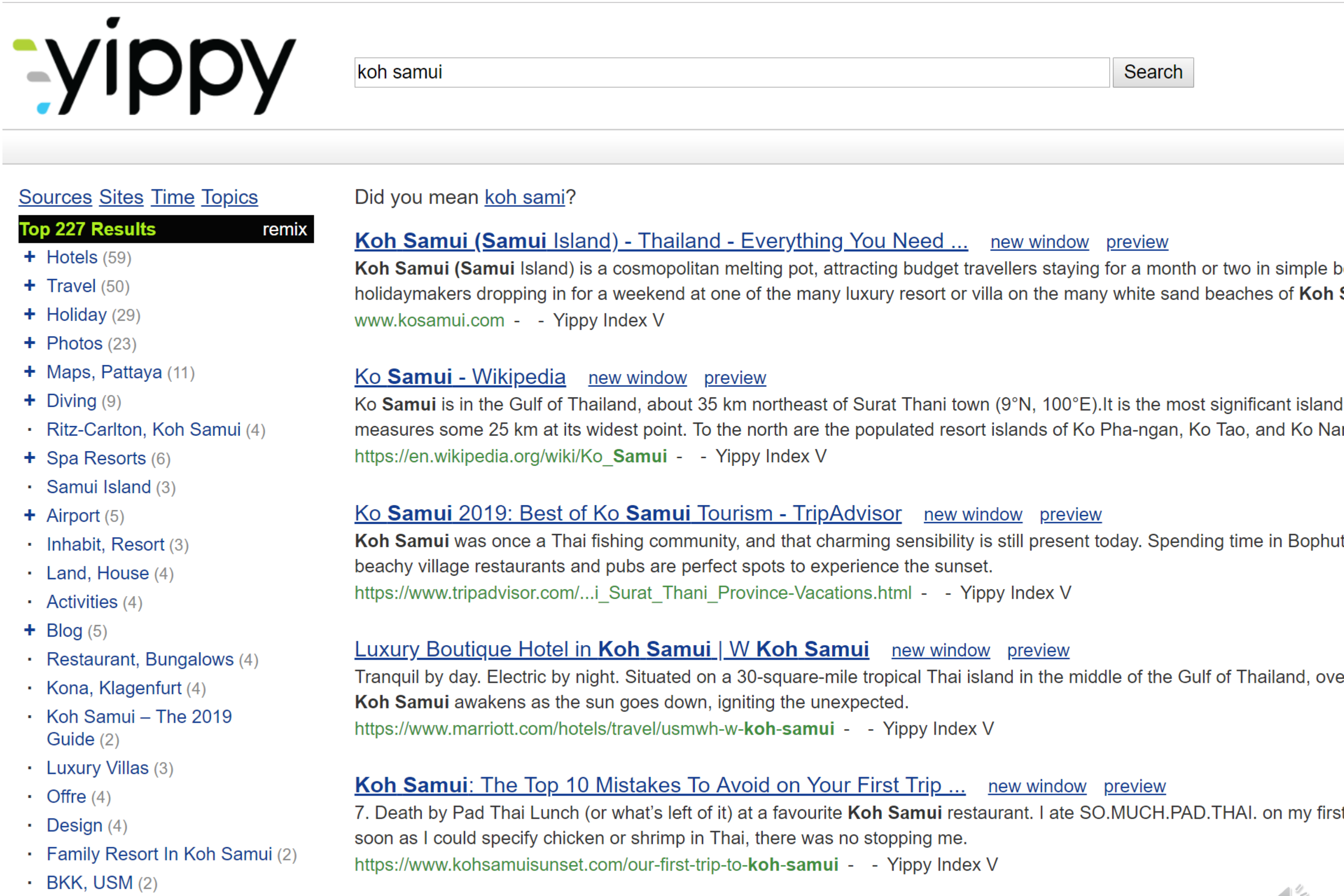
Why might this happen?

具有類似的文件特徵

For better navigation of search results

- For grouping search results thematically
  - clusty.com / Vivisimo (Enterprise Search – Velocity)




yippy
koh samui
Search
Sources Sites Time Topics
Top 227 Results
remix
+ Hotels (59)
+ Travel (50)
+ Holiday (29)
+ Photos (23)
+ Maps, Pattaya (11)
+ Diving (9)
· Ritz-Carlton, Koh Samui (4)
+ Spa Resorts (6)
· Samui Island (3)
+ Airport (5)
· Inhabit, Resort (3)
· Land, House (4)
· Activities (4)
+ Blog (5)
· Restaurant, Bungalows (4)
· Kona, Klagenfurt (4)
· Koh Samui – The 2019 Guide (2)
· Luxury Villas (3)
· Offre (4)
· Design (4)
· Family Resort In Koh Samui (2)
· BKK, USM (2)
Did you mean koh sami?
Koh Samui (Samui Island) - Thailand - Everything You Need ... new window preview
Koh Samui (Samui Island) is a cosmopolitan melting pot, attracting budget travellers staying for a month or two in simple b
holidaymakers dropping in for a weekend at one of the many luxury resort or villa on the many white sand beaches of Koh S
www.kosamui.com - - Yippy Index V
Ko Samui - Wikipedia new window preview
Ko Samui is in the Gulf of Thailand, about 35 km northeast of Surat Thani town (9°N, 100°E).It is the most significant island
measures some 25 km at its widest point. To the north are the populated resort islands of Ko Pha-ngan, Ko Tao, and Ko Nar
https://en.wikipedia.org/wiki/Ko_Samui - - Yippy Index V
Ko Samui 2019: Best of Ko Samui Tourism - TripAdvisor new window preview
Koh Samui was once a Thai fishing community, and that charming sensibility is still present today. Spending time in Bophut
beachy village restaurants and pubs are perfect spots to experience the sunset.
https://www.tripadvisor.com/...i_Surat_Thani_Province-Vacations.html - - Yippy Index V
Luxury Boutique Hotel in Koh Samui | W Koh Samui new window preview
Tranquil by day. Electric by night. Situated on a 30-square-mile tropical Thai island in the middle of the Gulf of Thailand, ove
Koh Samui awakens as the sun goes down, igniting the unexpected.
https://www.marriott.com/hotels/travel/usmwh-w-koh-samui - - Yippy Index V
Koh Samui: The Top 10 Mistakes To Avoid on Your First Trip ... new window preview
7. Death by Pad Thai Lunch (or what's left of it) at a favourite Koh Samui restaurant. I ate SO.MUCH.PAD.THAI. on my first
soon as I could specify chicken or shrimp in Thai, there was no stopping me.
https://www.kohsamuisunset.com/our-first-trip-to-koh-samui - - Yippy Index V


carrot2
open source framework for building search clustering engines
Home
Download
Documentation
FAQ
Support
Source Code
License
Algorithms
Publications
Powered by C2
Authors
Spin-off
Labs
Carrot2 is an Open Source Search Results Clustering Engine. It can automatically organize small collections of documents (search results but not only) into thematic categories.
Web
Wiki
Images
News
Yahoo
salsa
Search
More options
Tree
Visualization
All Topics (100)
Dance Salsa (18)
Salsa Classes (10)
Salsa Recipes (7)
Salsa Music (6)
Mambo (5)
Salsa Tanzschule (4)
Cluster Salsa Recipes with 7 documents
1
© Salsa Cycles 2008
Product index, salsa recipes, catalog r
http://www.salsacycles.com/
4
Salsa Recipes - Dips and Spreads - All
Looking for salsa recipes? Allrecipes ha
Search results clustered with Carrot2 (live demo)
Apart from two specialized document clustering algorithms, Carrot2 offers ready-to-use components for fetching search results from various sources including GoogleAPI, Bing API, eTools Meta Search, Lucene, SOLR, and more.
Download
Carrot2 API JavaDoc
Carrot2 C# API reference
News
Release 3.11.0 is available.
Sponsors and donors
Carrot2 is kindly supported by a number of companies and organizations
Spin-off company
CARROT search

For better understanding the data

Example

- 使用總體資料敘述統計做描述
  - 有4隻雞、4隻兔子
  - 每隻平均有3隻腳
- 先對資料做分群，再對各群做描述
  - 有4隻雞、4隻兔子
  - 依每隻的腳數可分為2群
  - 第1群每隻平均有2隻腳，第2群每隻平均有4隻腳

Issues for clustering (1)

- General goal: put related docs in the same cluster, put unrelated docs in different clusters.
- Representation for clustering
  - Document representation 如何表示一篇文件
  - Need a notion of similarity/distance 如何表示相似度

Issues for clustering (2)

- How to decide the number of clusters
  - Fixed a priori : assume the number of clusters K is given.
  - Data driven : semiautomatic methods for determining K
  - Avoid very small and very large clusters
- Define clusters that are easy to explain to the user

Clustering Algorithms

- Flat (Partitional) algorithms 無階層的聚類演算法
  - Usually start with a random (partial) partitioning
  - Refine it iteratively 不斷地修正調整
    - *K* means clustering
- Hierarchical algorithms 有階層的聚類演算法
  - Create a hierarchy
  - Bottom-up, agglomerative 由下往上聚合
  - Top-down, divisive 由上往下分裂

Flat (Partitioning) Algorithms

- Partitioning method: Construct a partition of *n* documents into a set of *K* clusters
  將 n 篇文件分到 K 群中
- Given: a set of documents and the number *K*
- Find: a partition of *K* clusters that optimizes the chosen partitioning criterion
  - Globally optimal: exhaustively enumerate all partitions
    找出最佳切割 → 通常很耗時
  - Effective heuristic methods: *K*-means and *K*-medoids algorithms 用經驗法則找出近似解即可

Hard vs. Soft clustering

- Hard clustering: Each document belongs to exactly one cluster.
  - More common and easier to do
- Soft clustering: A document can belong to more than one cluster.
  - For applications like creating browsable hierarchies
  - Ex. Put sneakers in two clusters: sports apparel, shoes
  - You can only do that with a soft clustering approach.

*only hard clustering is discussed in this class.

K-means algorithm

K-means

- Perhaps the best known clustering algorithm
- Simple, works well in many cases
- Use as default / baseline for clustering documents

K-means

- In vector space model, Assumes documents are real-valued vectors.
- Clusters based on *centroids* (aka the *center of gravity* 重心 or mean) of points in a cluster, *c*:

$$\vec{\mu}(c) = \frac{1}{|c|} \sum_{\vec{x} \in c} \vec{x}$$

- Reassignment of instances to clusters is based on distance to the current cluster centroids.

K-means algorithm

1. Select K random docs $\{s_1, s_2, \ldots s_K\}$ as seeds. 先挑選種子
2. Until clustering converges or other stopping criterion:

   重複下列步驟直到收斂或其它停止條件成立

   2.1 For each doc d_i: 針對每一篇文件

   Assign d_i to the cluster c_j such that $dist(x_i, s_j)$ is minimal.

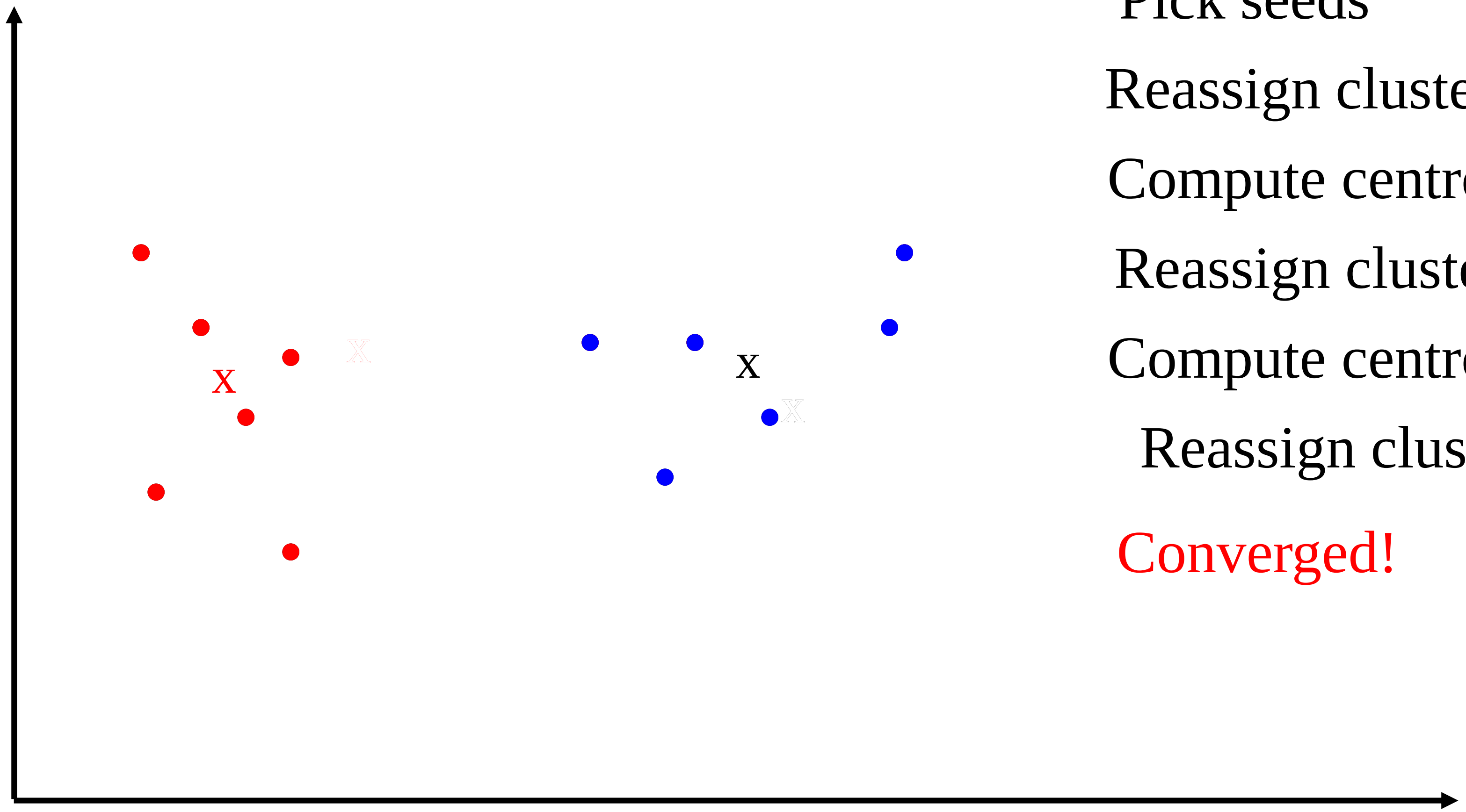
   將該文件加入最近的一群

   2.2 For each cluster c_j

   $s_j = \mu(c_j)$ 以各群的重心為種子，再做一次

   (*Update the seeds to the centroid of each cluster*)

K-means example (*K*=2)



Pick seeds

Reassign clusters

Compute centroids

Reassign clusters

Compute centroids

Reassign clusters

Converged!

通常做3至4回就大致穩定（但仍需視資料與群集多寡而調整）

K-means algorithm

$K\text{-MEANS}(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$

$(\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_K) \leftarrow \text{SELECTRANDOMSEEDS}(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$

for $k \leftarrow 1$ **to** K

do $\vec{\mu}_k \leftarrow \vec{s}_k$

while stopping criterion has not been met

do for $k \leftarrow 1$ **to** K

  do $\omega_k \leftarrow \{\}$

  for $n \leftarrow 1$ **to** N

  do $j \leftarrow \arg\min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$

    $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$ *(reassignment of vectors)*

  for $k \leftarrow 1$ **to** K

  do $\vec{\mu}_k \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$ *(recomputation of centroids)*

return $\{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$

Termination conditions

- Several possibilities, e.g.,
  - A fixed number of iterations. 只做固定幾回合
  - Doc partition unchanged. 群集不再改變
  - Centroid positions don't change. 重心不再改變

Convergence of *K*-Means

- Why should the *K*-means algorithm ever reach a *fixed point*?
  - A state in which clusters don't change. 收斂
- *K*-means is a special case of a general procedure known as the *Expectation Maximization (EM) algorithm*.
  - EM is known to converge.
  - Number of iterations could be large.
    在理論上一定會收斂，只是要做幾回合的問題
    (逼近法，且一開始逼近得快，之後逼近變慢)

Convergence of *K*-Means : 證明

- Define goodness measure of cluster *k* as sum of squared distances from cluster centroid:
  - $G_k = \Sigma_i (d_i - c_k)^2$ (sum over all d_i in cluster *k*)
  - $G = \Sigma_k G_k$

  計算每一群中文件與中心的距離平方，然後加總
- Reassignment monotonically decreases G since each vector is assigned to the closest centroid. 每回合的動作只會讓G越來越小

Time Complexity

- Computing distance between two docs is O*(m)* where *m* is the dimensionality of the vectors.
- Reassigning clusters: O*(Kn)* distance computations, or O*(Knm).*
- Computing centroids: Each doc gets added once to some centroid: O*(nm).*
- Assume these two steps are each done once for *I* iterations: O*(IKnm).*

執行 I 回合；分 K 群；n 篇文件；m 個詞 → 慢且不scalable

改善方法：用**近似估計**, **抽樣**, **選擇**等技巧來加速

Issue (1) Seed Choice

- Results can vary based on random seed selection.
- Some seeds can result in poor convergence rate, or convergence to sub-optimal clusterings.
  - Select good seeds using a heuristic (e.g., doc least similar to any existing mean)
  - Try out multiple starting points

Example showing sensitivity to seeds

In the above, if you start with B and E as centroids you converge to {A,B,C} and {D,E,F}
If you start with D and F you converge to {A,B,D,E} {C,F}

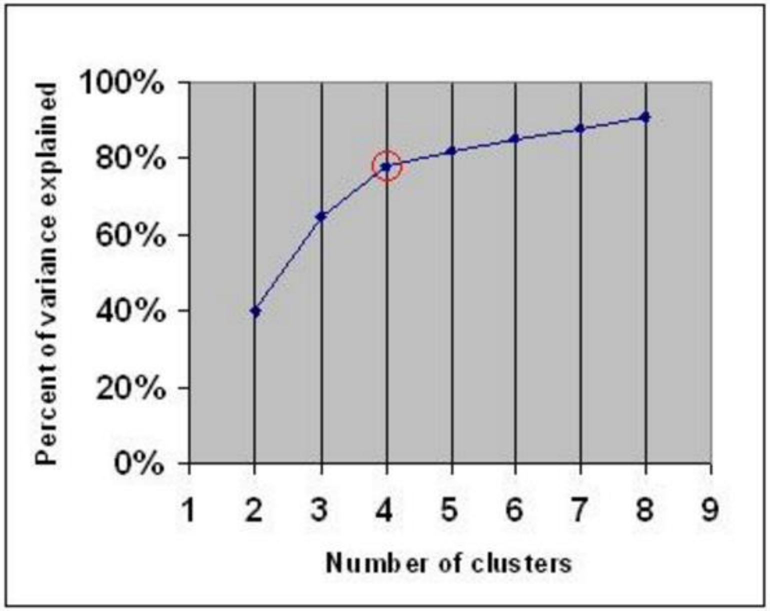
Issue (2) How Many Clusters?

- Number of clusters *K* is given
  - Partition *n* docs into predetermined number of clusters
- Finding the "right" number of clusters is part of the problem 假設連應該分成幾群都不知道
  - Given docs, partition into an "appropriate" number of subsets.
  - E.g., for query results - ideal value of *K* not known up front - though UI may impose limits. 查詢結果分群時通常不會預先知道該分幾群

If *K* not specified in advance

- Suggest *K* automatically
  - using heuristics based on *N*
    - 依經驗法則，例如每m筆分1群，缺點是可能很不準
  - using K vs. Cluster-size diagram 畫成圖表來分析
- Tradeoff between having less clusters (better focus within each cluster) and having too many clusters 如何取捨

- 方法: 以「群間變異對應於整體變異的百分比」來看 (即F檢驗)，每增加一群所能帶來的邊際變異開始下降的前一點。



Explained Variance. The "elbow" is indicated by the red circle. The number of clusters chosen should therefore be 4.

Ref: "Determining the number of clusters in a data set", Wikipedia.

- The Calinski-Harabasz index
  - 群內方差和WGSS : 加總各群內各點離各群中心距離之平方和
  - 群間方差和BGSS : 加總各群中心與全資料中心距離之平方和
  - BGSS越大越好，WGSS越小越好，因此得到的分數越高越好。

$$\boxed{\mathcal{C} = \frac{BGSS/(K-1)}{WGSS/(N-K)} = \frac{N-K}{K-1}\frac{BGSS}{WGSS}}$$

Ref: "Clustering Indices", clusterCrit package, R project.

K-means variations

- Recomputing the centroid after every assignment (rather than after all points are re-assigned) can improve speed of convergence of *K*-means

  每個點調整後就重算重心，可以加快收斂

Evaluation of Clustering

What Is A Good Clustering?

- Internal criterion: A good clustering will produce high quality clusters in which:
  - the intra-class (that is, intra-cluster) similarity is high 群內同質性越高越好
  - the inter-class similarity is low 群間差異大
    - The measured quality of a clustering depends on both the document representation and the similarity measure used

External criteria for clustering quality

- Based on a gold standard data set (ground truth)
  - e.g., the Reuters collection we also used for the evaluation of classification
- Goal: Clustering should reproduce the classes in the gold standard
- Quality measured by its ability to discover some or all of the hidden patterns

用挑出中間不符合的份子來評估分群好不好

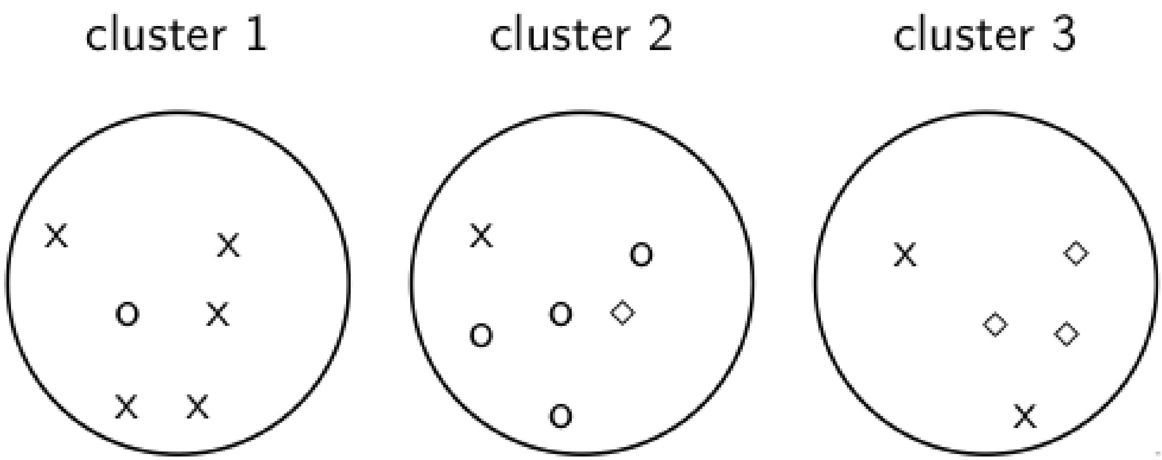
External criterion: Purity

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

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

  purity是群中最多一類佔該群總數之比例

Example for computing purity



To compute purity:
$5 = \max_j |\omega_1 \cap c_j|$ (class x, cluster 1)
$4 = \max_j |\omega_2 \cap c_j|$ (class o, cluster 2)
$3 = \max_j |\omega_3 \cap c_j|$ (class ⋄, cluster 3)
Purity is $(1/17) \times (5 + 4 + 3) \approx 0.71$.

Rand Index

Number of points	Same Cluster in clustering 分在同一群	Different Clusters in clustering 分在不同群
Same class in ground truth 已知同一類	A	C
Different classes in ground truth 已知不同類	B	D

Rand index: symmetric version

$$RI = \frac{A + D}{A + B + C + D}$$

Compare with standard Precision and Recall.

$$P = \frac{A}{A + B} \qquad R = \frac{A}{A + C}$$

Rand Index example: 0.68

Number of points	Same Cluster in clustering	Different Clusters in clustering
Same class in ground truth	20	24
Different classes in ground truth	20	72

DBSCAN algorithm

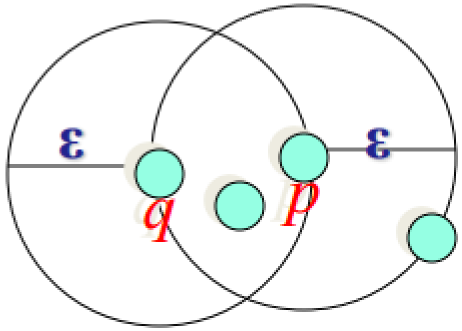
DBSCAN

- Density-based clustering : clusters are dense regions in the data space, separated by regions of lower object density. A cluster is defined as a maximal set of density-connected points
- May discovers clusters of arbitrary shape
  - c.f. K-mean is *spherical*

Ref: Density-Based Spatial Clustering of Applications with Noise

DBSCAN

- Definition
  - Eps-neighborhood of point p : points within radius eps from p
  - "High density" : Eps-neighborhood of a point contains at least MinPts of points



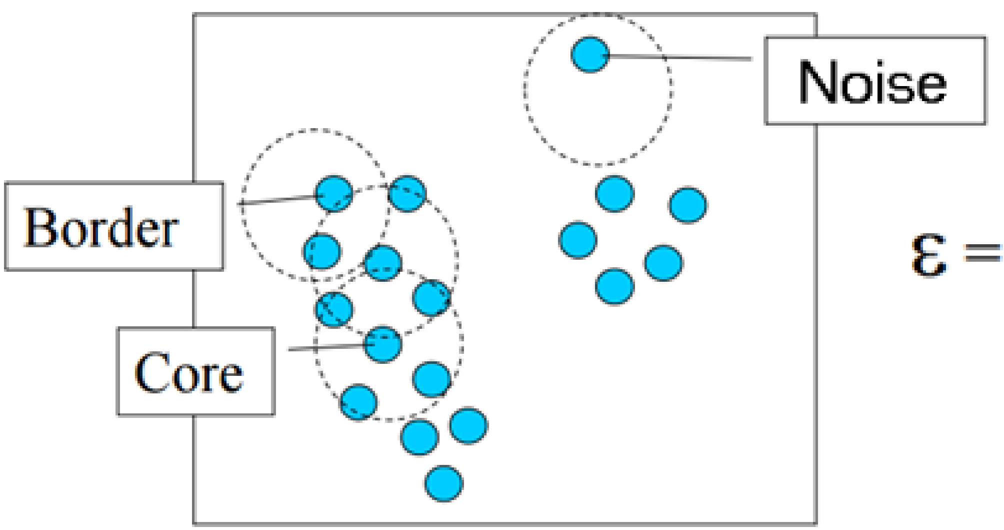
For radius ε, MinPts=4.

Density of p is "high"

Density of q is "low"

DBSCAN

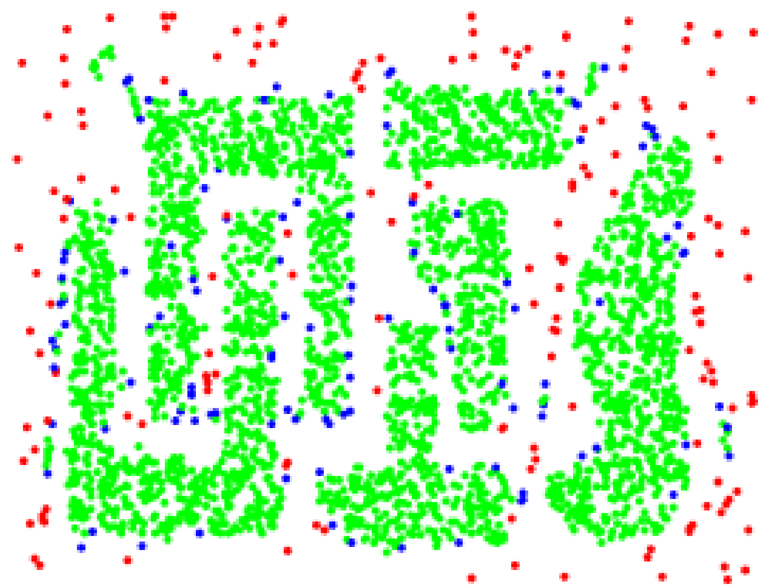
- Core points, Border points, and Noise points
  - A point is a core point if it has more than a specified number of points (MinPts) within Eps—These are points that are at the interior of a cluster
  - A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point
  - A noise point is any point that is not a core point nor a border point.



$\varepsilon = 1\text{unit}, \text{MinPts} = 5$

Example

Original Points



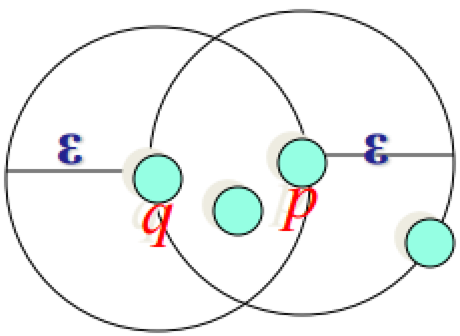
Point types: core, border and outliers

ε = 10, MinPts = 4

See http://www.cse.buffalo.edu/~jing/cse601/fa12/materials/clustering_density.pdf

DBSCAN

- Directly density-reachable
  - point q is directly density-reachable from point p if p is a core object and q is in eps-neighborhood of p



MinPts=4

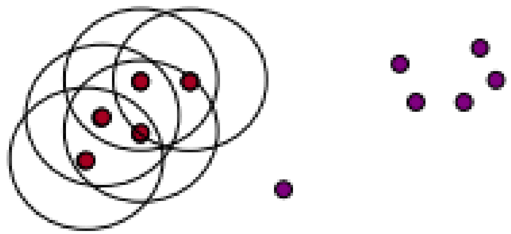
q is directly density-reachable from p

p is not directly density-reachable from q

density-reachability is <u>asymmetric</u>

DBSCAN Algorithm: Example

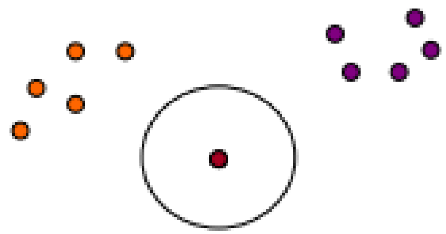
- **Parameter**
  - $\varepsilon = 2$ cm
  - $MinPts = 3$



for each $o \in D$ **do**
    if o is not yet classified **then**
        if o is a core-object **then**
            collect all objects density-reachable from o
            and assign them to a new cluster.
        else
            assign o to NOISE

DBSCAN Algorithm: Example

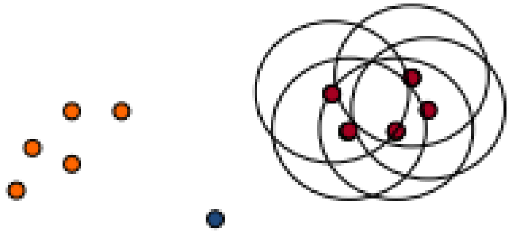
- **Parameter**
  - ε = 2 cm
  - *MinPts* = 3



for each $o \in D$ **do**
    if o is not yet classified **then**
        if o is a core-object **then**
            collect all objects density-reachable from o
            and assign them to a new cluster.
        else
            assign o to NOISE

DBSCAN Algorithm: Example

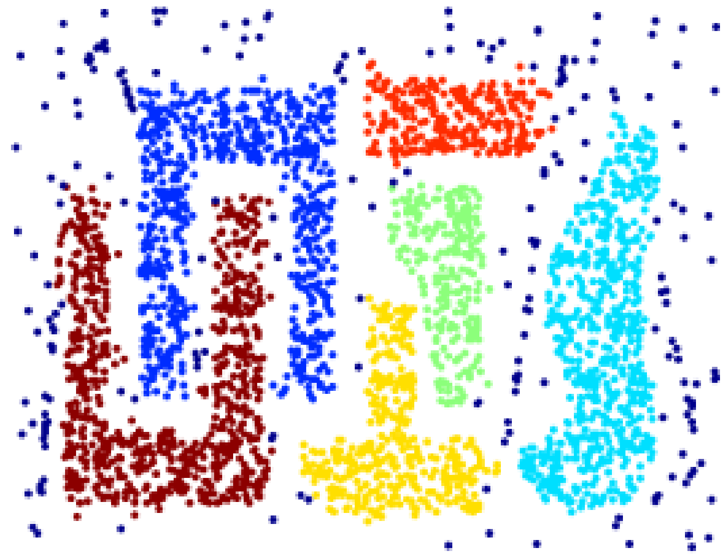
- **Parameter**
  - ε = 2 cm
  - *MinPts* = 3



```
for each o ∈ D do
    if o is not yet classified then
        if o is a core-object then
            collect all objects density-reachable from o
            and assign them to a new cluster.
        else
            assign o to NOISE
```

When DBSCAN Works Well

Original Points

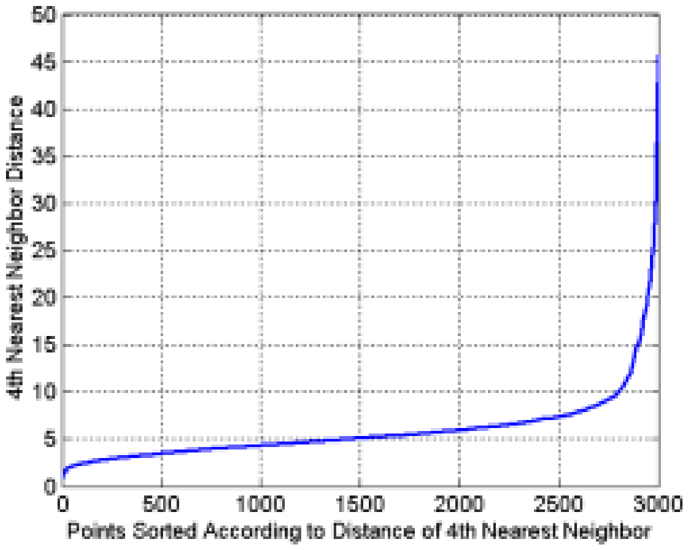


Clusters

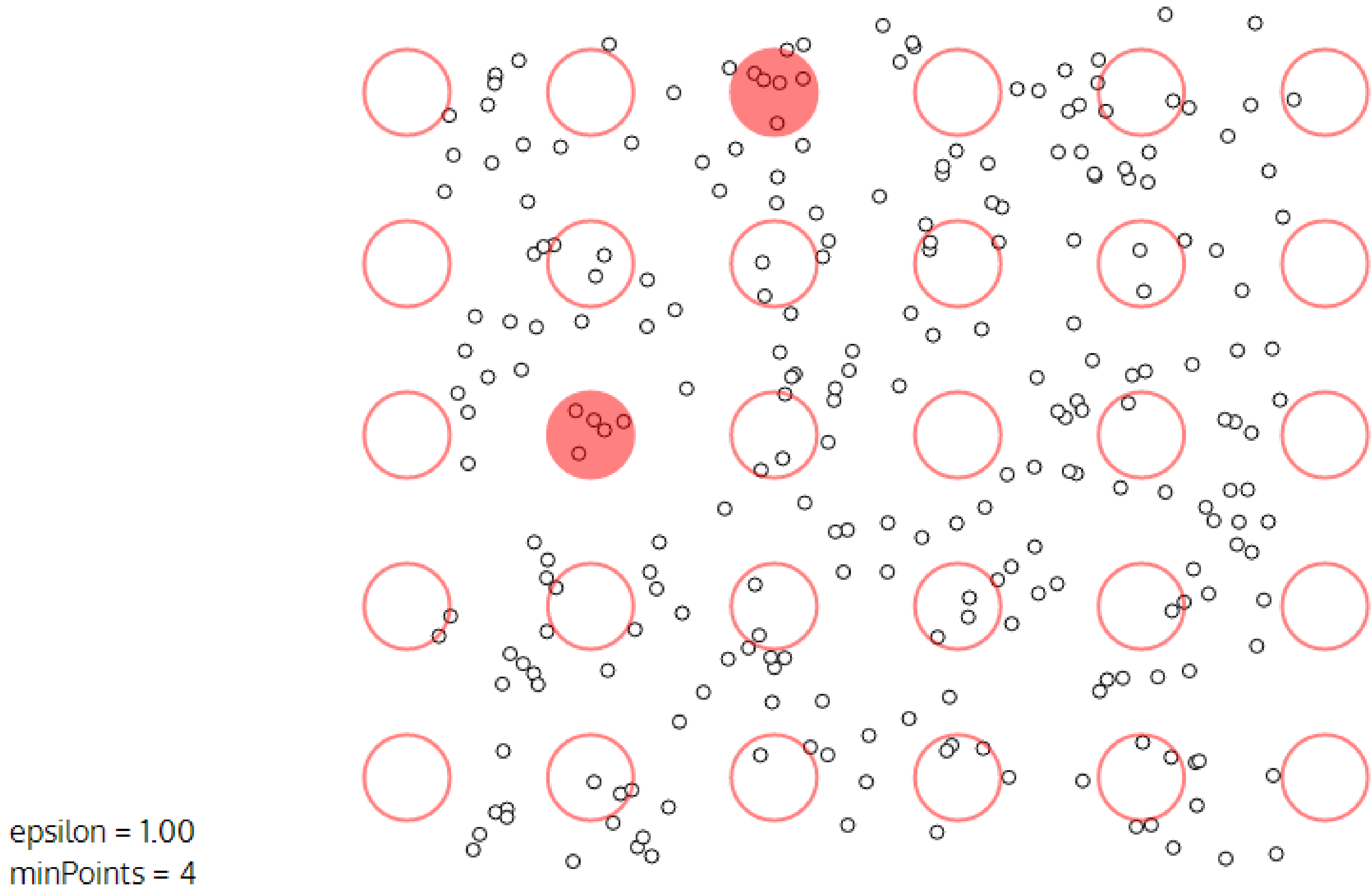
- Resistant to Noise
- Can handle clusters of different shapes and sizes

DBSCAN: Determining EPS and MinPts

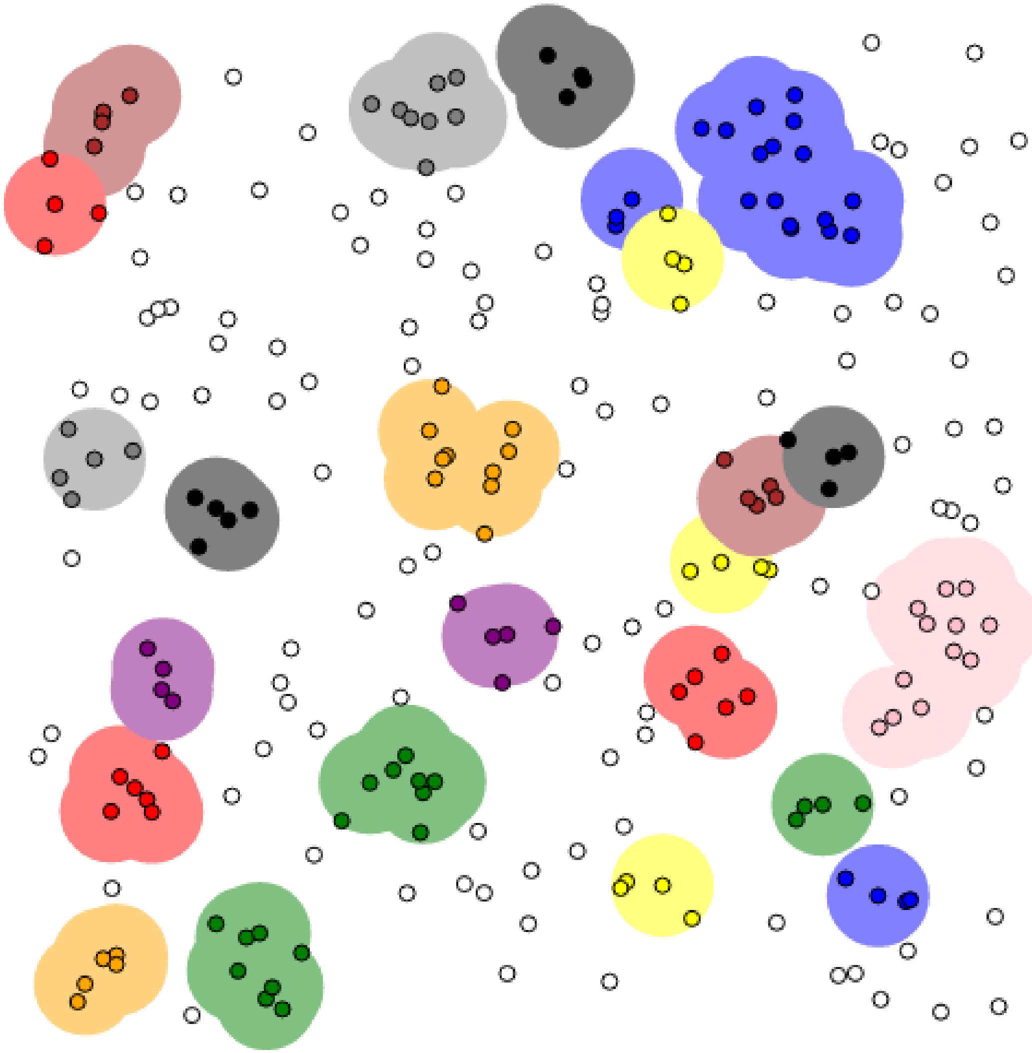
- Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
- Noise points have the k^{th} nearest neighbor at farther distance
- So, plot sorted distance of every point to its k^{th} nearest neighbor



Visualize the algorithm
http://www.naftaliharris.com/blog/visualizing-dbscan-clustering/



After clustering.



epsilon = 1.00
minPoints = 4

What kind of data would you like?

Uniform Points	Gaussian Mixture	Smiley Face
Density Bars	Packed Circles	Pimpled Smiley
DBSCAN Rings	Example A	

Restart

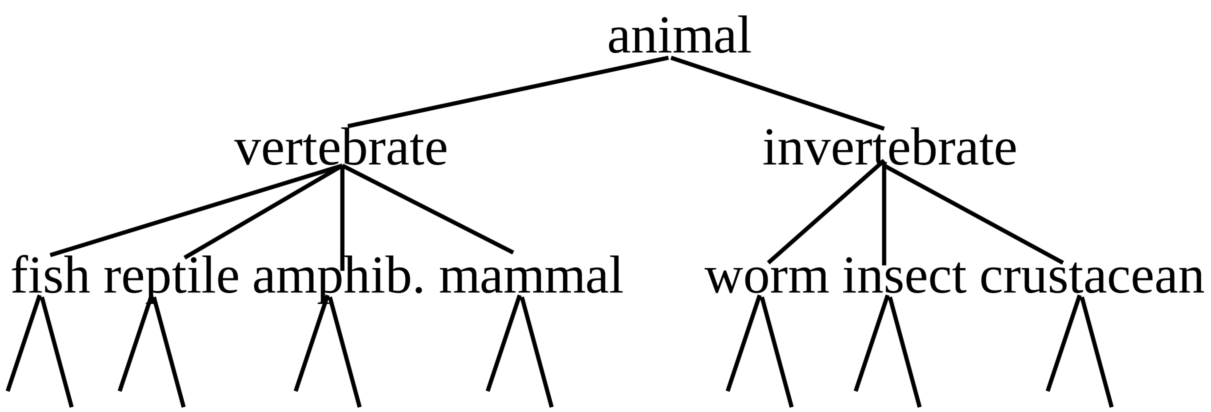
DBSCAN, (Density-Based Spatial Clustering of Applications with Noise), captures the insight that clusters are dense groups of points. The idea is that if a particular point belongs to a cluster, it should be near to lots of other points in that cluster.

It works like this: First we choose two parameters, a positive number epsilon and a

Hierarchical Clustering

Hierarchical Clustering

- Build a tree-based hierarchical taxonomy (*dendrogram*) from a set of documents.



- One approach: recursive application of a partitional clustering algorithm.

可由每一層不斷執行分群演算法所組成

Dendrogram: Hierarchical Clustering

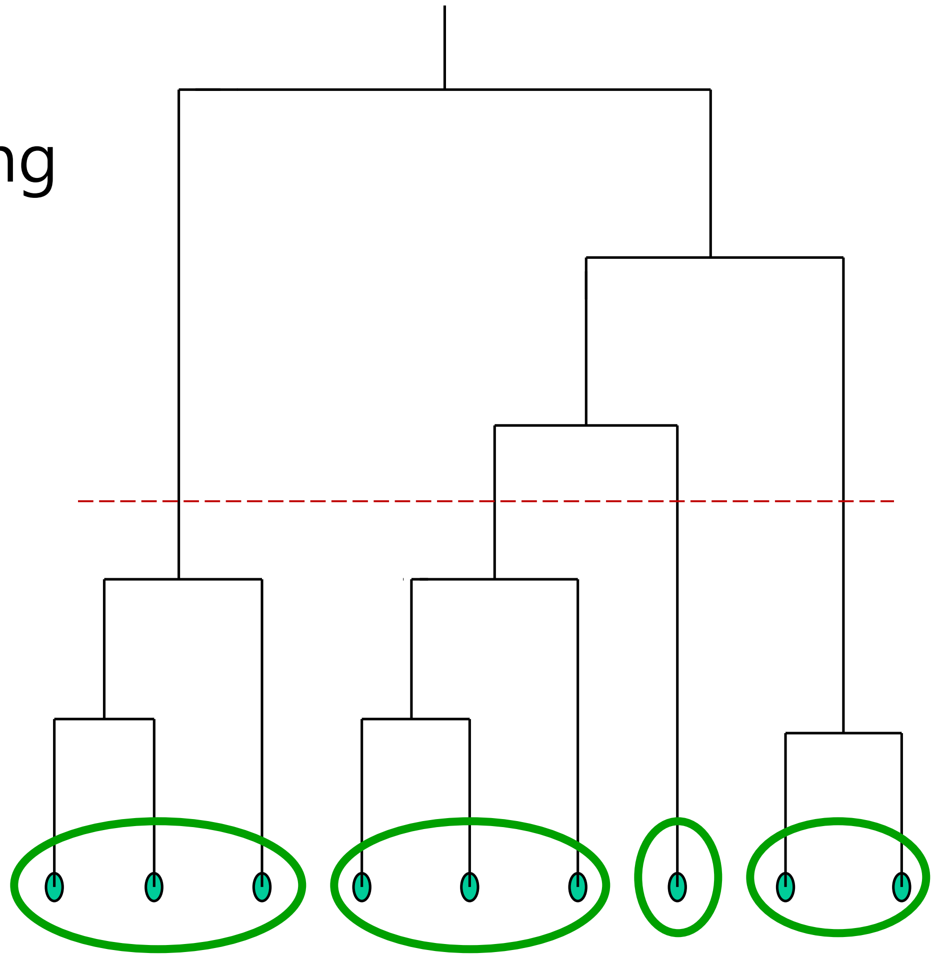
- Clustering obtained by cutting the dendrogram at a desired level: each connected component forms a cluster.

  另一種思考：

  對階層樹橫向切一刀,

  留下有連接在一起的,

  就構成一群



Hierarchical Clustering algorithms

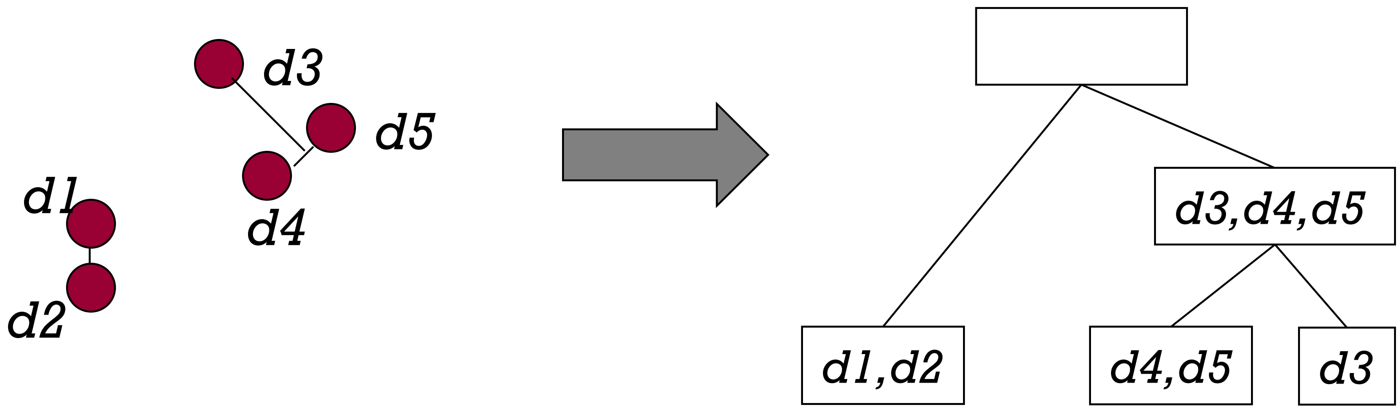
- Agglomerative (bottom-up): 由下往上聚合
  - Start with each document being a single cluster.
  - Eventually all documents belong to the same cluster.
- Divisive (top-down): 由上往下分裂
  - Start with all documents belong to the same cluster.
  - Eventually each node forms a cluster on its own.
- Does not require the number of clusters *k* in advance
  - 不需要先決定要分成幾群
- Needs a termination condition

Hierarchical Agglomerative Clustering (HAC) Algorithm

- Starts with each doc in a separate cluster
  每篇文件剛開始都自成一群
  - then repeatedly joins the *closest pair* of clusters, until there is only one cluster.
    不斷地將最近的二群做連接
- The history of merging forms a binary tree or hierarchy.
  連接的過程就構成一個二元階層樹

Dendrogram: Document Example

- As clusters *agglomerate*, docs likely to fall into a hierarchy of “topics” or concepts.



Closest pair of clusters 如何計算最近的二群

- Many variants to defining closest pair of clusters
- **Single-link** 挑群中最近的一點來代表
  - Similarity of the *most* cosine-similar (single-link)
- **Complete-link** 挑群中最遠的一點來代表
  - Similarity of the "furthest" points, the *least* cosine-similar
- **Centroid** 挑群中的重心來代表
  - Clusters whose centroids (centers of gravity) are the most cosine-similar
- **Average-link** 跟群中的所有點計算距離後取平均值
  - Average cosine between pairs of elements

Single Link Agglomerative Clustering

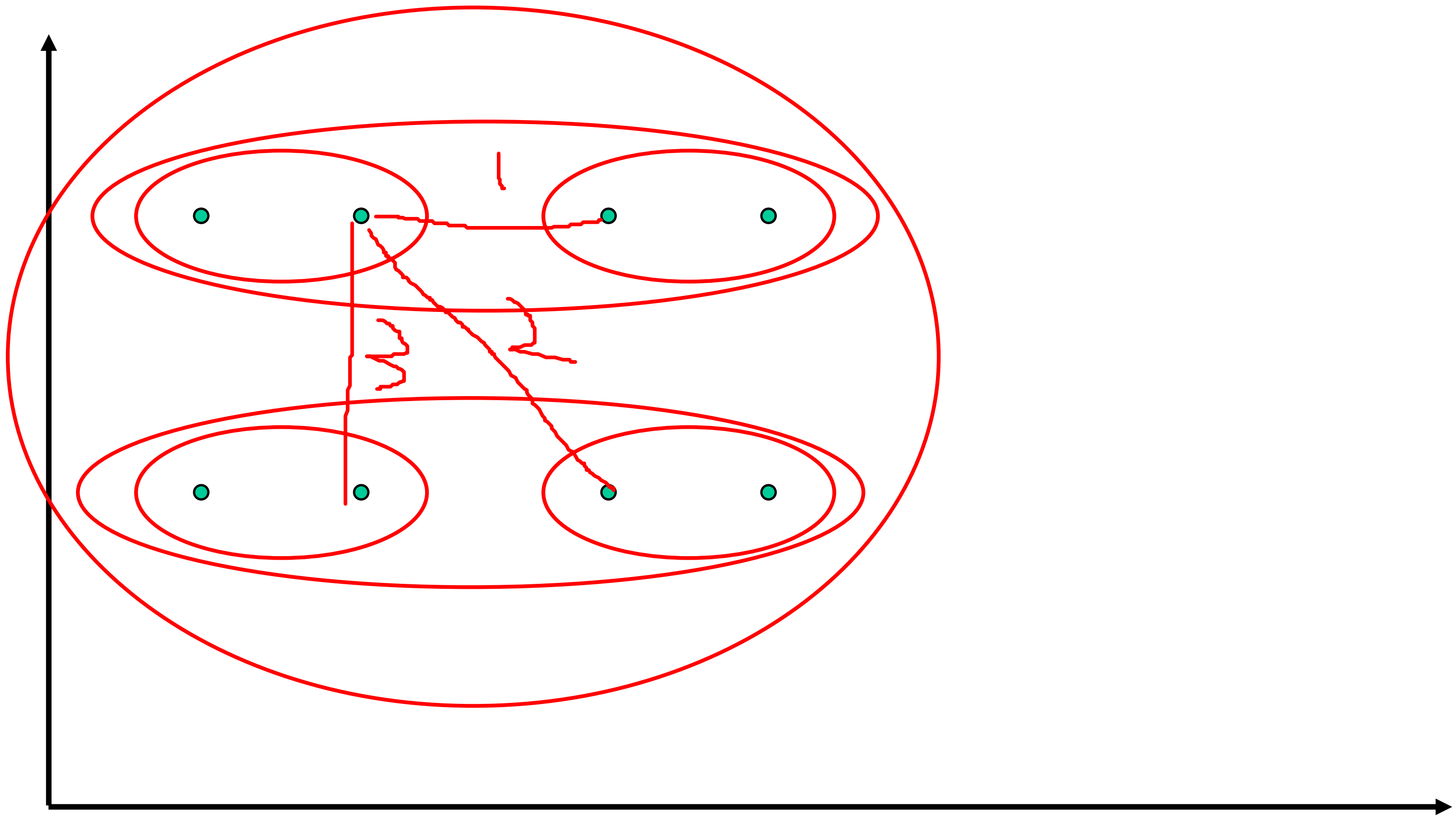
- Use maximum similarity of pairs:

$$sim(c_i, c_j) = \max_{x \in c_i, y \in c_j} sim(x, y)$$

- Can result in "straggly" (long and thin) clusters due to chaining effect. 長而鬆散的群集
- After merging c_i and c_j, the similarity of the resulting cluster to another cluster, c_k, is:

$$sim((c_i \cup c_j), c_k) = \max(sim(c_i, c_k), sim(c_j, c_k))$$

Single Link Example



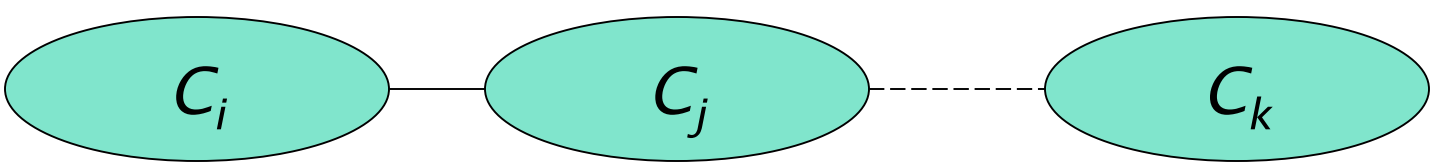
Complete Link Agglomerative Clustering

- Use minimum similarity of pairs:

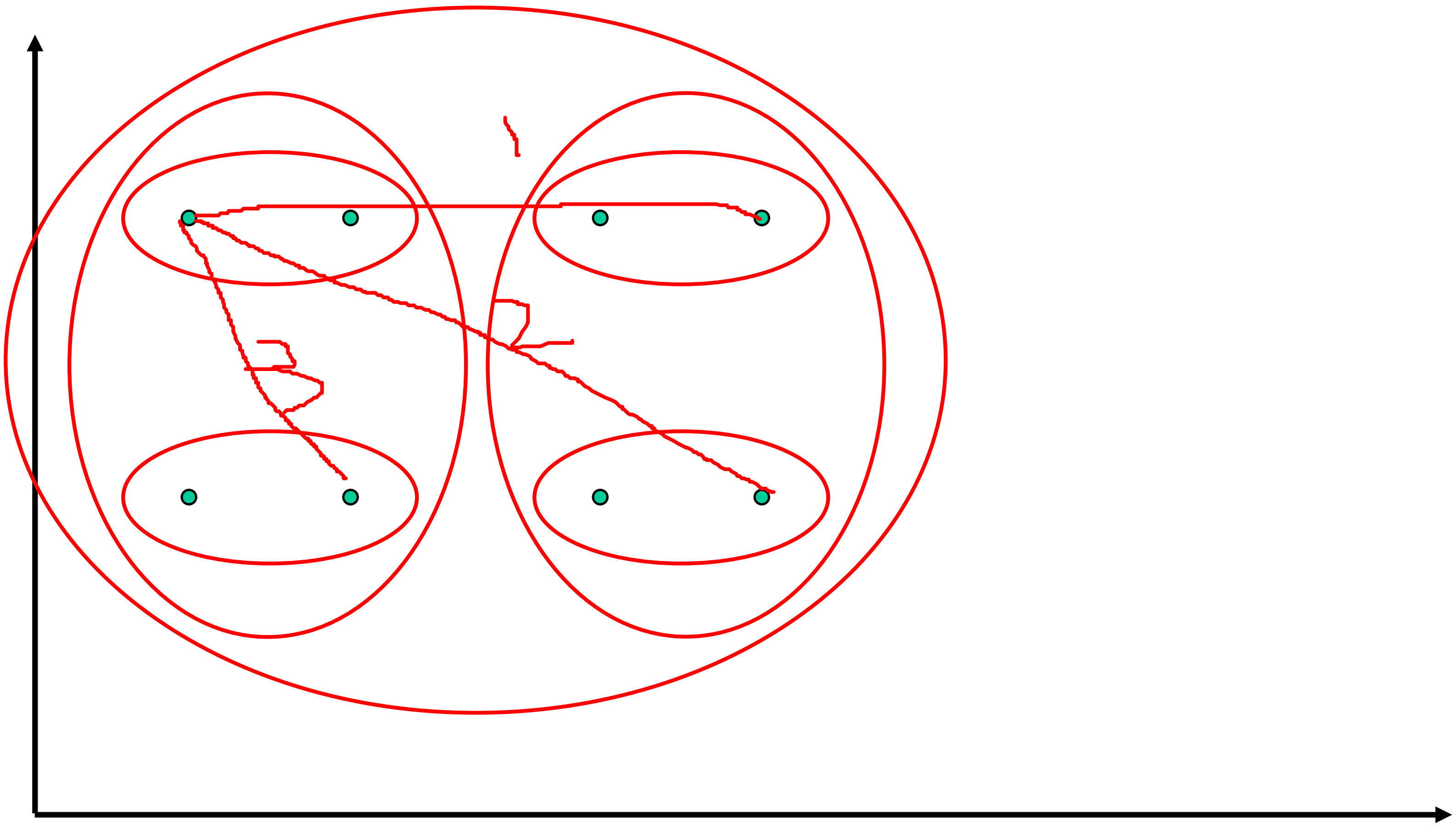
$$sim(c_i, c_j) = \min_{x \in c_i, y \in c_j} sim(x, y)$$

- Makes "tighter," spherical clusters that are typically preferable. 緊密一點的群集
- After merging c_i and c_j, the similarity of the resulting cluster to another cluster, c_k, is:

$$sim((c_i \cup c_j), c_k) = \min(sim(c_i, c_k), sim(c_j, c_k))$$



Complete Link Example



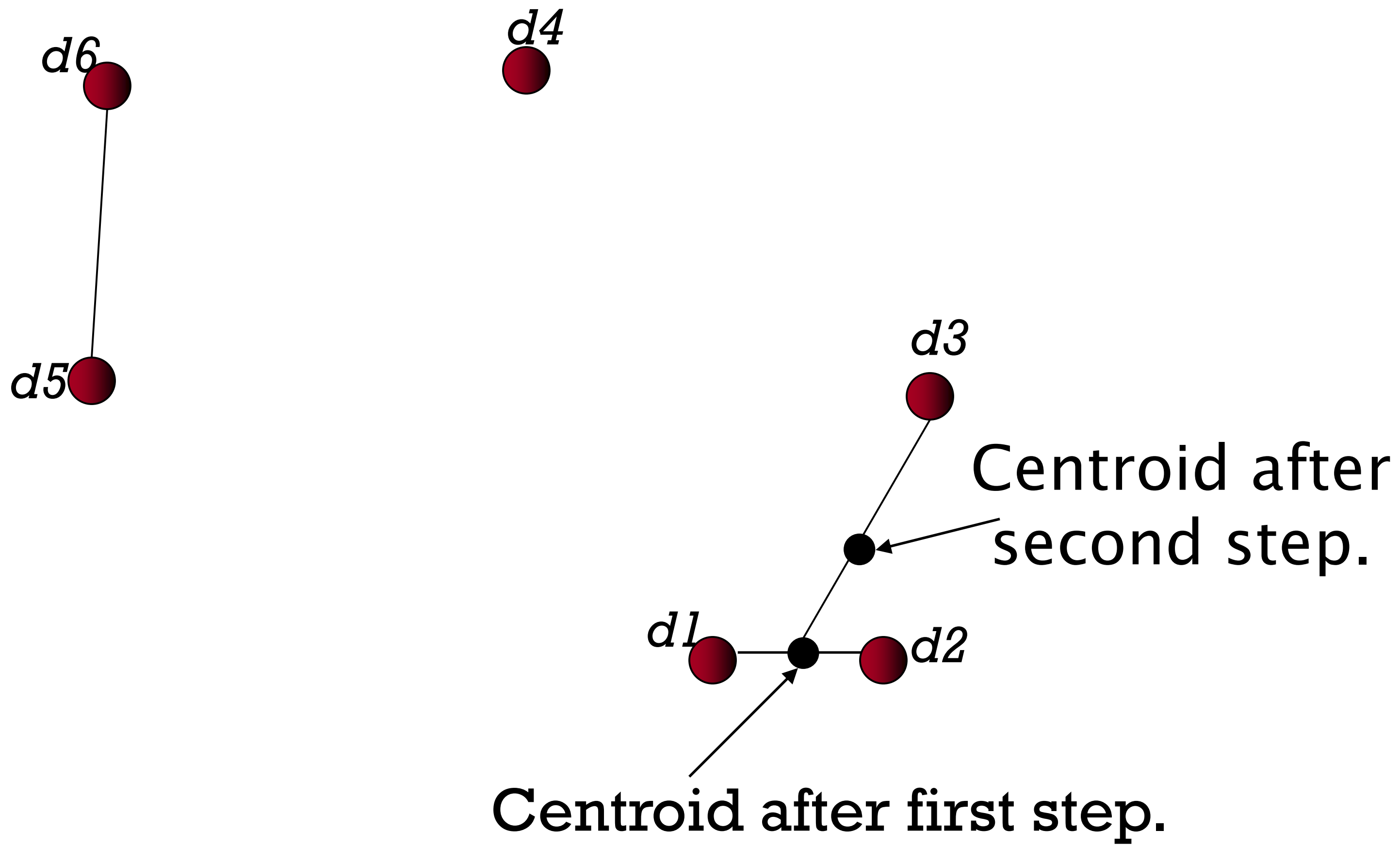
Computational Complexity

- In the first iteration, all HAC methods need to compute similarity of all pairs of n individual instances which is $O(n^2)$. 兩兩文件計算相似性
- In each of the subsequent n–2 merging iterations, compute the distance between the most recently created cluster and all other existing clusters.
  - 包含合併過程 $O(n^2 \log n)$

Key notion: *cluster representative*

- We want a notion of a representative point in a cluster
  如何代表該群→可以用中心或其它點代表
- Representative should be some sort of “typical” or central point in the cluster, e.g.,

Example: *n=6, k=3,* closest pair of centroids



Outliers in centroid computation

- Can ignore outliers when computing centroid.
- What is an outlier?
  - Lots of statistical definitions, e.g.
  - *moment* of point to centroid > M × some cluster *moment*.

Say 10.

簡單說就是距離太遠的點 (ex. 10倍遠)，直接忽略

Using Medoid As Cluster Representative

- The centroid does not have to be a document.
- Medoid: A cluster representative that is one of the documents 用以代表該群的某一份文件
  - Ex. the document closest to the centroid
- Why use Medoid ?
  - Consider the representative of a large cluster (>1000 documents)
  - The centroid of this cluster will be a *dense* vector
  - The medoid of this cluster will be a *sparse* vector

Clustering : discussion

Feature selection 選擇好的詞再來做分群

- Which terms to use as axes for vector space?
  - IDF is a form of feature selection
  - the most discriminating terms 鑑別力好的詞
    - Ex. use only nouns/noun phrases

Labeling 在分好的群上加標記

- After clustering algorithm finds clusters - how can they be useful to the end user?
- Need pithy label for each cluster 加上簡潔扼要的標記
  - In search results, say "Animal" or "Car" in the ***jaguar*** example.
  - In topic trees (Yahoo), need navigational cues.
    - Often done by hand, a posteriori. 事後以人工編輯

How to Label Clusters

- Show titles of typical documents

  用幾份代表文件的標題做標記

- Show words/phrases prominent in cluster

  用幾個較具代表性的詞做標記

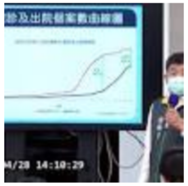
  – More likely to fully represent cluster
  – Use distinguishing words/phrases

    配合自動產生關鍵詞的技術

肺炎

全部 新聞 影片 圖片 購物 更多 設定 工具

約有 136,000,000 項結果 (搜尋時間：0.24 秒)



武漢肺炎28日零確診首度連3天無新增病例
中央社即時新聞 - 16 小時前
中央社記者陳偉婷台北28日電）中央流行疫情指揮中心宣布，台灣今天無新增武漢肺炎（2019冠狀病毒疾病，COVID-19）確診病例，疫情指揮中心 ...

武漢肺炎》台灣連3日零確診！ 全球確診破306萬死亡逾21萬
自由時報電子報 - 7 小時前

武漢肺炎全球確診破300萬各國封城解封情況一次看
中央社即時新聞 - 16 小時前

快訊》讚讚讚！台灣武漢肺炎連3天0確診307人解除隔離
新頭殼 - 16 小時前

直播／今日再度0確診！國內新冠肺炎維持429例
udn 聯合新聞網 (新聞發布) - 16 小時前

查看全部

武漢肺炎全球最新情報4/28
中央社即時新聞 - 22 小時前
中央社台北28日電）2019冠狀病毒疾病（COVID-19，武漢肺炎）疫情有緩和趨勢，多國都著眼放寬封鎖令，但民眾已不耐遲遲未解禁，冒險外出、抗命 ...

英牛津團隊進度領先9月前可能推出武漢肺炎疫苗
中央社即時新聞 - 11 小時前
中央社倫敦27日綜合外電報導）2019冠狀病毒疾病（COVID-19，武漢肺炎）肆虐全球，各國競相開發疫苗。根據「紐約時報」，英國牛津大學實驗室領先 ...

武漢肺炎》猴子實驗成功牛津大學：疫苗最快9月問世
自由時報電子報 - 13 小時前

【武漢肺炎】牛津疫苗為何如此快進入臨床階段？
立場新聞 - 7 小時前

查看全部

Labeling

- Common heuristics - list 5-10 most frequent terms in the centroid vector. 通常用5~10個詞來代表該群
- Differential labeling by frequent terms
  - Within a collection "Computers", clusters all have the word ***computer*** as frequent term.
  - Discriminant analysis of centroids.
  - 要挑選有鑑別力的詞

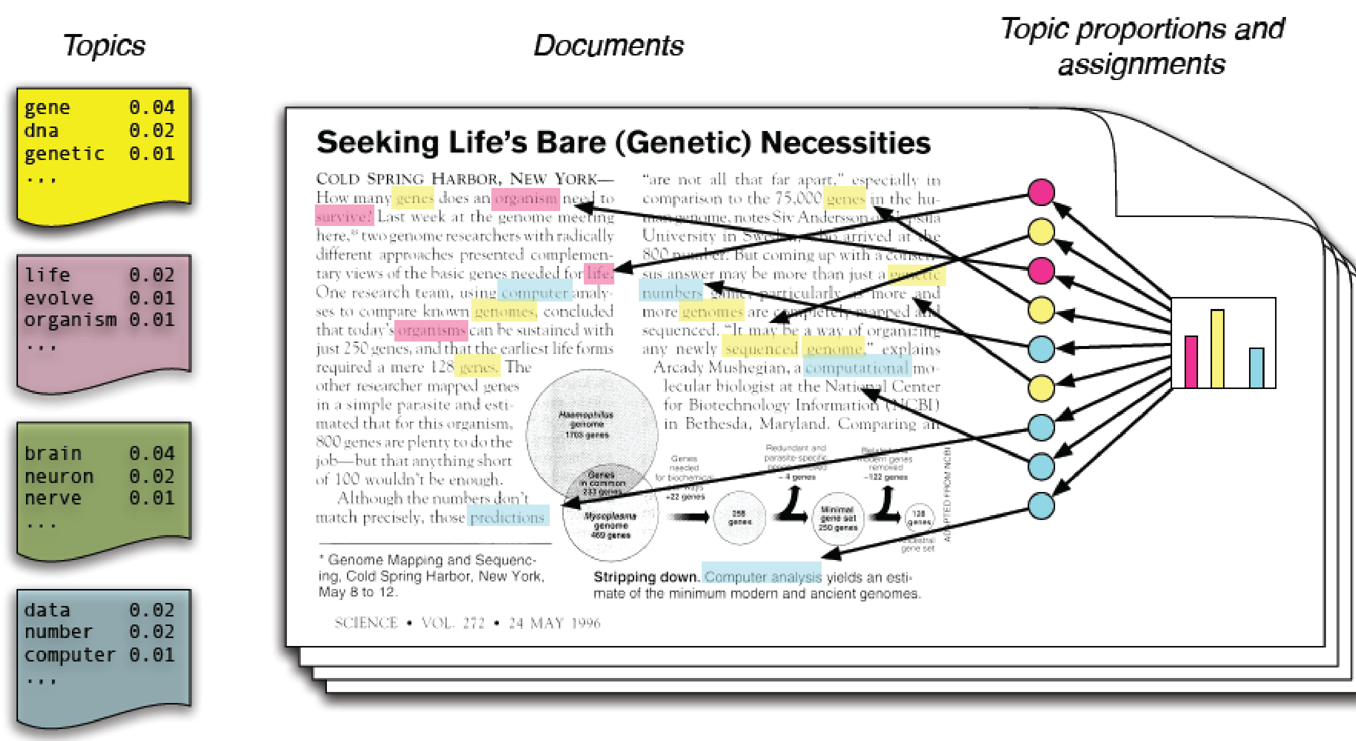
Topic Model 應用分群在主題建模上

- 在數量龐大的文件集合中自動地發現某些結構 (主題)，並將每個主題用某些關鍵字的形式表現 (註: 即Bag-of-Word模型)；隨後，還可以知道每篇文章中各個主題占得比重如何，並據此判斷兩篇文章的相關程度。
- 分群演算法就可以將關鍵字群聚成若干主題。

延伸學習

- Topic Modelling (with LSA, pLSA, or LDA)

brain
computer
data
dna
evolve
gene
genetic
life
organism
nerve
neuron
number
...



Clustering words into topics

Discussions