Nhập môn Học máy và Khai phá dữ liệu
(*IT3190*)

Nguyễn Nhật Quang

quang.nguyennhat@hust.edu.vn

Trường Đại học Bách Khoa Hà Nội

Viện Công nghệ thông tin và truyền thông

Năm học 2019-2020

Nội dung môn học:

Giới thiệu về Học máy và Khai phá dữ liệu

Tiền xử lý dữ liệu

Đánh giá hiệu năng của hệ thống

Hồi quy

Phân lớp

Phân cụm

- **Phân cụm dựa trên phân tách: k-Means**
- **Phân cụm phân cấp: HAC**

Phát hiện luật kết hợp

Học có vs. không có giám sát

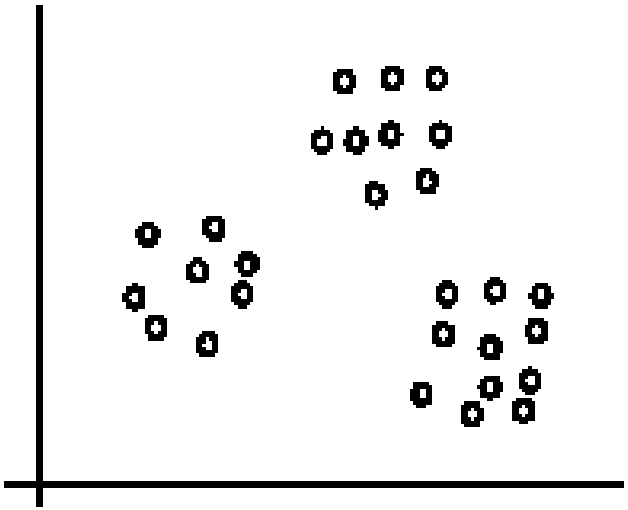
- **Học có giám sát (Supervised learning)**
  - Tập dữ liệu (dataset) bao gồm các ví dụ, mà mỗi ví dụ được *gắn kèm với một nhãn lớp/giá trị đầu ra mong muốn*
  - Mục đích là học (xấp xỉ) một giả thiết/hàm mục tiêu (vd: phân lớp, hồi quy) phù hợp với tập dữ liệu hiện có
  - Hàm mục tiêu học được (learned target function) sau đó sẽ được dùng để phân lớp/dự đoán đối với các ví dụ mới
- **Học không có giám sát (Unsupervised learning)**
  - Tập dữ liệu (dataset) bao gồm các ví dụ, mà mỗi ví dụ *không có thông tin về nhãn lớp/giá trị đầu ra mong muốn*
  - Mục đích là tìm ra (xác định) các cụm/các cấu trúc/các quan hệ tồn tại trong tập dữ liệu hiện có

Phân cụm

- Phân cụm/nhóm (Clustering) là phương pháp học không có giám sát được sử dụng phổ biến nhất
  - Tồn tại các phương pháp học không có giám sát khác, ví dụ: Lọc cộng tác (Collaborative filtering), Khai phá luật kết hợp (Association rule mining), ...
- Bài toán Phân cụm:
  - Đầu vào: Một tập dữ liệu không có nhãn (các ví dụ không có nhãn lớp/giá trị đầu ra mong muốn)
  - Đầu ra: Các cụm (nhóm) của các ví dụ
- Một **cụm (cluster)** là một tập các ví dụ:
  - Tương tự với nhau (theo một ý nghĩa, đánh giá nào đó)
  - Khác biệt với các ví dụ thuộc các cụm khác

Phân cụm – Ví dụ minh họa

Các ví dụ được phân chia thành 3 cụm



[Liu, 2006]

Phân cụm – Các thành phần

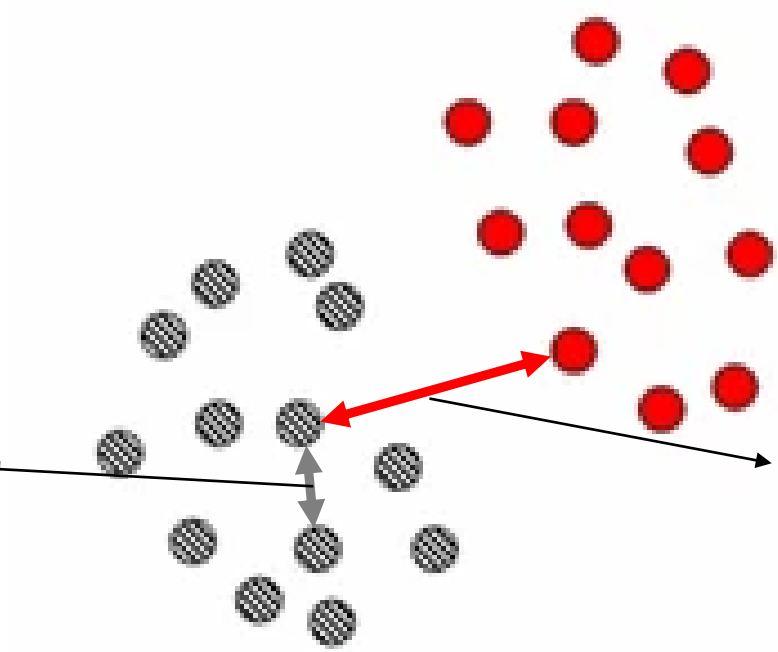
- Hàm tính khoảng cách (độ tương tự, độ khác biệt)
- Giải thuật phân cụm
  - **Dựa trên phân tách (Partition-based clustering)**
  - **Dựa trên tích tụ phân cấp (Hierarchical clustering)**
  - Bản đồ tự tổ thức (Self-organizing map – SOM)
  - Các mô hình hỗn hợp (Mixture models)
  - …
- Đánh giá chất lượng phân cụm (Clustering quality)
  - Khoảng cách/sự khác biệt *giữa các cụm* → Cần được *cực đại* hóa
  - Khoảng cách/sự khác biệt *bên trong một cụm* → Cần được *cực tiểu* hóa

Bài toán phân cụm: Đánh giá hiệu năng

- Làm sao để đánh giá hiệu quả phân cụm?
  - *External evaluation*: Sử dụng thêm thông tin bên ngoài (ví dụ: nhãn lớp của mỗi ví dụ)
    - Ví dụ: Accuracy, Precision,…
  - *Internal evaluation*: Chỉ dựa trên các ví dụ được phân cụm (mà không có thêm thông tin bên ngoài)
    - Rất thách thức!
    - Là trọng tâm được trình bày tiếp theo

Internal evaluation: Nguyên tắc

- Sự gắn kết (compactness/coherence)
  - Khoảng cách giữa các ví dụ trong cùng cụm (intra-cluster distance)
- Sự tách biệt (separation)
  - Khoảng cách giữa các ví dụ thuộc 2 cụm khác nhau (inter-cluster distance)



Khoảng cách giữa các ví dụ trong cùng cụm (intra-cluster distance)

Khoảng cách giữa các ví dụ thuộc 2 cụm khác nhau (inter-cluster distance)

Internal evaluation: Các độ đo (1)

- **RMSSTD** (Root-mean-square standard deviation)
  - Đánh giá sự gắn kết (compactness) của các cụm thu được
  - Mong muốn giá trị RMSSTD càng nhỏ càng tốt!

$$RMSSTD = \sqrt{\frac{\sum_{i=1}^{k} \sum_{x \in C_i} \|x - m_i\|^2}{P \sum_{i=1}^{k} (n_i - 1)}}$$

  - k: Số lượng các cụm
  - C_i: Cụm thứ i
  - m_i: Điểm trung tâm (center/centroid) của cụm C_i
  - P: Tổng số chiều (số lượng thuộc tính) biểu diễn ví dụ
  - n_i: Tổng số các ví dụ thuộc cụm C_i

Internal evaluation: Các độ đo (2)

- **R-squared**
  - Đánh giá sự phân tách (separation) giữa các cụm thu được
  - Mong muốn giá trị R-squared càng lớn càng tốt!

$$R\text{-}squared = \frac{\sum_{x \in D} \|x - g\|^2 - \sum_{i=1}^{k} \sum_{x \in C_i} \|x - m_i\|^2}{\sum_{x \in D} \|x - g\|^2}$$

  - k: Số lượng các cụm
  - C_i: Cụm thứ i
  - m_i: Điểm trung tâm (center/centroid) của cụm C_i
  - D: Tập toàn bộ các ví dụ
  - g: Điểm trung tâm (center/centroid) của toàn bộ các ví dụ

Internal evaluation: Các độ đo (3)

- **Dunn index**
  - ~ (Separation/Compactness): Tỷ lệ giữa khoảng cách nhỏ nhất giữa các cụm (minimum inter-cluster distance) và khoảng cách cực đại trong một cụm (maximum intra-cluster distance)
  - Mong muốn giá trị Dunn index càng lớn càng tốt!

$$Dunn - index = \frac{min_{1 \le i < j \le k} inter - distance(i,j)}{max_{1 \le h \le k} intra - distance(h)}$$

  - k: Số lượng các cụm
  - inter-distance(i,j): Khoảng cách giữa 2 cụm i và j
  - intra-distance(h): Khoảng cách (sự khác biệt) giữa các ví dụ thuộc cụm h

Internal evaluation: Các độ đo (4)

- **Davies–Bouldin index**
  - ~ (Compactness/Separation): Tỷ lệ giữa khoảng cách trung bình trong cụm (average intra-cluster distance) và khoảng cách giữa các cụm (inter-cluster distance)
  - Mong muốn giá trị Davies–Bouldin index càng nhỏ càng tốt!

$$DB - index = \frac{1}{k}\sum_{i=1}^{k} max_{j \neq i} \frac{\frac{1}{n_i}\sum_{x \in C_i} d(x, m_i) + \frac{1}{n_j}\sum_{x \in C_j} d(x, m_j)}{d(m_i, m_j)}$$

  - k: Số lượng các cụm
  - n_i, m_i: Tổng số các ví dụ và tâm cụm i
  - n_j, m_j: Tổng số các ví dụ và tâm cụm j
  - $d(m_i,m_j)$: Khoảng cách 2 tâm cụm m_i và m_j

Phân cụm k-Means

- Là phương pháp phổ biến nhất trong các phương pháp phân cụm dựa trên chia cắt (partition-based clustering)
- Tập dữ liệu $D=\{x_1,x_2,\ldots,x_r\}$
  - x_i là một ví dụ (một vectơ trong một không gian *n* chiều)
- Giải thuật k-means phân chia (partitions) tập dữ liệu thành k cụm
  - Mỗi cụm (cluster) có một điểm trung tâm, được gọi là **centroid**
  - k (tổng số các cụm thu được) là một giá trị được xác định trước (vd: được chỉ định bởi người thiết kế hệ thống phân cụm)

k-Means – Các bước chính

Với một giá trị k được xác định trước

- Bước 1. Chọn ngẫu nhiên k ví dụ (được gọi là **các hạt nhân – seeds**) để sử dụng làm *các điểm trung tâm ban đầu* (*initial centroids*) của *k* cụm
- Bước 2. Đối với mỗi ví dụ, *gán nó vào cụm* (trong số k cụm) có điểm trung tâm (centroid) gần ví dụ đó nhất
- Bước 3. Đối với mỗi cụm, *tính toán lại điểm trung tâm (centroid) của nó* dựa trên tất cả các ví dụ thuộc vào cụm đó
- Bước 4. Dừng lại nếu *điều kiện hội tụ* (*convergence criterion*) được thỏa mãn; nếu không, quay lại Bước 2

***k*-means**(D, k)

- D: Tập ví dụ học
- k: Số lượng cụm kết quả (thu được)

Lựa chọn ngẫu nhiên k ví dụ trong tập D để làm các điểm trung tâm ban đầu (initial centroids)

while not CONVERGENCE

  for each ví dụ $x \in D$

    Tính các khoảng cách từ x đến các điểm trung tâm (centroid)

    Gán x vào cụm có điểm trung tâm (centroid) gần x nhất

  end for

  for each cụm

    Tính (xác định) lại điểm trung tâm (centroid) dựa trên các ví dụ hiện thời đang thuộc vào cụm này

end while

return {k cụm kết quả}

Điều kiện hội tụ

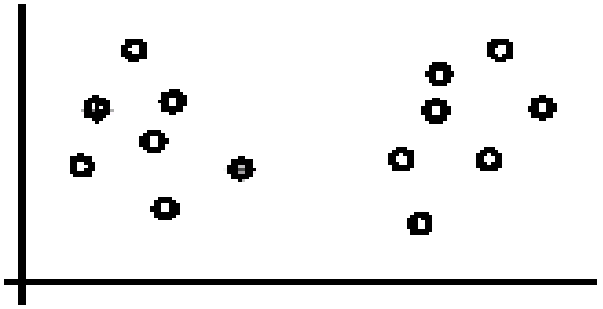
Quá trình phân cụm kết thúc, nếu:

- Không có (hoặc có không đáng kể) việc gán lại các ví dụ vào các cụm khác, *hoặc*
- Không có (hoặc có không đáng kể) thay đổi về các điểm trung tâm (centroids) của các cụm, *hoặc*
- Giảm không đáng kể về tổng lỗi phân cụm:

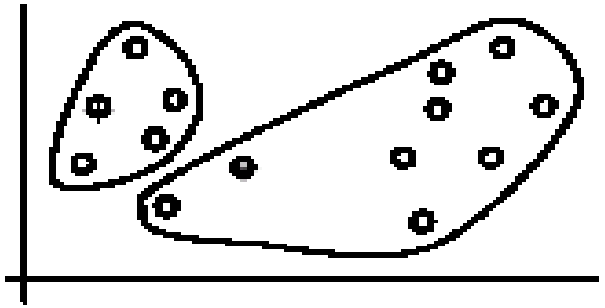
$$Error = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} d(\mathbf{x}, \mathbf{m_i})^2$$

  - C_i: Cụm thứ i
  - $\mathbf{m_i}$: Điểm trung tâm (centroid) của cụm C_i
  - $d(\mathbf{x}, \mathbf{m}_i)$: Khoảng cách (khác biệt) giữa ví dụ $\mathbf{x}$ và điểm trung tâm $\mathbf{m}_i$

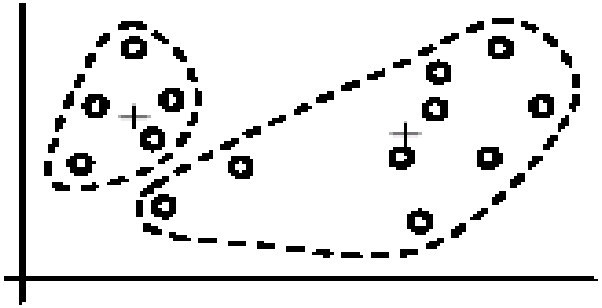
k-Means – Minh họa (1)



(A). Random selection of k centers



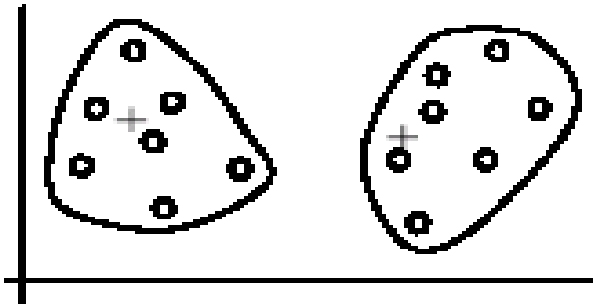
Iteration 1: (B). Cluster assignment



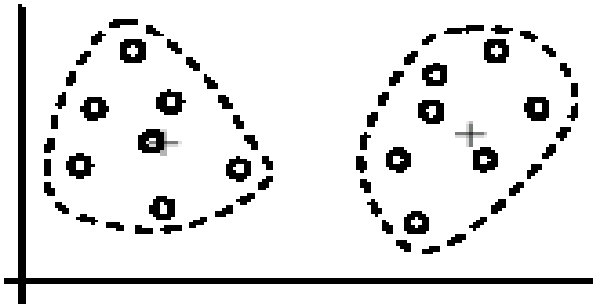
(C). Re-compute centroids

[Liu, 2006]

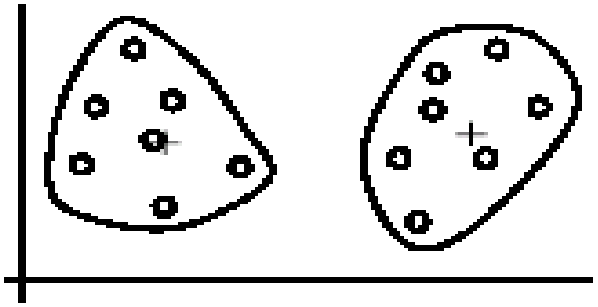
k-Means – Minh họa (2)



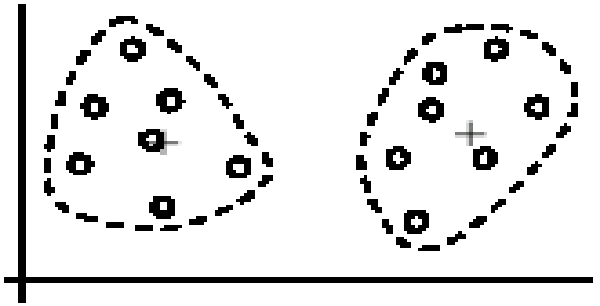
Iteration 2: (D). Cluster assignment



(E). Re-compute centroids



Iteration 3: (F). Cluster assignment



(G). Re-compute centroids

[Liu, 2006]

Điểm trung tâm, Hàm khoảng cách

- Xác định điểm trung tâm: Điểm trung bình (*Mean centroid*)

$$\mathbf{m_i} = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}$$

  - (vectơ) $\boldsymbol{m_i}$ là điểm trung tâm (centroid) của cụm C_i
  - $|C_i|$ kích thước của cụm C_i (tổng số ví dụ trong C_i)

- Hàm khoảng cách: *Euclidean distance*

$$d(\mathbf{x}, \mathbf{m_i}) = \|\mathbf{x} - \mathbf{m_i}\| = \sqrt{(x_1 - m_{i1})^2 + (x_2 - m_{i2})^2 + \ldots + (x_n - m_{in})^2}$$

  - (vectơ) $\boldsymbol{m_i}$ là điểm trung tâm (centroid) của cụm C_i
  - $d(\boldsymbol{x}, \boldsymbol{m_i})$ là khoảng cách giữa ví dụ $\boldsymbol{x}$ và điểm trung tâm $\boldsymbol{m_i}$

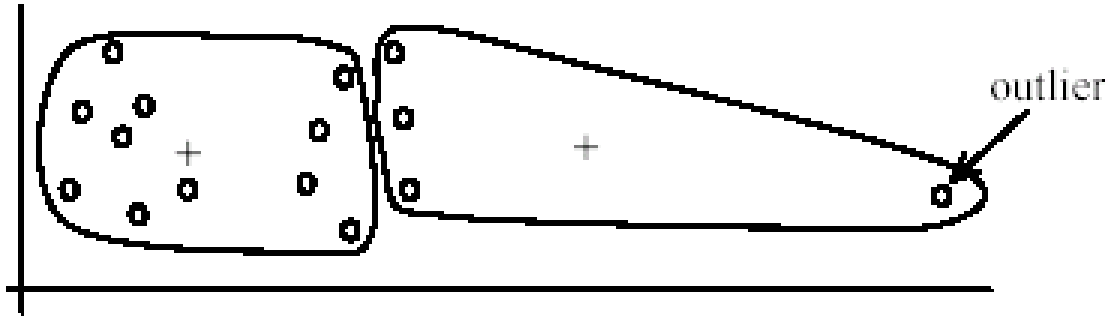
k-Means – Các ưu điểm

- Đơn giản
  - Rất dễ cài đặt
  - Rất dễ hiểu
- Hiệu quả
  - Độ phức tạp về thời gian ~ $O(r.k.t)$
    - r: Tổng số các ví dụ (kích thước của tập dữ liệu)
    - k: Tổng số cụm thu được
    - t: Tổng số bước lặp (của quá trình phân cụm)
  - Nếu cả 2 giá trị k và t đều nhỏ, thì giải thuật k-means được xem như là có độ phức tạp ở mức tuyến tính
- k-means là giải thuật phân cụm được dùng phổ biến nhất

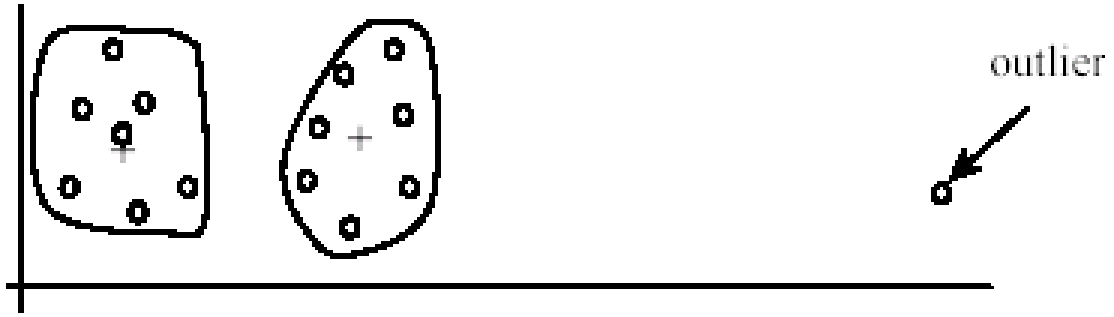
k-Means – Các nhược điểm (1)

- Giá trị k (số cụm thu được) phải được xác định trước
- Giải thuật k-means cần xác định cách tính điểm trung bình (centroid) của một cụm
  - Đối với các thuộc tính định danh (nominal attributes), giá trị trung bình có thể được xác định là giá trị phổ biến nhất
- Giải thuật k-means nhạy cảm (gặp lỗi) với ***các ví dụ ngoại lai (outliers)***
  - Các ví dụ ngoại lai là các ví dụ (rất) khác biệt với tất các ví dụ khác
  - Các ví dụ ngoại lai có thể do lỗi trong quá trình thu thập/lưu dữ liệu
  - Các ví dụ ngoại lai có các giá trị thuộc tính (rất) khác biệt với các giá trị thuộc tính của các ví dụ khác

k-Means – Các ví dụ ngoại lai



(A): Undesirable clusters



(B): Ideal clusters

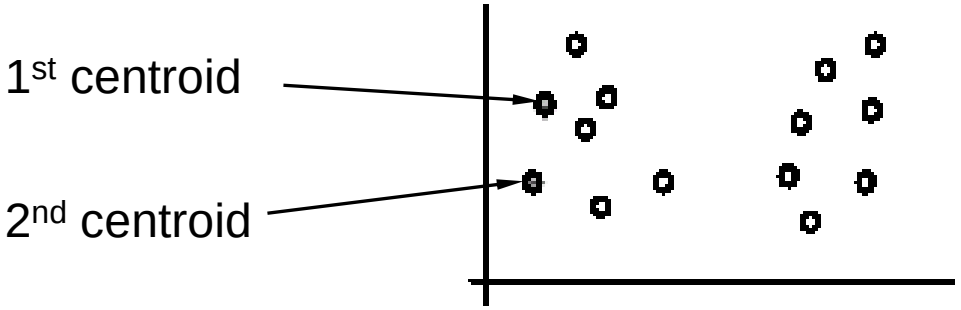
[Liu, 2006]

Giải quyết vấn đề ngoại lai

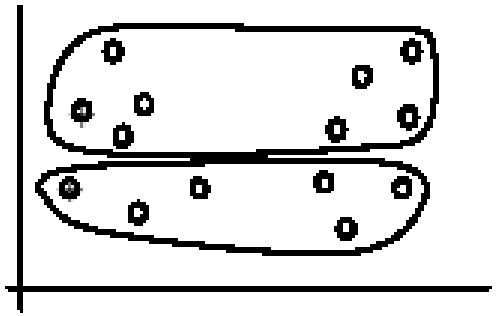
- Giải pháp 1. Trong quá trình phân cụm, cần loại bỏ một số các ví dụ quá khác biệt với (cách xa) các điểm trung tâm (centroids) so với các ví dụ khác
  - Để chắc chắn (không loại nhầm), theo dõi các ví dụ ngoại lai (outliers) qua một vài (thay vì chỉ 1) bước lặp phân cụm, trước khi quyết định loại bỏ
- Giải pháp 2. Thực hiện việc lấy mẫu ngẫu nhiên (a random sampling)
  - Do quá trình lấy mẫu chỉ lựa chọn một tập con nhỏ của tập dữ liệu ban đầu, nên khả năng một ngoại lai (outlier) được chọn là rất nhỏ
  - Gán các ví dụ còn lại của tập dữ liệu vào các cụm tùy theo đánh giá về khoảng cách (hoặc độ tương tự)

k-Means – Các nhược điểm (2)

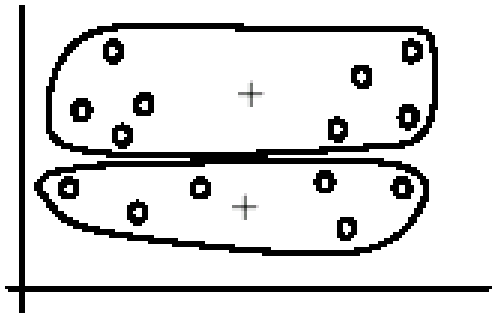
- Giải thuật k-means phụ thuộc vào việc chọn các điểm trung tâm ban đầu (initial centroids)



(A). Random selection of seeds (centroids)



(B). Iteration 1

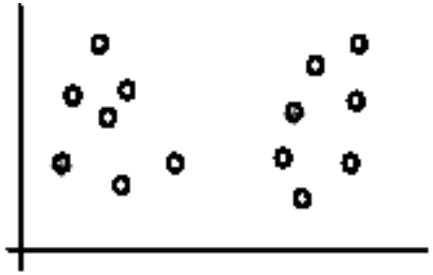


(C). Iteration 2

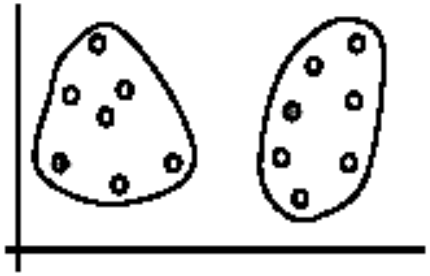
[Liu, 2006]

k-Means – Các hạt nhân ban đầu (1)

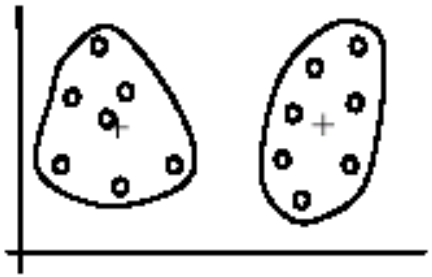
- Sử dụng các hạt nhân (seeds) khác nhau → Kết quả tốt hơn!
  - Thực hiện giải thuật k-means nhiều lần, mỗi lần bắt đầu với một tập (khác lần trước) các hạt nhân được chọn ngẫu nhiên



(A). Random selection of k seeds (centroids)



(B). Iteration 1



(C). Iteration 2

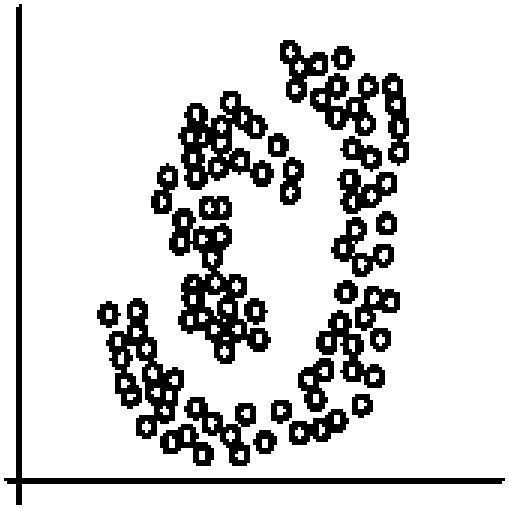
[Liu, 2006]

k-Means – Các hạt nhân ban đầu (2)

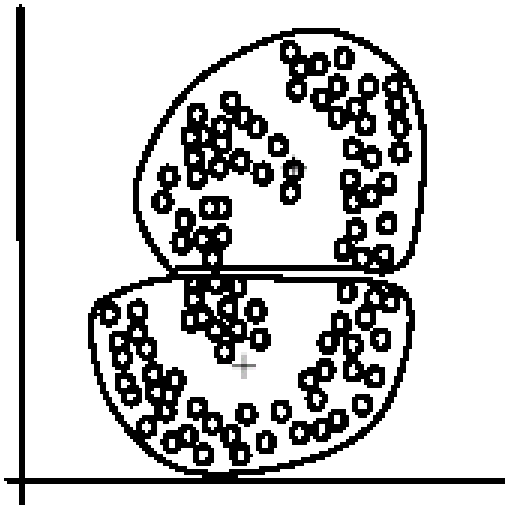
- Lựa chọn ngẫu nhiên hạt nhân thứ 1 ($\boldsymbol{m_1}$)
- Lựa chọn hạt nhân thứ 2 ($\boldsymbol{m_2}$) càng xa càng tốt so với hạt nhân thứ 1
- …
- Lựa chọn hạt nhân thứ *i* ($\boldsymbol{m_i}$) càng xa càng tốt so với hạt nhân gần nhất trong số $\{\boldsymbol{m_1}, \boldsymbol{m_2}, \ldots, \boldsymbol{m_{i-1}}\}$
- ...

k-Means – Các nhược điểm (3)

- Giải thuật k-means không phù hợp để phát hiện các cụm (nhóm) không có dạng hình elip hoặc hình cầu



(A): Two natural clusters



(B): k-means clusters

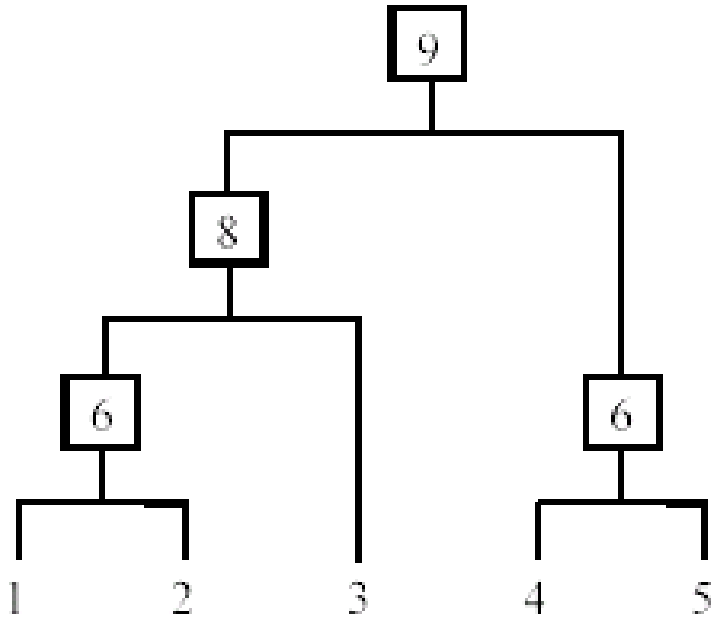
[Liu, 2006]

k-Means – Tổng kết

- Mặc dù có những nhược điểm như trên, k-means vẫn là giải thuật phổ biến nhất được dùng để giải quyết các bài toán phân cụm – do tính đơn giản và hiệu quả
  - Các giải thuật phân cụm khác cũng có các nhược điểm riêng
- Về tổng quát, không có lý thuyết nào chứng minh rằng một giải thuật phân cụm khác hiệu quả hơn k-means
  - Một số giải thuật phân cụm có thể phù hợp hơn một số giải thuật khác đối với một số kiểu tập dữ liệu nhất định, hoặc đối với một số bài toán ứng dụng nhất định
- So sánh hiệu năng của các giải thuật phân cụm là một nhiệm vụ khó khăn (thách thức)
  - Làm sao để biết được các cụm kết quả thu được là chính xác?

HAC (1)

- Sinh ra một chuỗi lồng nhau của các cụm, được gọi là **dendrogram**
  - Cũng được gọi là một phân loại (*taxonomy*)/phân cấp (*hierarchy*)/cây (*tree*) của các ví dụ

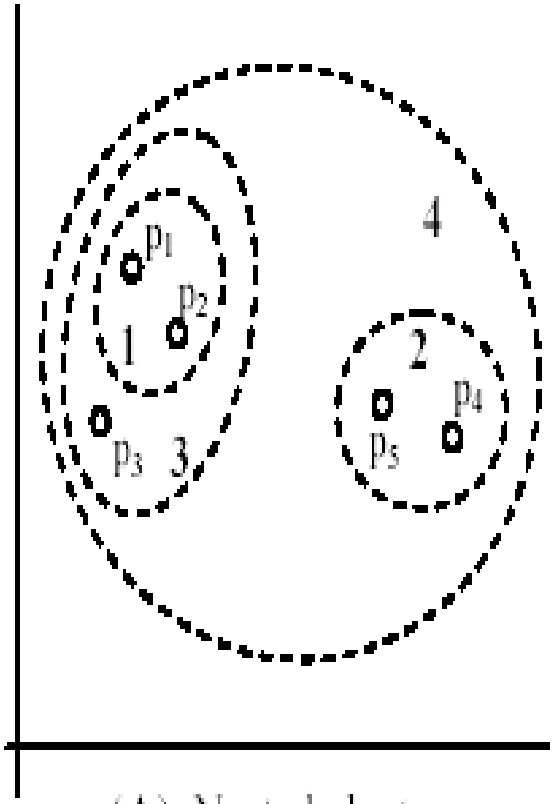


[Liu, 2006]

HAC (2)

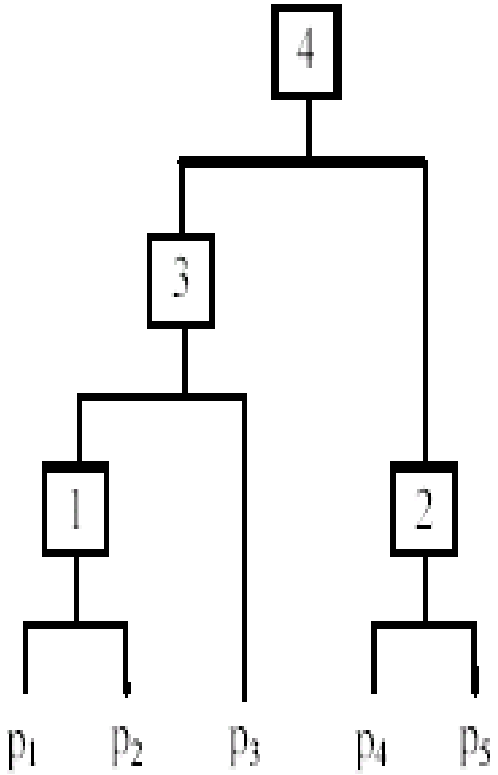
- Phân cụm dựa trên tích tụ phân cấp (Hierarchical Agglomerative Clustering – HAC) sẽ xây dựng dendrogram từ mức đáy (cuối) dần lên (bottom-up)
- Giải thuật HAC
  - Bắt đầu, mỗi ví dụ chính là một cụm (là một nút trong dendrogram)
  - Hợp nhất 2 cụm có mức độ tương tự (gần) nhau nhất
    - Cặp gồm 2 cụm có khoảng cách nhỏ nhất trong số các cặp cụm
  - Tiếp tục quá trình hợp nhất
  - Giải thuật kết thúc khi tất cả các ví dụ được hợp nhất thành một cụm duy nhất (là nút gốc trong dendrogram)

HAC – Ví dụ



(A). Nested clusters

(Venn diagram)



(B) Dendrogram

[Liu, 2006]

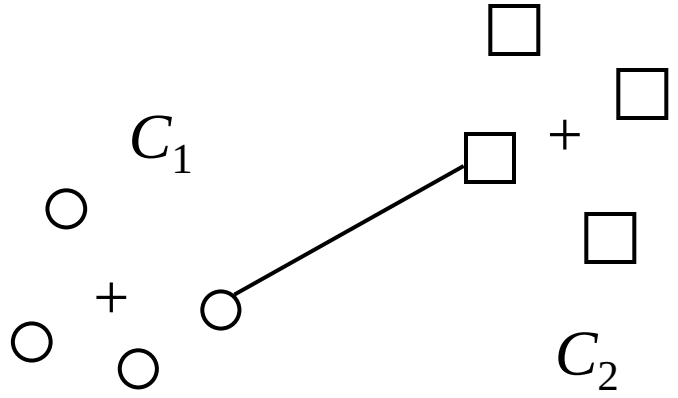
Khoảng cách giữa 2 cụm

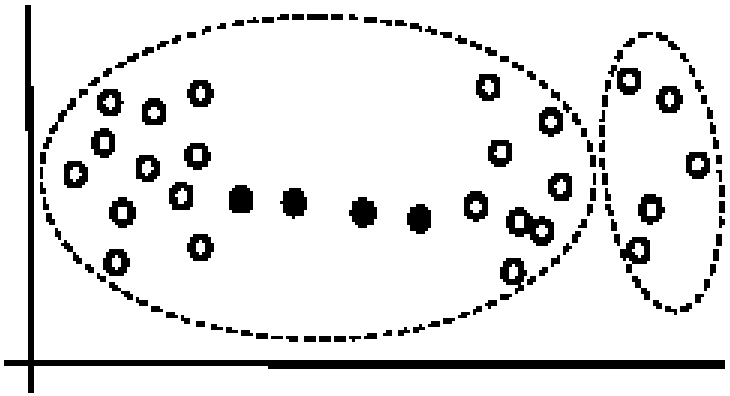
- Giải thuật HAC cần định nghĩa việc tính toán khoảng cách giữa 2 cụm
  - Trước khi hợp nhất, cần tính khoảng cách giữa mỗi cặp 2 cụm có thể
- Có nhiều phương pháp để đánh giá khoảng cách giữa 2 cụm – đưa đến các biến thể khác nhau của giải thuật HAC
  - Liên kết đơn (Single link)
  - Liên kết hoàn toàn (Complete link)
  - Liên kết trung bình (Average link)
  - Liên kết trung tâm (Centroid link)
  - …

HAC – Liên kết đơn

HAC liên kết đơn (Single link):

- Khoảng cách giữa 2 cụm là **khoảng cách nhỏ nhất** giữa các ví dụ (các thành viên) của 2 cụm đó
- Có xu hướng sinh ra các cụm có dạng "chuỗi dài" (long chain)



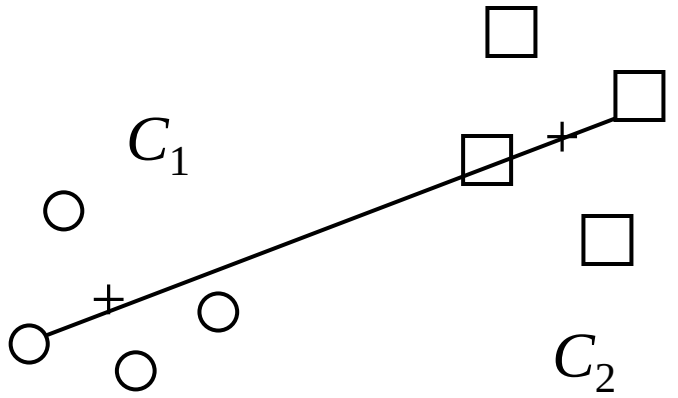


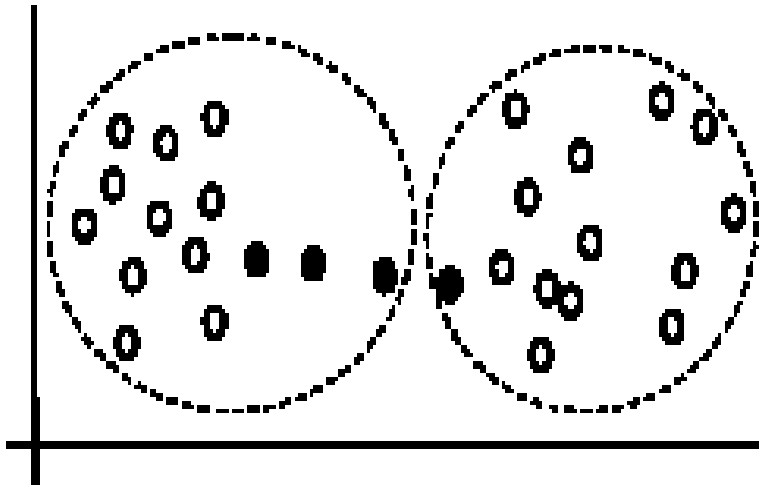
[Liu, 2006]

HAC – Liên kết hoàn toàn

HAC liên kết hoàn toàn (Complete link):

- Khoảng cách giữa 2 cụm là **khoảng cách lớn nhất** giữa các ví dụ (các thành viên) của 2 cụm đó
- Nhạy cảm (gặp lỗi phân cụm) đối với các ngoại lai (outliers)
- Có xu hướng sinh ra các cụm có dạng "bụi cây" (clumps)





[Liu, 2006]

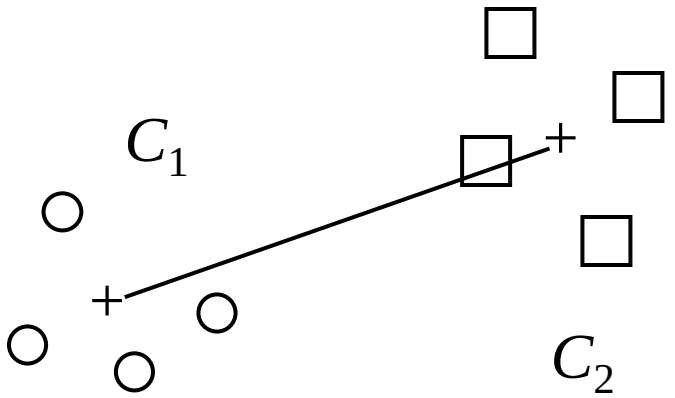
HAC – Liên kết trung bình

- Khoảng cách trong liên kết trung bình (Average-link) là sự thỏa hiệp giữa các khoảng cách trong liên kết hoàn toàn (Complete-link) và liên kết đơn (Single-link)
  - Để giảm mức độ nhạy cảm (khả năng lỗi) của phương pháp phân cụm dựa trên liên kết hoàn toàn đối với các ngoại lai (outliers)
  - Để giảm xu hướng sinh ra các cụm có dạng "chuỗi dài" của phương pháp phân cụm dựa trên liên kết đơn (dạng "chuỗi dài" không phù hợp với khái niệm tự nhiên của một cụm)
- Khoảng cách giữa 2 cụm là khoảng cách trung bình của tất cả các cặp ví dụ (mỗi ví dụ thuộc về một cụm)

HAC – Liên kết trung tâm

HAC liên kết trung tâm (Centroid link):

- Khoảng cách giữa 2 cụm là khoảng cách giữa 2 điểm trung tâm (centroids) của 2 cụm đó



Giải thuật HAC – Độ phức tạp

- Tất cả các biến thể của giải thuật HAC đều có độ phức tạp tối thiểu mức $O(r^2)$
  - r: Tổng số các ví dụ (kích thước của tập dữ liệu)
- Phương pháp phân cụm HAC liên kết đơn (Single-link) có độ phức tạp mức $O(r^2)$
- Các phương pháp phân cụm HAC liên kết hoàn toàn (Complete-link) và liên kết trung bình (Average-link) có độ phức tạp mức $O(r^2 \log r)$
- Do độ phức tạp cao, giải thuật HAC khó có thể áp dụng được đối với các tập dữ liệu có kích thước (rất) lớn

Các hàm khoảng cách

- Một thành phần quan trọng của các phương pháp phân cụm
  - Cần xác định các hàm tính độ khác biệt (dissimilarity/distance functions), hoặc các hàm tính độ tương tự (similarity functions)
- Các hàm tính khoảng cách khác nhau đối với
  - Các kiểu dữ liệu khác nhau
    - Dữ liệu kiểu số (Numeric data)
    - Dữ liệu kiểu định danh (Nominal data)
  - Các bài toán ứng dụng cụ thể

Hàm khoảng cách cho thuộc tính số

- Họ các hàm khoảng cách hình học (khoảng cách Minkowski)
- Các hàm được dùng phổ biến nhất
  - Khoảng cách Euclid
  - Khoảng cách Manhattan (khoảng cách City-block)
- Ký hiệu $d(\mathbf{x}_i, \mathbf{x}_j)$ là khoảng cách giữa 2 ví dụ (2 vectơ) $\mathbf{x}_i$ và $\mathbf{x}_j$
- Khoảng cách Minkowski (với p là một số nguyên dương)

$$d(\mathbf{x_i}, \mathbf{x_j}) = [(x_{i1} - x_{j1})^p + (x_{i2} - x_{j2})^p + ... + (x_{in} - x_{jn})^p]^{1/p}$$

Hàm k/c cho thuộc tính nhị phân

- Sử dụng một ma trận để biểu diễn hàm tính khoảng cách
  - *a*: Tổng số thuộc tính có giá trị là 1 trong cả $\mathbf{x_i}$ và $\mathbf{x_j}$
  - *b*: Tổng số các thuộc tính có giá trị là 1 trong $\mathbf{x_i}$ và có giá trị là 0 trong $\mathbf{x_j}$
  - *c*: Tổng số các thuộc tính có giá trị là 0 trong $\mathbf{x_i}$ và có giá trị là 1 trong $\mathbf{x_j}$
  - *d*: Tổng số các thuộc tính có giá trị là 0 trong cả $\mathbf{x_i}$ và $\mathbf{x_j}$

ví dụ $\mathbf{x_i}$ \ ví dụ $\mathbf{x_j}$	1	0
1	a	b
0	c	d

- **Hệ số phù hợp đơn giản (Simple matching coefficient)**. Tỷ lệ sai lệch giá trị của các thuộc tính giữa 2 ví dụ:

$$d(\mathbf{x_i}, \mathbf{x_j}) = \frac{b+c}{a+b+c+d}$$

Hàm k/c cho thuộc tính định danh

- Hàm khoảng cách cũng dựa trên phương pháp đánh giá tỷ lệ khác biệt giá trị thuộc tính giữa 2 ví dụ
- Với 2 ví dụ $\mathbf{x}_i$ và $\mathbf{x}_j$, ký hiệu p là tổng số các thuộc tính (trong tập dữ liệu), và q là số các thuộc tính mà giá trị là như nhau trong $\mathbf{x}_i$ và $\mathbf{x}_j$

$$d(\mathbf{x_i}, \mathbf{x_j}) = \frac{p - q}{p}$$

Tài liệu tham khảo

- B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Springer, 2006.