

# **MỘT SỐ THUẬT TOÁN XỬ LÝ DỮ LIỆU LỚN**

# Nội dung

---

- ❖ Thuật toán K-Means
- ❖ Xây dựng thuật toán MapReduce\_K-Means
- ❖ Thuật toán Naïve Bayes
- ❖ Xây dựng thuật toán MapReduce\_Bayes



# Thuật toán K-Means

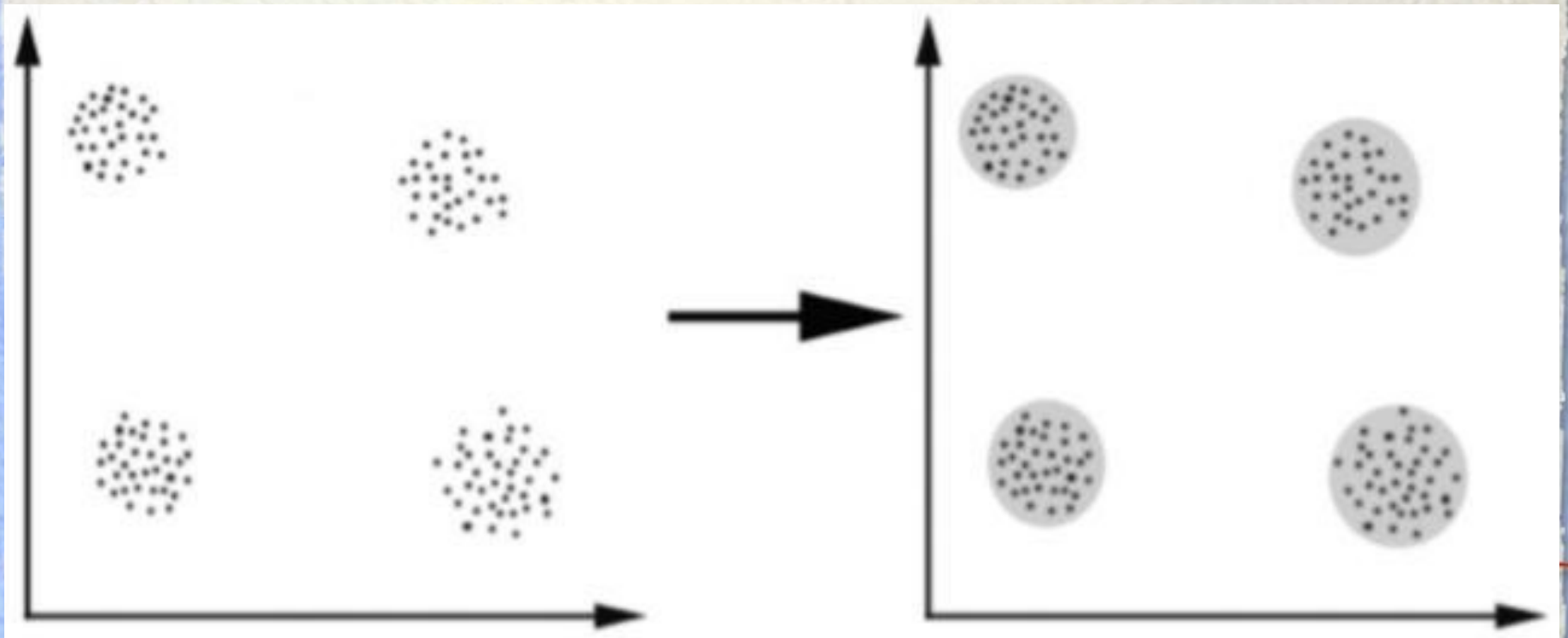
- ❖ Giới thiệu thuật toán K-Means
- ❖ Phát biểu bài toán phân cụm
- ❖ Các bước thuật toán K-Means
- ❖ Điều kiện dừng và chất lượng phân cụm
- ❖ Nhận xét thuật toán K-Means

# Giới thiệu thuật toán K-Means

- ❖ Là một trong các thuật toán phân cụm đơn giản và điển hình nhất
- ❖ Do MacQueen đề xuất trong lĩnh vực thống kê năm 1967
- ❖ Mục đích:
  - ❖ Sinh ra k cụm dữ liệu từ một tập dữ liệu ban đầu gồm n đối tượng trong không gian p chiều  $X_i = \{x_{i1}, x_{i2}, \dots, x_{ip}\}$ ,  $i = 1..n$ , sao cho hàm tiêu chuẩn E đạt giá trị tối thiểu
  - ❖ 
$$E = \sum_{i=1}^k \sum_{x \in C^i} d(x, m_i)^2$$
  - ❖  $m_i$  là vector trọng tâm của cụm  $C_i$ , giá trị của mỗi phần tử là trung bình cộng các thành phần tương ứng của các đối tượng vector dữ liệu trong cụm đang xét
  - ❖ d là khoảng cách Euclide giữa hai đối tượng

# Phát biểu bài toán phân cụm

- ❖ Input:  $n$  đối tượng và số các cụm  $k$
- ❖ Output: Các cụm  $C_i$  ( $i=1 \dots k$ ) sao cho hàm tiêu chuẩn  $E$  đạt giá trị tối thiểu

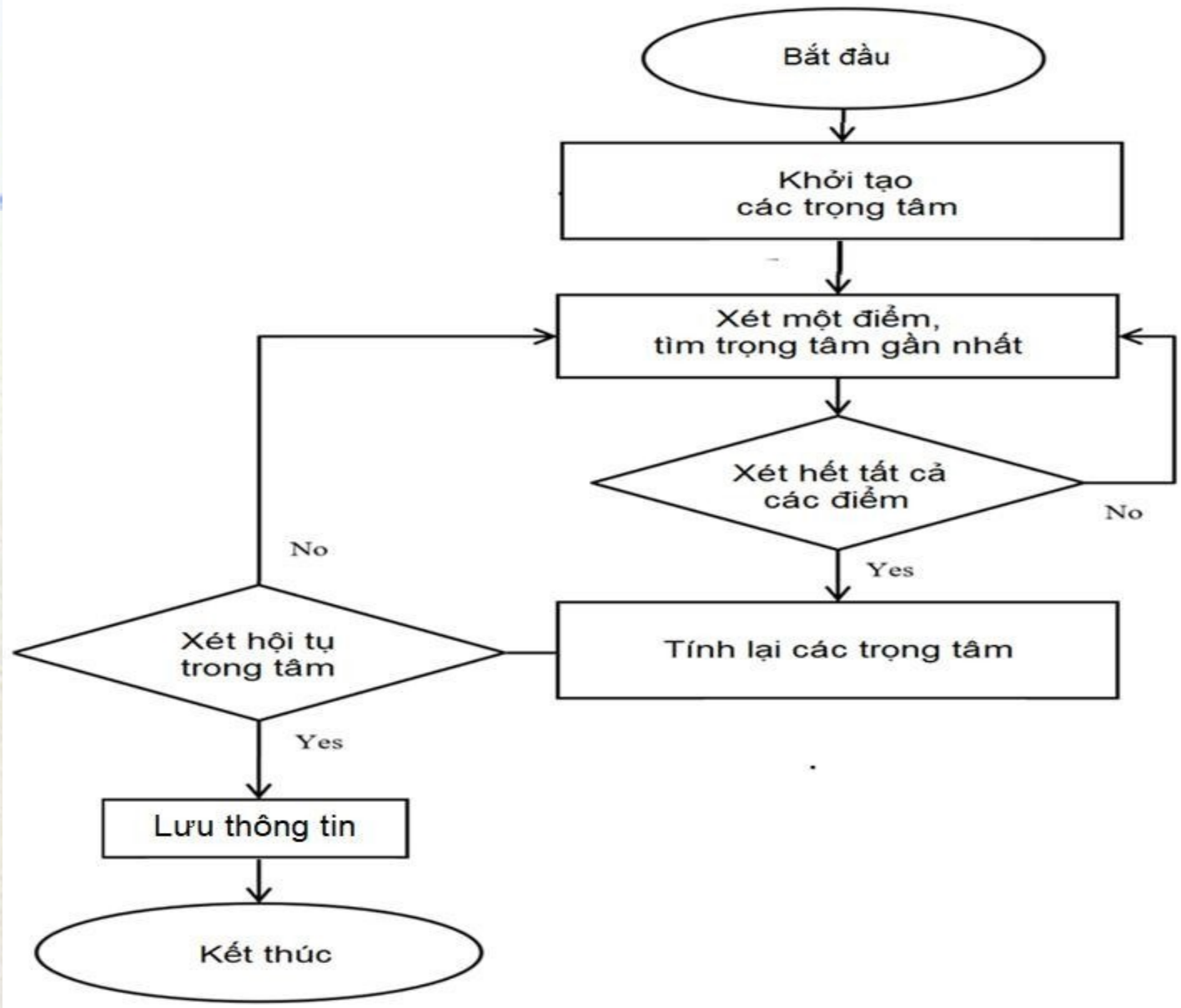




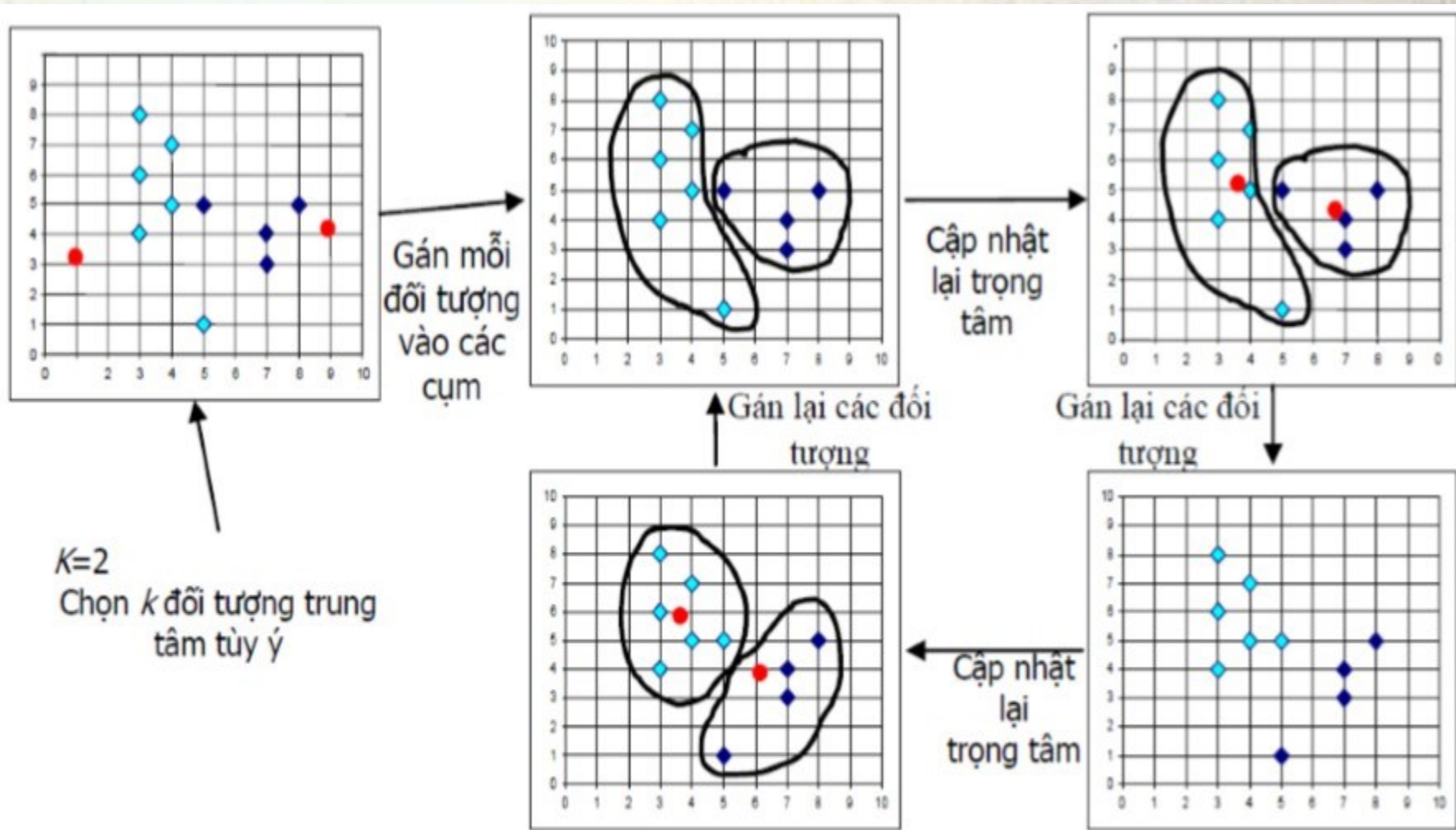
# Các bước thuật toán K-Means

- ❖ Bước 1: Khởi tạo tâm cụm
  - ❖ Chọn  $k$  đối tượng  $m_j$  ( $j=1\dots k$ ) là trọng tâm ban đầu của  $k$  cụm từ tập dữ liệu (Việc lựa chọn này có thể là ngẫu nhiên hoặc theo kinh nghiệm)
- ❖ Bước 2: Tính toán khoảng cách và gán cụm
  - ❖ Với mỗi đối tượng  $X_i$  ( $i=1\dots n$ ), tính toán khoảng cách từ nó tới mỗi trọng tâm  $m_j$  với  $j=1,\dots,k$ , sau đó tìm trọng tâm gần nhất đối với mỗi đối tượng
- ❖ Bước 3: Cập nhật lại trọng tâm
  - ❖ Với mỗi  $j=1,\dots,k$ , cập nhật trọng tâm cụm  $m_j$  bằng cách xác định trung bình cộng của các vector đối tượng dữ liệu
- ❖ Bước 4: Kiểm tra điều kiện dừng
  - ❖ Lặp các bước 2 và 3 cho đến khi các trọng tâm của cụm không thay đổi

# Lưu đồ thuật toán K-Means



# Minh họa quá trình phân cụm





# Điều kiện dừng và chất lượng phân cụm

## ❖ Điều kiện dừng

- ❖ Không có (hoặc có không đáng kể) việc gán lại các ví dụ vào các cụm khác
- ❖ Không có (hoặc có không đáng kể) thay đổi về các điểm trung tâm (centroids) của các cụm
- ❖ Không giảm hoặc giảm không đáng kể về tổng lỗi phân cụm  $E$

## ❖ Chất lượng phân cụm

- ❖ Phụ thuộc nhiều vào các tham số đầu vào như: **số cụm  $k$  và  $k$  trọng tâm khởi tạo ban đầu**
- ❖ Nếu các trọng tâm khởi tạo ban đầu quá lệch so với các trọng tâm cụm tự nhiên thì kết quả phân cụm của k-means là rất thấp => **các cụm dữ liệu được khám phá rất lệch so với các cụm trong thực tế**

# Nhận xét thuật toán K-Means

- ❖ Phần lớn khối lượng tính toán tập trung ở bước 2: tính khoảng cách từ mỗi điểm (đối tượng) tới các tâm cụm
- ❖ Số lượng đối tượng trong tập dữ liệu càng lớn, thời gian cần cho bước này càng nhiều
- ❖ Việc tính toán khoảng cách từ một điểm tới tâm cụm là độc lập, không phụ thuộc vào điểm khác
- ❖ => Việc tính khoảng cách từ các điểm đến các tâm cụm có thể thực hiện song song, đồng thời với nhau



# **Xây dựng thuật toán MapReduce\_K-Means**

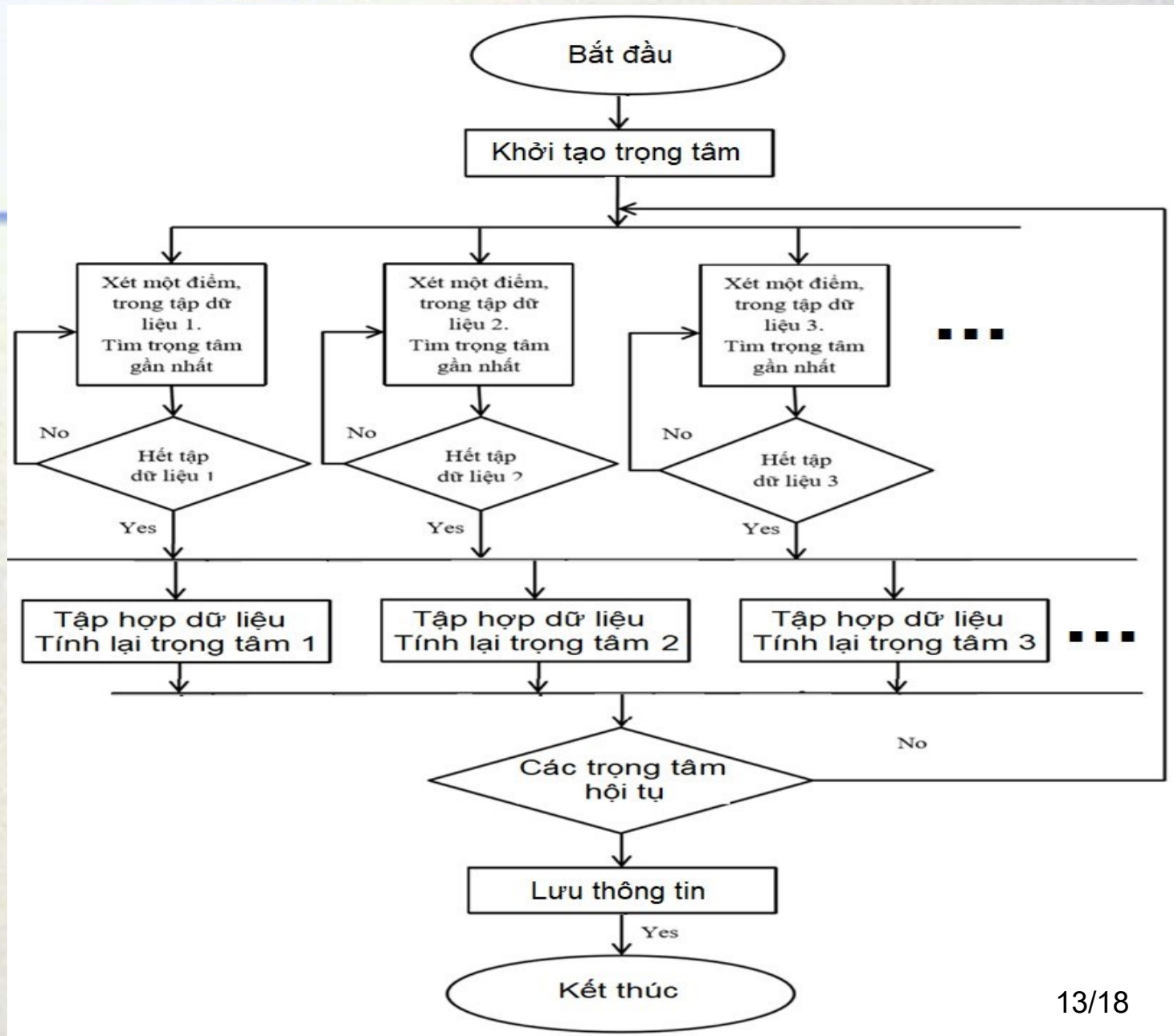
- ❖ Ý tưởng MapReduce hóa K-Means
- ❖ Lưu đồ MapReduce hóa thuật toán K-Means
- ❖ Giải pháp MapReduce hóa thuật toán K-Means
- ❖ Xây dựng hàm Map\_K-Means
- ❖ Xây dựng hàm Reduce\_K-Means
- ❖ Mã nguồn MapReduce\_K-Means



# Ý tưởng MapReduce hóa K-Means

- ❖ Tách dữ liệu thành các nhóm nhỏ
- ❖ Với mỗi vòng lặp
  - ❖ Map:
    - ❖ Phân cụm trên từng nhóm nhỏ dữ liệu
    - ❖ Với mỗi điểm dữ liệu, tìm trọng tâm gần nhất
  - ❖ Tất cả dữ liệu được gom theo từng tâm
  - ❖ Reduce:
    - ❖ Tính tâm mới của các dữ liệu được gom (theo từng tâm)

# Lưu đồ MapReduce thuật toán K-Means



# Thuật toán Naïve Bayes

- ❖ Giới thiệu thuật toán Naïve Bayes
- ❖ Định lý Bayes
- ❖ Thuật toán phân lớp Bayes
- ❖ Lưu đồ thuật toán phân lớp Bayes
- ❖ Ví dụ minh họa
- ❖ Thực thi thuật toán với ví dụ
- ❖ Phân tích thuật toán Bayes



# Giới thiệu thuật toán Naïve Bayes

- ❖ Là phương pháp phân loại dựa vào xác suất
- ❖ Sử dụng rộng rãi trong lĩnh vực học máy, phổ biến trong nhiều lĩnh vực như các công cụ tìm kiếm, các bộ lọc mail nói riêng và phân loại văn bản nói chung
- ❖ Ý tưởng cơ bản của cách tiếp cận Naïve Bayes là sử dụng xác suất có điều kiện giữa từ và chủ đề để dự đoán xác suất chủ đề của một văn bản cần phân loại
- ❖ Điểm quan trọng của phương pháp này chính:
  - ❖ Giả định rằng sự xuất hiện của tất cả các từ trong văn bản đều độc lập với nhau
  - ❖ Không khai thác sự phụ thuộc của nhiều từ vào trong một chủ đề cụ thể
- ❖ Được xem là thuật toán đơn giản nhất trong các phương pháp

# Định lý Bayes

- ❖ Định lý Bayes được phát biểu như sau:

$$❖ P(Y|X) = \frac{P(X|Y).P(Y)}{P(X)}$$

- ❖  $P(Y)$ : Xác suất của sự kiện Y xảy ra
- ❖  $P(X)$ : Xác suất của sự kiện X xảy ra
- ❖  $P(X|Y)$ : Xác suất (có điều kiện) của sự kiện X xảy ra, nếu biết rằng sự kiện Y đã xảy ra
- ❖  $P(Y|X)$ : Xác suất (có điều kiện) của sự kiện Y xảy ra, nếu biết rằng sự kiện X đã xảy ra



# Thuật toán phân lớp Bayes

## ❖ Dữ kiện cần có:

- ❖ D: tập dữ liệu huấn luyện, được vector hoá dưới dạng  $x \rightarrow = (x_1, x_2, \dots, x_n)$
- ❖  $C_i$ : tập các tài liệu của D thuộc lớp  $C_i$  với  $i=\{1,2,3,\dots\}$
- ❖ Các thuộc tính  $x_1, x_2, \dots, x_n$  độc lập xác suất đôi một với nhau

## ❖ Thuật toán Naïve Bayes cơ bản:

- ❖ Bước 1 : Huấn luyện Naïve Bayes (dựa vào tập dữ liệu)

- ❖ Tính xác suất  $P(C_i)$
- ❖ Tính xác suất  $P(x_k|C_i)$

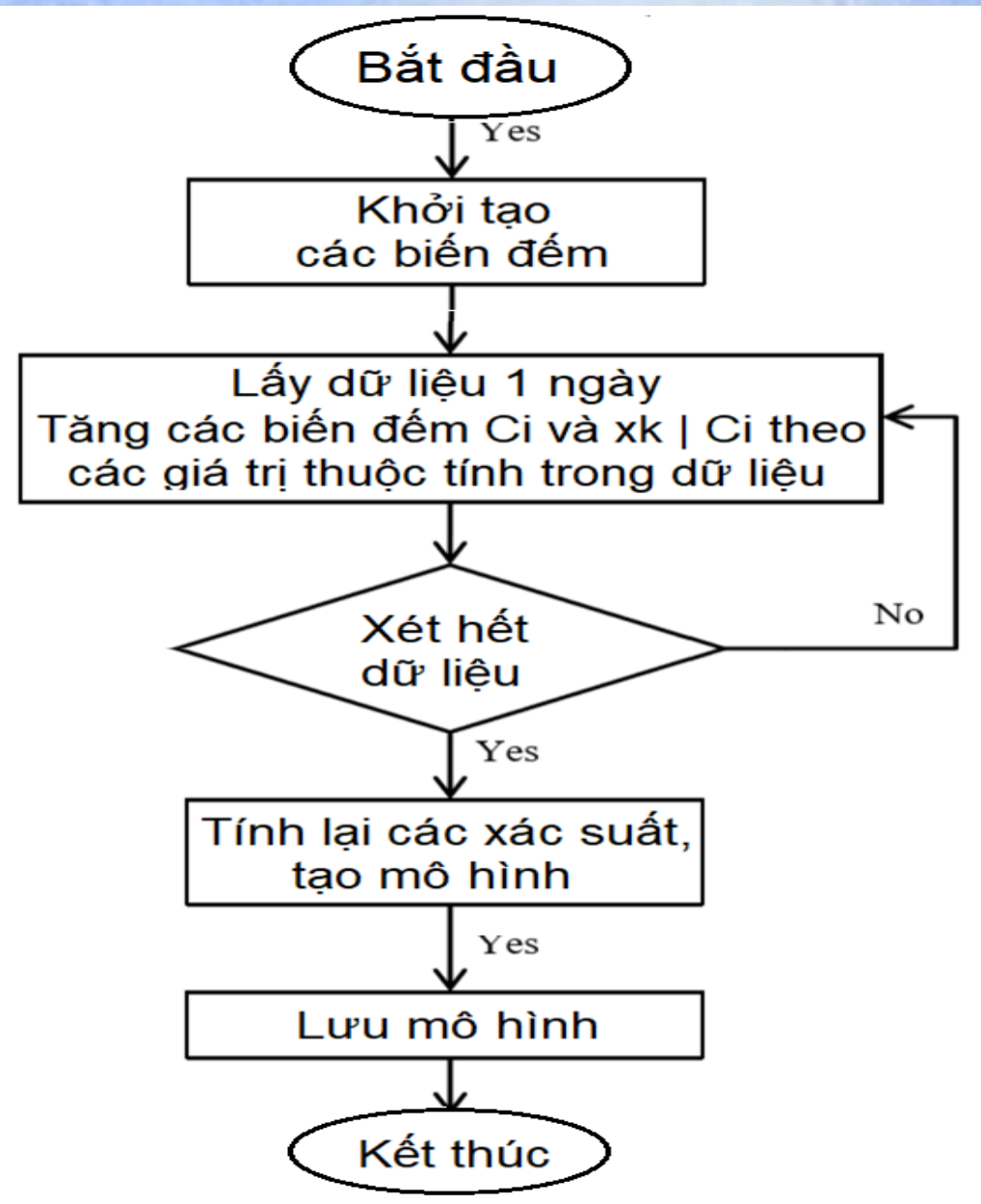
- ❖ Bước 2: Phân lớp  $X_{new}$

- ❖ Tính  $F(X_{new}, C_i) = P(C_i) \prod_{k=1}^n P(x_k|C_i)$
- ❖  $X_{new}$  được gán vào lớp  $C_q$  sao cho

$$❖ F(X_{new}, C_q) = \max(F(X_{new}, C_i))$$



# Lưu đồ thuật toán phân lớp Bayes



# Ví dụ minh họa thuật toán Bayes

- ❖ Yêu cầu: Dự đoán quyết định của người chơi có đi chơi Tennis hay không với các điều kiện về thời tiết đã được biết trước
- ❖ Bảng dữ liệu huấn luyện như sau:

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No

# Ví dụ minh họa

❖ Bảng dữ liệu huấn luyện như sau (tiếp):

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



# Thực thi thuật toán Bayes với ví dụ

- ❖ Có 2 lớp dự báo:
  - ❖  $C1 = \text{"yes"} \Rightarrow$  Có đi chơi Tennis
  - ❖  $C2 = \text{"no"} \Rightarrow$  Không đi chơi Tennis
- ❖ B1: Huấn luyện Bayes
- ❖ B2: Phân lớp Bayes

# B1: Huấn luyện Bayes

- ❖ Tính các xác suất  $P(C_i)$ 
  - ❖  $P(C1) = P(\text{"yes"}) = 9/14$
  - ❖  $P(C2) = P(\text{"no"}) = 5/14$
- ❖ Tính các xác suất  $P(x_k|C_i)$ 
  - ❖ Với thuộc tính Outlook
  - ❖ Với thuộc tính Temp
  - ❖ Với thuộc tính Humidity
  - ❖ Với thuộc tính Wind



# Với thuộc tính Outlook

- ❖ Có các giá trị: sunny, overcast, rain
- ❖  $P(\text{sunny} \mid \text{yes}) = 2/9$
- ❖  $P(\text{sunny} \mid \text{no}) = 3/5$
- ❖  $P(\text{overcast} \mid \text{yes}) = 4/9$
- ❖  $P(\text{overcast} \mid \text{no}) = 0/5$
- ❖  $P(\text{rain} \mid \text{yes}) = 3/9$
- ❖  $P(\text{rain} \mid \text{no}) = 2/5$

# Với thuộc tính Temp

- ❖ Có các giá trị: Hot, Cold, Mild
- ❖  $P(\text{hot} \mid \text{yes}) = 2/9$
- ❖  $P(\text{hot} \mid \text{no}) = 2/5$
- ❖  $P(\text{cold} \mid \text{yes}) = 3/9$
- ❖  $P(\text{cold} \mid \text{no}) = 1/5$
- ❖  $P(\text{mild} \mid \text{yes}) = 4/9$
- ❖  $P(\text{mild} \mid \text{no}) = 2/5$



# Với thuộc tính Humidity

- ❖ Có các giá trị: Normal, High
- ❖  $P(\text{normal} \mid \text{yes}) = 6/9$
- ❖  $P(\text{normal} \mid \text{no}) = 1/5$
- ❖  $P(\text{high} \mid \text{yes}) = 3/9$
- ❖  $P(\text{high} \mid \text{no}) = 4/5$

# Với thuộc tính Wind

- ❖ Có các giá trị: Weak, Strong
- ❖  $P(\text{weak} \mid \text{yes}) = 6/9$
- ❖  $P(\text{weak} \mid \text{no}) = 2/5$
- ❖  $P(\text{strong} \mid \text{yes}) = 3/9$
- ❖  $P(\text{strong} \mid \text{no}) = 3/5$



## B2: Phân lớp Bayes

- ❖  $X_{\text{new}} = \{\text{sunny, cool, high, strong}\}$
- ❖ Tính các xác suất
  - ❖  $F(X_{\text{new}} | \text{yes}) = P(\text{yes}) * P(\text{sunny} | \text{yes}) * P(\text{cool} | \text{yes}) * P(\text{high} | \text{yes}) * P(\text{strong} | \text{yes}) = 9/14 * 2/9 * 3/9 * 3/9 * 3/9 = 0.0053$
  - ❖  $F(X_{\text{new}} | \text{no}) = P(\text{no}) * P(\text{sunny} | \text{no}) * P(\text{cool} | \text{no}) * P(\text{high} | \text{no}) * P(\text{strong} | \text{no}) = 5/14 * 3/5 * 1/5 * 4/5 * 3/5 = 0.0206$
- ❖ Kết luận:  $X_{\text{new}}$  thuộc vào lớp No

# Phân tích thuật toán Bayes

- ❖ Tương tự như thuật toán k-means, Naïve bayes cũng có vấn đề khi xử lý lượng dữ liệu lớn
- ❖ Trong quá trình học (training), nếu số lượng dữ liệu quá lớn, dẫn đến các vấn đề về thiếu bộ nhớ, tốc độ xử lý
- ❖ Với lượng dữ liệu lớn (khoảng vài triệu bản ghi) thì hầu hết thời gian của Naïve bayes là đếm số lần xuất hiện của các biến => tính các xác suất cần thiết để xây dựng mô hình
- ❖ Tiêu thụ thời gian chủ yếu do tính: các  $P(C_i)$  và  $P(x_k|C_i)$
- ❖ Để tính các xác suất này, ta cần đếm số lần xuất hiện của các  $C_i$  và các  $x_k|C_i$
- ❖ Việc tính các xác suất, đếm các số lần xuất hiện của từng biến là độc lập => chia dữ liệu thành nhiều phần nhỏ và thực hiện song song



# Thuật toán MapReduce\_Bayes

- ❖ Ý tưởng MapReduce thuật toán Bayes
- ❖ Lưu đồ thuật toán MapReduce\_Bayes
- ❖ Giải pháp MapReduce hoá thuật toán Bayes
- ❖ Xây dựng hàm Map\_Bayes
- ❖ Xây dựng hàm Reduce\_Bayes

# Ý tưởng MapReduce hoá thuật toán Bayes

## ❖ Nhiệm vụ:

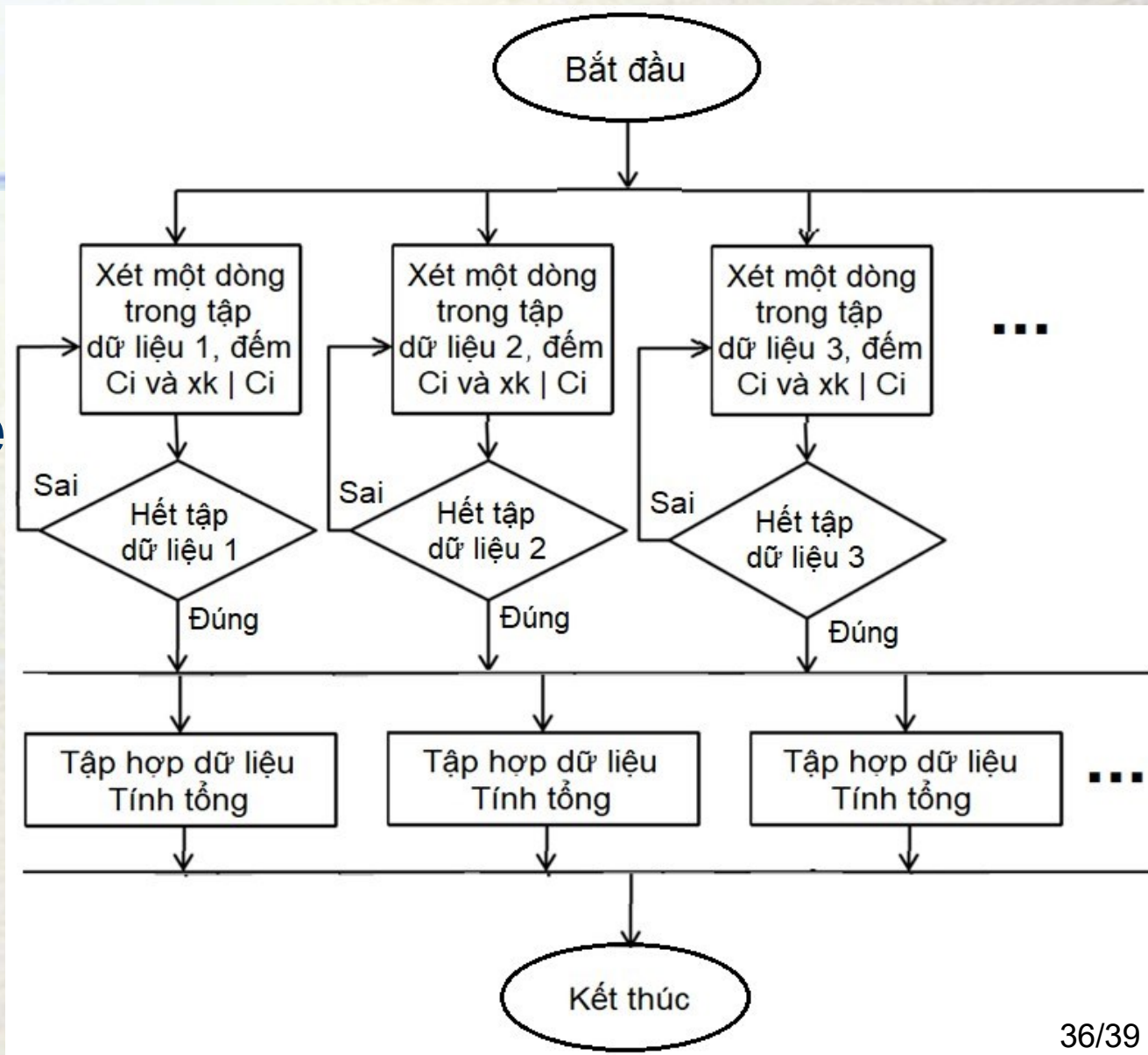
- ❖ MapReduce hóa việc đếm số lần xuất hiện của các  $C_i$  và các  $x_k|C_i$

## ❖ Ý tưởng:

- ❖ Chia dữ liệu thành nhiều phần nhỏ
- ❖ Đếm các số lần xuất hiện của từng biến  $C_i$  và  $x_k|C_i$  trong hàm Map
- ❖ Tập hợp kết quả và tính tổng theo từng biến trong hàm Reduce
- ❖ Lưu thông tin số lần xuất hiện của từng biến  $C_i$  và  $x_k|C_i$
- ❖ Giai đoạn phân lớp: tính các xác suất  $P(C_i)$  và  $P(x_k|C_i)$  dựa trên dữ liệu về số lần xuất hiện của từng biến  $C_i$  và  $x_k|C_i$  để tính các  $F(X_{new}, C_i)$



# Lưu đồ thuật toán MapReduce \_Bayes



# MapReduce hoá thuật toán Bayes

- ❖ Dữ liệu đầu vào:
  - ❖ Là danh sách các hàng (có thể lưu trên file txt)
  - ❖ Mỗi hàng là dữ liệu huấn luyện mô tả từng ngày: **Sunny**  
**Hot High Weak No**
  - ❖ Được chuyển sang kiểu **key/value** làm đầu vào cho thuật toán
- ❖ Mô hình cơ bản của MapReduce:
  - ❖ `map (keyIn, valIn) -> list (keyInt, valInt)`
  - ❖ `reduce (keyInt, list (valInt)) -> list (keyOut, valOut)`
- ❖ Áp dụng cho thuật toán Bayes:
  - ❖ Xây dựng hàm **Map\_Bayes**
  - ❖ Xây dựng hàm **Reduce\_Bayes**



# Xây dựng hàm Map\_Bayes

## ❖ Đầu vào:

- ❖ cặp **key/value** biểu diễn dữ liệu một ngày
- ❖ **keyIn** là giá trị byte offset của dòng
- ❖ **valIn** là text biểu dữ liệu một ngày (**Sunny Hot High Weak No**)

## ❖ Xử lý:

- ❖ Đếm 1 cho xuất hiện của  $C_i$
- ❖ Đếm 1 cho xuất hiện của  $x_k|C_i$

## ❖ Đầu ra:

- ❖ cặp **key/value** trung gian
- ❖ **keyInt** là  $C_i$  hoặc  $x_k|C_i$
- ❖ **valInt** là giá trị 1

# Xây dựng hàm Reduce\_Bayes

- ❖ Trước khi hàm reduce thực hiện
  - ❖ Kết quả của hàm map được trộn lại
  - ❖ Các **valInt** cùng **keyInt** được gom thành một nhóm
- ❖ Đầu vào:
  - ❖ **keyInt** được chuyển từ hàm map ( $C_i$  hoặc  $x_k|C_i$ )
  - ❖ **list(valInt)** là list các giá trị 1
- ❖ Xử lý:
  - ❖ Tính tổng các giá trị 1 trong **list(valInt)**
- ❖ Đầu ra:
  - ❖ **keyOut** là **keyInt** ( $C_i$  hoặc  $x_k|C_i$ )
  - ❖ **valOut** là tổng các giá trị 1 trong **list(valInt)**



# **ỨNG DỤNG PHÂN TÍCH DỮ LIỆU LỚN**

# Nội dung

- ❖ Phân lớp văn bản với Naïve Bayes
- ❖ Phân cụm ảnh với MapReduce\_K-Means
- ❖ Tra cứu thông tin từ internet



# Phân lớp văn bản với Naïve Bayes

- ❖ Giới thiệu về phân lớp văn bản
- ❖ Các bước thực hiện:
  - ❖ Huấn luyện mô hình phân lớp văn bản
  - ❖ Phân lớp văn bản
- ❖ Ví dụ minh họa phân lớp văn bản
- ❖ Thực thi phân văn bản với ví dụ

# Giới thiệu về phân lớp văn bản

- ❖ Dùng để dự đoán xem văn bản thuộc chủ đề nào, dựa trên nội dung văn bản
- ❖ Để áp dụng thuật toán Naïve Bayes vào phân loại văn bản, cần thực hiện các bước tiền xử lý và vector hoá các văn bản (trong tập huấn luyện hay văn bản dùng để phân lớp)
- ❖ Vector hoá các văn bản: đếm tần suất từ trong văn bản
- ❖ Nếu số chiều quá lớn, có thể thực hiện rút chọn các đặc trưng
- ❖ Vector đặc trưng được dùng để huấn luyện



# Huấn luyện mô hình phân lớp văn bản

- ❖ Từ tập huấn luyện, ta rút trích:
  - ❖ Tập từ vựng (các đặc trưng)  $\text{vocSet}$
  - ❖ Vector đặc trưng của từng văn bản
- ❖ Tính xác suất  $P(C_i)$  và  $P(x_k|C_i)$ 
  - ❖  $P(C_i) = \frac{\text{totalDoc}C_i}{\text{totalDocAll}}$
  - ❖  $P(x_k|C_i) = \frac{|x_k|}{|\text{Text}C_i|}$  hoặc  $P(x_k|C_i) = \frac{|x_k|+1}{n_{C_i}+|\text{Text}C_i|}$
  - ❖  $\text{totalDoc}C_i$ : số tài liệu của tập huấn luyện thuộc lớp  $C_i$
  - ❖  $\text{totalDocAll}$ : số tài liệu có trong tập huấn luyện
  - ❖  $n_{C_i}$ : tổng số từ đôi một khác nhau của lớp  $C_i$
  - ❖  $|x_k|$ : tổng số từ  $x_k$  trong lớp  $C_i$
  - ❖  $|\text{Text}C_i|$ : tổng số từ vựng (không phân biệt đôi một) trong lớp  $C_i$ ,  $|\text{Text}C_i| = \sum_{x_k \in \text{vocSet}} |x_k|$

# Phân lớp văn bản

- ❖ Cho văn bản  $doc^{new}$
- ❖ Yêu cầu:
  - ❖ Dự báo văn bản  $doc^{new}$  thuộc lớp nào ?
- ❖ Các bước thực hiện:
  - ❖ Tính:
    - ❖  $F(doc^{new}, C_i) = P(C_i) * \prod_{x_k \in \text{VocSet}} (P(x_k|C_i) * |x_k|^{new})$
    - ❖ Ta có:  $p(doc^{new}) = \max(F(doc^{new}, C_i)), i=1, \dots, n_C$



# Ví dụ minh họa phân lớp văn bản

- ❖ Có tập tài liệu để huấn luyện sau khi đã vector hoá (sử dụng phương pháp đơn giản đếm số lần xuất hiện) và rút trích đặc trưng như sau:
  - ❖ Bộ từ vựng (đặc trưng) : var, bit, chip, log

<b>Docs</b>	<b>Var</b>	<b>Bit</b>	<b>Chip</b>	<b>Log</b>	<b>Class</b>
Doc1	42	25	7	56	Math
Doc2	10	28	45	2	Comp
Doc3	11	25	22	4	Comp
Doc4	33	40	8	48	Math
Doc5	28	32	9	60	Math
Doc6	8	22	30	1	Comp

# Thực thi phân văn bản với ví dụ

- ❖ Có 2 lớp chủ đề:
  - ❖ C1 = “Comp”
  - ❖ C2 = “Math”
- ❖ B1: Huấn luyện chủ đề
- ❖ B2: Phân lớp chủ đề



# Huấn luyện chủ đề

- ❖ Tính xác suất các lớp  $C_i$ :
  - ❖  $P(C_1 = \text{"Comp"}) = 3/6 = 0.5$
  - ❖  $P(C_2 = \text{"Math"}) = 3/6 = 0.5$
- ❖ B1: Huấn luyện chủ đề
  - ❖ Các xác suất  $P(x_k|C_1)$
  - ❖ Các xác suất  $P(x_k|C_2)$

# Các xác suất $P(x_k|C_1)$

- ❖ Tổng số từ lớp  $C_1 = \text{"Comp"}$ :
  - ❖  $|\text{Text}C_1| = (10 + 11 + 8) + (28 + 25 + 22) + (45 + 22 + 30) + (2 + 4 + 1) = 208$
- ❖  $P(\text{var}|\text{Comp}) = (10 + 11 + 8) / 208 = 29/208$
- ❖  $P(\text{bit}|\text{Comp}) = (28 + 25 + 22) / 208 = 75/208$
- ❖  $P(\text{chip}|\text{Comp}) = (45 + 22 + 30) / 208 = 97/208$
- ❖  $P(\text{log}|\text{Comp}) = (2 + 4 + 1) / 208 = 7/208$



# Các xác suất $P(x_k|C_2)$

- ❖ Tổng số từ lớp  $C_2 = \text{"Math"}:$ 
  - ❖  $|\text{Text}C_2| = (42 + 33 + 28) + (25 + 40 + 32) + (7 + 8 + 9) + (56 + 48 + 60) = 388$
- ❖  $P(\text{var}|\text{Math}) = (42 + 33 + 28) / 388 = 103/388$
- ❖  $P(\text{bit}|\text{Math}) = (25 + 40 + 32) / 388 = 97/388$
- ❖  $P(\text{chip}|\text{Math}) = (7 + 8 + 9) / 388 = 24/388$
- ❖  $P(\text{log}|\text{Math}) = (56 + 48 + 60) / 388 = 164/388$

# Phân lớp chủ đề

- ❖ Cho văn bản có vector đặc trưng  $\text{Doc}^{\text{new}} = (23, 40, 15, 50)$
- ❖  $F(\text{doc}^{\text{new}}, C_1) = P(\text{Math}) * [P(\text{var}|\text{Math}) * 23 * P(\text{bit}|\text{Math}) * 40 * P(\text{chip}|\text{Math}) * 15 * P(\text{log}|\text{Math}) * 50] = 0.5 * [103/388 * 23 * 97/388 * 40 * 24/388 * 15 * 164/388 * 50] = 598.627$
- ❖  $F(\text{doc}^{\text{new}}, C_2) = P(\text{Comp}) * [P(\text{var}|\text{Comp}) * 23 * P(\text{bit}|\text{Comp}) * 40 * P(\text{chip}|\text{Comp}) * 15 * P(\text{log}|\text{Comp}) * 50] = 0.5 * [29/208 * 23 * 75/208 * 40 * 97/208 * 15 * 7/208 * 50] = 272.204$
- ❖ Kết luận: Văn bản  $\text{Doc}^{\text{new}}$  thuộc về lớp Math do  $p(\text{doc}^{\text{new}}) = \max(F(\text{doc}^{\text{new}}, C_i)) = 598,627$



# MapReduce hoá thuật toán Bayes

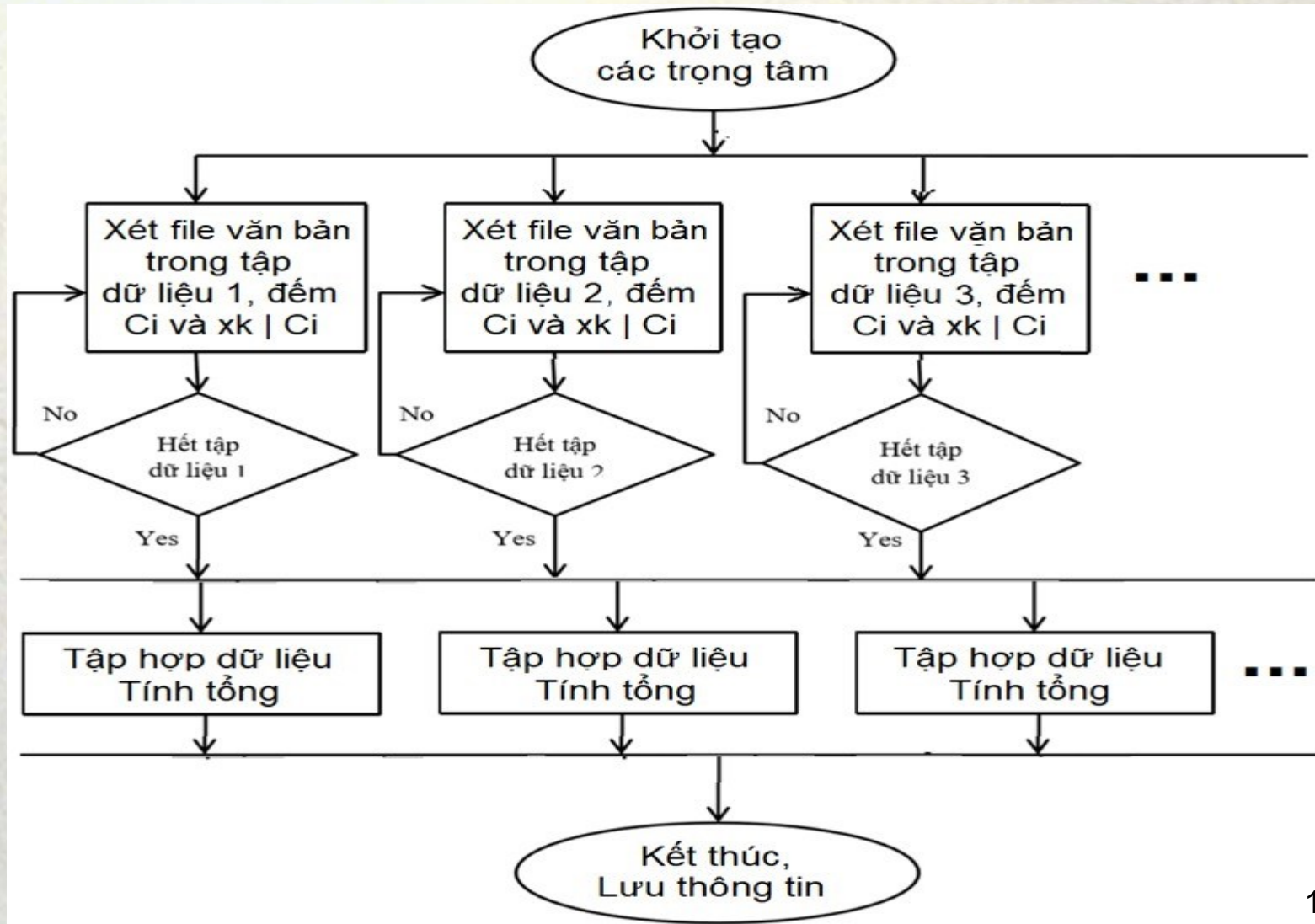
## ❖ Nhiệm vụ:

- ❖ MapReduce hóa việc đếm số lần xuất hiện của các  $C_i$  và các  $x_k|C_i$

## ❖ Ý tưởng:

- ❖ Chia dữ liệu thành nhiều phần nhỏ
- ❖ Đếm các số lần xuất hiện của từng biến  $C_i$  và  $x_k|C_i$  trong hàm Map
- ❖ Tập hợp kết quả và tính tổng theo từng biến trong hàm Reduce
- ❖ Lưu thông tin số lần xuất hiện của từng biến  $C_i$  và  $x_k|C_i$
- ❖ Giai đoạn phân lớp: tính các xác suất  $P(C_i)$  và  $P(x_k|C_i)$  dựa trên dữ liệu về số lần xuất hiện của từng biến  $C_i$  và  $x_k|C_i$  để tính các  $F(X_{new}, C_i)$

## Lưu đồ thuật toán MapReduce\_Bayes





# MapReduce hoá thuật toán Bayes

- ❖ Dữ liệu đầu vào:
  - ❖ Là danh sách các file văn bản (có thể lưu trên file txt)
  - ❖ Mỗi hàng là dữ liệu huấn luyện mô tả từng file văn bản, gồm tên lớp và tên file:
    - ❖ **number** D:\\Hadoop\\test\\input\\file1.txt
  - ❖ Được chuyển sang kiểu **key/value** làm đầu vào cho thuật toán
- ❖ Mô hình cơ bản của MapReduce:
  - ❖ **map** (**keyIn**, **valIn**) -> **list** (**keyInt**, **valInt**)
  - ❖ **reduce** (**keyInt**, **list** (**valInt**)) -> (**keyOut**, **valOut**)
- ❖ Áp dụng cho thuật toán Bayes:
  - ❖ Xây dựng hàm **Map\_TextBayes**
  - ❖ Xây dựng hàm **Reduce\_TextBayes**

# Xây dựng hàm Map\_TextBayes

## ❖ Đầu vào:

- ❖ cặp **key/value** biểu diễn dữ liệu mô tả file văn bản
- ❖ **keyIn** là giá trị byte offset của dòng
- ❖ **valIn** là text biểu dữ liệu một file văn bản (**number D:\\Hadoop\\test\\input\\file1.txt**)

## ❖ Xử lý: Tính ValInt

- ❖ Đếm **1** cho xuất hiện của  $C_i$
- ❖ Đếm **1** cho xuất hiện của  $x_k|C_i$

## ❖ Đầu ra:

- ❖ cặp **key/value** trung gian
- ❖ **keyInt** là  $C_i$  hoặc  $x_k|C_i$
- ❖ **valInt** là giá trị **1**



# Xây dựng hàm Reduce\_TextBayes

- ❖ Trước khi hàm reduce thực hiện
  - ❖ Kết quả của hàm map được trộn lại
  - ❖ Các cặp cùng **keyInt** được gom thành một nhóm
- ❖ Đầu vào:
  - ❖ **keyInt** được chuyển từ hàm map
  - ❖ **list(valInt)** là list các giá trị 1
- ❖ Xử lý:
  - ❖ Tính tổng các giá trị 1 trong **list(valInt)**
- ❖ Đầu ra:
  - ❖ **keyOut** là **keyInt** ( $C_i$  hoặc  $x_k|C_i$ )
  - ❖ **valOut** là tổng các giá trị 1 trong **list(valInt)**

# Phân cụm ảnh với MapReduce\_K-Means

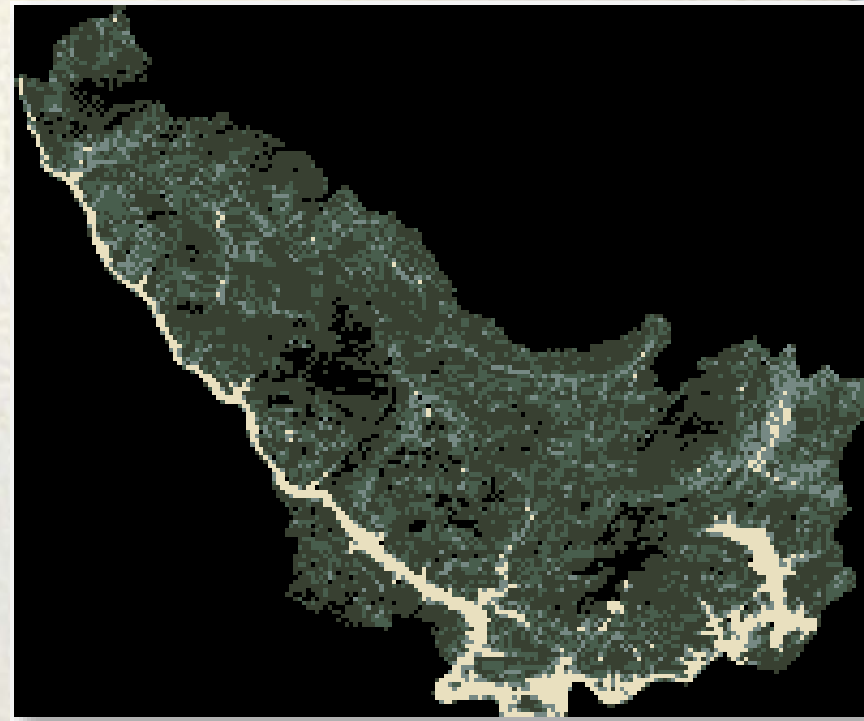
---

- ❖ Phát biểu bài toán phân cụm ảnh
- ❖ Giải pháp phân cụm ảnh với MapReduce\_K-Means



# Phát biểu bài toán phân cụm ảnh

- ❖ Input:  $n$  điểm ảnh và số các cụm  $k$
- ❖ Output: Các cụm  $C_i$  ( $i=1 \dots k$ ) (các cụm điểm ảnh) sao cho hàm tiêu chuẩn  $E$  đạt giá trị tối thiểu



# Phân cụm ảnh với MapReduce\_K-Means

- ❖ B1: Chuyển đổi dữ liệu
  - ❖ Chuyển đổi dữ liệu điểm ảnh thành list các hàng
  - ❖ Mỗi hàng là list giá trị là các thành phần của vector biểu diễn cho một điểm ảnh
- ❖ B2: Thực hiện phân cụm với MapReduce\_K-Means
- ❖ B3: Chuyển đổi kết quả phân cụm của MapReduce\_K-Means cho dữ liệu ảnh gốc



# Tra cứu thông tin từ internet

## ❖ Ví dụ:

- ❖ Tra cứu thông tin khách sạn, tham khảo:

<http://www.trivago.vn>

- ❖ Tra cứu thông tin sản phẩm điện máy

## ❖ Các bước xây dựng ứng dụng:

- ❖ B1: Thu thập dữ liệu từ internet
- ❖ B2: Lưu vào CSDL NoSQL
- ❖ B3: Xây dựng ứng dụng tra cứu thông tin truy xuất dữ liệu từ CSDL NoSQL