KHAI PHÁ DỮ LIỆU – CHƯƠNG 5
PHÂN CỤM DỮ LIỆU (DATA CLUSTERING)

Tổng hợp kiến thức, câu hỏi tự luận và trắc nghiệm

1. TỔNG QUAN VỀ PHÂN CỤM

1.1. Khái niệm phân cụm

Phân cụm (Clustering) là quá trình nhóm các đối tượng dữ liệu thành các cụm (cluster) sao cho:

- Các đối tượng **trong cùng một cụm** có độ tương đồng cao với nhau (khoảng cách nội cụm – intra-cluster distances – được **tối thiểu hóa**).
- Các đối tượng **thuộc các cụm khác nhau** có độ tương đồng thấp (khoảng cách liên cụm – inter-cluster distances – được **tối đa hóa**).

Ví dụ ứng dụng: phân nhóm khách hàng, phân tích mạng xã hội, phát hiện bất thường (outlier detection).

1.2. Đo lường độ bất tương đồng giữa các đối tượng

Dữ liệu được biểu diễn dưới dạng **ma trận dữ liệu** (n đối tượng, p thuộc tính) và **ma trận bất tương đồng** $d(i,j)$ thỏa:

- $d(i,i) = 0$
- $d(i,j) = d(j,i) \geq 0$
- $d(i,j) \leq d(i,k) + d(k,j)$ (bất đẳng thức tam giác)

Đo lường độ tương đồng bằng Cosine (cho đối tượng dạng vector):

$$s(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}^T\mathbf{y}}{|\mathbf{x}||\mathbf{y}|} = \frac{x_1y_1 + \cdots + x_py_p}{\sqrt{x_1^2 + \cdots + x_p^2} \cdot \sqrt{y_1^2 + \cdots + y_p^2}}$$

1.3. Tính khoảng cách theo từng loại thuộc tính

(a) Thuộc tính có thang đo khoảng (Interval-scaled): Để các thuộc tính có đơn vị và biên độ khác nhau không làm lệch khoảng cách, người ta **chuẩn hóa** theo dạng Z-score trên từng thuộc tính f ($f = 1,\ldots,p$) cho từng đối tượng i ($i = 1,\ldots,n$).

- **Trung bình (mean) của thuộc tính f:**

$$m_f = \frac{1}{n}\left(x_{1f} + x_{2f} + \cdots + x_{nf}\right) = \frac{1}{n}\sum_{i=1}^{n} x_{if}.$$

- **Độ lệch tuyệt đối trung bình (mean absolute deviation):**

$$s_f = \frac{1}{n}\Big(|x_{1f} - m_f| + |x_{2f} - m_f| + \cdots + |x_{nf} - m_f|\Big) = \frac{1}{n}\sum_{i=1}^{n} |x_{if} - m_f|.$$

- **Giá trị chuẩn hóa (Z-score measurement theo MAD):**

$$z_{if} = \frac{x_{if} - m_f}{s_f}.$$

  (Khi $s_f = 0$, tất cả x_{if} trên cột f bằng nhau; khi thực hành cần xử lý riêng, ví dụ bỏ qua thuộc tính đó hoặc đặt $z_{if} = 0$.)

Ghi chú: sau khi chuẩn hóa, dùng z_{if} **thay cho** x_{if} trong các bước tính khoảng cách; $i = 1..n$, $f = 1..p$. Tức là các công thức bên dưới áp dụng trên các **thành phần đã chuẩn hóa** z_{if}, z_{jf} (viết gọn dưới dạng tương đương với x để ngắn gọn, nhưng ý nghĩa là đã thay bằng z):

- **Euclidean:** $d(i,j) = \sqrt{(z_{i1} - z_{j1})^2 + \cdots + (z_{ip} - z_{jp})^2}$
- **Manhattan:** $d(i,j) = |z_{i1} - z_{j1}| + \cdots + |z_{ip} - z_{jp}|$
- **Minkowski:** $d(i,j) = \left(|z_{i1} - z_{j1}|^q + \cdots + |z_{ip} - z_{jp}|^q\right)^{1/q}$

(b) Thuộc tính nhị phân (Binary):

- **Khoảng cách đơn giản** (symmetric): $d(i,j) = \dfrac{b+c}{a+b+c+d}$
- **Khoảng cách Jaccard** (asymmetric): $d(i,j) = \dfrac{b+c}{a+b+c}$

Trong đó a: cả hai đều bằng 1; b: $i = 1, j = 0$; c: $i = 0, j = 1$; d: cả hai bằng 0.

(c) Thuộc tính danh mục (Categorical): $d_{ij}^{(f)} = 0$ nếu $x_{if} = x_{jf}$; bằng 1 trong các trường hợp còn lại.

(d) Thuộc tính thứ tự (Ordinal) hoặc tỉ lệ (Ratio-scaled): Chuyển đổi $z_{if} = \dfrac{r_{if} - 1}{M_f - 1}$ rồi áp dụng Minkowski/Euclidean/Manhattan.

(e) Thuộc tính hỗn hợp (Mixed types):

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} \cdot d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}$$

Trong đó $\delta_{ij}^{(f)} = 0$ nếu thiếu giá trị, ngược lại bằng 1.

1.4. Các yêu cầu thiết yếu của một phương pháp phân cụm tốt

- **Khả năng mở rộng (Scalability):** hoạt động tốt khi kích thước và loại dữ liệu thay đổi.
- **Xử lý đa kiểu dữ liệu** và dữ liệu chiều cao.
- **Phát hiện cụm có hình dạng tùy ý.**
- **Yêu cầu tối thiểu tham số** đầu vào.
- **Khả năng xử lý nhiễu và giá trị ngoại lệ (outlier).**
- **Phân cụm tăng dần** và không nhạy cảm với thứ tự đầu vào.
- **Tính khả giải thích và khả dụng.**

1.5. Các phương pháp phân cụm phổ biến

1. **Phân hoạch (Partitioning):** tạo và đánh giá các phân hoạch dựa trên tiêu chí cho trước.
2. **Phân cấp (Hierarchical):** phân chia tập dữ liệu theo thứ bậc.
3. **Dựa trên mật độ (Density-based):** dựa trên tính kết nối và mật độ phân bố.
4. **Dựa trên mô hình (Model-based):** đề xuất mô hình phân phối xác suất cho mỗi cụm.

1.6. Đánh giá kết quả phân cụm

- **Đánh giá ngoài (External validation):** so sánh với cấu trúc cụm đã biết trước. Các độ đo: Rand statistic, Jaccard coefficient, Folkes-Mallows index.
- **Đánh giá trong (Internal validation):** dựa trên ma trận khoảng cách. Các độ đo: Silhouette index, Dunn's index.

- **Đánh giá tương đối (Relative validation):** so sánh giữa các phương pháp.

Entropy (càng nhỏ càng tốt) để đánh giá độ thuần của cụm:

$$\text{Entropy}(P) = -\sum_i \frac{n_i}{n} \sum_j \frac{n_{ij}}{n_i} \log \frac{n_{ij}}{n_i}$$

trong đó $n_{ij} = |P_i \cap C_j|$ là số đối tượng trong cụm P_i thuộc lớp thực sự C_j.

2. PHÂN CỤM DỰA TRÊN PHÂN HOẠCH – K-MEANS

2.1. Thuật toán K-means

Ý tưởng: Chia n đối tượng thành k cụm không chồng lên nhau, tối thiểu hóa tổng bình phương khoảng cách từ mỗi đối tượng đến trọng tâm (centroid) cụm của nó.

Hàm mục tiêu:

$$s = \sum_{i=1}^{k} s_i = \sum_{i=1}^{k} \sum_{x \in C_i} \text{dist}(x, r_i)^2$$

trong đó r_i là trọng tâm của cụm C_i.

Các bước của thuật toán:

1. Chọn ngẫu nhiên k đối tượng làm tâm cụm ban đầu.
2. **Gán** mỗi đối tượng vào cụm có tâm gần nhất.
3. **Cập nhật** tâm cụm bằng cách tính trung bình tất cả đối tượng trong cụm.
4. Lặp lại bước 2–3 cho đến khi các tâm cụm không thay đổi (hoặc thay đổi dưới ngưỡng).

2.2. Đặc điểm của K-means

- **Tối ưu hóa cục bộ** – kết quả phụ thuộc vào khởi tạo ban đầu.
- **Độ phức tạp:** $O(nkt)$, với n là kích thước tập dữ liệu, k là số cụm, t là số vòng lặp ($k \ll n, t \ll n$).
- **Nhạy cảm** với nhiễu và giá trị ngoại lệ.
- **Không phù hợp** với cụm có hình dạng phi lồi (non-convex) hoặc kích thước khác nhau rõ rệt.

- Kết quả tạo ra các cụm có dạng **hình cầu** (hyperspherical) với kích thước tương đối đều nhau.
- Biến thể làm giảm nhạy cảm với ngoại lệ: **PAM (k-medoids)**.

2.3. PAM (Partitioning Around Medoids)

K-medoids (đại diện bằng medoid): thay vì dùng **trọng tâm (centroid)** là trung bình số học các vector trong cụm (có thể **không** trùng với bất kỳ đối tượng thực nào), k-medoids chọn **medoid** là một **đối tượng thực** trong tập dữ liệu làm đại diện cho cụm. Thông thường medoid của cụm C là điểm $o \in C$ sao cho tổng độ không tương đồng (hoặc tổng khoảng cách) từ o đến các đối tượng khác trong cụm là nhỏ nhất:

$$o^* = \arg\min_{o \in C} \sum_{x \in C} d(x, o)\,.$$

PAM là thuật toán phân hoạch kinh điển cho k-medoids, gồm hai pha chính:

1. **BUILD:** khởi tạo tập k medoid ban đầu bằng cách lần lượt chọn các điểm làm giảm mạnh hàm mục tiêu (tổng khoảng cách gán cụm).
2. **SWAP:** lặp lại việc thử đổi một medoid hiện tại với một đối tượng không phải medoid; nếu hoán đổi làm **giảm** tổng chi phí thì chấp nhận, cho đến khi không còn cải thiện.

So sánh trực tiếp K-means và PAM (k-medoids):

Tiêu chí	K-means	PAM (k-medoids)
Đại diện cụm	**Mean** (centroid): thường là điểm "ảo" trong không gian thuộc tính.	**Medoid**: **điểm dữ liệu thực**, dễ giải thích (ví dụ "khách hàng mẫu").
Hàm mục tiêu điển hình	Tối thiểu tổng bình phương khoảng cách tới centroid (SSE); gắn chặt với khoảng cách Euclidean.	Tối thiểu tổng khoảng cách (hoặc không tương đồng) tới medoid; dùng được với **ma trận khoảng cách** bất kỳ (Manhattan, v.v.).
Độ bền với ngoại lệ / nhiễu	**Nhạy:** một outlier có thể kéo centroid lệch mạnh.	**Bền hơn:** medoid phải là điểm thực ở "trung tâm" cụm, ít bị kéo bởi giá trị cực đoan.
Độ phức tạp	Thường $O(nkt)$, phù hợp dữ liệu **rất lớn**.	PAM đầy đủ tốn kém hơn (bậc $O(n^2)$ mỗi vòng trong các thao tác SWAP); có phiên bản xấp xỉ (CLARA, CLARANS) cho dữ liệu lớn.
Khởi tạo / tối ưu	Cục bộ, phụ thuộc khởi tạo.	Cục bộ tương tự; BUILD–SWAP vẫn không đảm bảo toàn cục.

Ưu điểm PAM so với K-means:

- **Ổn định hơn** trước nhiễu và điểm ngoại lệ nhờ medoid thay cho mean.
- **Có đại diện thực tế** (đối tượng có thể chỉ ra trong báo cáo nghiệp vụ).
- Dùng được khi chỉ có **ma trận không tương đồng** (không cần vector trong $\mathbb{R}^p$).

Nhược điểm PAM so với K-means:

- **Chậm hơn** trên cùng n, k khi dùng PAM "đầy đủ"; K-means mỗi bước chỉ cần tính mean và gán lại.
- Vẫn phải **chọn** k trước (giống K-means), vẫn có rủi ro **tối ưu cục bộ**.
- Với dữ liệu cực lớn, thường cần **CLARA/CLARANS** hoặc mẫu con thay vì PAM nguyên bản trên toàn bộ n.

K-means phù hợp khi dữ liệu sạch, lớn, Euclidean và cần tốc độ; PAM phù hợp khi cần **đại diện có thể giải thích**, dữ liệu **có nhiễu**, hoặc khoảng cách/không tương đồng **không gắn với mean bình phương**.

3. PHÂN CỤM DỰA TRÊN PHÂN CẤP (HIERARCHICAL CLUSTERING)

3.1. Khái niệm

Phân cụm phân cấp (*hierarchical clustering*) xây dựng một **cấu trúc lồng nhau** của các cụm: tại mức trên, cụm lớn **chứa** các cụm nhỏ hơn ở mức dưới. Toàn bộ quy trình được biểu diễn gọn bằng **dendrogram** (sơ đồ cây): trục thể hiện thứ tự/mức gộp hoặc tách, độ cao nút thường gắn với khoảng cách (hoặc độ không tương đồng) tại thời điểm gộp hai cụm. Nhờ đó, người dùng có thể **cắt** cây ở một mức bất kỳ để thu được phân hoạch với số cụm tương ứng – khác với K-means phải cố định k ngay từ đầu. Có hai hướng tiếp cận chính:

Agglomerative (bottom-up – từ dưới lên):

- **Khởi tạo:** mỗi trong n đối tượng tạo thành **một cụm đơn** $\Rightarrow$ ban đầu có đúng n cụm.
- **Vòng lặp:** ở mỗi bước, tìm **hai cụm** "gần nhau" nhất theo một **tiêu chí liên kết** (linkage) đã chọn (single-, complete-, average-linkage, Ward,...), rồi **gộp** chúng lại thành một cụm mới. Số cụm giảm đi 1 sau mỗi lần gộp.
- **Dừng:** sau $n-1$ lần gộp, còn **một cụm duy nhất** chứa toàn bộ dữ liệu – toàn bộ lịch sử gộp tạo nên cây phân cấp.
- **Đặc điểm:** cách làm tự nhiên, phổ biến trong phần mềm (ví dụ thuật toán **AGNES** ở mục sau). Chi phí tính toán thường **cao hơn nhiều** so với một lần chạy K-means khi n lớn ($O(n^2)$ hoặc $O(n^3)$ tùy cách cài đặt và loại linkage).

Divisive (top-down – từ trên xuống):

- **Khởi tạo: một cụm** duy nhất chứa tất cả đối tượng.
- **Vòng lặp:** ở mỗi bước, **tách** một cụm hiện có thành hai (hoặc nhiều) cụm con sao cho tiêu chí tách là tốt nhất (ví dụ tách theo điểm "xa" nhất, hoặc tối đa hóa khoảng cách liên cụm). Số cụm tăng dần cho đến khi đạt cấu trúc mong muốn hoặc mỗi đối tượng là một cụm riêng.
- **Đặc điểm:** ít dùng hơn agglomerative vì mỗi bước tách phải xem xét **nhiều cách phân chia** (tổ hợp) – thường **tốn kém** hơn; đại diện điển hình là **DIANA**.

So sánh nhanh hai hướng: bottom-up bắt đầu từ chi tiết rồi gộp dần; top-down bắt đầu từ toàn cục rồi tinh chỉnh dần. Cả hai đều cho cùng một **dạng kết quả** (cây), không bắt buộc biết k khi chạy thuật toán, nhưng về mặt tính toán agglomerative thường **dễ triển khai và phổ biến** hơn.

Ưu điểm chung: không cần cố định trước số cụm k; có được **nhiều phân hoạch lồng nhau** trong một lần chạy; trực quan khi minh họa bằng dendrogram.

Nhược điểm chung: mỗi bước gộp/tách là **quyết định cố định** – sai lầm ở bước đầu khó sửa (**không có thao tác undo**); độ phức tạp và bộ nhớ thường **không lý tưởng** cho dữ liệu cực lớn so với các phương pháp phân hoạch đơn giản (như K-means một lần).

3.2. AGNES và DIANA

AGNES (Agglomerative NESting): triển khai điển hình của chiến lược **bottom-up**. Giả sử đã có ma trận khoảng cách (hoặc không tương đồng) giữa từng cặp đối tượng.

- **Bước 1:** Khởi tạo n cụm, mỗi cụm chứa đúng một đối tượng.
- **Bước 2:** Tính ma trận khoảng cách **giữa các cụm** hiện có theo một **tiêu chí linkage** (mục dưới).
- **Bước 3:** Tìm cặp cụm có khoảng cách **nhỏ nhất** và **gộp** hai cụm đó thành một.
- **Bước 4:** Cập nhật lại khoảng cách từ cụm mới đến các cụm còn lại (công thức cập nhật tùy thuộc linkage; ví dụ Lance–Williams cho nhiều họ linkage).
- **Bước 5:** Lặp bước 3–4 cho đến khi còn đúng **một cụm** (đã thực hiện $n-1$ lần gộp).

Độ phức tạp: cài đặt thẳng thường là $O(n^3)$ hoặc tối ưu hơn tới khoảng $O(n^2 \log n)$ tùy cấu trúc dữ liệu và linkage. Bộ nhớ thường cần lưu ma trận $n \times n$ (hoặc danh sách kề), nên n rất lớn sẽ tốn.

DIANA (Divisive ANAlysis): ví dụ điển hình của **top-down**: bắt đầu từ một cụm gồm toàn bộ n điểm, lặp lại việc **tách** một cụm thành hai cụm con cho đến khi mỗi cụm chỉ còn một đối tượng (hoặc dừng sớm theo tiêu chí người dùng).

- **Bước 1:** Bắt đầu với cụm C chứa tất cả đối tượng.
- **Bước 2:** Trong cụm C, tìm cách phân chia thành hai nhóm con C_1 và C_2 sao cho **khoảng cách liên nhóm** (giữa C_1 và C_2) lớn nhất hoặc tổng độ không đồng nhất trong từng nhóm là nhỏ nhất—thường dùng ý tưởng: chọn **tâm tách** (điểm "xa" nhất so với trung tâm cụm) và gán từng điểm vào nhóm gần tâm nào hơn, rồi cập nhật lại đại diện hai nhóm lặp lại cho đến khi ổn định.
- **Bước 3:** Trong các cụm hiện có, chọn cụm **không đồng nhất nhất** (ví dụ đường kính lớn) để **tách tiếp**.
- **Bước 4:** Lặp cho đến khi đạt đủ n cụm đơn hoặc người dùng dừng.

Mỗi lần tách phải xét nhiều cách chia nên DIANA thường **tốn hơn** AGNES trên cùng dữ liệu nhỏ; với dữ liệu lớn, divisive ít được dùng hơn agglomerative nếu không có tối ưu bổ sung.

3.3. Tiêu chí gộp cụm (linkage)

Cho hai cụm không rỗng C_i và C_j, khoảng cách giữa cụm $D(C_i, C_j)$ được định nghĩa khác nhau tùy loại **liên kết**. Dưới đây $d(x, y)$ là khoảng cách (hoặc không tương đồng) giữa hai đối tượng.

- **Single-linkage (liên kết đơn / minimum):**

$$D(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y).$$

  Gộp hai cụm "gần nhau" nhất theo **cặp điểm gần nhất**. Dễ tạo chuỗi cụm dài (*chaining effect*) khi có cầu nối nhiễu giữa hai vùng dày.

- **Complete-linkage (liên kết đầy đủ / maximum):**

$$D(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y).$$

  Hai cụm chỉ được coi là gần khi **mọi** cặp điểm liên cụm đều không quá xa—thường tạo cụm **gọn** (compact), nhưng nhạy hơn với ngoại lệ kéo khoảng cách tối đa.

- **Average-linkage (liên kết trung bình / UPGMA):**

$$D(C_i, C_j) = \frac{1}{|C_i|\,|C_j|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y).$$

  Cân bằng hơn giữa single và complete; phản ánh mức độ tách rời **trung bình** giữa hai cụm.

- **Ward's method (phương pháp Ward):** Mỗi lần gộp chọn cặp cụm làm **tăng ít nhất** tổng bình phương sai số trong cụm (ESS – error sum of squares / within-cluster variance). Điều kiện thường gắn với khoảng cách Euclidean; thường cho các cụm có kích thước và phương sai tương đối **cân đối**, thiên hướng tối ưu hóa kiểu "cầu K-means" ở từng bước gộp.

Linkage khác nhau $\Rightarrow$ dendrogram và cụm khi cắt cây có thể khác mạnh—cần chọn và kiểm định theo miền ứng dụng.

3.4. Dendrogram

Dendrogram là biểu diễn đồ họa của toàn bộ chuỗi gộp (hoặc tách): mỗi **nút lá** là một đối tượng (hoặc cụm ban đầu); mỗi lần hai nhánh hợp nhất thành nút cha tương ứng một lần gộp. **Chiều cao** (hoặc vị trí trên trục dọc) tại điểm gộp thường tỷ lệ với khoảng cách D giữa hai cụm vừa gộp – càng gộp muộn (ở **cao** trên cây) thường là hai phần càng **khác biệt**.

Cách dùng thực tế:

- **Cắt ngang** dendrogram ở một mức h: mọi nhánh còn cách gốc "xa" hơn h tạo thành các cụm tách biệt $\Rightarrow$ thu được phân hoạch với một số cụm cố định.
- Có thể chọn mức cắt để có đúng k cụm, hoặc theo "khoảng trống" lớn trên trục chiều cao (khe lớn = tự nhiên tách thành k nhóm).

Muốn dendrogram nhất quán (không có **nghịch đảo** thứ tự gộp), nhiều thuật toán yêu cầu D thỏa tính **đơn điệu** theo thời gian gộp – linkage khác nhau cho tính chất khác nhau (ví dụ single hay có hiện tượng chuỗi).

3.5. Một số thuật toán / phương pháp phân cụm liên quan phân cấp

- **BIRCH** (*Balanced Iterative Reducing and Clustering using Hierarchies*): thiết kế cho dữ liệu **rất lớn**. Dữ liệu được tóm tắt thành các **clustering feature** (CF) trong cây cân bằng (CF tree) – lưu số đếm, tổng, bình phương tổng theo chiều, v.v. Phân cụm chủ yếu trên tóm tắt $\Rightarrow$ chỉ cần quét dữ liệu vài lần (linear scan). BIRCH có thể kết hợp bước tinh chỉnh (ví dụ K-means trên tâm lá) nhưng **ưu tiên khả năng mở rộng** hơn phân cụm phân cấp "nguyên bản" trên ma trận đầy đủ.
- **ROCK** (*Robust Clustering using linKs*): nhắm dữ liệu có thuộc tính **danh mục / rời rạc**. Thay vì chỉ dùng khoảng cách cặp điểm, ROCK dùng khái niệm **liên kết (link)**: hai điểm "liên kết" nếu cùng tương tự với một tập điểm thứ ba; số liên kết $\Rightarrow$ độ tương đồng cụm bền hơn với thuộc tính định tính.
- **Chameleon:** kết hợp mô hình **đồ thị** (mỗi điểm là đỉnh, cạnh theo láng giềng gần) và độ đo **tự thích ứng**: quyết định gộp hai cụm dựa trên **kết nối nội bộ** và **kết nối liên cụm** tương đối, chứ không chỉ một hàm D cố định. Phù hợp cụm có **hình dạng bất thường**, mật độ không đều (tương tự tinh thần density-based nhưng trong khung phân cụm phân cấp/người láng giềng).

3.6. Vấn đề và hạn chế của phân cụm phân cấp

- **Chọn điểm dừng và cách cắt cây:** thuật toán cho cả **cây đầy đủ**; người dùng phải quyết định cắt ở đâu (mức chiều cao, số cụm k, hoặc heuristics). Sai lựa chọn $\Rightarrow$ phân hoạch không phù hợp.
- **Sai số tích lũy (không undo):** mỗi lần gộp/tách là quyết định **vĩnh viễn**; nếu hai cụm "đáng lẽ" tách bị gộp sớm do linkage/nhiễu, các bước sau không sửa được cấu trúc cha.
- **Khả năng mở rộng:** ma trận khoảng cách đầy đủ là $O(n^2)$ bộ nhớ; từng bước agglomerative có thể $O(n^3)$ ở cài đặt đơn giản. Với luồng dữ liệu hoặc n cực lớn cần BIRCH, mẫu, hoặc xấp xỉ.
- **Nhạy cảm với nhiễu và linkage:** single-linkage rất nhạy chuỗi; complete nhạy outlier; Ward gắn Euclidean và giả định cầu – dữ liệu thực tế cần **so sánh nhiều linkage** hoặc chỉnh tiền xử lý/ chuẩn hóa.
- **Giải pháp thực tế:** kết hợp **chia nhỏ rồi gộp** (divide-and-conquer), phân cụm hai mức (ví dụ BIRCH $\rightarrow$ tinh chỉnh), hoặc dùng phân cụm phân cấp chỉ trên **mẫu** rồi gán điểm còn lại cho cụm gần nhất.

4. PHÂN CỤM DỰA TRÊN MẬT ĐỘ (DENSITY-BASED CLUSTERING)

4.1. Khái niệm cơ bản

- **Cụm** là vùng **dày đặc** các đối tượng, bao quanh bởi vùng thưa.
- Đối tượng ở vùng thưa là **nhiễu/ngoại lệ (noise/outlier)**.
- Hình dạng và kích thước cụm rất đa dạng (không bị giới hạn dạng cầu).

Các khái niệm trong DBSCAN:

- ε **(epsilon):** bán kính lân cận của một đối tượng.
- ε**-neighborhood:** tập tất cả đối tượng trong vùng bán kính ε của đối tượng p.
- **Core object (đối tượng lõi):** đối tượng p có số lượng đối tượng trong ε-neighborhood $\geq$ MinPts.
- **Directly density-reachable (trực tiếp đạt được theo mật độ):** q trực tiếp đạt được từ p nếu $q \in \varepsilon\text{-neighborhood}(p)$ và p là core object. *Quan hệ bất đối xứng.*

- **Density-reachable (đạt được theo mật độ):** q đạt được từ p nếu tồn tại chuỗi $p_1, \ldots, p_n$ với $p_1 = p$, $p_n = q$ sao cho p_{i+1} directly density-reachable từ p_i. *Quan hệ bất đối xứng.*
- **Density-connected (kết nối theo mật độ):** p và q density-connected nếu tồn tại o sao cho cả p và q đều density-reachable từ o. *Quan hệ đối xứng.*
- **Border object (đối tượng biên):** không phải core object nhưng nằm trong ε-neighborhood của một core object.
- **Noise:** đối tượng không thuộc bất kỳ cụm nào.

4.2. Thuật toán DBSCAN

Yêu cầu:

- Đầu vào: D, ε, MinPts.
- Đầu ra: Các cụm dựa trên mật độ và nhiễu.

Các bước hiện thực:

1. Xác định ε-neighborhood cho mỗi đối tượng $p \in D$.
2. Nếu p là core object $\Rightarrow$ tạo cụm cho p.
3. Từ bất kỳ core object p, tìm tất cả đối tượng density-reachable từ p và thêm vào cụm của p.
    - Các cụm density-reachable có thể gộp lại.
    - Dừng khi không còn đối tượng nào có thể thêm vào cụm.

4.3. Đặc điểm của DBSCAN

- Phát hiện cụm có kích thước và hình dạng đa dạng.
- Không giả định về phân phối đối tượng.
- **Không cần** xác định k ban đầu.
- Khởi tạo không ảnh hưởng đến kết quả.
- Cần định nghĩa **mật độ** qua ε và MinPts.
- Nhận diện nhiễu và ngoại lệ hiệu quả.

- **Độ phức tạp:** $O(n \log n)$ đến $O(n^2)$.

Thuật toán liên quan: OPTICS (Ordering Points To Identify the Clustering Structure), DENCLUE (DENsity-based CLUstEring based on distribution functions).

5. PHÂN CỤM DỰA TRÊN MÔ HÌNH (MODEL-BASED CLUSTERING)

5.1. Khái niệm

Phương pháp này **tối ưu hóa sự khớp** giữa dữ liệu và một số mô hình toán học, dựa trên giả định rằng dữ liệu được sinh ra từ một hỗn hợp các mô hình phân phối xác suất.

Các hướng tiếp cận:

- **Thống kê:** thuật toán EM (Expectation-Maximization) – mở rộng của K-means.
- **Học máy:** phân cụm khái niệm (conceptual clustering).
- **Dựa trên mạng nơ-ron:** Self-Organizing Feature Map (SOM).

5.2. Thuật toán EM (Expectation-Maximization)

Giả định: Dữ liệu được sinh ra từ hỗn hợp K phân phối Gaussian, mỗi phân phối có tham số $\Theta_j = (\mu_j, \sigma_j)$.

EM là thuật toán lặp để tìm Maximum Likelihood (ML) kể cả khi có dữ liệu bị thiếu:

- **E-step (Expectation):** Ước lượng xác suất mỗi đối tượng x_i thuộc cụm j:

$$E[z_{ij}] = p(x_i \in C_j | x_i) = \frac{p(x_i | x_i \in C_j) \cdot p_j}{\sum_{n=1}^{K} p(x_i | x_i \in C_n) \cdot p_n}$$

- **M-step (Maximization):** Cập nhật tham số mô hình:

$$\mu_j = \frac{\sum_{i=1}^{m} E[z_{ij}] \cdot x_i}{\sum_{i=1}^{m} E[z_{ij}]}, \quad p_j = \frac{1}{m} \sum_{i=1}^{m} E[z_{ij}]$$

Tóm tắt thuật toán EM:

1. **Khởi tạo:** Chọn ngẫu nhiên K đối tượng làm tâm cụm ban đầu; ước lượng $\Theta_j = (\mu_j, \sigma_j)$.
2. **Lặp:**

- **E-step:** Gán x_i vào cụm C_k với xác suất $P(x_i \in C_k)$, $k = 1 \dots K$.
- **M-step:** Ước lượng lại $\Theta_j = (\mu_j, \sigma_j)$.
- Dừng khi đạt điều kiện hội tụ (ví dụ: ML không tăng thêm đáng kể).

5.3. EM so với K-means

- EM sử dụng xác suất thuộc cụm (soft assignment) thay vì gán cứng (hard assignment) như K-means.
- EM phù hợp với dữ liệu có phân phối Gaussian.
- EM tổng quát hơn K-means, nhưng tính toán phức tạp hơn.

6. CÁC PHƯƠNG PHÁP PHÂN CỤM KHÁC

6.1. Phân cụm cứng và phân cụm mờ

- **Phân cụm cứng (Hard clustering):**
  - Mỗi đối tượng thuộc **đúng một cụm**.
  - Mức độ thành viên (Degree of Membership – DoM): 0 hoặc 1.
  - Ranh giới giữa các cụm rõ ràng.
- **Phân cụm mờ (Fuzzy clustering):**
  - Một đối tượng có thể thuộc **nhiều cụm** với mức độ thành viên từ 0 đến 1.
  - Ranh giới giữa các cụm không rõ ràng (mờ – fuzzy).
  - Phù hợp với dữ liệu có ranh giới tự nhiên không rõ ràng.

7. TÓM TẮT

- **Phân cụm** là kỹ thuật học không giám sát, nhóm dữ liệu dựa trên độ tương đồng.
- Đo lường độ tương đồng phụ thuộc vào **kiểu dữ liệu** của các thuộc tính.
- **K-means:** đơn giản, hiệu quả nhưng nhạy cảm với khởi tạo và nhiễu; tạo cụm hình cầu.
- **Hierarchical:** không cần k, kết quả trực quan qua dendrogram; khó mở rộng.
- **DBSCAN:** phát hiện cụm hình dạng tùy ý, xử lý nhiễu tốt; cần chọn ε và MinPts phù hợp.

- **EM:** phân cụm mềm (soft) dựa trên mô hình xác suất; tổng quát hơn K-means.
- Đánh giá kết quả phân cụm bằng external, internal hoặc relative validation.

8. CÂU HỎI TỰ LUẬN

Câu 1. Phân cụm dữ liệu (data clustering) là gì? Trình bày mục tiêu của phân cụm và nêu ít nhất 3 ứng dụng thực tiễn của phân cụm trong khai phá dữ liệu.

Câu 2. Giải thích sự khác biệt giữa **ma trận dữ liệu** và **ma trận bất tương đồng** trong phân cụm. Ma trận bất tương đồng $d(i, j)$ cần thỏa mãn những điều kiện nào?

Câu 3. Trình bày các công thức tính khoảng cách Euclidean, Manhattan và Minkowski cho các thuộc tính có thang đo khoảng. Khi nào nên chuẩn hóa dữ liệu trước khi tính khoảng cách?

Câu 4. Giải thích khoảng cách đơn giản (simple distance) và khoảng cách Jaccard dùng cho thuộc tính nhị phân. Khi nào dùng khoảng cách đối xứng, khi nào dùng khoảng cách Jaccard? Cho ví dụ minh họa.

Câu 5. Trình bày đầy đủ các bước của thuật toán K-means. Hàm mục tiêu của K-means là gì? Thuật toán dừng khi nào?

Câu 6. Phân tích các ưu điểm và nhược điểm của thuật toán K-means. Tại sao K-means không phù hợp với cụm có hình dạng phi lồi (non-convex)?

Câu 7. So sánh phân cụm phân cấp theo hướng từ dưới lên (AGNES) và từ trên xuống (DIANA). Phân cụm phân cấp có ưu điểm gì so với K-means? Hạn chế là gì?

Câu 8. Giải thích tiêu chí **single-linkage** và **complete-linkage** trong gộp cụm phân cấp. Mỗi tiêu chí tạo ra loại cụm như thế nào? Minh họa bằng ví dụ cụ thể.

Câu 9. **Dendrogram** là gì? Làm thế nào để đọc và sử dụng dendrogram để xác định số cụm? Nêu ưu điểm của cách biểu diễn này.

Câu 10. Trình bày và phân biệt các khái niệm trong DBSCAN: **ε-neighborhood**, **core object**, **directly density-reachable**, **density-reachable** và **density-connected**. Tính đối xứng của mỗi quan hệ?

Câu 11. Mô tả chi tiết thuật toán DBSCAN. Đối tượng nào được xem là **border object**, đối tượng nào là **noise**? Các tham số ε và MinPts ảnh hưởng thế nào đến kết quả?

Câu 12. So sánh DBSCAN với K-means về: (a) khả năng phát hiện hình dạng cụm tùy ý, (b) xử lý nhiễu, (c) nhu cầu chỉ định số cụm trước, (d) độ phức tạp tính toán.

Câu 13. Phân cụm dựa trên mô hình (model-based clustering) là gì? Trình bày giả định về mô hình phân phối dữ liệu trong phương pháp EM. Ưu điểm của phương pháp này so với K-means là gì?

Câu 14. Giải thích thuật toán EM (Expectation-Maximization). Mô tả chi tiết E-step và M-step. Thuật toán EM được khởi tạo và dừng như thế nào?

Câu 15. So sánh **phân cụm cứng (hard clustering)** và **phân cụm mờ (fuzzy clustering)**. Trong trường hợp nào nên sử dụng phân cụm mờ? Nêu ví dụ thực tế.

Câu 16. Trình bày các yêu cầu thiết yếu mà một phương pháp phân cụm tốt cần đáp ứng. Với mỗi yêu cầu, cho biết phương pháp nào (K-means, DBSCAN, Hierarchical, EM) đáp ứng tốt nhất.

Câu 17. Giải thích ba phương pháp đánh giá kết quả phân cụm: **external validation**, **internal validation** và **relative validation**. Nêu ít nhất một độ đo cụ thể cho mỗi phương pháp.

Câu 18. Entropy trong đánh giá kết quả phân cụm được tính như thế nào? Một giá trị entropy nhỏ có nghĩa là gì? Cho ví dụ tính entropy cho một kết quả phân cụm cụ thể.

Câu 19. Trình bày cách tính khoảng cách cho thuộc tính **hỗn hợp (mixed types)**. Khi một giá trị thuộc tính bị thiếu, hệ số $\delta_{ij}^{(f)}$ được xử lý như thế nào?

Câu 20. Phân tích các vấn đề (challenges) trong phân cụm phân cấp về khả năng mở rộng. Giải pháp nào có thể được áp dụng để cải thiện hiệu suất? Trình bày thuật toán BIRCH và ý tưởng chính của nó.

9. CÂU HỎI TRẮC NGHIỆM

Câu 1. Mục tiêu chính của phân cụm dữ liệu là:

A. Tối đa hóa khoảng cách nội cụm và tối thiểu hóa khoảng cách liên cụm.

B. Tối thiểu hóa khoảng cách nội cụm và tối đa hóa khoảng cách liên cụm.

C. Gán nhãn cho từng đối tượng dựa trên tập huấn luyện.

D. Tìm các luật kết hợp giữa các thuộc tính.

Câu 2. Phân cụm là kỹ thuật học:

A. Có giám sát (supervised learning).

B. Không giám sát (unsupervised learning).

C. Bán giám sát (semi-supervised learning).

D. Học tăng cường (reinforcement learning).

Câu 3. Ma trận bất tương đồng $d(i,j)$ **không** thỏa điều kiện nào sau đây?

A. $d(i,i) = 0$

B. $d(i,j) = d(j,i) \geq 0$

C. $d(i,j) \leq d(i,k) + d(k,j)$

D. $d(i,j) = d(i,k) \cdot d(k,j)$

Câu 4. Khoảng cách Minkowski với $q = 1$ tương đương với:

A. Khoảng cách Euclidean.

B. Khoảng cách Manhattan.

C. Khoảng cách Cosine.

D. Khoảng cách Jaccard.

Câu 5. Khoảng cách Minkowski với $q = 2$ tương đương với:

A. Khoảng cách Manhattan.

B. Khoảng cách Chebyshev.

C. Khoảng cách Euclidean.

D. Khoảng cách Hamming.

Câu 6. Khoảng cách Jaccard dùng cho thuộc tính nhị phân **bất đối xứng** có công thức:

A. $d(i,j) = \dfrac{b+c}{a+b+c+d}$

B. $d(i,j) = \dfrac{b+c}{a+b+c}$

C. $d(i,j) = \dfrac{a}{a+b+c}$

D. $d(i,j) = \dfrac{a+d}{a+b+c+d}$

Câu 7. Trong đo lường tương đồng bằng Cosine, kết quả $s(\mathbf{x},\mathbf{y}) = 1$ có nghĩa là:

A. Hai vector hoàn toàn trực giao (vuông góc).

B. Hai vector hoàn toàn giống nhau (cùng hướng).

C. Hai vector hoàn toàn ngược chiều.

D. Hai vector không có quan hệ gì với nhau.

Câu 8. Trong đánh giá kết quả phân cụm, **internal validation** dựa trên:

A. Cấu trúc cụm được định nghĩa sẵn từ bên ngoài.

B. Ma trận khoảng cách (proximity matrix) tính từ tập dữ liệu.

C. So sánh hiệu suất giữa các phương pháp khác nhau.

D. Nhãn lớp thực sự của dữ liệu.

Câu 9. Thuật toán K-means dừng khi:

A. Số vòng lặp đạt đúng bằng k.

B. Các tâm cụm không thay đổi (hoặc thay đổi dưới ngưỡng).

C. Tất cả đối tượng được gán vào đúng 1 cụm.

D. Số cụm tăng lên bằng n.

Câu 10. Độ phức tạp tính toán của thuật toán K-means là:

A. $O(n^2)$

B. $O(n \log n)$

C. $O(nkt)$

D. $O(k^n)$

Câu 11. Nhược điểm của K-means là:

A. Không cần xác định số cụm k trước.

B. Chỉ tạo ra cụm có hình dạng phi lồi (non-convex).

C. Nhạy cảm với nhiễu và giá trị ngoại lệ.

D. Tự động phát hiện số cụm tối ưu.

Câu 12. K-means tạo ra các cụm có dạng:

A. Hình dạng tùy ý (arbitrary shapes).

B. Hình cầu (hyperspherical) với kích thước tương đối đều nhau.

C. Hình elipse kéo dài.

D. Chuỗi (chain-like).

Câu 13. Phương pháp **PAM (k-medoids)** khác K-means ở chỗ:

A. Sử dụng **medoid** (đối tượng thực trong tập dữ liệu) thay vì mean làm tâm cụm.

B. Không cần xác định số cụm k trước.

C. Sử dụng mật độ để xác định cụm.

D. Sử dụng phân phối Gaussian.

Câu 14. Hàm mục tiêu của K-means là tối thiểu hóa:

A. Tổng khoảng cách giữa các tâm cụm.

B. Tổng bình phương khoảng cách từ mỗi đối tượng đến tâm cụm của nó.

C. Số lượng cụm k.

D. Khoảng cách liên cụm trung bình.

Câu 15. Phân cụm phân cấp theo hướng agglomerative (AGNES) bắt đầu bằng:

A. Một cụm chứa tất cả đối tượng, sau đó tách dần.

B. Mỗi đối tượng là một cụm riêng, sau đó gộp dần.

C. k cụm ngẫu nhiên, sau đó cập nhật tâm cụm.

D. Các cụm dựa trên mật độ vùng.

Câu 16. Ưu điểm của phân cụm phân cấp so với K-means là:

A. Không cần xác định số cụm k trước.

B. Có độ phức tạp thấp hơn K-means.

C. Không bị ảnh hưởng bởi nhiễu.

D. Luôn tìm được cụm tối ưu toàn cục.

Câu 17. Tiêu chí **single-linkage** trong gộp cụm dựa trên:

A. Khoảng cách trung bình giữa tất cả cặp đối tượng.

B. Khoảng cách ngắn nhất giữa hai đối tượng bất kỳ trong hai cụm.

C. Khoảng cách dài nhất giữa hai đối tượng bất kỳ trong hai cụm.

D. Khoảng cách giữa hai tâm cụm.

Câu 18. Tiêu chí **complete-linkage** trong gộp cụm dựa trên:

A. Khoảng cách ngắn nhất giữa hai đối tượng bất kỳ trong hai cụm.

B. Khoảng cách trung bình giữa các tâm cụm.

C. Khoảng cách dài nhất giữa hai đối tượng bất kỳ trong hai cụm.

D. Khoảng cách Jaccard giữa hai cụm.

Câu 19. Dendrogram được sử dụng để:

A. Biểu diễn quá trình phân cụm phân cấp theo dạng cây.

B. Biểu diễn phân phối mật độ của dữ liệu.

C. Tính khoảng cách giữa các cụm.

D. Khởi tạo tâm cụm ban đầu cho K-means.

Câu 20. Thuật toán **BIRCH** được thiết kế để giải quyết vấn đề:

A. Phân cụm dữ liệu phân phối Gaussian.

B. Phân cụm phân cấp trên tập dữ liệu lớn (scalability).

C. Phát hiện cụm có hình dạng tùy ý.

D. Phân cụm dữ liệu thuộc tính danh mục.

Câu 21. Trong DBSCAN, **core object** là đối tượng:

A. Có khoảng cách tới tâm cụm nhỏ nhất.

B. Có số đối tượng trong ε-neighborhood $\geq$ MinPts.

C. Nằm trên ranh giới giữa hai cụm.

D. Không thuộc bất kỳ cụm nào.

Câu 22. Trong DBSCAN, đối tượng **noise (nhiễu)** là:

A. Đối tượng nằm ở biên giới của cụm.

B. Đối tượng có ít nhất MinPts láng giềng.

C. Đối tượng không thuộc bất kỳ cụm nào.

D. Đối tượng có khoảng cách lớn nhất tới tâm cụm.

Câu 23. Quan hệ **density-reachable** trong DBSCAN có tính:

A. Đối xứng (symmetric).

B. Bất đối xứng (asymmetric).

C. Phản xạ (reflexive) nhưng không bắc cầu.

D. Cả đối xứng và phản xạ.

Câu 24. Quan hệ **density-connected** trong DBSCAN có tính:

A. Bất đối xứng.

B. Đối xứng.

C. Chỉ đúng với core object.

D. Chỉ đúng với border object.

Câu 25. Ưu điểm của DBSCAN so với K-means là:

A. Cần ít tham số đầu vào hơn.

B. Phát hiện cụm có hình dạng tùy ý và xử lý nhiễu tốt.

C. Chỉ hoạt động với dữ liệu phân phối Gaussian.

D. Có độ phức tạp $O(n)$ trong mọi trường hợp.

Câu 26. Độ phức tạp của DBSCAN trong trường hợp tốt nhất là:

A. $O(n)$

B. $O(n \log n)$

C. $O(n^2)$

D. $O(nk)$

Câu 27. Thuật toán **OPTICS** trong phân cụm dựa trên mật độ được dùng để:

A. Thay thế hoàn toàn K-means trong mọi trường hợp.

B. Sắp xếp các điểm để xác định cấu trúc phân cụm.

C. Phân cụm dữ liệu theo mô hình Gaussian.

D. Tính khoảng cách Jaccard giữa các cụm.

Câu 28. Trong DBSCAN, nếu tăng giá trị ε thì:

A. Số cụm tăng lên và số nhiễu tăng lên.

B. Số cụm giảm xuống và số nhiễu giảm xuống.

C. Số cụm không thay đổi.

D. Chỉ ảnh hưởng đến các border object.

Câu 29. Phân cụm dựa trên mô hình (model-based) giả định rằng:

A. Dữ liệu phân bố đều trong không gian.

B. Dữ liệu được sinh ra từ hỗn hợp các mô hình phân phối xác suất.

C. Tất cả cụm có kích thước bằng nhau.

D. Dữ liệu không có nhiễu.

Câu 30. Trong thuật toán EM, **E-step** thực hiện:

A. Cập nhật tham số mô hình $\Theta_j = (\mu_j, \sigma_j)$.

B. Ước lượng xác suất mỗi đối tượng thuộc mỗi cụm.

C. Gán cứng mỗi đối tượng vào cụm gần nhất.

D. Tính khoảng cách Euclidean giữa các đối tượng.

Câu 31. Trong thuật toán EM, **M-step** thực hiện:

A. Ước lượng xác suất thuộc cụm của mỗi đối tượng.

B. Khởi tạo ngẫu nhiên tâm cụm ban đầu.

C. Cực đại hóa hàm log-likelihood bằng cách cập nhật tham số mô hình.

D. Xóa bỏ các đối tượng nhiễu.

Câu 32. So với K-means, thuật toán EM sử dụng:

A. Gán cứng (hard assignment) – mỗi đối tượng thuộc đúng 1 cụm.

B. Gán mềm (soft assignment) – mỗi đối tượng có xác suất thuộc mỗi cụm.

C. Khoảng cách Manhattan thay vì Euclidean.

D. Medoid thay vì mean làm tâm cụm.

Câu 33. Thuật toán EM dừng khi:

A. Đã chạy đúng k vòng lặp.

B. Đạt điều kiện hội tụ (ví dụ: log-likelihood không tăng thêm đáng kể).

C. Tất cả đối tượng có cùng xác suất thuộc mỗi cụm.

D. Số cụm tăng lên đến n.

Câu 34. Trong **phân cụm mờ (fuzzy clustering)**, Degree of Membership (DoM) của một đối tượng là:

A. Luôn bằng 0 hoặc 1.

B. Một giá trị thực trong khoảng $[0, 1]$.

C. Một giá trị nguyên dương.

D. Luôn bằng $1/k$ với k là số cụm.

Câu 35. Điểm khác biệt chính giữa **phân cụm cứng** và **phân cụm mờ** là:

A. Phân cụm cứng không cần xác định số cụm k.

B. Trong phân cụm mờ, một đối tượng có thể thuộc nhiều cụm với mức độ thành viên khác nhau.

C. Phân cụm cứng xử lý tốt hơn dữ liệu có ranh giới không rõ ràng.

D. Phân cụm mờ chỉ áp dụng được với dữ liệu nhị phân.

Câu 36. Trong đánh giá kết quả phân cụm, **Entropy** càng nhỏ thì:

A. Cụm càng không thuần, kết quả càng tệ.

B. Cụm càng thuần, kết quả phân cụm càng tốt.

C. Số cụm càng ít.

D. Khoảng cách liên cụm càng nhỏ.

Câu 37. **Silhouette index** được dùng trong:

A. External validation.

B. Internal validation.

C. Relative validation.

D. Không dùng trong đánh giá phân cụm.

Câu 38. Thuật toán **ROCK** trong phân cụm phân cấp được thiết kế cho:

A. Dữ liệu có thang đo khoảng (interval-scaled).

B. Dữ liệu danh mục/rời rạc (categorical/discrete).

C. Dữ liệu phân phối Gaussian.

D. Dữ liệu nhị phân đối xứng.

Câu 39. Phương pháp nào sau đây **không cần** xác định số cụm k trước?

A. K-means.

B. PAM (k-medoids).

C. EM algorithm.

D. DBSCAN.

Câu 40. Phương pháp phân cụm nào phù hợp nhất để phát hiện cụm có hình dạng bất kỳ (arbitrary shapes)?

A. K-means.

B. K-medoids (PAM).

C. DBSCAN.

D. EM algorithm.

Câu 41. Chuẩn hóa Z-score trong xử lý thuộc tính có thang đo khoảng được thực hiện bằng công thức:

A. $z_{if} = \dfrac{x_{if}}{m_f}$

B. $z_{if} = \dfrac{x_{if} - m_f}{s_f}$

C. $z_{if} = x_{if} - \min_f$

D. $z_{if} = \dfrac{x_{if} - \min_f}{\max_f - \min_f}$

ĐÁP ÁN

Câu hỏi tự luận – Hướng dẫn trả lời

Câu	Nội dung cần trình bày
1	Phân cụm là nhóm các đối tượng sao cho nội cụm tương đồng cao, liên cụm tương đồng thấp. Ứng dụng: phân nhóm khách hàng, phân tích mạng xã hội, phát hiện bất thường.
2	Ma trận dữ liệu: $n \times p$ (đối tượng $\times$ thuộc tính). Ma trận bất tương đồng: $n \times n$ lưu $d(i,j)$. Điều kiện: $d(i,i) = 0$; $d(i,j) = d(j,i) \geq 0$; bất đẳng thức tam giác.
3	Euclidean ($q = 2$), Manhattan ($q = 1$), Minkowski (q tùy ý). Nên chuẩn hóa (Z-score) khi các thuộc tính có đơn vị và biên độ khác nhau để tránh thiên vị.
4	Simple distance (đối xứng) dùng khi cả hai giá trị 0 và 1 đều có ý nghĩa như nhau (ví dụ: giới tính). Jaccard (bất đối xứng) dùng khi giá trị "0–0" không có ý nghĩa (ví dụ: xét nghiệm y tế).
5	(1) Chọn k tâm ngẫu nhiên; (2) Gán đối tượng vào cụm gần nhất; (3) Cập nhật tâm cụm = mean; (4) Lặp đến khi tâm không đổi. Hàm mục tiêu: $\sum_{i=1}^{k} \sum_{x \in C_i} \text{dist}(x, r_i)^2$.
6	Ưu: đơn giản, hiệu quả $O(nkt)$. Nhược: tối ưu cục bộ, nhạy cảm nhiễu, chỉ tạo cụm dạng cầu, cần chọn k. K-means không phù hợp non-convex vì tâm cụm là trung bình – không nắm bắt được hình dạng phức tạp.
7	AGNES: bottom-up, gộp từng bước. DIANA: top-down, tách từng bước. Ưu: không cần k, kết quả trực quan qua dendrogram. Nhược: không quay lại được, khó mở rộng.
8	Single-linkage: khoảng cách ngắn nhất, có thể tạo chuỗi dài. Complete-linkage: khoảng cách dài nhất, tạo cụm compact hơn.
9	Dendrogram là sơ đồ cây biểu diễn quá trình gộp/tách cụm. Cắt ngang dendrogram ở ngưỡng tương đồng nhất định để chọn số cụm.
10	ε-neighborhood: vùng bán kính ε. Core: $\lvert\varepsilon\text{-nhbd}\rvert \geq$ MinPts. Directly density-reachable: bất đối xứng. Density-reachable: bắc cầu, bất đối xứng. Density-connected: đối xứng.
11	DBSCAN: (1) xác định ε-neighborhood; (2) tạo cụm từ core object; (3) mở rộng theo density-reachable. Border: trong nhbd của core nhưng không phải core. Noise: không thuộc cụm nào. ε nhỏ $\rightarrow$ nhiều nhiễu, ε lớn $\rightarrow$ ít cụm.

Câu	Nội dung cần trình bày
12	DBSCAN phát hiện hình dạng tùy ý, xử lý nhiễu tốt, không cần k; $O(n\log n)$–$O(n^2)$. K-means: chỉ hình cầu, nhạy nhiễu, cần k; $O(nkt)$.
13	Model-based: tối ưu hóa khớp dữ liệu với mô hình phân phối xác suất (Gaussian). EM là thuật toán iterative tìm ML. Ưu: soft assignment, tổng quát hơn K-means.
14	E-step: tính $P(x_i \in C_j \mid x_i)$ (xác suất thuộc cụm). M-step: cập nhật μ_j, σ_j, p_j để maximize log-likelihood. Khởi tạo: random. Dừng: khi log-likelihood hội tụ.
15	Hard clustering: DoM $\in \{0,1\}$, ranh giới rõ. Fuzzy: DoM $\in [0,1]$, ranh giới mờ. Dùng fuzzy khi dữ liệu có ranh giới tự nhiên không rõ ràng (ví dụ: phân loại xe theo tốc độ và trọng lượng).
16	Scalability, multi-type data, arbitrary shapes, ít tham số, xử lý nhiễu, incremental, interpretability. K-means: scalability, ít tham số. DBSCAN: arbitrary shapes, nhiễu. Hierarchical: không cần k. EM: interpretability.
17	External: so với nhãn thực (Rand, Jaccard, Folkes-Mallows). Internal: dựa trên proximity matrix (Silhouette, Dunn's index). Relative: so sánh hiệu quả giữa các phương pháp.
18	Entropy $= -\sum_i \frac{n_i}{n} \sum_j \frac{n_{ij}}{n_i} \log \frac{n_{ij}}{n_i}$. Entropy nhỏ = cụm thuần. Ví dụ: cụm có 12/12 đối tượng cùng lớp $\rightarrow$ entropy = 0 (hoàn hảo).
19	$d(i,j) = \frac{\sum_f \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_f \delta_{ij}^{(f)}}$. Khi x_{if} hoặc x_{jf} bị thiếu thì $\delta_{ij}^{(f)} = 0$ (bỏ qua thuộc tính đó).
20	Phân cụm phân cấp khó mở rộng vì mỗi bước gộp phải đánh giá nhiều đối tượng. Giải pháp: divide-and-conquer, BIRCH – dùng Clustering Feature (CF) tree để tóm tắt dữ liệu, cho phép phân cụm một lần qua (single scan) với bộ nhớ hạn chế.

Câu hỏi trắc nghiệm – Đáp án

Câu	ĐA	Câu	ĐA	Câu	ĐA	Câu	ĐA
1	B	11	C	21	B	31	B
2	B	12	B	22	C	32	B
3	D	13	A	23	B	33	C
4	B	14	B	24	B	34	B
5	C	15	B	25	B	35	B

Câu	ĐA	Câu	ĐA	Câu	ĐA	Câu	ĐA
6	B	16	A	26	B	36	B
7	B	17	B	27	B	37	B
8	B	18	C	28	B	38	B
9	B	19	A	29	B	39	D
10	C	20	B	30	A	40	B

Giải thích đáp án trắc nghiệm

Câu	Giải thích
1	(B) – Mục tiêu phân cụm: tối thiểu hóa khoảng cách nội cụm, tối đa hóa khoảng cách liên cụm.
2	(B) – Phân cụm là học không giám sát vì không có nhãn lớp trong quá trình học.
3	(D) – Điều kiện $d(i,j) = d(i,k) \cdot d(k,j)$ không phải điều kiện của ma trận bất tương đồng (đúng là bất đẳng thức tam giác).
4	(B) – Minkowski với $q = 1$ là Manhattan distance.
5	(C) – Minkowski với $q = 2$ là Euclidean distance.
6	(B) – Jaccard distance (asymmetric): $d(i,j) = (b+c)/(a+b+c)$, bỏ qua trường hợp cả hai bằng 0.
7	(B) – Cosine similarity $= 1$ khi hai vector cùng hướng (hoàn toàn giống nhau về hướng).
8	(B) – Internal validation dựa trên proximity matrix tính từ chính tập dữ liệu.
9	(B) – K-means dừng khi tâm cụm không thay đổi hoặc thay đổi dưới ngưỡng.
10	(C) – $O(nkt)$: n đối tượng, k cụm, t vòng lặp.
11	(C) – K-means nhạy cảm với nhiễu và outliers vì dùng mean.
12	(B) – K-means tạo cụm dạng hình cầu với kích thước tương đối đều nhau.
13	(A) – PAM dùng medoid (đối tượng thực) thay vì mean, bền hơn với nhiễu.
14	(B) – Hàm mục tiêu K-means: tổng bình phương khoảng cách tới tâm cụm.
15	(B) – AGNES bắt đầu với mỗi đối tượng là một cụm riêng, sau đó gộp dần.
16	(A) – Hierarchical không cần xác định k trước.

Câu	Giải thích
17	(B) – Single-linkage dùng khoảng cách ngắn nhất giữa hai đối tượng trong hai cụm.
18	(C) – Complete-linkage dùng khoảng cách dài nhất (maximum distance).
19	(A) – Dendrogram biểu diễn quá trình phân cụm phân cấp dạng cây.
20	(B) – BIRCH giải quyết vấn đề khả năng mở rộng (scalability) của phân cụm phân cấp.
21	(B) – Core object: số đối tượng trong ε-neighborhood $\geq$ MinPts.
22	(C) – Noise: đối tượng không thuộc bất kỳ cụm nào.
23	(B) – Density-reachable là quan hệ bất đối xứng (asymmetric).
24	(B) – Density-connected là quan hệ đối xứng (symmetric).
25	(B) – DBSCAN phát hiện cụm hình dạng tùy ý và xử lý nhiễu hiệu quả.
26	(B) – Độ phức tạp tốt nhất của DBSCAN là $O(n \log n)$.
27	(B) – OPTICS sắp xếp các điểm để xác định cấu trúc phân cụm.
28	(B) – Tăng ε: vùng lân cận rộng hơn $\rightarrow$ nhiều đối tượng được gộp $\rightarrow$ số cụm giảm, số nhiễu giảm.
29	(B) – Model-based giả định dữ liệu từ hỗn hợp các mô hình phân phối xác suất.
30	(A) – EM E-step: ước lượng (tính) xác suất thuộc cụm của mỗi đối tượng.
31	(B) – EM M-step: cực đại hóa log-likelihood bằng cách cập nhật tham số.
32	(B) – EM dùng soft assignment (gán mềm), K-means dùng hard assignment.
33	(C) – EM dừng khi đạt điều kiện hội tụ (log-likelihood không tăng đáng kể).
34	(B) – Fuzzy clustering: DoM $\in [0,1]$, một đối tượng có thể thuộc nhiều cụm.
35	(B) – Fuzzy: đối tượng thuộc nhiều cụm với mức độ thành viên khác nhau.
36	(B) – Entropy nhỏ $\rightarrow$ cụm thuần nhất (objects trong cùng cụm thuộc cùng lớp thực).
37	(B) – Silhouette index là độ đo internal validation.
38	(B) – ROCK xử lý dữ liệu thuộc tính danh mục/rời rạc.
39	(D) – DBSCAN không cần xác định k trước; K-means, PAM, EM đều cần k.
40	(B) – Z-score: $z_{if} = (x_{if} - m_f)/s_f$ với m_f là trung bình, s_f là độ lệch tuyệt đối trung bình.