VNUHCM - UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY

KNOWLEDGE ENGINEERING DEPARTMENT

Report Lab 3

Project 2: INTRODUCTION TO OPENSSL

Course: Introduction to Cryptography

Students:
Lê Trường Thịnh (23127018)
Trần Lý Nhật Hào (23127187)

Instructor:
Trịnh Văn Minh

Ngày 14 tháng 1 năm 2026

Mục lục

1 Giới thiệu

Phân cụm là bài toán học không giám sát đã được nghiên cứu sâu rộng trong nhiều thập kỷ qua. Trong số các mục tiêu phân cụm khác nhau, một trong những công thức phổ biến nhất là phân cụm k-means. Trong phân cụm k-means, bài toán được cho là một tập P các điểm dữ liệu trong không gian Euclid d chiều, và mục tiêu là tính toán một tập $C \subset \mathbb{R}^d$ các tâm với kích thước tối đa k sao cho tổng bình phương khoảng cách từ các điểm dữ liệu trong P đến các tâm gần nhất trong C là nhỏ nhất. Bài toán k-means rất được quan tâm và được chứng minh là NP-khó Dasgupta (2008). Hơn nữa, kết quả các nghiên cứu chỉ ra rằng ngay cả việc tìm một nghiệm cho bài toán k-means với tỷ lệ xấp xỉ nhỏ hơn 1.07 cũng là NP-khó Cohen-Addad and Karthik (2019). Tỷ lệ xấp xỉ tốt nhất hiện nay cho bài toán k-means là 5.912 Cohen-Addad et al. (2022), dựa trên các phương pháp đối ngẫu nguyên thủy (primal-dual) và tập độc lập tựa lồng nhau (nested quasi-independent set). Đối với số chiều d hoặc số cụm k cố định, đã có một số thuật toán xấp xỉ $(1+\varepsilon)$ như Jaiswal et al. (2014); Friggstad et al. (2019). Tuy nhiên, các thuật toán này với ngưỡng xấp xỉ rất chặt thường không mở rộng (scale) tốt cho các tập dữ liệu quy mô lớn. Do đó, nhiều phương pháp thực tiễn với thời gian tuyến tính đã được đề xuất, chẳng hạn như phương pháp k-means++ với tỷ lệ xấp xỉ $O(\log k)$ Arthur and Vassilvitskii (2007) và các phương pháp local search xấp xỉ hệ số hằng số Lattanzi and Sohler (2019). Mặc dù các thuật toán thời gian tuyến tính này được sử dụng rộng rãi, tỷ lệ xấp xỉ lớn của chúng có thể làm giảm hiệu suất phân cụm trong các kịch bản đòi hỏi nghiệm phải rất tốt.

Để vượt qua rào cản không thể xấp xỉ và phát triển các thuật toán thực tiễn hơn, một loạt các nghiên cứu đã tập trung vào các thuật toán có hỗ trợ học (learning-augmented) Mitzenmacher and Vassilvitskii (2022). Đối với bài toán phân cụm, Gamlath và tác giả cộng sự Gamlath et al. (2022) đã đề xuất khôi phục phân cụm với nhãn nhiễu, trong đó các nhãn phân cụm dự đoán là thông tin bổ trợ. Bộ dự đoán có tham số tỷ lệ lỗi $\alpha \in [0,1)$, sao cho kích thước của hiệu đối xứng giữa cụm dự đoán và cụm tối ưu tương ứng được chặn bởi α lần kích thước cụm tối ưu. Dựa trên mô hình này, người ta đề ra một thuật toán xấp xỉ $(1+O(\alpha))$ với thời gian chạy đa thức khi giả định k và d cố định. Ergun và tác giả cộng sự Ergun et al. (2021) đã giới thiệu một mô hình phân cụm có hỗ trợ học khác, hướng đến việc thiết kế các thuật toán nhanh và thực tiễn. Trong mô hình này, bộ dự đoán cung cấp thông tin cho mỗi điểm dữ liệu dưới dạng nhãn dự đoán với độ tin cậy α, đảm bảo rằng có tối đa một phần α dương tính giả và âm tính giả trong mỗi cụm dự đoán. Dựa trên mô hình này, một cải tiến xấp xỉ $(1+O(\alpha))$ có thể đạt được với thời gian chạy gần tuyến tính.

Trong bài báo này, tác giả tập trung chủ yếu vào bài toán phân cụm có hỗ trợ học (learning-augmented) được đề xuất bởi Ergun và tác giả cộng sự Ergun et al. (2021). Động lực nghiên cứu bài toán này đến từ hai phía. Về mặt lý thuyết, k-means có hỗ trợ học có thể vượt qua rào cản không thể xấp xỉ, cho ra ngưỡng chất lượng rất chặt cùng khả năng mở rộng cao. Về mặt thực tế, các bộ dự đoán đáng tin cậy luôn có cho nhiều loại dữ liệu tự nhiên. Ví dụ, nhãn huấn luyện có thể đóng vai trò thông tin bổ trợ để tăng cường chất lượng phân cụm trên tập kiểm tra. Ngay cả khi không có nhãn, thực nghiệm cho thấy nhãn từ các phương pháp hiện có như k-means++ Arthur and Vassilvitskii (2007) hoặc các phương pháp heuristic như Lloyd Lloyd (1982) cũng có thể là bộ dự đoán tốt. Tuy nhiên, như đã chỉ ra bởi Nguyen và tác giả cộng sự Nguyen et al. (2022), ngay cả khi bộ dự đoán gần như tối ưu, chỉ cần một điểm dương tính giả nằm xa các tâm tối ưu

cũng có thể ảnh hưởng nghiêm trọng đến cấu trúc phân cụm. Do đó, thách thức then chốt là thiết kế các thuật toán bền vững để giảm thiểu tác động của dương tính giả.

Dựa trên các phương pháp thống kê, Ergun và tác giả cộng sự Ergun et al. (2021) đề xuất thuật toán ngẫu nhiên đạt mức xấp xỉ $(1+20\alpha)$ trong thời gian $O(md\log m)$. Tuy nhiên, thuật toán này yêu cầu các điều kiện khắt khe về tỷ lệ lỗi α và kích thước cụm tối ưu. Để khắc phục, Nguyen và tác giả cộng sự Nguyen et al. (2022) đề xuất phương pháp tìm kiếm xác định đạt kết quả xấp xỉ $(1+O(\alpha))$ tốt hơn trong thời gian $O(md\log m)$ với $\alpha \in [0,1/2)$. Các thuật toán hiện tại chủ yếu dựa trên chiến lược sắp xếp để xấp xỉ tâm tối ưu. Vì sắp xếp yêu cầu thời gian logarit và có cận dưới là $O(m\log m)$, điều này hạn chế khả năng mở rộng khi xử lý dữ liệu quy mô cực lớn. Ngoài ra, thời gian chạyđ này không thể cải thiện thông qua các kỹ thuật giảm chiều như phương pháp JL vì bản thân việc chiếu cũng tốn ít nhất $O(md\log m)$. Thách thức trung tâm trong việc thiết kế các thuật toán nhanh hơn là xấp xỉ hiệu quả các tâm tối ưu trong từng chiều mà không cần sử dụng các chiến lược dựa trên sắp xếp.

Đằng sau các thuật toán: Việc sử dụng sắp xếp trong các nghiên cứu trước đây nhằm xác định vị trí các điểm lân cận để lọc, nhưng điều này dẫn đến chi phí nhân với $\log m$. Mục tiêu của tác giả trong bài báo này là thay thế việc sắp xếp bằng các kỹ thuật lấy mẫu (sampling) và ước lượng (estimation). Bằng cách này, tác giả có thể xác định các khoảng chứa tâm tối ưu và tinh chỉnh chúng trong thời gian tuyến tính $O(md)$. Điều này cho phép thuật toán phá vỡ rào cản tính toán của các phương pháp dựa trên sắp xếp truyền thống mà vẫn giữ được độ chính xác xấp xỉ trong phạm vi sai số dự đoán α.

Bảng 1: Kết quả so sánh các thuật toán k-means có hỗ trợ học

Phương pháp và Tài liệu tham khảo	Tỷ lệ xấp xỉ	Khoảng lỗi nhãn α	Độ phức tạp thời gian
Phân vùng và Sắp xếp Ergun et al. (2021)	$1+20\alpha$	$[\frac{10\log m}{\sqrt{m}}, 1/7]$	$O(md\log m)$
Sắp xếp Nguyen et al. (2022)	$1+\frac{\alpha}{1-\alpha}+\frac{4\alpha}{(1-2\alpha)(1-\alpha)}$	$[0,1/2)$	$O(md\log m)$
Fast-Sampling (Tác giả)	$1+\frac{\alpha}{1-\alpha}+\frac{4\alpha+\alpha\varepsilon}{(1-2\alpha)(1-\alpha)}$	$[0,1/2)$	$O(\varepsilon^{-1}md\log(kd))$
Fast-Estimation (Tác giả)	$1+\frac{\alpha}{1-\alpha}+\frac{13\alpha-15\alpha^2}{(1-3\alpha-\varepsilon)(1-2\alpha-\varepsilon)}$	$(0,1/3-\varepsilon)$	$O(md)+\tilde{O}(\varepsilon^{-5}kd/\alpha)$
Fast-Filtering (Tác giả)	$1+6\sqrt{\alpha}+\frac{36(\sqrt{\alpha}+\alpha)}{1-(3+\varepsilon)\alpha}+\frac{9(\sqrt{\alpha}+\alpha)}{(1-\alpha)(1-(3+\varepsilon)\alpha)}$	$O(md)+\tilde{O}(\varepsilon^{-5}kd/\alpha)$	

Nếu so sánh tỷ lệ xấp xỉ theo lý thuyết, các thuật toán của tác giả không tốt hơn so với các thuật toán sắp xếp.

Tuy nhiên ở phần thực nghiệm Fast-Filtering cho thấy chi phí phân cụm giảm đi 1.5%. Điều này có thể do mặc dù theo lí thuyết là tệ nhưng trong thực tế tỷ lệ xấp xỉ thường nhỏ hơn (tốt hơn).

2 Các công trình liên quan

Thuật toán k-means++ có tỷ lệ xấp xỉ $\Theta(\log k)$ so với đáp án phân cụm tối ưu, đồng thời cải thiện tốc độ, độ chính xác (accuracy) nếu biết nhãn. Mặc dù các thuật toán thời gian tuyến tính như k-means++ được sử dụng rộng rãi, nhưng không phù hợp để dùng nếu cần độ chính xác cao (như kết quả thực nghiệm của bài

báo này). Thuật toán chỉ khác k-means ở bước đầu "seeding"(chọn các tâm ban đầu), việc chọn các tâm là lấy mẫu D^2 (D^2-sampling):

1. Chọn tâm cụm đầu tiên c_1 ngẫu nhiên từ tập dữ liệu P.
2. Chọn tâm cụm tiếp theo c_i, chọn $c_i = x$ với xác suất $P(x) \propto D(x)^2$:

$$P(x) = \frac{D(x)^2}{\sum_{y \in P} D(y)^2}$$

3. Chạy thuật toán k-means.

2.1 Các thuật toán dựa trên sắp xếp

- **Thuật toán của Ergun và cộng sự Ergun et al. (2021):** Phương pháp ngẫu nhiên này đạt được tỷ lệ xấp xỉ $(1+20\alpha)$ trong thời gian $O(md\log m)$.
- **Thuật toán của Nguyen và cộng sự Nguyen et al. (2022):** Tác giả đề xuất phương pháp tìm kiếm định tính để đạt được tỷ lệ xấp xỉ cải tiến $(1+O(\alpha))$ cho $\alpha \in [0, 1/2)$ trong cùng thời gian $O(md\log m)$.

Ngoài ra trong 2 bài báo trên, các tác giả còn đề xuất thuật toán k-medians. Các phương pháp này đều cần phí sắp xếp $O(\log m)$

2.2 Clustering tương tác

Giống với mô hình của bài báo này yêu cầu có 1 oracle cho biết các nhãn hay các cụm dự đoán P_i. Thuật toán có thể đưa ra các câu hỏi dạng: "Điểm p và điểm q có thuộc cùng một cụm tối ưu hay không?"Balcan and Blum (2008). Một hướng tiếp cận nâng cao khác là phân cụm phân cấp Bayesian có tương tác Vikram and Dasgupta (2016)

3 Kiến thức cơ sở

Gọi $P \subset \mathbb{R}^d$ và k lần lượt là tập dữ liệu và số lượng cụm. Gọi m là kích thước của tập dữ liệu. Đối với hai điểm $p, q \in \mathbb{R}^d$ bất kỳ, ký hiệu $\delta(p,q)$ và $\delta^2(p,q)$ lần lượt là khoảng cách và bình phương khoảng cách giữa chúng. Cho một điểm $p \in \mathbb{R}^d$ và một tập các tâm $C = \{c_1, c_2, \ldots, c_k\}$, gọi $\delta(p,C) = \min_{c \in C} \delta(p,c)$ là khoảng cách từ p đến tâm gần nhất trong C.

Để ý có thể có nhiều cách phân cụm tối ưu nhưng ở đây tác giả chọn 1 cách cố định để phân tích.

Gọi $C^* = \{c_1^*, \ldots, c_k^*\}$ và $P(C^*) = \{P_1^*, \ldots, P_k^*\}$ là tập các tâm tối ưu và phân hoạch phân cụm tối ưu tương ứng. Mỗi tâm tối ưu $c_i^* \in C^*$ được biểu diễn bởi d tọa độ, tức là $c_i^* = (c_{i1}^*, c_{i2}^*, \ldots, c_{id}^*)$. Chi phí phân cụm

của tập P đối với tập tâm C được định nghĩa là:

$$\delta^2(P,C) = \sum_{x \in P} \delta^2(x,C) \tag{1}$$

Cho một tập hợp $L(P) = \{P_1, P_2, \ldots, P_k\}$ đóng vai trò là bộ dự đoán, gọi $Q_i = P_i \cap P_i^*$ là tập các điểm dữ liệu trong cụm dự đoán P_i thuộc cụm tối ưu P_i^*. Gọi tọa độ chiếu của các điểm dữ liệu trong P_i và Q_i lên chiều thứ j lần lượt là P_{ij} và Q_{ij}. Gọi P_{ij}^* là tọa độ chiếu của các điểm trong P_i^* lên chiều thứ j. Gọi m_i và m lần lượt là kích thước của P_i và P. Với một tập điểm dữ liệu $V \subset \mathbb{R}^d$, gọi $\overline{V}$ là tâm hình học của tập V. Gọi $P(j)$ là tọa độ chiếu của toàn bộ các điểm trong P lên chiều thứ j. Gọi Δ_{max} là tỷ lệ chiều tối đa của các điểm dữ liệu được chiếu, được xác định bởi:

$$\Delta_{max} = \max_{1 \le j \le d} \frac{\max_{x,y \in P(j)} \delta(x,y)}{\min_{x,y \in P(j), x \neq y} \delta(x,y)} \tag{2}$$

Cụm từ "Aspect Ratio" được tác giả đề cập khi dịch về tiếng Việt để quen thuộc nhất thì chúng em sẽ gọi là **tỷ lệ khung hình**. Về bản chất thì cũng chỉ là tỉ lệ giữa khoảng cách lớn nhất của 2 điểm và khoảng cách nhỏ nhất của 2 điểm.

Với một số nguyên dương t, gọi $[t]$ là tập hợp các số nguyên từ 1 đến t.

Bài toán k-means hỗ trợ học: Cho tập dữ liệu $P \subset \mathbb{R}^d$ gồm m điểm, gọi C^* và $P(C^*) = \{P_1^*, P_2^*, \ldots, P_k^*\}$ lần lượt là một lời giải tối ưu và phân hoạch tương ứng. Trong thiết lập có hỗ trợ học, giả định rằng có quyền truy cập vào một bộ dự đoán dưới dạng phân hoạch nhãn $L(P) = \{P_1, P_2, \ldots, P_k\}$ được tham số hóa bởi tỷ lệ lỗi nhãn $\alpha \in [0,1)$, thỏa mãn điều kiện $|P_i \cap P_i^*| \ge (1-\alpha)\max\{|P_i|, |P_i^*|\}$. Mục tiêu của bài toán là tìm tập $C \subset \mathbb{R}^d$ các tâm sao cho $\delta^2(P,C)$ đạt giá trị nhỏ nhất.

Các hệ quả toán học trong phần này dựa trên tính chất cơ bản của không gian Euclid, trọng tâm $\overline{V}$ là điểm duy nhất tối thiểu hóa tổng bình phương khoảng cách (SSE) tới mọi điểm trong tập V. Logic của các bổ đề dưới đây cho phép phân rã chi phí phân cụm thành hai thành phần: chi phí trong cụm (độ liên kết - cohesion) và chi phí do khoảng cách từ tâm dự đoán đến tâm tối ưu.

Các bổ đề dưới đây là "dân gian truyền miệng"(folklore) rất phổ biến liên quan bài toán k-means clustering

Lemma 1. *Cho tập $X \subset \mathbb{R}^d$ có kích thước m và một điểm dữ liệu bất kỳ $c \in \mathbb{R}^d$, ta luôn có:*

$$\delta^2(X,c) = \delta^2(X,\overline{X}) + m \cdot \delta^2(c,\overline{X}) \tag{3}$$

Arthur and Vassilvitskii (2007)

Bổ đề 1 xuất phát từ quan sát trọng tâm $\overline{P_i}$ của các cụm dự đoán không đủ là nghiệm của bài toán. Vì dự đoán không đúng, có thể tồn tại 1 số điểm trong P_i nằm ngoài P_i^*. Nếu các điểm trong $P_i \setminus P_i^*$ nằm **rất xa** $\overline{P_i^*}$, trọng tâm cụm dự đoán sẽ bị lệch tuỳ ý dẫn đến chi phí tăng cụm tăng lên tuỳ ý.

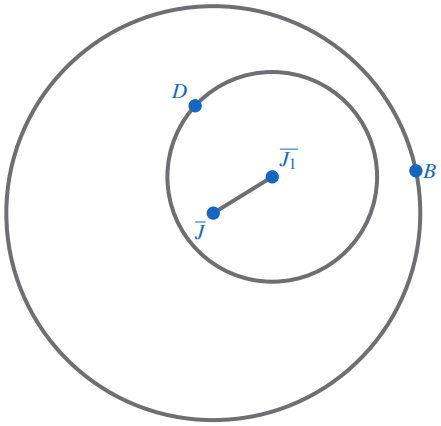
Lemma 2. *Cho tập $J \subset \mathbb{R}$, gọi $J_1 \subseteq J$ với $|J_1| \ge (1-\zeta)|J|$, trong đó $0 \le \zeta < 1$. Khi đó, mối liên hệ giữa*

chi phí của tập con và tập tổng thể được chặn bởi:

$$\delta^2(\bar{J},\overline{J_1}) \leq \frac{\zeta}{(1-\zeta)|J|}\delta^2(J,\bar{J}) \tag{4}$$

Nguyen et al. (2022)

Bổ đề 2 cũng đúng với $J \subset \mathbb{R}^d$, mặc dù tác giả chỉ ghi trên $\mathbb{R}$, xem chứng minh trong bài báo của Nguyen và tác giả cộng sự (Nguyen et al. (2022), phần chứng minh Bổ đề 3 trong mục Appendix 5.1).



Hình 1: Minh họa Bổ đề 2 trong không gian $\mathbb{R}^d$: Khoảng cách giữa tâm tập tổng thể $\bar{J}$ và tâm tập con $\overline{J_1}$ được chặn bởi hàm chi phí của J.

Bổ đề 2 xuất phát từ quan sát liên hệ chi phí và kích thước tập con của cụm tối ưu. Ta muốn tìm $Q_i = P_i \cap P_i^*$ và lấy $\overline{Q_i}$ làm đáp án cho cụm i, điều này tự nhiên xuất phát từ dữ kiện Q_i của bài toán có hỗ trợ học.

$$\begin{aligned} |Q_i| &\geq (1-\alpha)\max\{|P_i|,|P_i^*|\} \geq (1-\alpha)|P_i^*| \\ \Rightarrow |P_i^* \setminus Q_i| &\leq \alpha m_i^* \end{aligned}$$

Dùng bổ đề trên, ta có chặn trên chi phí phân cụm:

$$\begin{aligned} \delta^2(P_i^*,\overline{Q_i}) &= \delta^2(P_i^*,\overline{P_i^*}) + m_i^*\delta^2(\overline{P_i^*},\overline{Q_i}) \text{ (Bổ đề 1)} \\ &\leq \delta^2(P_i^*,\overline{P_i^*}) + m_i^*\frac{\alpha}{1-\alpha}\frac{\delta^2(P_i^*,\overline{P_i^*})}{m_i^*} \text{ (Bổ đề 2)} \\ &= \left(1+\frac{\alpha}{1-\alpha}\right)\delta^2(P_i^*,\overline{P_i^*}) \end{aligned}$$

Như vậy ta có xấp xỉ $\left(1+\frac{\alpha}{1-\alpha}\right)$ cho 1 cụm và cũng như cho bài toán. Cũng chính là chặn dưới và lí do xuất hiện của nó trong tỷ lệ xấp xỉ trong Bảng 1.

Khó khăn là Q_i chưa biết, vì vậy các thuật toán chính dưới đây tập trung vào việc loại bỏ các điểm outlier trong P_i, tìm một trọng tâm gần $\overline{Q_i}$, như vậy đồng thời giảm được khoảng cách đến $\overline{P_i^*}$ và chi phí.

Lemma 3. *Cho tập $X \subset \mathbb{R}^d$ và một giá trị $\alpha \in (0,1]$, gọi $X' = \arg\min_{X'' \subseteq X, |X''|=\alpha|X|} \delta^2(X'', \overline{X''})$. Khi đó, ta có:*

$$\delta^2(X', \overline{X'}) \leq \alpha \cdot \delta^2(X, \overline{X}) \tag{5}$$

Nguyen et al. (2022)

Kết quả của Bổ đề 3 được chứng minh bằng phương pháp xác suất (probabilistic method), xem chi tiết tại phần đầu trong chứng minh của Bổ đề 5 (Appendix 5.1) trong bài báo của Nguyen và tác giả cộng sự Nguyen et al. (2022).

3.1 Các công cụ (toán)

Background Theorem 1 (Bất đẳng thức tam giác nới lỏng)**.** *Với mọi số thực $a, b \in \mathbb{R}$ và một tham số dương $\lambda > 0$, bất đẳng thức sau luôn thỏa mãn:*

$$(a+b)^2 \leq \left(1+\frac{1}{\lambda}\right) a^2 + (1+\lambda) b^2$$

Trong không gian vector $\mathbb{R}^d$ với chuẩn Euclid $\|\cdot\|$, bất đẳng thức này tương đương với:

$$\|u+v\|^2 \leq \left(1+\frac{1}{\lambda}\right) \|u\|^2 + (1+\lambda)\|v\|^2$$

Chứng minh. Ở đây nhóm em chứng minh cho 1 chiều, còn lại cũng tương tự. Ta bắt đầu bằng việc khai triển vế trái của bất đẳng thức:

$$(a+b)^2 = a^2 + 2ab + b^2$$

Ta áp dụng bất đẳng thức AM-GM. Với hai số thực dương x, y, ta luôn có $x^2 + y^2 \geq 2xy$. Chọn $x = \frac{a}{\sqrt{\lambda}}$ và $y = b\sqrt{\lambda}$. Khi đó:

$$\left(\frac{a}{\sqrt{\lambda}}\right)^2 + (b\sqrt{\lambda})^2 \geq 2\left(\frac{a}{\sqrt{\lambda}}\right)(b\sqrt{\lambda})$$

$$\frac{a^2}{\lambda} + \lambda b^2 \geq 2ab$$

Thay thế chặn trên của $2ab$ vào khai triển ban đầu của $(a+b)^2$:

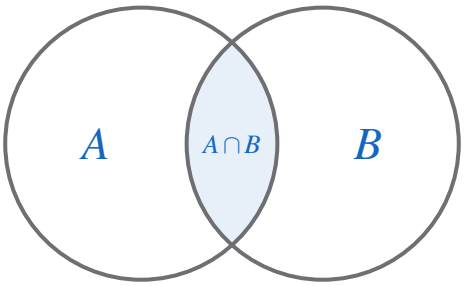
$$\begin{aligned}(a+b)^2 &= a^2 + 2ab + b^2 \\ &\leq a^2 + \left(\frac{a^2}{\lambda} + \lambda b^2\right) + b^2 \\ &= \left(a^2 + \frac{a^2}{\lambda}\right) + (b^2 + \lambda b^2) \\ &= \left(1 + \frac{1}{\lambda}\right) a^2 + (1+\lambda) b^2\end{aligned}$$

□

Background Theorem 2 (Chặn hợp - Bất đẳng thức Boole)**.** *Nếu $A_1, A_2, \ldots, A_n$ là các sự kiện thì xác suất để ít nhất một sự kiện xảy ra nhỏ hơn tổng xác suất của tất cả các sự kiện.*

$$\Pr\left(\bigcup_{i=1}^{n} A_i\right) \leq \sum_{i=1}^{n} \Pr(A_i)$$

Định lý 2 khá tự nhiên và dễ thấy. Xét ví dụ với hai sự kiện A và B. Xác suất để ít nhất một trong hai sự kiện xảy ra là xác suất của hợp $A \cup B$. Ta có thể biểu diễn mối quan hệ này trên biểu đồ Venn như Hình 2.



Hình 2: Minh họa chặn hợp với hai sự kiện A và B. Phần giao $A \cap B$ (được tô màu) được tính hai lần trong tổng $\Pr(A) + \Pr(B)$.

Từ hình 2, ta thấy diện tích của phần hợp $A \cup B$ bằng tổng diện tích của A và B trừ đi diện tích phần giao nhau $A \cap B$ (vì phần này được tính hai lần trong tổng). Chuyển sang công thức xác suất:

$$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$$

Vì xác suất của phần giao luôn không âm ($\Pr(A \cap B) \geq 0$), ta suy ra bất đẳng thức:

$$\Pr(A \cup B) \leq \Pr(A) + \Pr(B)$$

Mở rộng tương tự cho n sự kiện, ta có định lý chặn hợp (union bound).

Trong bài báo, tác giả dùng định lý 2 để gộp xác suất thất bại trên từng chiều dữ liệu (d) và trên từng cụm (k), đảm bảo rằng thuật toán Fast-Sampling vẫn đúng trên toàn bộ không gian dữ liệu.

Background Theorem 3 (Chặn Chernoff)**.** *Cho $X_1, X_2, \ldots, X_n$ là các biến ngẫu nhiên độc lập nhận giá trị trong khoảng $[0,1]$. Gọi $X = \sum_{i=1}^{n} X_i$ và $\mu = \mathbb{E}[X]$.*

1. *Với mọi $\delta > 0$, ta có:*
$$\Pr(X \geq (1+\delta)\mu) \leq \exp\left(-\frac{\delta^2 \mu}{2+\delta}\right)$$
2. *Với mọi $0 < \delta < 1$, ta có:*
$$\Pr(X \leq (1-\delta)\mu) \leq \exp\left(-\frac{\delta^2 \mu}{2}\right)$$

Trong bối cảnh của bài báo này (và các thuật toán ngẫu nhiên nói chung), chặn Chernoff cực kỳ quan trọng để chứng minh tính hiệu quả của phương pháp lấy mẫu (sampling). Cụ thể, các thuật toán được đề xuất (như Fast-Sampling) không duyệt qua toàn bộ dữ liệu mà chỉ lấy một tập mẫu ngẫu nhiên S. Chúng ta cần đảm bảo rằng các đặc tính của mẫu S (ví dụ: tỷ lệ các điểm thuộc cụm tối ưu) xấp xỉ tốt các đặc tính của tập dữ liệu gốc P. Chặn Chernoff cho ta biết xác suất để giá trị thực nghiệm lệch khỏi giá trị kỳ vọng sẽ giảm theo hàm mũ. Điều này đảm bảo rằng chỉ cần kích thước mẫu $O(\log m)$, ta có thể đạt được độ chính xác mong muốn với xác suất rất cao (high probability).

Background Theorem 4 (Bất đẳng thức Cauchy-Schwarz)**.** *Với mọi u, v thuộc không gian tích vô hướng thực, ta có:*
$$|\langle u, v \rangle|^2 \leq \langle u, u \rangle \cdot \langle v, v \rangle$$

Hay viết dưới dạng chuẩn (norm) trong không gian Euclid $\mathbb{R}^d$:

$$|\langle u, v \rangle| \leq \|u\| \cdot \|v\|$$

Dấu đẳng thức xảy ra khi và chỉ khi u và v phụ thuộc tuyến tính (cùng phương).

Bất đẳng thức này thường xuyên được sử dụng trong các chứng minh liên quan đến k-means để tách và đánh giá các thành phần của hàm chi phí. Ví dụ, khi khai triển $\delta^2(x,c) = \|x-c\|^2 = \|(x-c^*)+(c^*-c)\|^2$, ta sẽ gặp số hạng chéo $2\langle x-c^*, c^*-c\rangle$. Cauchy-Schwarz giúp chặn trên giá trị tuyệt đối của số hạng này, từ đó giúp thiết lập các mối quan hệ giữa chi phí của lời giải xấp xỉ và lời giải tối ưu.

3.2 Các thuật toán con

3.2.1 Chọn phần tử hạng thứ *i* trong dãy

Trong các thuật toán được đề xuất, có nhiều bước yêu cầu tìm một giá trị ngưỡng để lọc bớt dữ liệu hoặc xác định khoảng cách (ví dụ: tìm bán kính bao phủ một tỷ lệ nhất định các điểm). Nếu sử dụng thuật toán sắp xếp (Sorting) thông thường (như QuickSort, MergeSort), độ phức tạp sẽ là $O(m \log m)$. Tuy nhiên, để đạt được mục tiêu thiết kế thuật toán chạy trong thời gian tuyến tính $O(m)$, ta cần sử dụng thuật toán chọn

lọc (selection algorithm), cụ thể là thuật toán **Median of Medians** (gốc là thuật toán **PICK** trong bài báo Blum et al. (1973)).

Bài toán: Cho một tập hợp S gồm n phần tử chưa được sắp xếp và một số nguyên i ($1 \leq i \leq n$). Hãy tìm phần tử nhỏ thứ i trong S.

Mô tả thuật toán:

1. **Chia nhóm:** Chia n phần tử của S thành các nhóm có 5 phần tử (nhóm cuối cùng có thể ít hơn 5 phần tử).
2. **Tìm trung vị con:** Với mỗi nhóm, tìm trung vị của nó (bằng cách sắp xếp cục bộ). Tập hợp các trung vị này được gọi là T (kích thước $\approx n/5$).
3. **Tìm chốt (pivot):** Gọi đệ quy thuật toán này để tìm trung vị của tập T. Gọi giá trị tìm được là m (đây chính là "trung vị của các trung vị").
4. **Phân hoạch:** Chia tập S ban đầu thành 3 tập con dựa trên chốt m:
   - $L = \{x \in S \mid x < m\}$ (Các phần tử nhỏ hơn m).
   - $E = \{x \in S \mid x = m\}$ (Các phần tử bằng m).
   - $G = \{x \in S \mid x > m\}$ (Các phần tử lớn hơn m).
5. **Chọn lọc đệ quy:**
   - Nếu $k \leq |L|$: Kết quả nằm trong L. Gọi đệ quy tìm phần tử thứ k trong L.
   - Nếu $|L| < k \leq |L| + |E|$: Trả về m.
   - Nếu $k > |L| + |E|$: Kết quả nằm trong G. Gọi đệ quy tìm phần tử thứ $(k - |L| - |E|)$ trong G.

Giải thích một chút về độ phức tạp: Trong bài báo gốc của Blum và tác giả cộng sự Blum et al. (1973), tác giả chứng minh rằng thuật toán Median of Medians chạy trong thời gian $O(n)$ trong trường hợp xấu nhất. Thuật toán khắc phục vấn đề chọn chốt (pivot) ngẫu nhiên của QuickSelect bằng cách tốn một chút chi phí để chọn ra một chốt m "đủ tốt" (nằm gần giữa dãy số).

- Vì m là trung vị của các trung vị, nó lớn hơn trung vị của một nửa số nhóm.
- Trong mỗi nhóm đó, trung vị lại lớn hơn 2 phần tử khác.
- Suy ra: m chắc chắn lớn hơn khoảng 30% số phần tử của S. Tương tự, m chắc chắn nhỏ hơn khoảng 30% số phần tử.
- Do đó, ở bước phân hoạch, ta luôn loại bỏ được ít nhất 30% dữ liệu. Kích thước bài toán giảm xuống rất nhanh theo cấp số nhân, đảm bảo tổng thời gian là tuyến tính.

Ý nghĩa: Trong các thuật toán đề xuất (Fast-Sampling, Fast-Estimation), tác giả cần liên tục tìm tập hợp "nearest neighbors" (ví dụ: tìm $(1-\alpha)m_i$ điểm gần nhất). Việc sử dụng thuật toán này cho phép tìm ngưỡng khoảng cách trong $O(m)$ thay vì phải sắp xếp toàn bộ dữ liệu tốn $O(m \log m)$. Đây là chìa khóa để

đạt được tốc độ xử lý nhanh trên dữ liệu lớn.

4 Thuật toán Fast-Sampling

Tóm tắt: abc

Ý tưởng tổng quát của thuật toán Fast-Sampling là xấp xỉ hiệu quả các tâm tối ưu trong từng chiều bằng cách xác định các tọa độ chất lượng cao mà không cần sử dụng các chiến lược dựa trên sắp xếp. Thách thức kỹ thuật chính nằm ở việc xử lý các âm tính giả mà không làm ảnh hưởng đáng kể đến các đảm bảo xấp xỉ. Mặc dù việc lấy mẫu trực tiếp một tập con nhỏ các tọa độ từ mỗi chiều của các cụm dự đoán có thể giúp xác định các điểm gần tâm tối ưu, các tọa độ được lấy mẫu theo phân phối đều có thể không xấp xỉ chính xác các tâm tối ưu, tiềm ẩn nguy cơ dẫn đến mất mát hằng số trong các đảm bảo xấp xỉ. Để giải quyết vấn đề này, thuật toán Fast-Sampling trước tiên xác định các tọa độ ứng viên gần với tọa độ của từng tâm tối ưu trong thời gian chạy tuyến tính đối với quy mô dữ liệu. Sau đó, các tọa độ ứng viên được xây dựng sẽ được sử dụng để xác định các khoảng phủ chính xác vị trí của các tâm tối ưu, cho phép xấp xỉ tốt hơn thông qua việc chia nhỏ các khoảng này kĩ (fine-grained).

Thuật toán Fast-Sampling đề xuất chủ yếu bao gồm hai giai đoạn sau: (1) ước lượng khoảng (bước 3-6 của Thuật toán 1); (2) xây dựng tọa độ ứng viên (bước 7 của Thuật toán 1). Trong giai đoạn ước lượng khoảng, đối với mỗi chiều của các cụm dự đoán, độ dài khoảng được ước tính thông qua các chiến lược lấy mẫu ngẫu nhiên. Các mẫu sau đó được điều chỉnh đối xứng (qua tâm) dựa trên các ước tính độ dài khoảng để xây dựng các khoảng có thể bao quanh tọa độ của các tâm tối ưu. Trong giai đoạn thứ hai, các khoảng thu được được chia thành các phần nhỏ hơn, mỗi phần tương ứng với một tọa độ ứng viên mới, cho phép xấp xỉ mịn các tâm tối ưu. Dưới đây là phân tích chi tiết cho thuật toán được đề xuất.

Giải thích thuật toán: Trình tự của Fast-Sampling dựa trên việc khai thác cấu trúc của tập điểm trong từng chiều không gian. Việc lấy mẫu $O(\log(kd))$ điểm đảm bảo rằng với xác suất cao, ít nhất một điểm ứng viên u sẽ rơi vào vùng mật độ cao của cụm tối ưu. Tại bước 6, giá trị l_{ij} được tính toán dựa trên độ lệch chuẩn của tập lân cận gần nhất $\mathcal{N}_{ij}(u)$, đóng vai trò là thước đo khoảng cách đặc trưng để thiết lập lưới tìm kiếm. Kỹ thuật chia lưới trong bước 7 giúp chuyển đổi bài toán tìm kiếm liên tục thành rời rạc với sai số được kiểm soát bởi ε', từ đó tránh được việc phải sắp xếp toàn bộ dữ liệu, giúp duy trì độ phức tạp tuyến tính.

Đầu tiên, tác giả xem xét một chiều đơn lẻ $j \in [d]$ của một cụm dự đoán bất kỳ P_i với $i \in [k]$. Gọi $Q'_{ij} \subseteq Q_{ij}$ là tập hợp các tọa độ có kích thước $(1-\alpha)m_i$ và chi phí phân cụm nhỏ nhất. Bắt đầu từ bước 3 của Thuật toán 1, một tập U_{ij} được xây dựng bằng cách lấy mẫu ngẫu nhiên và độc lập $O(\log(kd))$ mẫu từ P_{ij}. Mục tiêu ở đây là tìm các tọa độ gần với tọa độ của các tâm tối ưu. Tác giả sẽ chứng minh rằng, với xác suất nhất định, tồn tại ít nhất một tọa độ $u \in U_{ij}$ có thể xấp xỉ tốt trọng tâm của Q'_{ij}. Để phân tích xác suất thành công, tác giả định nghĩa G^{μ}_{ij} là tập hợp các tọa độ gần với Q'_{ij}. Bằng cách áp dụng chặn Union Bound cho xác suất thành công trên tất cả các chiều và các cụm dự đoán, tác giả có thể lập luận rằng với xác suất hằng số, tồn tại ít nhất một tọa độ $u \in U_{ij}$ sao cho $u \in G^2_{ij} \cap U_{ij}$.

Thuật toán 1 Fast-Sampling

Đầu vào: Một bài toán k-means (P,k,d), một tập $(P_1,...,P_k)$ các cụm với tỷ lệ lỗi α, và một tham số $\varepsilon \in (0,1]$.

Đầu ra: Một tập $C \subset \mathbb{R}^d$ các tâm với $|C| = k$.

for $i \in [k]$ **do**
  for $j \in [d]$ **do**
    Lấy mẫu ngẫu nhiên và độc lập để tạo tập U_{ij} từ P_{ij} với kích thước $O(\log(kd))$.
    for $u \in U_{ij}$ **do**
      Gọi $\mathcal{N}_{ij}(u)$ là tập $(1-\alpha)|P_i|$ tọa độ trong P_{ij} gần u nhất.
      $l_{ij} = \sqrt{\frac{2\delta^2(\mathcal{N}_{ij}(u), \overline{\mathcal{N}_{ij}(u)})}{(1-\alpha)|P_i|}}$.
      $s(u) = \{u + \varepsilon' \lambda l_{ij} : \lambda \in [-\frac{1}{\varepsilon'}, \frac{1}{\varepsilon'}] \cap \mathbb{Z}\}$, với $\varepsilon' = \sqrt{\frac{\varepsilon}{48}}$.
    $U'_{ij} = \bigcup_{u \in U_{ij}} s(u)$.
    $u_1 = \arg\min_{u \in U'_{ij}} \delta^2(\mathcal{N}_{ij}(u), \overline{\mathcal{N}_{ij}(u)})$.
    $I_{ij} = \mathcal{N}_{ij}(u_1)$.
  $\hat{c}_i = (\overline{I}_{ij})_{j \in [d]}$.
return $\{\hat{c}_1, \hat{c}_2, ..., \hat{c}_k\}$.

Dựa trên các tọa độ đã lấy mẫu, trong các bước còn lại của giai đoạn ước lượng khoảng (bước 4-6), thuật toán Fast-Sampling ước tính độ dài khoảng để xác định các vùng tiềm năng có thể bao quanh trọng tâm của Q'_{ij}. Theo các hệ quả rút ra, có thể giả định rằng luôn tồn tại ít nhất một tọa độ $u \in U_{ij} \cap G^2_{ij}$. Sau đó, trong bước 5, thuật toán xác định tập $\mathcal{N}_{ij}(u)$ gồm $(1-\alpha)m_i$ tọa độ trong P_{ij} gần u nhất. Các bổ đề cho thấy cả chặn dưới và chặn trên cho $\delta^2(Q'_{ij}, \overline{Q'_{ij}})$ đều có thể được thiết lập bằng cách sử dụng $\mathcal{N}_{ij}(u)$. Nếu điểm được lấy mẫu u thuộc G^2_{ij}, bằng cách xác định tập $\mathcal{N}_{ij}(u)$, tác giả có thể thu được các khoảng bao quanh Q'_{ij} với độ dài xác định.

Trong giai đoạn xây dựng tọa độ ứng viên (bước 7), thuật toán tiếp tục chia các khoảng thành các khối nhỏ hơn, trong đó độ dài mỗi khối được tham số hóa bởi $\varepsilon' = \sqrt{\varepsilon/48}$. Do đó, tập ứng viên U'_{ij} được xây dựng ở bước 8 sẽ chứa ít nhất một tọa độ u' đủ gần với trọng tâm của Q'_{ij}.

Bắt đầu từ bước 9, thuật toán Fast-Sampling liệt kê tất cả các tọa độ ứng viên đã xây dựng và $(1-\alpha)m_i$ lân cận gần nhất của chúng để xác định tập hợp tọa độ có chi phí phân cụm nhỏ nhất. Sau đó, trọng tâm của tập hợp này được chọn để làm tọa độ cho tâm. Gọi I_{ij} là tập hợp các tọa độ được tìm thấy ở bước 10. Các bổ đề chứng minh rằng khoảng cách giữa Q_{ij} và I_{ij} có thể được chặn bằng cách sử dụng $I_{ij} \cap Q_{ij}$ để bắc cầu. Kết hợp các kết quả này, tác giả thiết lập được chặn cho khoảng cách giữa I_{ij} và P^*_{ij}. Tổng hợp lại, thuật toán có thể đưa ra nghiệm xấp xỉ $(1+O(\alpha))$ cho bài toán k-means có hỗ trợ học trong thời gian $O(\varepsilon^{-1} md \log(kd))$.

Lemma 4. *Với mọi $Q_{ij} = P^*_{ij} \cap P_{ij}$, gọi Q'_{ij} là tập con của Q_{ij} có kích thước $(1-\alpha)m_i$ và chi phí phân cụm nhỏ nhất. Gọi $G^{\mu}_{ij} = \{x \in Q'_{ij} : \delta^2(x, \overline{Q'_{ij}}) \leq \mu \delta^2(Q'_{ij}, \overline{Q'_{ij}})/|Q'_{ij}|\}$ là tập hợp các tọa độ "tốt"với hằng số $\mu > 1$. Khi đó:*

$$|G^{\mu}_{ij}| \geq \frac{\mu - 1}{\mu} |Q'_{ij}|$$

Corollary 0.1. *Với xác suất hằng số, đối với mỗi cụm dự đoán và mỗi chiều , tồn tại ít nhất một tọa độ sao cho .*

Lemma 5. *Cho một tọa độ bất kỳ $u \in G^2_{ij} \cap U_{ij}$, bất đẳng thức sau luôn thỏa mãn:*

$$\delta^2(Q'_{ij}, \overline{Q'_{ij}}) \leq \delta^2(\mathcal{N}_{ij}(u), \overline{\mathcal{N}_{ij}(u)}) \leq 3\delta^2(Q'_{ij}, \overline{Q'_{ij}}) \tag{6}$$

trong đó $\overline{Q'_{ij}}$ và $\overline{\mathcal{N}_{ij}(u)}$ lần lượt là tâm hình học của tập Q'_{ij} và $\mathcal{N}_{ij}(u)$.

Corollary 0.2. *Với xác suất hằng số, đối với mỗi chiều $j \in [d]$ của mỗi cụm $i \in [k]$, tồn tại ít nhất một tọa độ $u' \in U'_{ij}$ sao cho:*

$$\delta(u', \overline{Q'_{ij}}) \leq \sqrt{\frac{\varepsilon \delta^2(Q'_{ij}, \overline{Q'_{ij}})}{2(1-\alpha)m_i}} \tag{7}$$

trong đó $\overline{Q'_{ij}}$ là tâm hình học của tập tối ưu Q'_{ij}.

Lemma 6. *Giới hạn sau đây luôn đúng đối với tập hợp các tọa độ I_{ij} được xác định bởi thuật toán*

Fast-Sampling so với tập giao Q_{ij}:

$$\delta^2(\overline{I_{ij}}, \overline{Q_{ij}}) \leq \frac{(4\alpha + \alpha\varepsilon)\delta^2(Q_{ij}, \overline{Q_{ij}})}{|Q_{ij}|(1-2\alpha)}$$

Lemma 7. *Khoảng cách giữa tọa độ của tâm thuật toán $\overline{I_{ij}}$ và tâm tối ưu $\overline{P^*_{ij}}$ bị chặn bởi:*

$$\delta^2(\overline{I_{ij}}, \overline{P^*_{ij}}) \leq \left(\frac{\alpha}{1-\alpha} + \frac{\alpha(4+\varepsilon)}{(1-2\alpha)(1-\alpha)}\right) \frac{\delta^2(P^*_{ij}, \overline{P^*_{ij}})}{|P^*_{ij}|}$$

Theorem 1. *Tồn tại một thuật toán k-means có hỗ trợ học (Fast-Sampling) trả về một giải pháp xấp xỉ $(1+O(\alpha))$ trong thời gian $O(\varepsilon^{-1} md \log(kd))$ với xác suất hằng số, trong đó tỷ lệ lỗi nhãn thỏa mãn $\alpha \in [0, 1/2)$.*

5 Fast-Estimation

Mặc dù thuật toán Fast-Sampling có thời gian chạy tuyến tính trong khi vẫn duy trì các đảm bảo về mặt xấp xỉ, nhưng vẫn có $O(\log(kd))$ khi thực hiện chặn hội tụ xác suất, có thể ảnh hưởng trong thực tế của thuật toán khi xử lý các tập dữ liệu quy mô cực lớn. Để giải quyết vấn đề này, trong phần này, tác giả đề xuất một thuật toán dựa trên lấy mẫu nhanh hơn mang tên Fast-Estimation. Thuật toán Fast-Estimation có thể xấp xỉ hiệu quả tọa độ của từng cụm dự đoán trong thời gian chạy tuyến tính, với một sự đánh đổi nhỏ trong các đảm bảo về chất lượng phân cụm.

Ý tưởng chính: trước tiên tạo ra các tọa độ ứng viên có khả năng xấp xỉ chặt chẽ tọa độ của các tâm tối ưu. Sau đó, trong mỗi chiều của từng cụm dự đoán, một bộ ước lượng (estimator) được xây dựng bằng cách lấy mẫu theo phân phối đều. Bộ ước lượng này được thiết kế để cung cấp các ước tính chi phí phân cụm chính xác cho các tập con tọa độ có kích thước $(1-\alpha)m_i$. Cụ thể, đối với mỗi chiều của từng cụm dự đoán, bộ ước lượng được xây dựng bằng cách chọn ngẫu nhiên một tập S_{ij} từ P_{ij}. Mỗi tọa độ được lấy mẫu sau đó được gán một trọng số bằng nhau, vì vậy xấp xỉ chi phí phân cụm thông qua các mẫu trọng số thay vì tính toàn bộ cụm dự đoán. Với các bộ ước lượng đã xây dựng, việc tìm kiếm tập hợp các tọa độ có chi phí phân cụm tối thiểu có thể được thực hiện trong thời gian hạ tuyến tính (sub-linear), loại bỏ nhân với $O(\log(kd))$ khỏi thời gian chạy của thuật toán Fast-Sampling.

Thuật toán 2 Fast-Estimation

Đầu vào: Một bài toán k-means (P,k,d), một tập các phân vùng $(P_1,P_2,...,P_k)$ với tỷ lệ lỗi α, và tham số $0<\varepsilon<0.5$.

Đầu ra: Một tập $C\subset\mathbb{R}^d$ các tâm với $|C|=k$.

1: **for** $i\in[k]$ **do**
2: **for** $j\in[d]$ **do**
3: Lấy mẫu ngẫu nhiên và độc lập một tập U_{ij} từ P_{ij} với kích thước $O(\log(kd))$, sau đó khởi tạo $U'_{ij}=\emptyset$ và $\varepsilon_1=\frac{\varepsilon}{126}$.
4: **for** $q=0$ to $O(\log(m\Delta^2_{max}))$ **do**
5: $l_{ij}=\sqrt{\frac{2^{q-1}}{(1-\alpha)m_i}}$.
6: **for** $u\in U_{ij}$ **do**
7: $s(u)=\{u+\varepsilon_2\lambda l_{ij}:\lambda\in[-\frac{1}{\varepsilon_2},\frac{1}{\varepsilon_2}]\cap\mathbb{Z}\}$, với $\varepsilon_2=\sqrt{\frac{\varepsilon_1}{32}}$.
8: $U'_{ij}=U'_{ij}\cup s(u)$.
9: Lấy mẫu ngẫu nhiên và độc lập một tập S_{ij} từ P_{ij} với kích thước $O\left(\frac{\log(m^3d\log^3(m\Delta^2_{max})/\varepsilon_1^2)\log(m\Delta^2_{max})}{\alpha\varepsilon_1^4}\right)$, gán cho mỗi điểm trong S_{ij} một trọng số $\frac{m_i}{|S_{ij}|}$.
10: Xây dựng bộ ước lượng ω sao cho $\forall u\in U'_{ij}$, $\omega(u)=\sum_{p\in S_{ij}\setminus F(u)}\frac{m_i}{|S_{ij}|}\delta^2(p,u)$, trong đó $F(u)$ là tập hợp $(1+3\varepsilon_1)\alpha|S_{ij}|$ điểm xa u nhất trong S_{ij}.
11: $c_{ij}=\arg\min_{u\in U'_{ij}}\omega(u)$.
12: Gọi I_{ij} là tập hợp $(1-2\alpha-\alpha\varepsilon)m_i$ tọa độ gần c_{ij} nhất từ P_{ij}.
13: $\hat{c}_i=(I_{ij})_{j\in[d]}$.
14: **return** $\{\hat{c}_1,\hat{c}_2,...,\hat{c}_k\}$.

Phân tích thuật toán:

Trong bước 3, đối với mỗi chiều của cụm dự đoán, thuật toán chọn một mẫu ngẫu nhiên U_{ij} để xấp xỉ tọa độ của các tâm tối ưu. Theo Lemma 4, với xác suất hằng số, tồn tại ít nhất một tọa độ được lấy mẫu $u\in U_{ij}$ sao cho $\delta(u,Q'_{ij})\le\sqrt{2\delta^2(Q'_{ij},\overline{Q'_{ij}})/|Q'_{ij}|}$. Sau đó, từ bước 4 liệt kê tất cả các độ dài khoảng ứng viên để xây dựng tập hợp các tọa độ ứng viên. Không mất tính tổng quát, có thể giả sử khoảng cách cặp tối thiểu giữa các tọa độ trong P_{ij} là 1 và khoảng cách cặp tối đa là Δ_{max}. Do đó, trong bước 5, tồn tại ít nhất một lần đoán q cho độ dài thoả $\sqrt{\frac{2\delta^2(Q'_{ij},\overline{Q'_{ij}})}{(1-\alpha)m_i}}\le l_{ij}\le\sqrt{\frac{4\delta^2(Q'_{ij},\overline{Q'_{ij}})}{(1-\alpha)m_i}}$. Tiếp theo, trong các bước 7-8, theo Lemma 5, cũng tồn tại ít nhất một tọa độ $u'\in U'_{ij}$ sao cho u' đủ gần với trọng tâm của Q'_{ij}, tức là $\delta(u',Q'_{ij})\le\sqrt{\varepsilon_1\delta^2(Q'_{ij},\overline{Q'_{ij}})/|Q'_{ij}|}$.

Đối với mỗi $u\in U'_{ij}$, gọi $\mathcal{N}_{ij}(u)$ là tập hợp $(1-\alpha)m_i$ tọa độ gần nhất từ P_{ij} đến u. Gọi $O(u)=P_{ij}\setminus\mathcal{N}_{ij}(u)$ là tập hợp αm_i tọa độ xa nhất từ P_{ij} đến u. Trước khi xây dựng bộ ước lượng ω (bước 9-10), tác giả bắt đầu bằng cách chia $\mathcal{N}_{ij}(u)$ thành $\gamma=\frac{(1+\varepsilon_1)\log(m\Delta^2_{max})}{\varepsilon_1}$ khối. Cụ thể, đối với mỗi $u\in U'_{ij}$, $\mathcal{N}_{ij}(u)$ được phân rã thành γ khối (ký hiệu là $\mathscr{B}^1_u,\mathscr{B}^2_u,\ldots,\mathscr{B}^\gamma_u$) dựa trên khoảng cách từ các tọa độ trong $\mathcal{N}_{ij}(u)$ đến u, trong đó $\mathscr{B}^l_u=\{x\in\mathcal{N}_{ij}(u):(1+\varepsilon_1)^l\le\delta^2(x,u)<(1+\varepsilon_1)^{l+1}\}$.

Các "khối"này có thể hình dung là các phần tiếp nối giữa khối cầu có bán kính $(1+\varepsilon_1)^l$ và $(1+\varepsilon_1)^{l+1}$ trong $\mathbb{R}^d$

Sau đó, các khối này được chia tiếp thành hai nhóm dựa trên kích thước: $\mathscr{L}(u) = \{\mathscr{B}_u^l : |\mathscr{B}_u^l| \geq \frac{\varepsilon_1^2 \alpha m_i}{(1+\varepsilon_1)\log(m_i\Delta_{max}^2)}, l \in [\gamma]\}$ là nhóm các khối lớn và $\mathscr{S}(u) = \{\mathscr{B}_u^1, \ldots, \mathscr{B}_u^\gamma\} \setminus \mathscr{L}(u)$ là nhóm các khối nhỏ.

Sự hội tụ xác suất:

Mục tiêu là xấp xỉ tốt từng khối lớn trong $\mathscr{L}(u)$ đồng thời cho phép bỏ qua các tọa độ trong các khối nhỏ.

1. **Biến ngẫu nhiên:** Đối với mỗi mẫu $p \in S_{ij}$, xét biến ngẫu nhiên chỉ thị cho việc p rơi vào một khối cụ thể.
2. **Áp dụng Bất đẳng thức Chernoff:** Với kích thước mẫu $|S_{ij}|$ được chọn, kỳ vọng số điểm rơi vào mỗi khối lớn đủ lớn để xác suất sai lệch quá ε_1 lần kỳ vọng bị chặn bởi một hàm mũ âm. Cụ thể, $Pr(|X - \mathbb{E}[X]| \geq \varepsilon_1 \mathbb{E}[X]) \leq 2e^{-\varepsilon_1^2 \mathbb{E}[X]/3}$.
3. **Chặn hội tụ (Union Bound):** Bằng cách lấy tổng xác suất lỗi trên tất cả các khối và các tọa độ ứng viên, tác giả đảm bảo rằng bộ ước lượng ω hoạt động chính xác với xác suất cao trên toàn không gian ứng viên.

Với bộ ước lượng đã được chứng minh là hội tụ về giá trị thực, việc tìm c_{ij} tại bước 11 nhanh hơn vì số lượng ứng viên $|U'_{ij}|$ chỉ phụ thuộc logarit vào Δ_{max} và m, trong khi việc tính toán mỗi giá trị $\omega(u)$ chỉ tốn thời gian phụ thuộc vào kích thước mẫu $|S_{ij}|$ thay vì kích thước toàn bộ dữ liệu m_i. Cuối cùng, bằng cách sử dụng Lemma 7, Theorem 2 có thể được chứng minh để độ phức tạp thời gian tuyến tính $O(md) + \tilde{O}(\varepsilon^{-5}kd/\alpha)$ cho bài toán có hỗ trợ học.

Lemma 8. *Giả sử S_{ij} là một mẫu được lấy ngẫu nhiên từ cụm dự đoán P_{ij} với kích thước mẫu $|S_{ij}| = \tilde{O}(1/\alpha\varepsilon_1^4)$. Với xác suất ít nhất $1 - \frac{\varepsilon_1}{m^3 d \log^2(m\Delta_{\max}^2)}$, các bất đẳng thức sau đây đồng thời xảy ra cho mọi khối lớn $\mathscr{B}_u^l \in \mathscr{L}(u)$ và tập các điểm xa nhất $\mathscr{O}(u)$:*

$$(1-\varepsilon_1)\mathbb{E}[|\mathscr{B}_u^l \cap S_{ij}|] \leq |\mathscr{B}_u^l \cap S_{ij}| \leq (1+\varepsilon_1)\mathbb{E}[|\mathscr{B}_u^l \cap S_{ij}|]$$

$$(1-\varepsilon_1)\mathbb{E}[|\mathscr{O}(u) \cap S_{ij}|] \leq |\mathscr{O}(u) \cap S_{ij}| \leq (1+\varepsilon_1)\mathbb{E}[|\mathscr{O}(u) \cap S_{ij}|]$$

Lemma 9. *Gọi $\mathscr{J}(u)$ là tập hợp các tọa độ nằm trong các khối nhỏ đối với một tọa độ ứng viên u. Với xác suất ít nhất $1 - \frac{\varepsilon_1}{m^3 d \log^2(m\Delta_{\max}^2)}$, giao của tập mẫu S_{ij} và $\mathscr{J}(u)$ bị chặn như sau:*

$$|\mathscr{J}(u) \cap S_{ij}| \leq 2\varepsilon_1 \alpha |S_{ij}|$$

Lemma 10. *Cho một tọa độ ứng viên bất kỳ $u \in U'_{ij}$. Với xác suất cao (xác suất hằng số), ước lượng $\omega(u)$ thỏa mãn các chặn sau:*

$$\frac{\delta^2(P_{ij} \setminus \mathscr{F}^\dagger(u), u)}{1+7\varepsilon_1} \leq \omega(u) \leq (1+\varepsilon_1)^2 \delta^2(\mathscr{N}_{ij}(u), u)$$

trong đó:

- $\mathscr{F}^{\dagger}(u)$ *là tập hợp gồm* $(2+20\varepsilon_1)\alpha m_i$ *tọa độ xa nhất từ* P_{ij} *đến u.*
- $\mathscr{N}_{ij}(u)$ *là tập hợp gồm* $(1-\alpha)m_i$ *tọa độ gần nhất trong* P_{ij} *đến u.*

Lemma 11. *Với tập hợp các tọa độ* I_{ij} *được xác định bởi thuật toán Fast-Estimation, chặn sau đây luôn thỏa mãn:*

$$\delta^2(\overline{I_{ij}}, \overline{Q_{ij}}) \leq \frac{13\alpha - 15\alpha^2}{(1-3\alpha-\varepsilon)(1-2\alpha-\varepsilon)} \frac{\delta^2(Q_{ij}, \overline{Q_{ij}})}{|Q_{ij}|}$$

Theorem 2. *Thuật toán Fast-Estimation xấp xỉ* $(1+O(\alpha))$ *cho bài toán k-means có hỗ trợ học (learning-augmented) trong thời gian* $O(md)+\tilde{O}(\varepsilon^{-5}kd/\alpha)$ *với xác suất hằng số, với tỷ lệ lỗi nhãn* $\alpha \in (0, 1/3-\varepsilon)$.

6 Fast-Filtering

Đối với Fast-Sampling và Fast-Estimation, các tâm được tạo ra bằng cách tìm xấp xỉ tọa độ trong từng chiều không gian. Tuy nhiên, quy trình lấy mẫu này có thể làm phát sinh các sai số tích lũy, dẫn đến sự suy giảm chất lượng phân cụm tổng thể. Trong phần này, dựa trên các thuật toán Fast-Sampling và Fast-Estimation, tác giả đề xuất một thuật toán heuristic thực tiễn hơn mang tên Fast-Filtering nhằm bảo toàn tốt hơn chất lượng phân cụm trong khi vẫn duy trì được thời gian chạy hiệu quả.

Thuật toán đề xuất được trình bày trong Thuật toán 3, với ý tưởng chủ đạo là trực tiếp tìm kiếm các xấp xỉ tâm cho từng cụm dự đoán thay vì xấp xỉ từng chiều độc lập. Tại bước 2, một tập hợp các mẫu được rút ra một cách ngẫu nhiên và độc lập từ mỗi cụm dự đoán để đóng vai trò là các tâm ứng viên. Sau đó, trong các bước 3-4, các bộ ước lượng được xây dựng dựa trên những ý tưởng tương tự từ thuật toán Fast-Estimation. Dựa trên các bộ ước lượng này, tâm ứng viên có chi phí phân cụm tối thiểu được lựa chọn tại bước 5 để xác định các khoảng chứa $(1-\alpha)m_i$ điểm gần nhất. Cuối cùng, tại bước 7, các trọng tâm của các tập điểm đã xác định được chọn làm các tâm cuối cùng. Trong Phụ lục A.4, tác giả cung cấp phân tích lý thuyết cho thuật toán Fast-Filtering và chỉ ra rằng, với việc điều chỉnh số lượng lân cận gần nhất cùng kích thước mẫu R_1 và R_2, thuật toán này có thể đưa ra một nghiệm xấp xỉ $(1+O(\sqrt{\alpha}))$.

Giải thích thuật toán:

Thuật toán này giải quyết vấn đề "sai số tích lũy"bằng cách làm trực tiếp trên vecto thay vì gộp kết quả từ d bài toán đơn chiều.

- **Ước lượng nhanh:** Ý tưởng giống Fast-Estimation. Tại bước 4, thay vì tính toán tổng bình phương khoảng cách δ^2 trên toàn bộ tập dữ liệu P_i (vốn tốn thời gian $O(m_i d)$), tác giả sử dụng tập mẫu S_i có kích thước R_2 nhỏ hơn nhiều. Trọng số $\frac{m_i}{|S_i|}$ đảm bảo rằng kỳ vọng của bộ ước lượng $\omega(u)$ sẽ hội tụ về giá trị chi phí thực tế của cụm.
- **Loại bỏ nhiễu (Filtering):** Một đóng góp quan trọng của tác giả là việc định nghĩa tập $F(u)$ gồm các điểm xa nhất. Trong bài toán có hỗ trợ học, cụm dự đoán P_i có thể chứa tới αm_i điểm âm tính giả (nhiễu). Nếu các điểm nhiễu này nằm rất xa tâm thực, chúng sẽ kéo trọng tâm lệch khỏi vị trí tối

Thuật toán 3 Fast-Filtering

Đầu vào: Bài toán k-means (P,k,d), tập các phân vùng $(P_1,P_2,\ldots,P_k)$ với tỷ lệ lỗi α, các tham số $R_1 > 0, R_2 > 0$ và $0 < \varepsilon < 1$.

Đầu ra: Một tập $C \subset \mathbb{R}^d$ các tâm với $|C| \leq k$.

1: **for** $i = 1..k$ **do**
2: Lấy mẫu ngẫu nhiên và độc lập một tập U_i từ P_i với kích thước R_1.
3: Lấy mẫu ngẫu nhiên và độc lập một tập S_i từ P_i với kích thước R_2, và gán cho mỗi điểm trong S_i một trọng số $\frac{m_i}{|S_i|}$.
4: Xây dựng bộ ước lượng ω sao cho $\forall u \in U_i, \omega(u) = \sum_{p \in S_i \setminus F(u)} \frac{m_i}{|S_i|} \delta^2(p,u)$, trong đó $F(u)$ là tập hợp $(1+\varepsilon)\alpha|S_i|$ điểm trong S_i có khoảng cách xa nhất đối với u.
5: $c_i = \arg\min_{u \in U_i} \omega(u)$.
6: Gọi I_i là tập hợp $(1-\alpha)m_i$ điểm trong P_i gần c_i nhất.
7: $\hat{c}_i = \bar{I}_i$.
return $\{\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_k\}$.

ưu. Việc loại bỏ $F(u)$ trong quá trình ước lượng giúp "cô lập"ảnh hưởng của các điểm ngoại lệ này, giúp việc chọn c_i trở nên bền bỉ (robust) hơn.

- **Trọng tâm:** Sau khi đã xác định được một tâm ứng viên tốt c_i, thuật toán thực hiện một bước tinh chỉnh tại bước 6 và 7. Tập I_i đại diện cho phần "lõi"sạch nhất của cụm. Theo Lemma 1, trọng tâm $\bar{I}_i$ là điểm duy nhất tối thiểu hóa tổng bình phương khoảng cách tới tất cả các điểm trong tập đó. Do đó, $\hat{c}_i$ chính là nghiệm tối ưu địa phương cho tập điểm đã được lọc nhiễu.

Sự kết hợp giữa lấy mẫu ngẫu nhiên để tìm ứng viên và bộ lọc thống kê để đánh giá chi phí cho phép Fast-Filtering đạt được sự cân bằng giữa tốc độ tính toán và độ chính xác phân cụm.

Theorem 3. *Cho* $R_1 = O\left(\frac{\log k}{1-2\alpha}\right)$ *và* $R_2 = O\left(\frac{\log(m^3 d \log^3(m\Delta^2)/\varepsilon^2)\log(m\Delta^2)}{\alpha\varepsilon^4}\right)$*, trong đó* Δ *là tỷ lệ chiều của tập dữ liệu. Với xác suất hằng số, Thuật toán 4 (Fast-Filtering) trả về nghiệm xấp xỉ* $(1+O(\sqrt{\alpha}))$ *cho bài toán k-means có hỗ trợ học trong thời gian* $O(md) + \tilde{O}\left(\frac{kd}{\varepsilon^4(1-2\alpha)\alpha}\right)$ *với* $\alpha \in (0, 1/3-\varepsilon)$.

Corollary 3.1. *Cho kích thước mẫu* $R_1 = \Theta\left(\frac{\log k}{1-2\alpha}\right)$*. Với mỗi cụm dự đoán* $i \in [k]$*, với xác suất hằng số, tồn tại ít nhất một điểm dữ liệu u trong tập mẫu* U_i *sao cho* $u \in G_2(P_i^*)$*, trong đó* $G_2(P_i^*)$ *là tập hợp các điểm nằm gần tâm tối ưu.*

Corollary 3.2. *Cho*

$$R_2 = O\left(\frac{\log(m^3 d \log^3(m\Delta^2)/\varepsilon_1^2)\log(m\Delta^2)}{\alpha\varepsilon_1^4}\right)$$

Với một điểm dữ liệu bất kỳ $u \in U_i$*, với xác suất cao, bộ ước lượng* $\omega(u)$ *thỏa mãn:*

$$\frac{\delta^2(P_i \setminus \mathscr{Z}^\dagger(u), u)}{1+7\varepsilon_1} \leq \omega(u) \leq (1+\varepsilon_1)^2 \delta^2(H_i(u), u)$$

trong đó $H_i(u)$ *là tập hợp* $(1-\alpha)m_i$ *điểm gần u nhất trong* P_i*, và* $\mathscr{Z}^\dagger(u)$ *là tập hợp* $(2+20\varepsilon_1)\alpha m_i$ *điểm xa u nhất.*

Lemma 12. *Khoảng cách giữa trọng tâm của tập hợp đã lọc $\overline{I}_i$ và tâm tối ưu c_i^* bị chặn như sau:*

$$\delta^2(\overline{I}_i, c_i^*) \leq \frac{9\delta^2(P_i^*, c_i^*)}{(1-(3+\varepsilon)\alpha)m_i}$$

Lemma 13. *Chi phí phân cụm của tập Q_i đối với tâm của tập hợp đã lọc $\overline{I}_i$ thỏa mãn chặn cụ thể sau:*

$$\delta^2(Q_i, \overline{I}_i) \leq \delta^2(Q_i, c_i^*) + \left(5\sqrt{\alpha} + \frac{36(\sqrt{\alpha}+\alpha)}{1-(3+\varepsilon)\alpha}\right)\delta^2(P_i^*, c_i^*)$$

Lemma 14. *Tổng chi phí phân cụm của cụm tối ưu P_i^* đối với trọng tâm $\overline{I}_i$ bị chặn bởi:*

$$\delta^2(P_i^*, \overline{I}_i) \leq \left(1 + 6\sqrt{\alpha} + \frac{36(\sqrt{\alpha}+\alpha)}{1-(3+\varepsilon)\alpha} + \frac{9(\sqrt{\alpha}+\alpha)}{(1-\alpha)(1-(3+\varepsilon)\alpha)}\right)\delta^2(P_i^*, c_i^*)$$

7 Fast-Sampling (k-median)

Trong mục này, tác giả trình bày cách mở rộng các phương pháp dựa trên lấy mẫu được đề xuất cho bài toán k-median có hỗ trợ học. Thách thức chính ở đây nảy sinh từ sự khác biệt trong các mục tiêu tối ưu hóa. Cụ thể, đối với một tập hợp tọa độ $S \subset \mathbb{R}^d$ bất kỳ, tâm hình học của S không còn đóng vai trò là tâm phân cụm tối ưu cho S theo mục tiêu k-median, khiến việc xác định các tọa độ hoặc tâm ứng viên chất lượng cao trở nên khó khăn. Kết quả là, các thuật toán k-median có hỗ trợ học hiện có thường gặp khó khăn trong việc đạt được các đảm bảo xấp xỉ chất lượng cao.

Để vượt qua thách thức này, mục tiêu của tác giả là sử dụng các chiến lược dựa trên lấy mẫu để xây dựng một tập U_i các tâm nằm gần các tâm phân cụm tối ưu cho mỗi cụm dự đoán P_i. Sau đó, bằng cách rời rạc hóa lưới, tác giả có thể tạo ra các tâm ứng viên có khả năng xấp xỉ tốt các tâm phân cụm tối ưu. Cuối cùng, bằng cách liệt kê các tâm ứng viên đã xây dựng, tác giả chứng minh rằng chi phí phân cụm của mỗi cụm tối ưu có thể được xấp xỉ tốt bằng cách sử dụng tâm tốt nhất được chọn từ quá trình liệt kê.

Bảng 2: Kết quả so sánh các thuật toán k-median có hỗ trợ học

Phương pháp và Tài liệu tham khảo	**Tỷ lệ xấp xỉ**	**Khoảng lỗi nhãn α**	**Độ phức tạp thời gian**
Phân vùng và Sắp xếp Ergun et al. (2021)	$1+\tilde{O}((k\alpha)^{1/4})$	Hằng số nhỏ	$O(md\log^3 m + \text{poly}(k,\log m))$
Sắp xếp Nguyen et al. (2022)	$1+\frac{\alpha(7+10\alpha-10\alpha^2)}{(1-\alpha)(1-2\alpha)}$	$[0,1/2)$	$O\left(\frac{md\log^3 m\log^2(k/\delta)}{1-2\alpha}\right)$
Fast-Sampling (Tác giả)	$1+\frac{\alpha(6+4\varepsilon-4\alpha-3\varepsilon\alpha)}{(1-\alpha)(1-2\alpha)}$	$(0,1/2)$	$O\left(\frac{md\log(kd)\log(m\Delta)}{1-2\alpha}\cdot\left(\frac{\sqrt{d}}{\varepsilon\alpha}\right)^{O(d)}\right)$

Bảng 7 cung cấp một so sánh chi tiết các kết quả cho bài toán k-median có hỗ trợ học. Tác giả cũng đưa ra một biểu đồ (Hình 2) về tỷ lệ xấp xỉ so với tỷ lệ lỗi α. Có thể thấy từ bảng rằng kết quả tốt nhất hiện nay đạt được xấp xỉ $(1+O(\alpha))$ với $\alpha \in [0,1/2)$ Nguyen et al. (2022). So với các kết quả tiên tiến nhất, thuật toán Fast-Sampling có thể đạt được các đảm bảo chất lượng phân cụm tốt hơn với thời gian chạy kém hơn

một chút đối với số chiều d cố định.

Mô tả cụ thể cho thuật toán k-median có hỗ trợ học được trình bày trong Thuật toán 5. Ý tưởng chung đằng sau thuật toán là trước tiên tạo ra các tâm ứng viên có thể xấp xỉ chặt chẽ các tâm phân cụm tối ưu cho mỗi cụm dự đoán. Sau đó, bằng cách chọn tâm tốt nhất với chi phí k-median nhỏ nhất, tác giả chứng minh rằng thuật toán đề xuất có thể đưa ra các đảm bảo xấp xỉ tốt hơn cho bài toán k-median có hỗ trợ học. Dưới đây, tác giả đưa ra phân tích cụ thể cho thuật toán được đề xuất.

Thuật toán 4 Fast-Sampling (k-median)

Đầu vào: Bài toán k-median (P,k,d), tập các phân vùng $(P_1,\ldots,P_k)$ với tỷ lệ lỗi α, tham số $\varepsilon \in (0,1]$.
Đầu ra: Tập $C \subset \mathbb{R}^d$ gồm k tâm sao cho $|C| = k$.

1: **for** mỗi $i \in [k]$ **do**
2: Lấy mẫu ngẫu nhiên và độc lập tập U_i từ P_i với kích thước $O\left(\frac{\log(kd)}{1-2\alpha}\right)$, sau đó khởi tạo $U_i' = \emptyset$.
3: **for** $q = 0$ đến $O(\log(m\Delta))$ **do**
4: $l_i = 2^{q-1}/(1-\alpha)m_i$.
5: **for** mỗi $u \in U_i$ **do**
6: Gọi $G(u)$ là lưới tâm u với độ dài cạnh $2l_i$.
7: Phân rã $G(u)$ thành các lưới con nhỏ hơn với độ dài cạnh $(1-\alpha)\alpha\varepsilon_1 l_i/\sqrt{d}$, và gọi $s(u)$ là tập các tâm của các lưới con này, với $\varepsilon_1 < \varepsilon/4$.
8: $U_i' = U_i' \cup s(u)$.
9: $u_i = \arg\min_{u \in U_i'} \delta(\mathcal{N}_i(u), u)$, trong đó $\mathcal{N}_i(u)$ là tập $(1-\alpha)m_i$ điểm trong P_i gần u nhất.
10: $\hat{c}_i = u_i$.
11: **return** $\{\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_k\}$.

Không mất tính tổng quát, tác giả có thể giả định rằng khoảng cách từng đôi tối thiểu giữa các điểm dữ liệu trong P là 1 trong khi khoảng cách từng đôi tối đa là Δ. Lưu ý rằng điều này có thể thực hiện được bằng các kỹ thuật tỉ lệ chuẩn. Theo Bổ đề 4, trong mỗi bước 2 của Thuật toán 5, với xác suất ít nhất $1 - 1/k$, có thể tìm thấy ít nhất một tâm $u \in U_i$ sao cho $\delta(u, c_i^*) \le 2\delta(P_i^*, c_i^*)/|P_i^*| \le \frac{2\delta(P_i^*, c_i^*)}{(1-\alpha)m_i}$, trong đó bước cuối cùng tuân theo thực tế là $|P_i^*| \ge |Q_i| \ge (1-\alpha)m_i$. Sau đó, trong bước 3 của Thuật toán 5, vì thuật toán liệt kê tất cả các giá trị có thể giữa 1 và $\log(m\Delta)$, tồn tại ít nhất một dự đoán cho bán kính phân cụm (bước 4 của Thuật toán 5) sao cho $\delta(P_i^*, c_i^*)/(1-\alpha)m_i \le l_i \le 2\delta(P_i^*, c_i^*)/(1-\alpha)m_i$. Do đó, trong bước 6 của Thuật toán 5, lưới có tâm tại u với độ dài cạnh $2l_i$ ($G(u)$) sẽ chứa tâm phân cụm tối ưu c_i^*. Sau đó, trong bước 7 của Thuật toán 5, bằng cách phân rã lưới $G(u)$ thành các lưới con nhỏ hơn với độ dài cạnh $(1-\alpha)\alpha\varepsilon_1 l_i/\sqrt{d}$ cho một số $\varepsilon_1 < \varepsilon/4$, tâm phân cụm tối ưu c_i^* cũng phải thuộc về một trong các lưới con. Vì lưới con có độ dài cạnh $(1-\alpha)\alpha\varepsilon_1 l_i/\sqrt{d}$, cũng tồn tại ít nhất một $u' \in U_i'$ sao cho u' đủ gần với c_i^*, tức là $\delta(u', c_i^*) \le (1-\alpha)\alpha\varepsilon_1 l_i \le \alpha\varepsilon\delta(P_i^*, c_i^*)/m_i$. Gọi u_i là điểm được chọn trong bước 9 của Thuật toán 5. Đối với bất kỳ điểm dữ liệu nào $u \in U_i'$, gọi $N_i(u)$ là tập hợp các điểm gần nhất $(1-\alpha)m_i$ trong P_i tới u. Do đó, ta có

$$\begin{aligned}\delta(\mathcal{N}_i(u_i),u_i) &\leq \delta(\mathcal{N}_i(u'),u') \\ &\leq \delta(\mathcal{N}_i(u'),c_i^*) + |\mathcal{N}_i(u')|\delta(u',c_i^*) \\ &\leq \delta(\mathcal{N}_i(u_i),c_i^*) + m_i \cdot \left(\frac{\alpha\varepsilon\delta(P_i^*,c_i^*)}{m_i}\right) \\ &\leq \delta(\mathcal{N}_i(u_i),c_i^*) + \alpha\varepsilon\delta(P_i^*,c_i^*)\end{aligned}$$

Trong đó:

- Bất đẳng thức đầu tiên tuân theo tính tối tiểu của u_i trong tập U_i'.
- Bất đẳng thức thứ hai áp dụng bất đẳng thức tam giác cho từng điểm trong $\mathcal{N}_i(u')$.
- Bước cuối cùng sử dụng giới hạn $|\mathcal{N}_i(u_i)| \leq m_i$ và khoảng cách tâm $\delta(u',c_i^*)$ đã thiết lập.

Corollary 3.3. *Đối với một cụm dự đoán P_i, với xác suất ít nhất $1-1/k$, tâm u_i được chọn bởi thuật toán Fast-Sampling cho mục tiêu k-median thỏa mãn:*

$$\delta(\mathcal{N}_i(u_i),u_i) \leq \delta(\mathcal{N}_i(u_i),c_i^*) + \alpha\varepsilon\delta(P_i^*,c_i^*)$$

trong đó $\mathcal{N}_i(u_i)$ là tập hợp $(1-\alpha)m_i$ điểm gần nhất trong P_i đến u_i, và c_i^* là tâm phân cụm tối ưu cho cụm thứ i.

Lemma 15. *Đối với hàm mục tiêu k-median, khoảng cách giữa tâm thuật toán u_i và tâm phân cụm tối ưu c_i^* thỏa mãn:*

$$\delta(u_i,c_i^*) \leq \frac{(2+\alpha\varepsilon)\delta(P_i^*,c_i^*)}{(1-2\alpha)m_i}$$

Lemma 16. *Đối với hàm mục tiêu k-median, chi phí phân cụm của tập Q_i đối với tâm được chọn u_i thỏa mãn chặn sau:*

$$\delta(Q_i,u_i) \leq \delta(Q_i,c_i^*) + \frac{\alpha(4+3\varepsilon)}{1-2\alpha}\delta(P_i^*,c_i^*)$$

trong đó c_i^* là tâm tối ưu của cụm thứ i.

Lemma 17. *Với mỗi cụm $i \in [k]$, với xác suất ít nhất $1-1/k$, chi phí phân cụm k-median của cụm tối ưu P_i^* đối với tâm thuật toán u_i bị chặn bởi:*

$$\delta(P_i^*,u_i) \leq \left(1 + \frac{6\alpha+4\alpha\varepsilon-4\alpha^2-3\varepsilon\alpha^2}{(1-\alpha)(1-2\alpha)}\right)\delta(P_i^*,c_i^*)$$

trong đó c_i^* là tâm tối ưu của cụm thứ i.

8 Thực nghiệm

8.1 Thực nghiệm của tác giả

Trong phần này, tác giả đưa ra các đánh giá thực nghiệm về hiệu suất của các thuật toán đề xuất. Tất cả các thuật toán được cài đặt và thực thi bằng ngôn ngữ Python. Các thực nghiệm được thực hiện trên máy tính có bộ vi xử lý i7-12700KF và RAM 256GB. Dựa trên các nghiên cứu trước đó của Ergun et al. (2021); Nguyen et al. (2022), tác giả thực hiện mỗi thuật toán 10 lần và báo cáo kết quả trung bình cùng với độ lệch chuẩn. (Các nghiên cứu trước đó đề cập đều chạy thực nghiệm như thế này nên tác giả cũng sẽ thực hiện theo nhằm đảm bảo công bằng)

Tập dữ liệu. Dựa vào các nghiên cứu của Ergun et al. (2021) và Nguyen et al. (2022), tác giả cũng kiểm tra các thuật toán trên các tập dữ liệu CIFAR10 ($m = 10,000$, $d = 3,072$), PHY ($m = 10,000$, $d = 50$) và MNIST ($m = 1,797$, $d = 64$) với các tỷ lệ lỗi α và số lượng cụm k khác nhau. Ngoài ra, tác giả cũng đánh giá hiệu suất trên các tập dữ liệu lớn khác từ Kho lưu trữ Học máy UCI[1], bao gồm SUSY ($m = 5,000,000$, $d = 18$) và HIGGS ($m = 11,000,000$, $d = 27$), cùng một tập dữ liệu quy mô cực lớn SIFT ($m = 100,000,000$, $d = 128$) từ nghiên cứu của Matsui et al. (2017).

Thuật toán. Trong các thực nghiệm, tác giả chủ yếu so sánh các thuật toán Fast-Sampling, Fast-Estimation và Fast-Filtering (phiên bản có đảm bảo lý thuyết) với các thuật toán có hỗ trợ học khác, bao gồm thuật toán trong Ergun et al. (2021) (ký hiệu là Ergun) và thuật toán trong Nguyen et al. (2022) (ký hiệu là Det). Đối với thuật toán Fast-Sampling, kích thước mẫu được thiết lập là 4 và cố định $\varepsilon = 1$. Đối với Fast-Filtering và Fast-Estimation, tác giả cố định $R_1 = 10$, $R_2 = m/20$ và $\varepsilon = 0.3$, trong đó m là kích thước của bài toán phân cụm cụ thể (số lượng điểm dữ liệu). Để chứng minh ưu thế của mô hình phân cụm có hỗ trợ học, tác giả cũng thực hiện so sánh với phương pháp k-means++ Arthur and Vassilvitskii (2007) không sử dụng thông tin dự đoán.

Mô tả bộ dự đoán. Kế thừa phương pháp của Nguyen et al. (2022), bộ dự đoán được tạo ra như sau: với mỗi tập dữ liệu, đầu tiên tác giả chạy phương pháp k-means++ Arthur and Vassilvitskii (2007) để khởi tạo, sau đó chạy thuật toán Lloyd Lloyd (1982) cho đến khi hội tụ; các nhãn thu được được coi là phân hoạch nhãn tối ưu (ký hiệu là $\{P_1, \ldots, P_k\}$). Để kiểm tra hiệu suất dưới các tỷ lệ lỗi khác nhau, tác giả thay đổi ngẫu nhiên nhãn của αm_i điểm gần c_i nhất trong mỗi cụm P_i để tạo ra các phân hoạch nhãn bị nhiễu $\{P'_1, \ldots, P'_k\}$ làm bộ dự đoán, với α chạy từ 0.1 đến 0.5. Cách này là tương tự với cách triển khai trong Nguyen et al. (2022).

Chi tiết cài đặt thuật toán. Như đã được chỉ ra trong Nguyen et al. (2022), trong hầu hết các tình huống thực tế, chúng ta không thể biết được tỷ lệ lỗi α thực sự và phải thử các giá trị đoán khác nhau để chọn ra kết quả có chi phí tốt nhất. Do đó, đối với mỗi thuật toán, tác giả thực hiện lặp qua 15 giá trị tiềm năng của α phân bố đều trong khoảng $[0.01, 0.5]$ làm đầu vào. Giá trị α cho chi phí phân cụm thấp nhất sẽ được chọn làm kết quả cuối cùng. Thời gian chạy của mỗi thuật toán sẽ bao gồm tổng thời gian của 15 lần thử

[1] https://archive.ics.uci.edu/

nghiệm này. Ngoài ra, tác giả cũng so sánh các giá trị ARI và NMI để đánh giá chất lượng phân cụm so với nhãn thực tế.

Bảng 3: So sánh các thuật toán trên tập dữ liệu SIFT với $k = 20$ và các α khác nhau

Phương pháp	Lloyd	α	Chi phí	NMI	ARI	Thời gian (s)
k-means++	1.0542E+13(844.18s)	0.1	1.6884E+13 ± 1.45E+11	0.3285 ± 0.0138	0.1530 ± 0.0137	**1000.89 ± 10.84**
Ergun			9.9799E+12 ± 1.03E+05	0.9243 ± 0.0000	0.9181 ± 0.0001	16748.88 ± 5776.25
Det			9.7791E+12 ± 0.00E+00	0.9490 ± 0.0000	0.9491 ± 0.0000	13152.95 ± 2160.94
Fast-Sampling			9.7666E+12 ± 0.00E+00	**0.9519 ± 0.0000**	**0.9531 ± 0.0000**	13057.36 ± 1717.68
Fast-Filtering			**9.7150E+12 ± 2.90E+08**	0.9316 ± 0.0090	0.9333 ± 0.0107	**1006.31 ± 43.79**
Fast-Estimation			9.8007E+12 ± 3.74E+08	0.9465 ± 0.0003	0.9466 ± 0.0002	8874.66 ± 2871.26
k-means++	9.7055E+12(1011.24s)	0.2	1.6585E+13 ± 7.91E+10	0.3634 ± 0.0182	0.1940 ± 0.0221	**1077.12 ± 71.57**
Ergun			1.0210E+13 ± 2.89E+08	0.9043 ± 0.0000	0.8901 ± 0.0000	17410.75 ± 6132.76
Det			9.9919E+12 ± 0.00E+00	0.9019 ± 0.0000	0.8867 ± 0.0000	13681.80 ± 2073.36
Fast-Sampling			9.9576E+12 ± 0.00E+00	0.9037 ± 0.0000	0.8895 ± 0.0000	13270.53 ± 1989.65
Fast-Filtering			**9.7914E+12 ± 9.59E+08**	0.8690 ± 0.0116	0.8515 ± 0.0146	**1088.53 ± 92.09**
Fast-Estimation			1.0004E+13 ± 6.51E+08	**0.9093 ± 0.0002**	**0.8979 ± 0.0002**	9567.52 ± 2691.74
k-means++	9.2478E+12(1330.99s)	0.3	1.6561E+13 ± 9.48E+10	0.3531 ± 0.0278	0.1814 ± 0.0206	**927.07 ± 31.75**
Ergun			1.0526E+13 ± 4.08E+07	0.8663 ± 0.0000	0.8361 ± 0.0000	17586.20 ± 6488.30
Det			1.0291E+13 ± 0.00E+00	0.8625 ± 0.0000	0.8299 ± 0.0000	13214.91 ± 1914.86
Fast-Sampling			1.0238E+13 ± 0.00E+00	**0.8743 ± 0.0000**	**0.8496 ± 0.0000**	13032.26 ± 1657.72
Fast-Filtering			**9.9098E+12 ± 7.74E+09**	0.8180 ± 0.0048	0.7833 ± 0.0064	**1095.48 ± 66.17**
Fast-Estimation			1.0300E+13 ± 3.31E+08	0.8663 ± 0.0002	0.8371 ± 0.0002	8618.38 ± 2378.02
k-means++	8.9739E+12(1342.73s)	0.4	1.6814E+13 ± 4.91E+11	0.3582 ± 0.0111	0.1752 ± 0.0126	**991.80 ± 148.57**
Ergun			1.0924E+13 ± 4.27E+08	0.8273 ± 0.0000	0.7801 ± 0.0001	16291.70 ± 5926.28
Det			1.0683E+13 ± 0.00E+00	0.8248 ± 0.0000	0.7749 ± 0.0000	12999.81 ± 2144.98
Fast-Sampling			1.0613E+13 ± 0.00E+00	**0.8353 ± 0.0000**	**0.7930 ± 0.0000**	13658.40 ± 1766.14
Fast-Filtering			**1.0125E+13 ± 3.14E+09**	0.7879 ± 0.0048	0.7393 ± 0.0032	**1091.53 ± 94.64**
Fast-Estimation			1.0687E+13 ± 8.25E+08	0.8260 ± 0.0001	0.7781 ± 0.0003	8725.94 ± 2691.41
k-means++	8.7576E+12(1412.61s)	0.5	1.7542E+13 ± 2.81E+11	0.3313 ± 0.0073	0.1580 ± 0.0065	**972.59 ± 60.40**
Ergun			1.1414E+13 ± 4.92E+08	0.7885 ± 0.0000	0.7140 ± 0.0000	17256.11 ± 6160.91
Det			1.1156E+13 ± 0.00E+00	0.7863 ± 0.0000	0.7105 ± 0.0000	13121.68 ± 1901.27
Fast-Sampling			1.1089E+13 ± 0.00E+00	**0.7963 ± 0.0000**	**0.7290 ± 0.0000**	13042.91 ± 1762.42
Fast-Filtering			**1.0504E+13 ± 5.81E+09**	0.7086 ± 0.0103	0.6133 ± 0.0097	**1051.20 ± 34.37**
Fast-Estimation			1.1169E+13 ± 1.68E+09	0.7886 ± 0.0005	0.7153 ± 0.0012	8532.96 ± 2152.19

Kết quả. Bảng 3 so sánh các thuật toán đề xuất với các phương pháp có hỗ trợ học khác trên tập dữ liệu SIFT trên số cụm cố định và nhiều tỷ lệ lỗi khác nhau. Ngoài ra, còn nhiều kết quả trên các tập dữ liệu khác được trình bày ở phụ lục A.6 trong Huang et al. (2025).

Kết quả cho thấy Fast-Sampling đạt chi phí tương đương với các phương pháp hiện đại nhất, trong khi Fast-Filtering liên tục vượt trội hơn với việc giảm trung bình 1.5% chi phí phân cụm trên tất cả các tập dữ liệu. Về thời gian chạy, Fast-Filtering nhanh hơn đáng kể, đặc biệt trên các tập dữ liệu lớn và số chiều cao, đạt tốc độ nhanh hơn ít nhất là gấp 3 lần so với các phương pháp hiện hành. Trên tập dữ liệu SIFT, đây là phương pháp duy nhất nhanh hơn thuật toán Lloyd, với tốc độ nhanh hơn ít nhất là gấp 10 lần so với các phương pháp khác. Về giá trị NMI và ARI, các thuật toán của tác giả duy trì ổn định trên mức 0.80 ở hầu hết các tập dữ liệu, đặc biệt tốt hơn trên MNIST và SIFT do sự chặt chẽ về mặt không gian của chúng. Trong khi đó, thuật toán Det hoạt động tốt hơn trên các tập dữ liệu có số chiều cao (SUSY, HIGGS và PHY), còn thuật toán của Ergun lại vượt trội trên CIFAR10 do tập dữ liệu có các đặc trưng hình ảnh phức tạp.

Nhận xét

Phần thực nghiệm này cho thấy tác giả đang cố gắng công bằng hết mức có thể đối với các thuật toán hỗ trợ học hiện tại, đồng thời cũng kế thừa tư duy thực nghiệm từ các nghiên cứu trước đó. Kết

quả cho ra thông qua bảng 3 cho thấy các thuật toán tác giả đề xuất tốt hơn. Ngoài ra, tác giả cũng đưa ra các nguyên nhân như **sự chặt chẽ về không gian**, **đặc trưng hình ảnh phức tạp** để giải thích cho sự vượt trội của một số thuật toán ở các tập dữ liệu cụ thể. Có thể thấy các tập dữ liệu này có kích thước khá lớn và thời gian chạy được thống kê khá lớn. Do đó, có thể thấy được tác giả cũng đã rất kiên trì trong việc thực nghiệm và thống kê các số liệu.

Tuy nhiên để hiểu rõ hơn phần thực nghiệm này thì chúng em sẽ bổ sung thêm một số thông tin:

1. Tác giả có đề cập rằng chúng ta trong thực tế không thể biết được tỷ lệ lỗi α. Điều này có thể hiểu trong thực tế là nguồn tin cung cấp để giúp chúng ta thực hiện một bài toán nào đó thường sẽ chỉ đúng một phần. Tuy là nó cũng đúng hoàn toàn nhưng sẽ giúp ít một phần nhỏ trong việc gợi ý hướng giải quyết. Do đó, việc thực nghiệm giả lập việc nguồn tin cung cấp có thể sai số bằng việc thử nhiều tỷ lệ lỗi α khác nhau. Ngoài ra với nhiều mức nhiễu mà các thuật toán của tác giả vẫn cho ra kết quả ổn định thì chứng minh được sức mạnh của chúng.
2. Trong bảng 3, cột **Lloyd** đại diện cho chi phí và thời gian để sử dụng thuật toán Lloyd để tìm nhãn tối ưu cho bộ dữ liệu. Nó chỉ được chạy 1 lần duy nhất. Ngoài ra trong bài báo gốc thì tác giả để cột này là "Ref" đại diện cho chi phí và thời gian của thuật toán Lloyd nên chúng em đã sửa lại để tường minh hơn.
3. Tuy thấy được các thuật toán tác giả đề xuất có phần cải thiện nhưng cuối cùng k-means++ vẫn là thuật toán có tốc độ chạy nhanh nhất bởi sự đơn giản dù chi phí cao hơn các thuật toán khác.

8.2 Thực nghiệm của nhóm

8.2.1 Định hướng

Ở phần này, chúng em sẽ đi tiến hành đi kiểm chứng lại thực nghiệm của tác giả bằng cách chạy lại các thuật toán qua một số tập dữ liệu. Nhóm chúng em sẽ chọn tập dữ liệu MNIST và PHY để kiểm chứng. Lí do vì các tập dữ liệu này được tác giả sử dụng và có quy mô vừa đủ cho máy tính cá nhân của chúng em chạy được.

Ngoài ra, chúng em sẽ chạy các thuật toán cho một tập dữ liệu mới mà tác giả chưa chạy qua đó là USPS với kích thước $m = 9298$ và $d = 256$. Đây là tập dữ liệu chữ số viết tay từ bao thư bưu điện Mỹ gồm các ảnh trắng đen có kích thước 16×16.

Về phần tập dữ liệu, đường dẫn đến nơi tải về các tập dữ liệu đã được xử lý này ở phần phụ lục **??**.

8.2.2 Triển khai

Về phần triển khai, rất may mắn là tác giả có cung cấp mã nguồn mà tác giả đã chạy ra được các bảng kết quả như bảng 3. Tuy nhiên thì mã nguồn ở đây giống như bản nháp hơn, do đó chúng em đã tinh chỉnh và bổ sung một số chỗ để phần triển khai của chúng em diễn ra như mong muốn. Sau đây thì chúng em sẽ

trình bày một số hàm quan trọng của mã nguồn này:

- **Hàm** `algo1`: Thuật toán của Ergun et al. (2021)

```
def algo1(points, oracle_labels, k, eps):
    """
    points: Tập dữ liệu đầu vào (n mẫu, d chiều).
    oracle_labels: Nhãn dự đoán (có thể bị nhiễu).
    k: Số lượng cụm.
    eps: Tham số epsilon (ngưỡng sai số).
    """
    # Code của tác giả
    return centers
```

- **Hàm** `detAlg`: Thuật toán Det của Nguyen et al. (2022)

```
def detAlg(points, oracle_labels, k, eps):
    """
    Sử dụng hàm con smallCluster để tối ưu hóa
    trên từng chiều dữ liệu.
    """
    # Code của tác giả
    return centers
```

- **Hàm** `Ours`: Thuật toán Fast-Sampling của tác giả.

```
def Ours(points, oracle_labels, k, p_ours):
    """
    points: Dữ liệu; p_ours: Tỷ lệ lỗi alpha ước lượng.
    Sử dụng find_minimum hoặc find_center trên mẫu ngẫu nhiên.
    """
    # Code của tác giả
    return centers
```

- **Hàm** `Ours1`: Thuật toán Fast-Filtering của tác giả.

```
def Ours1(points, oracle_labels, k, p_ours):
    """
    Sử dụng find_minimum1 với tập ứng viên omega_j.
    """
    # Code của tác giả
```

```
    return centers
```

- **Hàm** `Ours2`: Thuật toán Fast-Estimation của tác giả.

```
def Ours2(points, oracle_labels, k, p_ours):
    """
    Sử dụng generate_center_candidates để tạo tập ứng viên,
    sau đó dùng find_minimum2 để chọn tâm tốt nhất.
    """
    # Code của tác giả
    return centers
```

- **Hàm** `kpp`: Thuật toán *k*-means++ chuẩn của thư viện Scikit-Learn

```
from sklearn.cluster import kmeans_plusplus as kpp
```

Ngoài ra tác giả có tăng tốc độ xử lý của chương trình bằng dòng lệnh:

```
@jit(nopython=True)
```

Về phần chạy mã nguồn thì tác giả đã thiết kế sẵn luồng xử lý như đã đề cập ở 8.1. Tuy nhiên thì chúng em điều chỉnh lại chỉ chạy các thuật toán qua ba tập dữ liệu là MNIST, PHY và USPS. Tập dữ liệu USPS thì chúng em dùng hàm `fetch_openml` của thư viện Scikit-Learn để lấy dữ liệu từ tập dữ liệu USPS.

Xem phụ lục 9.5 để biết nơi tải các tập dữ liệu này

Về quy trình thì chúng em sẽ chạy 2 lần:

- Lần đầu tiên chạy với $k = 20$ cố định và các giá trị $alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$
- Lần thứ hai chạy với $alpha = 0.2$ cố định và các giá trị $k \in \{10, 20, 30, 40, 50\}$

Kết quả. Đối chiếu bảng 4, 5, 7, và 8 với các bảng thực nghiệm của tác giả trong phụ lục A.6 Huang et al. (2025) thì thấy kết quả tương đồng cho thấy số liệu của tác giả là chuẩn trong hai tập dữ liệu MNIST và PHY. Tuy có một số sự khác biệt nhưng có thể hiểu được vì có sự tác động của sự ngẫu nhiên trong lúc thực nghiệm. Nhưng nhìn chung thì các số liệu gần như tương đương nhau và các thuật toán của tác giả cho ra kết quả tốt. Đa số là cho ra chi phí thấp nhất và chạy nhanh nhất, đặc biệt là thuật toán Fast-Filtering.

Đối với tập dữ liệu USPS chúng em thêm vào, thuật toán Fast-Filtering cho thấy sự vượt trội trong tối ưu chi phí và tốc độ chạy. Tuy NMI và ARI trong một vài trường hợp thua các thuật toán khác nhưng thua không quá nghiêm trọng.

Bảng 4: So sánh các thuật toán trên tập dữ liệu MNIST với $k = 20$ và các α khác nhau

Phương pháp	α	Chi phí	NMI	ARI	Thời gian (s)
k-means++	0.1	1.5669E+6 ± 0.05E+6	0.6936 ± 0.0137	0.4696 ± 0.0296	**0.01 ± 0.00**
Ergun		1.0007E+6 ± 0.00E+6	0.9626 ± 0.0042	0.9557 ± 0.0053	0.37 ± 0.88
Det		9.8546E+5 ± 0.00E+5	0.9750 ± 0.0000	0.9726 ± 0.0000	0.13 ± 0.23
Fast-Sampling		9.8542E+5 ± 0.00E+5	0.9747 ± 0.0027	0.9727 ± 0.0033	**0.09 ± 0.10**
Fast-Filtering		**9.7759E+5 ± 0.00E+5**	**0.9789 ± 0.0055**	**0.9739 ± 0.0077**	0.92 ± 2.70
Fast-Estimation		9.8559E+5 ± 0.00E+5	0.9758 ± 0.0026	0.9735 ± 0.0028	1.54 ± 0.50
k-means++	0.2	1.5546E+6 ± 0.03E+6	0.6902 ± 0.0187	0.4789 ± 0.0312	**0.01 ± 0.00**
Ergun		1.0026E+6 ± 0.00E+6	0.9504 ± 0.0045	0.9319 ± 0.0068	0.08 ± 0.00
Det		9.8569E+5 ± 0.00E+5	0.9644 ± 0.0000	0.9476 ± 0.0000	0.06 ± 0.00
Fast-Sampling		9.8983E+5 ± 0.02E+5	0.9590 ± 0.0039	0.9420 ± 0.0066	0.06 ± 0.00
Fast-Filtering		**9.6316E+5 ± 0.01E+5**	**0.9681 ± 0.0082**	**0.9588 ± 0.0115**	**0.02 ± 0.00**
Fast-Estimation		9.8988E+5 ± 0.01E+5	0.9566 ± 0.0029	0.9403 ± 0.0053	1.42 ± 0.02
k-means++	0.3	1.5309E+6 ± 0.04E+6	0.7022 ± 0.0258	0.5314 ± 0.0510	**0.01 ± 0.00**
Ergun		1.0633E+6 ± 0.00E+6	0.9302 ± 0.0050	0.9160 ± 0.0073	0.08 ± 0.00
Det		1.0356E+6 ± 0.00E+6	0.9413 ± 0.0000	0.9293 ± 0.0000	0.06 ± 0.00
Fast-Sampling		1.0460E+6 ± 0.00E+6	0.9307 ± 0.0057	0.9145 ± 0.0094	0.06 ± 0.00
Fast-Filtering		**9.8494E+5 ± 0.04E+5**	**0.9501 ± 0.0072**	**0.9418 ± 0.0088**	**0.02 ± 0.00**
Fast-Estimation		1.0550E+6 ± 0.00E+6	0.9331 ± 0.0041	0.9192 ± 0.0055	1.43 ± 0.02
k-means++	0.4	1.5537E+6 ± 0.03E+6	0.6881 ± 0.0223	0.4902 ± 0.0349	**0.01 ± 0.00**
Ergun		1.1871E+6 ± 0.00E+6	0.8747 ± 0.0065	0.8358 ± 0.0108	0.08 ± 0.00
Det		1.1360E+6 ± 0.00E+6	0.8853 ± 0.0000	0.8549 ± 0.0000	0.06 ± 0.00
Fast-Sampling		1.1573E+6 ± 0.01E+6	0.8681 ± 0.0092	0.8278 ± 0.0136	0.06 ± 0.00
Fast-Filtering		**1.0807E+6 ± 0.01E+6**	**0.8920 ± 0.0107**	**0.8593 ± 0.0178**	0.02 ± 0.00
Fast-Estimation		1.1713E+6 ± 0.00E+6	0.8773 ± 0.0075	0.8422 ± 0.0093	1.45 ± 0.01
k-means++	0.5	1.5460E+6 ± 0.05E+6	0.7087 ± 0.0243	0.5384 ± 0.0391	**0.01 ± 0.00**
Ergun		1.2815E+6 ± 0.01E+6	**0.8817 ± 0.0027**	**0.8458 ± 0.0060**	0.08 ± 0.00
Det		1.2188E+6 ± 0.00E+6	0.8766 ± 0.0000	0.8398 ± 0.0000	0.06 ± 0.00
Fast-Sampling		1.2408E+6 ± 0.01E+6	0.8612 ± 0.0134	0.8161 ± 0.0207	0.06 ± 0.00
Fast-Filtering		**1.1258E+6 ± 0.01E+6**	0.8779 ± 0.0114	0.8317 ± 0.0215	**0.02 ± 0.00**
Fast-Estimation		1.2599E+6 ± 0.01E+6	0.8736 ± 0.0092	0.8314 ± 0.0138	1.46 ± 0.01

Bảng 5: So sánh các thuật toán trên tập dữ liệu PHY với $k = 20$ và các α khác nhau

Phương pháp	α	Chi phí	NMI	ARI	Thời gian (s)
k-means++	0.1	4.2706E+11 ± 0.22E+11	0.8498 ± 0.0164	0.6740 ± 0.0440	**0.10 ± 0.01**
Ergun		3.0476E+11 ± 0.00E+11	0.9916 ± 0.0011	0.9913 ± 0.0013	1.97 ± 0.02
Det		3.0413E+11 ± 0.00E+11	**0.9955 ± 0.0000**	**0.9957 ± 0.0000**	1.86 ± 0.03
Fast-Sampling		3.0841E+11 ± 0.02E+11	0.9658 ± 0.0055	0.9516 ± 0.0097	1.86 ± 0.02
Fast-Filtering		**3.0226E+11 ± 0.00E+11**	0.9905 ± 0.0019	0.9894 ± 0.0027	**0.63 ± 0.02**
Fast-Estimation		3.0482E+11 ± 0.00E+11	0.9858 ± 0.0020	0.9830 ± 0.0026	4.85 ± 0.06
k-means++	0.2	4.3996E+11 ± 0.26E+11	0.8409 ± 0.0158	0.6499 ± 0.0406	**0.10 ± 0.00**
Ergun		3.0243E+11 ± 0.00E+11	0.9825 ± 0.0023	0.9777 ± 0.0039	2.01 ± 0.02
Det		2.9970E+11 ± 0.00E+11	**0.9847 ± 0.0000**	**0.9819 ± 0.0000**	1.88 ± 0.02
Fast-Sampling		3.1242E+11 ± 0.04E+11	0.9437 ± 0.0162	0.9070 ± 0.0353	1.88 ± 0.02
Fast-Filtering		**2.9327E+11 ± 0.00E+11**	0.9791 ± 0.0058	0.9715 ± 0.0103	**0.62 ± 0.01**
Fast-Estimation		2.9884E+11 ± 0.00E+11	0.9759 ± 0.0043	0.9684 ± 0.0064	4.80 ± 0.04
k-means++	0.3	4.4495E+11 ± 0.28E+11	0.8514 ± 0.0181	0.6803 ± 0.0504	**0.10 ± 0.00**
Ergun		3.0865E+11 ± 0.00E+11	0.9822 ± 0.0034	0.9795 ± 0.0055	1.97 ± 0.03
Det		3.0638E+11 ± 0.00E+11	**0.9904 ± 0.0000**	**0.9901 ± 0.0000**	1.90 ± 0.03
Fast-Sampling		3.6131E+11 ± 0.19E+11	0.9217 ± 0.0122	0.8585 ± 0.0298	1.89 ± 0.03
Fast-Filtering		**2.9510E+11 ± 0.00E+11**	0.9776 ± 0.0031	0.9700 ± 0.0049	**0.62 ± 0.02**
Fast-Estimation		3.0643E+11 ± 0.01E+11	0.9646 ± 0.0067	0.9498 ± 0.0112	4.89 ± 0.07
k-means++	0.4	4.3429E+11 ± 0.20E+11	0.8438 ± 0.0106	0.6570 ± 0.0266	**0.09 ± 0.01**
Ergun		3.2982E+11 ± 0.00E+11	0.9790 ± 0.0025	0.9729 ± 0.0036	1.82 ± 0.06
Det		**3.2517E+11 ± 0.00E+11**	**0.9833 ± 0.0000**	**0.9799 ± 0.0000**	1.74 ± 0.05
Fast-Sampling		4.4896E+11 ± 0.52E+11	0.8902 ± 0.0183	0.7773 ± 0.0474	1.74 ± 0.04
Fast-Filtering		3.3861E+11 ± 0.04E+11	0.9440 ± 0.0058	0.9106 ± 0.0104	**0.52 ± 0.02**
Fast-Estimation		3.2754E+11 ± 0.03E+11	0.9531 ± 0.0069	0.9267 ± 0.0123	4.49 ± 0.16
k-means++	0.5	4.4461E+11 ± 0.29E+11	0.8472 ± 0.0187	0.6679 ± 0.0517	**0.10 ± 0.01**
Ergun		3.4484E+11 ± 0.01E+11	**0.9672 ± 0.0037**	**0.9574 ± 0.0062**	1.99 ± 0.07
Det		**3.4259E+11 ± 0.00E+11**	0.9640 ± 0.0000	0.9460 ± 0.0000	1.91 ± 0.06
Fast-Sampling		6.6790E+11 ± 0.66E+11	0.8730 ± 0.0138	0.7459 ± 0.0378	1.90 ± 0.06
Fast-Filtering		5.3871E+11 ± 0.07E+11	0.9152 ± 0.0075	0.8537 ± 0.0128	**0.62 ± 0.03**
Fast-Estimation		3.6809E+11 ± 0.07E+11	0.9167 ± 0.0087	0.8504 ± 0.0221	4.97 ± 0.20

Bảng 6: So sánh các thuật toán trên tập dữ liệu USPS với $k = 20$ và các α khác nhau

Phương pháp	α	Chi phí	NMI	ARI	Thời gian (s)
k-means++	0.1	4.3698E+5 ± 0.10E+5	0.5902 ± 0.0166	0.4172 ± 0.0358	**0.05 ± 0.00**
Ergun		2.9351E+5 ± 0.00E+5	0.9122 ± 0.0014	0.9019 ± 0.0024	1.57 ± 0.02
Det		2.9024E+5 ± 0.00E+5	0.9362 ± 0.0000	0.9330 ± 0.0000	1.75 ± 0.01
Fast-Sampling		2.9045E+5 ± 0.00E+5	0.9316 ± 0.0051	0.9284 ± 0.0073	1.75 ± 0.01
Fast-Filtering		**2.8835E+5 ± 0.00E+5**	**0.9499 ± 0.0067**	0.9441 ± 0.0096	**0.48 ± 0.01**
Fast-Estimation		2.9025E+5 ± 0.00E+5	0.9448 ± 0.0008	**0.9454 ± 0.0010**	11.14 ± 0.09
k-means++	0.2	4.3995E+5 ± 0.10E+5	0.5858 ± 0.0289	0.4319 ± 0.0585	**0.05 ± 0.00**
Ergun		3.0753E+5 ± 0.00E+5	0.8712 ± 0.0005	0.8502 ± 0.0011	1.56 ± 0.02
Det		2.9987E+5 ± 0.00E+5	0.8847 ± 0.0000	0.8728 ± 0.0000	1.75 ± 0.02
Fast-Sampling		3.0054E+5 ± 0.01E+5	0.8733 ± 0.0074	0.8610 ± 0.0099	1.76 ± 0.02
Fast-Filtering		**2.9203E+5 ± 0.01E+5**	**0.9078 ± 0.0067**	**0.9046 ± 0.0087**	**0.46 ± 0.01**
Fast-Estimation		2.9967E+5 ± 0.00E+5	0.8931 ± 0.0013	0.8853 ± 0.0018	11.46 ± 0.17
k-means++	0.3	4.3711E+5 ± 0.06E+5	0.5828 ± 0.0138	0.4463 ± 0.0484	**0.05 ± 0.00**
Ergun		3.2889E+5 ± 0.00E+5	0.8395 ± 0.0019	0.8159 ± 0.0020	1.58 ± 0.03
Det		3.1710E+5 ± 0.00E+5	0.8477 ± 0.0000	0.8323 ± 0.0000	1.77 ± 0.04
Fast-Sampling		3.1709E+5 ± 0.01E+5	0.8276 ± 0.0178	0.8094 ± 0.0216	1.77 ± 0.03
Fast-Filtering		**3.0416E+5 ± 0.01E+5**	0.8598 ± 0.0075	**0.8574 ± 0.0086**	**0.47 ± 0.01**
Fast-Estimation		3.1749E+5 ± 0.00E+5	**0.8622 ± 0.0017**	0.8505 ± 0.0022	11.88 ± 0.45
k-means++	0.4	4.4033E+5 ± 0.07E+5	0.5888 ± 0.0189	0.4547 ± 0.0661	**0.05 ± 0.00**
Ergun		3.5682E+5 ± 0.00E+5	0.8233 ± 0.0013	0.7967 ± 0.0017	1.58 ± 0.02
Det		3.4399E+5 ± 0.00E+5	0.8251 ± 0.0000	0.8010 ± 0.0000	1.78 ± 0.02
Fast-Sampling		3.3824E+5 ± 0.03E+5	0.7841 ± 0.0206	0.7537 ± 0.0287	1.76 ± 0.02
Fast-Filtering		**3.2096E+5 ± 0.03E+5**	0.8176 ± 0.0143	0.8080 ± 0.0228	**0.48 ± 0.01**
Fast-Estimation		3.4234E+5 ± 0.01E+5	**0.8338 ± 0.0057**	**0.8149 ± 0.0066**	11.80 ± 0.17
k-means++	0.5	4.4058E+5 ± 0.06E+5	0.5882 ± 0.0210	0.4522 ± 0.0548	**0.05 ± 0.00**
Ergun		3.9300E+5 ± 0.01E+5	0.7852 ± 0.0010	0.7355 ± 0.0020	1.58 ± 0.02
Det		3.8272E+5 ± 0.00E+5	0.7910 ± 0.0000	0.7465 ± 0.0000	1.76 ± 0.01
Fast-Sampling		3.7367E+5 ± 0.04E+5	0.7071 ± 0.0245	0.6449 ± 0.0356	1.77 ± 0.02
Fast-Filtering		**3.5936E+5 ± 0.02E+5**	0.7524 ± 0.0159	0.7180 ± 0.0161	**0.45 ± 0.01**
Fast-Estimation		3.7803E+5 ± 0.00E+5	**0.7955 ± 0.0015**	**0.7533 ± 0.0019**	11.82 ± 0.14

Bảng 7: So sánh các thuật toán trên tập dữ liệu MNIST với $\alpha = 0.2$ và các k khác nhau

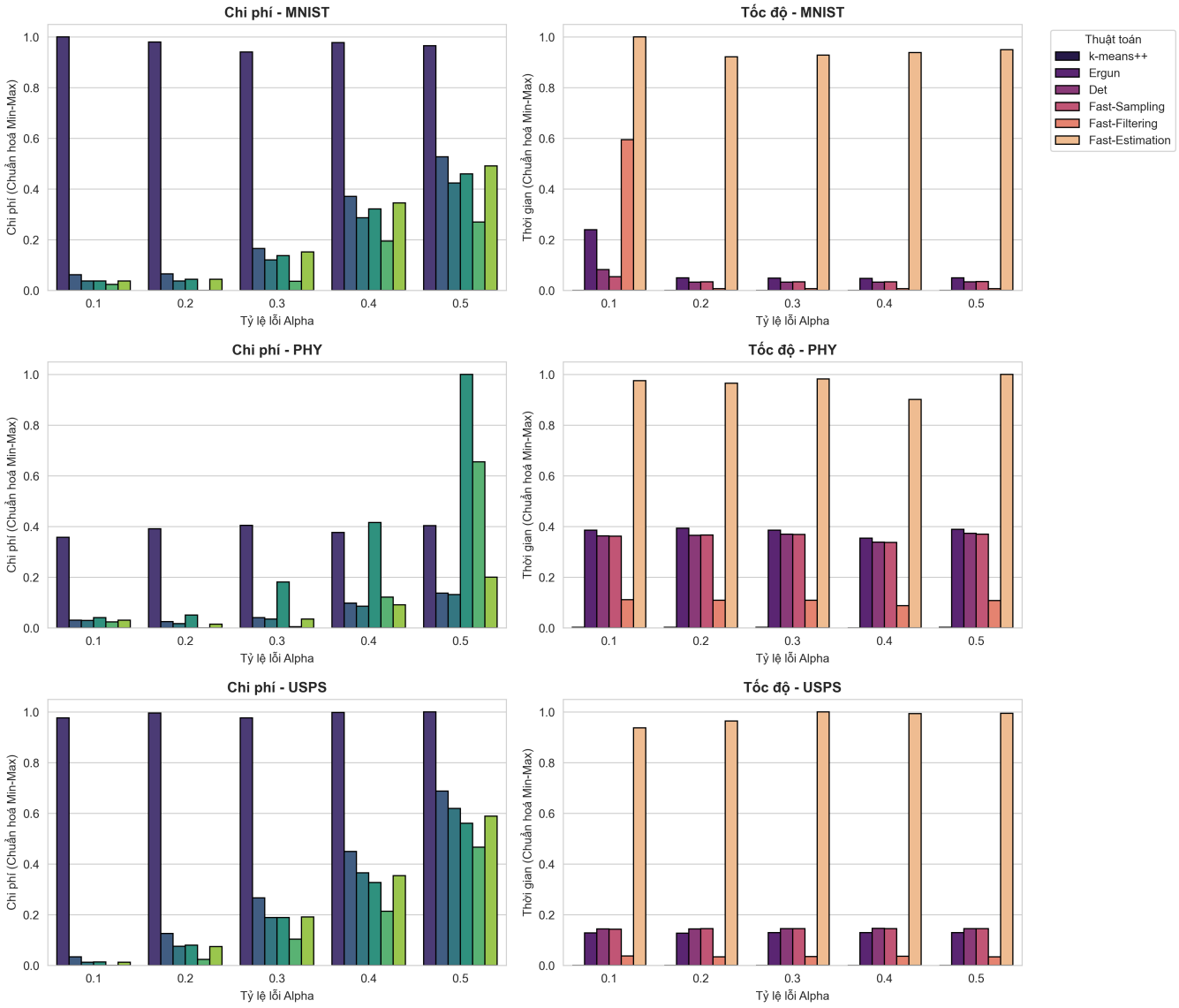
Phương pháp	k	Chi phí	NMI	ARI	Thời gian (s)
k-means++	10	2.0212E+6 ± 0.08E+6	0.6144 ± 0.0449	0.4931 ± 0.0529	**0.01 ± 0.00**
Ergun		1.2227E+6 ± 0.00E+6	0.9413 ± 0.0063	0.9395 ± 0.0068	0.37 ± 0.87
Det		1.2099E+6 ± 0.00E+6	0.9430 ± 0.0000	0.9405 ± 0.0000	0.16 ± 0.32
Fast-Sampling		1.2155E+6 ± 0.00E+6	**0.9459 ± 0.0091**	**0.9435 ± 0.0109**	**0.10 ± 0.13**
Fast-Filtering		**1.1812E+6 ± 0.00E+6**	0.9435 ± 0.0094	0.9416 ± 0.0121	1.08 ± 3.18
Fast-Estimation		1.2126E+6 ± 0.00E+6	0.9440 ± 0.0031	0.9413 ± 0.0035	1.13 ± 0.58
k-means++	20	1.5656E+6 ± 0.04E+6	0.6983 ± 0.0222	0.5214 ± 0.0452	**0.01 ± 0.00**
Ergun		1.0160E+6 ± 0.00E+6	0.9473 ± 0.0051	0.9417 ± 0.0063	0.10 ± 0.00
Det		9.9998E+5 ± 0.00E+5	**0.9590 ± 0.0000**	**0.9564 ± 0.0000**	0.06 ± 0.00
Fast-Sampling		1.0032E+6 ± 0.00E+6	0.9533 ± 0.0041	0.9494 ± 0.0050	0.07 ± 0.00
Fast-Filtering		**9.7239E+5 ± 0.01E+5**	0.9518 ± 0.0048	0.9462 ± 0.0057	**0.02 ± 0.00**
Fast-Estimation		1.0053E+6 ± 0.00E+6	0.9521 ± 0.0022	0.9490 ± 0.0027	1.69 ± 0.03
k-means++	30	1.3291E+6 ± 0.02E+6	0.7227 ± 0.0154	0.4762 ± 0.0259	**0.01 ± 0.00**
Ergun		9.0843E+5 ± 0.02E+5	0.9552 ± 0.0059	0.9374 ± 0.0085	0.12 ± 0.00
Det		8.8720E+5 ± 0.00E+5	0.9580 ± 0.0000	0.9410 ± 0.0000	0.08 ± 0.00
Fast-Sampling		8.9110E+5 ± 0.02E+5	0.9607 ± 0.0055	0.9445 ± 0.0096	0.08 ± 0.00
Fast-Filtering		**8.6222E+5 ± 0.02E+5**	**0.9630 ± 0.0108**	**0.9471 ± 0.0180**	**0.03 ± 0.00**
Fast-Estimation		8.9397E+5 ± 0.01E+5	0.9616 ± 0.0049	0.9471 ± 0.0071	2.41 ± 0.06
k-means++	40	1.2056E+6 ± 0.02E+6	0.7269 ± 0.0083	0.4355 ± 0.0228	**0.02 ± 0.00**
Ergun		8.4165E+5 ± 0.02E+5	0.9526 ± 0.0034	0.9212 ± 0.0066	0.15 ± 0.01
Det		8.1441E+5 ± 0.00E+5	**0.9686 ± 0.0000**	**0.9480 ± 0.0000**	0.09 ± 0.01
Fast-Sampling		8.2021E+5 ± 0.01E+5	0.9645 ± 0.0026	0.9432 ± 0.0043	0.10 ± 0.01
Fast-Filtering		**7.8947E+5 ± 0.02E+5**	0.9588 ± 0.0079	0.9282 ± 0.0162	**0.03 ± 0.00**
Fast-Estimation		8.2268E+5 ± 0.01E+5	0.9619 ± 0.0036	0.9391 ± 0.0068	3.18 ± 0.23
k-means++	50	1.1208E+6 ± 0.01E+6	0.7411 ± 0.0104	0.4398 ± 0.0202	**0.02 ± 0.00**
Ergun		8.0009E+5 ± 0.03E+5	0.9431 ± 0.0041	0.9009 ± 0.0074	0.17 ± 0.01
Det		7.6656E+5 ± 0.00E+5	0.9545 ± 0.0000	0.9236 ± 0.0000	0.10 ± 0.01
Fast-Sampling		7.7268E+5 ± 0.01E+5	**0.9586 ± 0.0046**	**0.9292 ± 0.0087**	0.11 ± 0.00
Fast-Filtering		**7.4156E+5 ± 0.02E+5**	0.9521 ± 0.0065	0.9096 ± 0.0143	**0.04 ± 0.00**
Fast-Estimation		7.7500E+5 ± 0.01E+5	0.9577 ± 0.0023	0.9282 ± 0.0047	3.78 ± 0.10

Bảng 8: So sánh các thuật toán trên tập dữ liệu PHY với $\alpha = 0.2$ và các k khác nhau

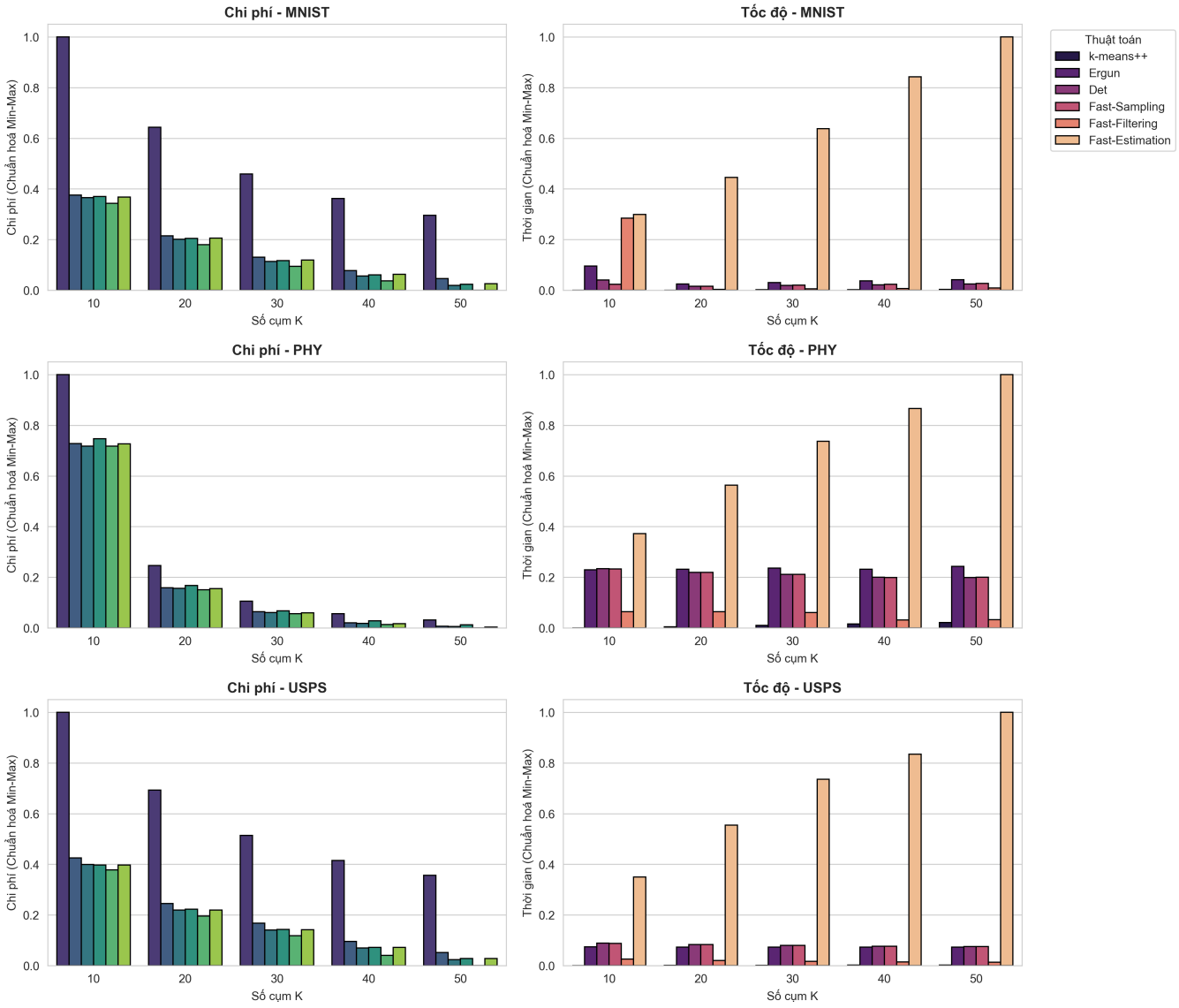
Phương pháp	k	Chi phí	NMI	ARI	Thời gian (s)
k-means++	10	1.4087E+12 ± 0.10E+12	0.7990 ± 0.0339	0.6595 ± 0.0619	**0.07 ± 0.01**
Ergun		1.0539E+12 ± 0.00E+12	0.9734 ± 0.0063	0.9690 ± 0.0117	1.94 ± 0.04
Det		**1.0403E+12 ± 0.00E+12**	**0.9799 ± 0.0000**	**0.9804 ± 0.0000**	1.98 ± 0.04
Fast-Sampling		1.0789E+12 ± 0.01E+12	0.9432 ± 0.0075	0.9272 ± 0.0088	1.97 ± 0.04
Fast-Filtering		1.0408E+12 ± 0.00E+12	0.9748 ± 0.0106	0.9726 ± 0.0158	**0.59 ± 0.02**
Fast-Estimation		1.0514E+12 ± 0.01E+12	0.9634 ± 0.0090	0.9585 ± 0.0121	3.11 ± 0.09
k-means++	20	4.2512E+11 ± 0.13E+11	0.8497 ± 0.0182	0.6706 ± 0.0475	**0.10 ± 0.00**
Ergun		3.1060E+11 ± 0.00E+11	0.9820 ± 0.0029	0.9766 ± 0.0045	1.96 ± 0.03
Det		3.0749E+11 ± 0.00E+11	**0.9869 ± 0.0000**	**0.9855 ± 0.0000**	1.86 ± 0.03
Fast-Sampling		3.2244E+11 ± 0.07E+11	0.9491 ± 0.0113	0.9193 ± 0.0225	1.86 ± 0.02
Fast-Filtering		**3.0117E+11 ± 0.00E+11**	0.9779 ± 0.0029	0.9697 ± 0.0048	**0.59 ± 0.01**
Fast-Estimation		3.0655E+11 ± 0.00E+11	0.9756 ± 0.0046	0.9681 ± 0.0081	4.68 ± 0.05
k-means++	30	2.4229E+11 ± 0.11E+11	0.8522 ± 0.0141	0.6502 ± 0.0415	**0.15 ± 0.01**
Ergun		1.8732E+11 ± 0.00E+11	0.9798 ± 0.0023	0.9697 ± 0.0030	2.00 ± 0.03
Det		1.8261E+11 ± 0.00E+11	**0.9818 ± 0.0000**	0.9746 ± 0.0000	1.80 ± 0.04
Fast-Sampling		1.9178E+11 ± 0.04E+11	0.9364 ± 0.0068	0.8813 ± 0.0173	1.79 ± 0.03
Fast-Filtering		**1.7722E+11 ± 0.00E+11**	0.9815 ± 0.0038	**0.9748 ± 0.0065**	**0.57 ± 0.03**
Fast-Estimation		1.8170E+11 ± 0.00E+11	0.9772 ± 0.0031	0.9670 ± 0.0049	6.09 ± 0.09
k-means++	40	1.7786E+11 ± 0.07E+11	0.8563 ± 0.0106	0.6366 ± 0.0357	**0.20 ± 0.01**
Ergun		1.3065E+11 ± 0.00E+11	**0.9826 ± 0.0020**	**0.9736 ± 0.0038**	1.96 ± 0.03
Det		1.2732E+11 ± 0.00E+11	0.9815 ± 0.0000	0.9702 ± 0.0000	1.71 ± 0.03
Fast-Sampling		1.4000E+11 ± 0.03E+11	0.9353 ± 0.0089	0.8650 ± 0.0251	1.70 ± 0.02
Fast-Filtering		**1.2100E+11 ± 0.00E+11**	0.9806 ± 0.0043	0.9667 ± 0.0091	**0.32 ± 0.01**
Fast-Estimation		1.2631E+11 ± 0.00E+11	0.9808 ± 0.0040	0.9712 ± 0.0078	7.15 ± 0.17
k-means++	50	1.4562E+11 ± 0.04E+11	0.8507 ± 0.0080	0.6185 ± 0.0245	**0.24 ± 0.01**
Ergun		1.1308E+11 ± 0.00E+11	0.9781 ± 0.0025	**0.9653 ± 0.0053**	2.06 ± 0.05
Det		1.1061E+11 ± 0.00E+11	0.9754 ± 0.0000	0.9579 ± 0.0000	1.70 ± 0.03
Fast-Sampling		1.2001E+11 ± 0.04E+11	0.9312 ± 0.0073	0.8495 ± 0.0213	1.70 ± 0.03
Fast-Filtering		**1.0376E+11 ± 0.00E+11**	**0.9783 ± 0.0029**	0.9615 ± 0.0061	**0.33 ± 0.01**
Fast-Estimation		1.0889E+11 ± 0.00E+11	0.9774 ± 0.0011	0.9646 ± 0.0024	8.24 ± 0.18

Bảng 9: So sánh các thuật toán trên tập dữ liệu USPS với $\alpha = 0.2$ và các k khác nhau

Phương pháp	k	Chi phí	NMI	ARI	Thời gian (s)
k-means++	10	5.3185E+5 ± 0.14E+5	0.5027 ± 0.0375	0.3823 ± 0.0577	**0.03 ± 0.01**
Ergun		3.5996E+5 ± 0.00E+5	0.8820 ± 0.0009	0.8812 ± 0.0012	1.55 ± 0.05
Det		3.5246E+5 ± 0.00E+5	0.8908 ± 0.0000	0.8937 ± 0.0000	1.84 ± 0.04
Fast-Sampling		3.5163E+5 ± 0.01E+5	0.8767 ± 0.0102	0.8808 ± 0.0116	1.82 ± 0.03
Fast-Filtering		**3.4595E+5 ± 0.01E+5**	0.8894 ± 0.0205	0.8957 ± 0.0237	**0.55 ± 0.02**
Fast-Estimation		3.5178E+5 ± 0.00E+5	**0.8943 ± 0.0021**	**0.8971 ± 0.0026**	7.12 ± 0.32
k-means++	20	4.4016E+5 ± 0.11E+5	0.5860 ± 0.0199	0.4624 ± 0.0555	**0.05 ± 0.02**
Ergun		3.0632E+5 ± 0.00E+5	0.8722 ± 0.0013	0.8561 ± 0.0013	1.53 ± 0.06
Det		2.9857E+5 ± 0.00E+5	0.8834 ± 0.0000	0.8730 ± 0.0000	1.72 ± 0.06
Fast-Sampling		2.9945E+5 ± 0.01E+5	0.8788 ± 0.0078	0.8709 ± 0.0102	1.72 ± 0.05
Fast-Filtering		**2.9159E+5 ± 0.00E+5**	**0.9084 ± 0.0128**	**0.9076 ± 0.0166**	**0.44 ± 0.03**
Fast-Estimation		2.9859E+5 ± 0.00E+5	0.8957 ± 0.0011	0.8909 ± 0.0013	11.28 ± 0.48
k-means++	30	3.8653E+5 ± 0.07E+5	0.6173 ± 0.0172	0.4570 ± 0.0550	**0.07 ± 0.01**
Ergun		2.8307E+5 ± 0.00E+5	0.8864 ± 0.0015	0.8738 ± 0.0017	1.52 ± 0.06
Det		2.7520E+5 ± 0.00E+5	0.9078 ± 0.0000	0.9051 ± 0.0000	1.65 ± 0.06
Fast-Sampling		2.7580E+5 ± 0.00E+5	0.8960 ± 0.0067	0.8904 ± 0.0083	1.65 ± 0.05
Fast-Filtering		**2.6844E+5 ± 0.00E+5**	**0.9226 ± 0.0105**	**0.9250 ± 0.0120**	**0.37 ± 0.02**
Fast-Estimation		2.7547E+5 ± 0.00E+5	0.9152 ± 0.0011	0.9122 ± 0.0013	14.95 ± 0.61
k-means++	40	3.5717E+5 ± 0.03E+5	0.6466 ± 0.0108	0.4499 ± 0.0504	**0.08 ± 0.01**
Ergun		2.6158E+5 ± 0.00E+5	0.8792 ± 0.0012	0.8372 ± 0.0023	1.51 ± 0.02
Det		2.5376E+5 ± 0.00E+5	0.9007 ± 0.0000	0.8743 ± 0.0000	1.59 ± 0.02
Fast-Sampling		2.5444E+5 ± 0.00E+5	0.8832 ± 0.0085	0.8456 ± 0.0189	1.59 ± 0.02
Fast-Filtering		**2.4527E+5 ± 0.00E+5**	**0.9371 ± 0.0093**	**0.9194 ± 0.0180**	0.33 ± 0.01
Fast-Estimation		2.5447E+5 ± 0.00E+5	0.9070 ± 0.0013	0.8824 ± 0.0022	16.95 ± 0.16
k-means++	50	3.3968E+5 ± 0.03E+5	0.6425 ± 0.0069	0.3851 ± 0.0237	**0.09 ± 0.01**
Ergun		2.4837E+5 ± 0.00E+5	0.8847 ± 0.0018	0.8233 ± 0.0032	1.51 ± 0.03
Det		2.3998E+5 ± 0.00E+5	0.9070 ± 0.0000	0.8643 ± 0.0000	1.56 ± 0.02
Fast-Sampling		2.4151E+5 ± 0.01E+5	0.8896 ± 0.0066	0.8292 ± 0.0145	1.55 ± 0.02
Fast-Filtering		**2.3296E+5 ± 0.00E+5**	**0.9361 ± 0.0080**	**0.9075 ± 0.0126**	**0.31 ± 0.01**
Fast-Estimation		2.4137E+5 ± 0.00E+5	0.9109 ± 0.0008	0.8716 ± 0.0014	20.31 ± 0.25



Hình 3: Biểu đồ tổng hợp chi phí và tốc độ của các thuật toán trên k cố định



Hình 4: Biểu đồ tổng hợp chi phí và tốc độ của các thuật toán trên α cố định

Nhận xét

Qua phần thực nghiệm của nhóm, chúng em thấy các thuật toán của tác giả hoạt động khá tốt, cho thấy các Heuristic của tác giả hoạt động tốt trong việc giảm thiểu chi phí và thời gian của thuật toán. Tuy nhiên trong triển khai code thực nghiệm của tác giả thì chúng em nghĩ có thể cải tiến thêm bằng chạy song song để giảm thời gian chờ đợi.
Ngoài ra, dựa vào hai biểu đồ 3 và 4 thì thấy được đầu ra chi phí và tốc độ có bị ảnh hưởng bởi hai tham số là k và α. Cụ thể thì ta quan sát thấy:

1. Khi cố định k thì tốc độ gần như không thấy thay đổi khi tăng α. Còn chi phí của các thuật toán thì có xu hướng tăng lên.
2. Khi cố định α thì tốc độ của các thuật toán vẫn ổn định khi thay đổi k ngoại trừ thuật toán Fast-Estimation. Còn chi phí thì có xu hướng giảm khi k giảm.

9 Các chứng minh

9.1 Kiến thức cơ sở

Lemma 1. *Cho tập $X \subset \mathbb{R}^d$ có kích thước m và một điểm dữ liệu bất kỳ $c \in \mathbb{R}^d$, ta luôn có:*

$$\delta^2(X,c) = \delta^2(X,\overline{X}) + m \cdot \delta^2(c,\overline{X}) \tag{8}$$

Arthur and Vassilvitskii (2007)

Chứng minh. Ta khai triển vế trái dựa trên định nghĩa khoảng cách Euclid:

$$\begin{aligned}
\delta^2(X,c) &= \sum_{x\in X} \|x-c\|^2 \\
&= \sum_{x\in X} \|(x-\overline{X}) + (\overline{X}-c)\|^2 \\
&= \sum_{x\in X} \left(\|x-\overline{X}\|^2 + \|\overline{X}-c\|^2 + 2\langle x-\overline{X}, \overline{X}-c\rangle\right) \\
&= \sum_{x\in X} \|x-\overline{X}\|^2 + \sum_{x\in X} \|\overline{X}-c\|^2 + 2\left\langle \sum_{x\in X}(x-\overline{X}), \overline{X}-c \right\rangle
\end{aligned}$$

Theo định nghĩa trọng tâm, ta có $\overline{X} = \frac{1}{|X|}\sum_{x\in X} x$, suy ra $\sum_{x\in X}(x-\overline{X}) = \sum x - |X|\overline{X} = 0$. Do đó, thành phần tích vô hướng (số hạng thứ 3) bằng 0. Ta thu được:

$$\delta^2(X,c) = \delta^2(X,\overline{X}) + m \cdot \|\overline{X}-c\|^2$$

□

9.2 FAST-SAMPLING

Lemma 4. *Với mọi $Q_{ij} = P^*_{ij} \cap P_{ij}$, gọi Q'_{ij} là tập con của Q_{ij} có kích thước $(1-\alpha)m_i$ và chi phí phân cụm nhỏ nhất. Gọi $G^{\mu}_{ij} = \{x \in Q'_{ij} : \delta^2(x, \overline{Q'_{ij}}) \leq \mu\delta^2(Q'_{ij}, \overline{Q'_{ij}})/|Q'_{ij}|\}$ là tập hợp các tọa độ "tốt"với hằng số $\mu > 1$. Khi đó:*

$$|G^{\mu}_{ij}| \geq \frac{\mu - 1}{\mu}|Q'_{ij}|$$

Chứng minh. Chứng minh này dựa trên phương pháp phản chứng thông qua đánh giá tổng chi phí. Chúng ta sẽ phân tích tổng chi phí phân cụm bằng cách chia tập Q'_{ij} thành hai phần rời nhau: tập tọa độ tốt G^{μ}_{ij} và tập tọa độ "xấu"(phần bù của G^{μ}_{ij} trong Q'_{ij}).

1. **Chi phí** Tổng bình phương khoảng cách từ các điểm trong Q'_{ij} đến trọng tâm $\overline{Q'_{ij}}$ chắc chắn lớn hơn hoặc bằng tổng chi phí đóng góp bởi các điểm nằm ngoài tập G^{μ}_{ij}:

$$\delta^2(Q'_{ij}, \overline{Q'_{ij}}) = \sum_{x \in G^{\mu}_{ij}} \delta^2(x, \overline{Q'_{ij}}) + \sum_{x \in Q'_{ij} \setminus G^{\mu}_{ij}} \delta^2(x, \overline{Q'_{ij}}) \geq \sum_{x \in Q'_{ij} \setminus G^{\mu}_{ij}} \delta^2(x, \overline{Q'_{ij}})$$

2. Theo định nghĩa của tập G^{μ}_{ij}, bất kỳ tọa độ x nào không thuộc tập này ($x \in Q'_{ij} \setminus G^{\mu}_{ij}$) đều thỏa mãn điều kiện khoảng cách lớn hơn:

$$\delta^2(x, \overline{Q'_{ij}}) > \mu \frac{\delta^2(Q'_{ij}, \overline{Q'_{ij}})}{|Q'_{ij}|}$$

   Số lượng phần tử nằm ngoài tập tốt là $|Q'_{ij} \setminus G^{\mu}_{ij}| = |Q'_{ij}| - |G^{\mu}_{ij}|$. Thay thế chặn dưới này vào bất đẳng thức ở Bước 1:

$$\begin{aligned}\delta^2(Q'_{ij}, \overline{Q'_{ij}}) &\geq \sum_{x \in Q'_{ij} \setminus G^{\mu}_{ij}} \left(\mu \frac{\delta^2(Q'_{ij}, \overline{Q'_{ij}})}{|Q'_{ij}|} \right) \\ &= (|Q'_{ij}| - |G^{\mu}_{ij}|) \cdot \frac{\mu\delta^2(Q'_{ij}, \overline{Q'_{ij}})}{|Q'_{ij}|}\end{aligned}$$

   Ta có thể viết lại dưới dạng tỷ lệ:

$$\delta^2(Q'_{ij}, \overline{Q'_{ij}}) \geq |Q'_{ij}| \left(1 - \frac{|G^{\mu}_{ij}|}{|Q'_{ij}|}\right) \frac{\mu\delta^2(Q'_{ij}, \overline{Q'_{ij}})}{|Q'_{ij}|}$$

3. **Chặn dưới** Giả sử chi phí phân cụm $\delta^2(Q'_{ij}, \overline{Q'_{ij}}) > 0$ (trường hợp bằng 0 thì bổ đề hiển nhiên đúng

vì tất cả các điểm trùng nhau), ta chia cả hai vế cho $\delta^2(Q'_{ij}, \overline{Q'_{ij}})$:

$$1 \geq \mu \left(1 - \frac{|G^{\mu}_{ij}|}{|Q'_{ij}|}\right)$$

$$\frac{|G^{\mu}_{ij}|}{|Q'_{ij}|} \geq \frac{\mu - 1}{\mu}$$

Như vậy

$$|G^{\mu}_{ij}| \geq \frac{\mu - 1}{\mu} |Q'_{ij}|$$

□

Corollary 0.1. *Với xác suất hằng số, đối với mỗi cụm dự đoán và mỗi chiều , tồn tại ít nhất một tọa độ sao cho .*

Chứng minh. Chứng minh dựa trên việc ước lượng xác suất thất bại và áp dụng chặn hợp (chặn hợp).

1. **Ước lượng tỷ lệ điểm tốt:** Theo Bổ đề 4, tập các tọa độ "tốt"chiếm ít nhất một nửa số lượng các tọa độ trong tập tối ưu con . Do , ta có chặn dưới cho tỷ lệ điểm tốt trong :

$$\frac{|G^{ij}_2|}{|P_{ij}|} = \frac{|G^{ij}_2|}{m_i} \geq \frac{1}{m_i} \cdot \frac{1}{2} |Q'_{ij}| = \frac{1 - \alpha}{2} \tag{9}$$

2. **Tính xác suất thất bại trên tập mẫu U_{ij}:** Với kích thước mẫu $|U_{ij}| = \frac{2}{1-\alpha} \ln(\frac{kd}{\eta})$, xác suất để *tất cả* các điểm trong U_{ij} đều không thuộc G^{ij}_2 là:

$$\begin{aligned} \Pr(\text{Thất bại tại } i, j) &= \left(1 - \frac{|G^{ij}_2|}{m_i}\right)^{|U_{ij}|} \\ &= e^{|U_{ij}| \ln\left(1 - \frac{|G^{ij}_2|}{m_i}\right)} \end{aligned}$$

Áp dụng bất đẳng thức $\ln(1 - x) \leq -x$ với $x \in (0, 1)$, ta có:

$$\Pr(\text{Thất bại tại } i, j) \leq e^{-\frac{|G^{ij}_2|}{m_i}|U_{ij}|} \leq e^{-\frac{1-\alpha}{2} \cdot \frac{2}{1-\alpha} \ln(\frac{kd}{\eta})} = e^{-\ln(\frac{kd}{\eta})} = \frac{\eta}{kd} \tag{10}$$

3. **Áp dụng Chặn hợp (chặn hợp):**

$$\Pr(\exists i, j : U_{ij} \cap G^{ij}_2 = \emptyset) \leq \sum_{i=1}^{k} \sum_{j=1}^{d} \frac{\eta}{kd} = \eta \tag{11}$$

Do đó, xác suất thành công là ít nhất $1-\eta$.

□

Lemma 5. *Cho một tọa độ bất kỳ $u \in G_{ij}^2 \cap U_{ij}$, bất đẳng thức sau luôn thỏa mãn:*

$$\delta^2(Q'_{ij}, \overline{Q'_{ij}}) \leq \delta^2(\mathcal{N}_{ij}(u), \overline{\mathcal{N}_{ij}(u)}) \leq 3\delta^2(Q'_{ij}, \overline{Q'_{ij}}) \tag{12}$$

trong đó $\overline{Q'_{ij}}$ và $\overline{\mathcal{N}_{ij}(u)}$ lần lượt là tâm hình học của tập Q'_{ij} và $\mathcal{N}_{ij}(u)$.

Chứng minh.

1. **Cận dưới:**
   Theo định nghĩa, Q'_{ij} là tập con của P_{ij} có kích thước $(1-\alpha)m_i$ tối thiểu hóa tổng bình phương khoảng cách đến tâm của nó. Vì $\mathcal{N}_{ij}(u)$ cũng là một tập con của P_{ij} với cùng kích thước $(1-\alpha)m_i$ (được xác định tại Bước 5 của Thuật toán 1), chi phí phân cụm của $\mathcal{N}_{ij}(u)$ không thể nhỏ hơn chi phí tối ưu của Q'_{ij}. Do đó:
   $$\delta^2(\mathcal{N}_{ij}(u), \overline{\mathcal{N}_{ij}(u)}) \geq \delta^2(Q'_{ij}, \overline{Q'_{ij}}) \tag{13}$$

2. **Cận trên:**
   Ta áp dụng các tính chất của tâm hình học và định nghĩa về lân cận gần nhất để biến đổi biểu thức:

   $$\begin{aligned} \delta^2(\mathcal{N}_{ij}(u), \overline{\mathcal{N}_{ij}(u)}) &\leq \delta^2(\mathcal{N}_{ij}(u), u) && (14)\\ &\leq \delta^2(Q'_{ij}, u) && (15)\\ &= \delta^2(Q'_{ij}, \overline{Q'_{ij}}) + |Q'_{ij}| \cdot \delta^2(u, \overline{Q'_{ij}}) && (16) \end{aligned}$$

   Trong đó:

   - (15) đúng vì $\mathcal{N}_{ij}(u)$ là tập hợp các điểm trong P_{ij} gần u nhất, nên tổng khoảng cách từ nó đến u nhỏ hơn hoặc bằng tổng khoảng cách từ bất kỳ tập nào khác cùng kích thước (như Q'_{ij}) đến u.
   - (16) sử dụng Bổ đề 1 (phân rã khoảng cách).

   Vì $u \in G_{ij}^2$, theo định nghĩa của tập G_{ij}^{μ} với $\mu = 2$, ta có điều kiện:

   $$|Q'_{ij}| \cdot \delta^2(u, \overline{Q'_{ij}}) \leq 2\delta^2(Q'_{ij}, \overline{Q'_{ij}})$$

   Thay thế chặn trên này vào phương trình (16), ta thu được kết quả cuối cùng:

   $$\delta^2(\mathcal{N}_{ij}(u), \overline{\mathcal{N}_{ij}(u)}) \leq \delta^2(Q'_{ij}, \overline{Q'_{ij}}) + 2\delta^2(Q'_{ij}, \overline{Q'_{ij}}) = 3\delta^2(Q'_{ij}, \overline{Q'_{ij}}) \tag{17}$$

Từ hai phần trên, bổ đề được chứng minh hoàn toàn. □

Corollary 0.2. *Với xác suất hằng số, đối với mỗi chiều $j \in [d]$ của mỗi cụm $i \in [k]$, tồn tại ít nhất một tọa độ $u' \in U'_{ij}$ sao cho:*

$$\delta(u', \overline{Q'_{ij}}) \leq \sqrt{\frac{\varepsilon \delta^2(Q'_{ij}, \overline{Q'_{ij}})}{2(1-\alpha)m_i}} \tag{18}$$

trong đó $\overline{Q'_{ij}}$ là tâm hình học của tập tối ưu Q'_{ij}.

Chứng minh. Chứng minh dựa trên sai số lượng tử hóa của lưới tọa độ được xây dựng xung quanh điểm mẫu.

1. **Sự tồn tại của khoảng chứa tâm tối ưu:** Theo bổ đề 5, tồn tại $u \in U_{ij} \cap G_2^{ij}$. Từ Bổ đề 5, trọng tâm $\overline{Q'_{ij}}$ nằm trong khoảng $[u - l_{ij}, u + l_{ij}]$ với độ dài l_{ij} được ước lượng từ tập lân cận $\mathcal{N}_{ij}(u)$.
2. **Sai số do chia lưới:** Khoảng này được chia thành các khoảng nhỏ $\varepsilon' l_{ij}$. Do lưới bao phủ toàn bộ khoảng, luôn tồn tại một điểm lưới $u' \in U'_{ij}$ nằm đủ gần $\overline{Q'_{ij}}$. Khoảng cách này bị chặn bởi:

$$\delta(u', \overline{Q'_{ij}}) \leq \varepsilon' l_{ij} \tag{19}$$

3. Thay thế các giá trị tham số $\varepsilon' = \sqrt{\frac{\varepsilon}{48}}$ và chặn trên của $l_{ij} \leq 2\sqrt{\frac{6\delta^2(Q'_{ij}, \overline{Q'_{ij}})}{(1-\alpha)m_i}}$ (từ Bổ đề 5), ta có:

$$\delta(u', \overline{Q'_{ij}}) \leq \sqrt{\frac{\varepsilon}{48}} \cdot 2\sqrt{\frac{6\delta^2(Q'_{ij}, \overline{Q'_{ij}})}{(1-\alpha)m_i}} = \sqrt{\frac{24\varepsilon\delta^2(Q'_{ij}, \overline{Q'_{ij}})}{48(1-\alpha)m_i}} = \sqrt{\frac{\varepsilon\delta^2(Q'_{ij}, \overline{Q'_{ij}})}{2(1-\alpha)m_i}} \tag{20}$$

□

Lemma 6. *Giới hạn sau đây luôn đúng đối với tập hợp các tọa độ I_{ij} được xác định bởi thuật toán Fast-Sampling so với tập giao Q_{ij}:*

$$\delta^2(\overline{I_{ij}}, \overline{Q_{ij}}) \leq \frac{(4\alpha + \alpha\varepsilon)\delta^2(Q_{ij}, \overline{Q_{ij}})}{|Q_{ij}|(1-2\alpha)}$$

Chứng minh. Chứng minh này dựa trên việc sử dụng một tập hợp trung gian (giao điểm của hai tập hợp) để bắc cầu để đánh giá khoảng cách giữa hai tâm. Quá trình chứng minh gồm 3 bước chính.

1. **Chặn trên cho chi phí của tập ứng viên I_{ij}**

   Trước hết, ta xét tập hợp I'_{ij} bao gồm $(1-\alpha)m_i$ tọa độ trong P_{ij} gần nhất với một điểm mẫu $u' \in U'_{ij}$. Theo Hệ quả 2 (Corollary 2), với xác suất hằng số, tồn tại u' sao cho u' nằm rất gần tâm $\overline{Q'_{ij}}$:

$$\delta(u', \overline{Q'_{ij}}) \leq \sqrt{\frac{\varepsilon\delta^2(Q'_{ij}, \overline{Q'_{ij}})}{2(1-\alpha)m_i}}$$

Áp dụng Bổ đề 1 và tính tối ưu của trọng tâm, ta có chặn chi phí cho I'_{ij} :

$$\begin{aligned}\delta^2(I'_{ij},\overline{I'_{ij}}) &\le \delta^2(I'_{ij},u') \\ &\le \delta^2(Q'_{ij},u') \quad \text{(Do } I'_{ij} \text{ là tập những điểm gần } u' \text{ nhất)} \\ &= \delta^2(Q'_{ij},\overline{Q'_{ij}}) + |Q'_{ij}|\delta^2(u',\overline{Q'_{ij}}) \\ &\le \delta^2(Q'_{ij},\overline{Q'_{ij}}) + |Q'_{ij}|\left(\frac{\varepsilon\delta^2(Q'_{ij},\overline{Q'_{ij}})}{2(1-\alpha)m_i}\right)\end{aligned}$$

Vì $|Q'_{ij}| = (1-\alpha)m_i$, ta thu được:

$$\delta^2(I'_{ij},\overline{I'_{ij}}) \le (1+\frac{\varepsilon}{2})\delta^2(Q'_{ij},\overline{Q'_{ij}})$$

Trong Bước 10 của Thuật toán 1, tập I_{ij} được chọn là tập có chi phí nhỏ nhất trong số các ứng viên. Do đó, chi phí của nó không vượt quá chi phí của I'_{ij} :

$$\delta^2(I_{ij},\overline{I_{ij}}) \le (1+\frac{\varepsilon}{2})\delta^2(Q'_{ij},\overline{Q'_{ij}})$$

2. **Sử dụng tập giao để bắc cầu**

Gọi $S = I_{ij} \cap Q_{ij}$ là tập giao giữa tập được chọn và tập tối ưu thực sự trong cụm dự đoán. Theo định nghĩa, $|P_{ij} \setminus Q_{ij}| \le \alpha m_i$. Do đó, khi xét giao của I_{ij} (có kích thước $(1-\alpha)m_i$) với Q_{ij}, số lượng phần tử bị mất đi tối đa là αm_i. Suy ra kích thước của tập giao:

$$|S| \ge |I_{ij}| - \alpha m_i = (1-2\alpha)m_i$$

Đặt tỷ lệ trùng lặp $\zeta = \frac{|S|}{|I_{ij}|}$. Ta áp dụng Bổ đề 2 (về quan hệ giữa chi phí của tập con và tập cha):

2a: Khoảng cách từ tâm $\overline{I_{ij}}$ đến tâm giao $\overline{S}$. Áp dụng Bổ đề 2 với $J = I_{ij}$ và $J_1 = S$:

$$\begin{aligned}\delta^2(\overline{S},\overline{I_{ij}}) &\le \frac{1-\zeta}{\zeta}\cdot\frac{\delta^2(I_{ij},\overline{I_{ij}})}{|I_{ij}|} \\ &= \frac{|I_{ij}|-|S|}{|S|}\cdot\frac{\delta^2(I_{ij},\overline{I_{ij}})}{|I_{ij}|}\end{aligned}$$

Vì $|I_{ij}| - |S| \le \alpha m_i$ và $|S| \ge (1-2\alpha)m_i$, ta có chặn trên :

$$\delta^2(\overline{S},\overline{I_{ij}}) \le \frac{\alpha m_i}{(1-2\alpha)m_i}\cdot\frac{(1+\varepsilon/2)\delta^2(Q'_{ij},\overline{Q'_{ij}})}{|I_{ij}|} \le \frac{\alpha+0.5\alpha\varepsilon}{1-2\alpha}\cdot\frac{\delta^2(Q_{ij},\overline{Q_{ij}})}{|Q_{ij}|}$$

(Lưu ý: Ta có $|I_{ij}| = |Q'_{ij}| = (1-\alpha)m_i$ và sử dụng tính chất $\delta^2(Q'_{ij},\overline{Q'_{ij}}) \le \delta^2(Q_{ij},\overline{Q_{ij}})$).

2b: Khoảng cách từ tâm $\overline{Q_{ij}}$ đến tâm giao $\overline{S}$. Tương tự, áp dụng Bổ đề 2 với $J = Q_{ij}$ và $J_1 = S$. Gọi $\zeta' = |S|/|Q_{ij}|$. Phần bù là các điểm thuộc Q_{ij} nhưng không thuộc I_{ij}, kích thước tối đa là αm_i. Ta có :

$$\delta^2(\overline{S}, \overline{Q_{ij}}) \leq \frac{\alpha m_i}{|S|} \cdot \frac{\delta^2(Q_{ij}, \overline{Q_{ij}})}{|Q_{ij}|} \leq \frac{\alpha}{1-2\alpha} \cdot \frac{\delta^2(Q_{ij}, \overline{Q_{ij}})}{|Q_{ij}|}$$

3. **Kết hợp bằng bất đẳng thức tam giác**

   Áp dụng bất đẳng thức tam giác cho khoảng cách Euclid:

$$\delta(\overline{I_{ij}}, \overline{Q_{ij}}) \leq \delta(\overline{I_{ij}}, \overline{S}) + \delta(\overline{S}, \overline{Q_{ij}})$$

   Bình phương hai vế và thế các chặn trên tìm được ở Bước 2. Đặt $K = \frac{\delta^2(Q_{ij}, \overline{Q_{ij}})}{|Q_{ij}|(1-2\alpha)}$. Ta có:

$$\delta(\overline{I_{ij}}, \overline{S}) \leq \sqrt{(\alpha + 0.5\alpha\varepsilon)K} \quad \text{và} \quad \delta(\overline{S}, \overline{Q_{ij}}) \leq \sqrt{\alpha K}$$

   Khi đó:

$$\begin{aligned}
\delta^2(\overline{I_{ij}}, \overline{Q_{ij}}) &\leq \left(\sqrt{\alpha + 0.5\alpha\varepsilon} + \sqrt{\alpha}\right)^2 K \\
&= \left(\alpha + 0.5\alpha\varepsilon + \alpha + 2\sqrt{\alpha(\alpha + 0.5\alpha\varepsilon)}\right) K \\
&= \left(2\alpha + 0.5\alpha\varepsilon + 2\alpha\sqrt{1 + 0.5\varepsilon}\right) K
\end{aligned}$$

Sử dụng bất đẳng thức $\sqrt{1+x} \leq 1 + x/2$ với $x = 0.5\varepsilon$, ta có $2\alpha\sqrt{1+0.5\varepsilon} \leq 2\alpha(1+0.25\varepsilon) = 2\alpha + 0.5\alpha\varepsilon$. Thay thế vào trên:

$$\delta^2(\overline{I_{ij}}, \overline{Q_{ij}}) \leq (2\alpha + 0.5\alpha\varepsilon + 2\alpha + 0.5\alpha\varepsilon)K = (4\alpha + \alpha\varepsilon)K$$

Thay K trở lại, ta thu được kết quả cuối cùng:

$$\delta^2(\overline{I_{ij}}, \overline{Q_{ij}}) \leq \frac{(4\alpha + \alpha\varepsilon)\delta^2(Q_{ij}, \overline{Q_{ij}})}{|Q_{ij}|(1-2\alpha)}$$

□

Lemma 7. *Khoảng cách giữa tọa độ của tâm thuật toán $\overline{I_{ij}}$ và tâm tối ưu $\overline{P^*_{ij}}$ bị chặn bởi:*

$$\delta^2(\overline{I_{ij}}, \overline{P^*_{ij}}) \leq \left(\frac{\alpha}{1-\alpha} + \frac{\alpha(4+\varepsilon)}{(1-2\alpha)(1-\alpha)}\right) \frac{\delta^2(P^*_{ij}, \overline{P^*_{ij}})}{|P^*_{ij}|}$$

Chứng minh.

1. **Quan hệ giữa các tâm**

Ta biết rằng $Q_{ij} = P_{ij} \cap P_{ij}^*$ là tập con của P_{ij}^*. Ta có thể biểu diễn tâm $\overline{P_{ij}^*}$ dưới dạng trung bình của tâm phần giao $\overline{Q_{ij}}$ và tâm phần còn lại $\overline{P_{ij}^* \setminus Q_{ij}}$:

$$|P_{ij}^*|\overline{P_{ij}^*} = |P_{ij}^* \setminus Q_{ij}|\overline{P_{ij}^* \setminus Q_{ij}} + |Q_{ij}|\overline{Q_{ij}}$$

Đặt $\gamma = \frac{|P_{ij}^* \setminus Q_{ij}|}{|P_{ij}^*|}$. Khi đó $\frac{|Q_{ij}|}{|P_{ij}^*|} = 1 - \gamma$. Phương trình trên trở thành:

$$\overline{P_{ij}^*} = \gamma\overline{P_{ij}^* \setminus Q_{ij}} + (1-\gamma)\overline{Q_{ij}}$$

Từ đó suy ra mối liên hệ khoảng cách giữa các tâm:

$$\overline{P_{ij}^*} - \overline{P_{ij}^* \setminus Q_{ij}} = -\frac{1-\gamma}{\gamma}(\overline{P_{ij}^*} - \overline{Q_{ij}})$$

Bình phương vô hướng hai vế (vì là vô hướng nên cũng là δ^2), ta được:

$$\delta^2(\overline{P_{ij}^*}, \overline{P_{ij}^* \setminus Q_{ij}}) = \left(\frac{1-\gamma}{\gamma}\right)^2 \delta^2(\overline{P_{ij}^*}, \overline{Q_{ij}})$$

2. **Phân rã chi phí phân cụm tối ưu**

$$\delta^2(P_{ij}^*, \overline{P_{ij}^*}) = \delta^2(P_{ij}^* \setminus Q_{ij}, \overline{P_{ij}^*}) + \delta^2(Q_{ij}, \overline{P_{ij}^*})$$

Tiếp tục áp dụng Bổ đề 1 cho từng số hạng:

- Với số hạng thứ nhất:

$$\delta^2(P_{ij}^* \setminus Q_{ij}, \overline{P_{ij}^*}) = \delta^2(P_{ij}^* \setminus Q_{ij}, \overline{P_{ij}^* \setminus Q_{ij}}) + |P_{ij}^* \setminus Q_{ij}|\delta^2(\overline{P_{ij}^* \setminus Q_{ij}}, \overline{P_{ij}^*})$$

- Với số hạng thứ hai:

$$\delta^2(Q_{ij}, \overline{P_{ij}^*}) = \delta^2(Q_{ij}, \overline{Q_{ij}}) + |Q_{ij}|\delta^2(\overline{Q_{ij}}, \overline{P_{ij}^*})$$

Thay thế các kết quả từ Bước 1 vào (lưu ý $|P_{ij}^* \setminus Q_{ij}| = \gamma|P_{ij}^*|$ và $|Q_{ij}| = (1-\gamma)|P_{ij}^*|$):

$$\begin{aligned}\delta^2(P_{ij}^*, \overline{P_{ij}^*}) &= \delta^2(P_{ij}^* \setminus Q_{ij}, \overline{P_{ij}^* \setminus Q_{ij}}) + \gamma|P_{ij}^*|\left(\frac{1-\gamma}{\gamma}\right)^2 \delta^2(\overline{Q_{ij}}, \overline{P_{ij}^*}) \\ &\quad + \delta^2(Q_{ij}, \overline{Q_{ij}}) + (1-\gamma)|P_{ij}^*|\delta^2(\overline{Q_{ij}}, \overline{P_{ij}^*}) \\ &= \delta^2(P_{ij}^* \setminus Q_{ij}, \overline{P_{ij}^* \setminus Q_{ij}}) + \delta^2(Q_{ij}, \overline{Q_{ij}}) + \frac{1-\gamma}{\gamma}|P_{ij}^*|\delta^2(\overline{Q_{ij}}, \overline{P_{ij}^*})\end{aligned}$$

Bỏ qua số hạng đầu tiên (không âm) và sử dụng giả thiết mô hình $\gamma \leq \alpha$ (do đó $\frac{1-\gamma}{\gamma} \geq \frac{1-\alpha}{\alpha}$), ta có chặn dưới:

$$\delta^2(P_{ij}^*, \overline{P_{ij}^*}) \geq \delta^2(Q_{ij}, \overline{Q_{ij}}) + \frac{1-\alpha}{\alpha}|P_{ij}^*|\delta^2(\overline{Q_{ij}}, \overline{P_{ij}^*})$$

3. **Kết hợp với Bổ đề 6**

Từ Bổ đề 6, ta có:

$$\delta^2(\overline{I_{ij}}, \overline{Q_{ij}}) \leq \frac{(4\alpha + \alpha\varepsilon)\delta^2(Q_{ij}, \overline{Q_{ij}})}{|Q_{ij}|(1-2\alpha)}$$

Suy ra:

$$\delta^2(Q_{ij}, \overline{Q_{ij}}) \geq \frac{|Q_{ij}|(1-2\alpha)}{4\alpha + \alpha\varepsilon}\delta^2(\overline{I_{ij}}, \overline{Q_{ij}})$$

Lại có $|Q_{ij}| \geq (1-\alpha)|P_{ij}^*|$. Thay vào bất đẳng thức cuối cùng của Bước 2:

$$\delta^2(P_{ij}^*, \overline{P_{ij}^*}) \geq |P_{ij}^*| \left[\underbrace{\frac{(1-\alpha)(1-2\alpha)}{\alpha(4+\varepsilon)}}_{C_1} \delta^2(\overline{I_{ij}}, \overline{Q_{ij}}) + \underbrace{\frac{1-\alpha}{\alpha}}_{C_2} \delta^2(\overline{Q_{ij}}, \overline{P_{ij}^*}) \right]$$

4. **Áp dụng bất đẳng thức Cauchy-Schwarz**

Ta cần tìm chặn trên cho $\delta^2(\overline{I_{ij}}, \overline{P_{ij}^*})$. Theo bất đẳng thức tam giác:

$$\delta(\overline{I_{ij}}, \overline{P_{ij}^*}) \leq \delta(\overline{I_{ij}}, \overline{Q_{ij}}) + \delta(\overline{Q_{ij}}, \overline{P_{ij}^*})$$

Đặt $x = \delta(\overline{I_{ij}}, \overline{Q_{ij}})$ và $y = \delta(\overline{Q_{ij}}, \overline{P_{ij}^*})$. Từ Bước 3, ta có:

$$C_1x^2 + C_2y^2 \leq \frac{\delta^2(P_{ij}^*, \overline{P_{ij}^*})}{|P_{ij}^*|}$$

Ta muốn chặn trên giá trị $(x+y)^2$. Áp dụng bất đẳng thức Cauchy-Schwarz cho hai vectơ $\mathbf{u} = (\sqrt{C_1}x, \sqrt{C_2}y)$ và $\mathbf{v} = (\frac{1}{\sqrt{C_1}}, \frac{1}{\sqrt{C_2}})$:

$$(x+y)^2 = \left(\sqrt{C_1}x \cdot \frac{1}{\sqrt{C_1}} + \sqrt{C_2}y \cdot \frac{1}{\sqrt{C_2}} \right)^2 \leq (C_1x^2 + C_2y^2)\left(\frac{1}{C_1} + \frac{1}{C_2}\right)$$

Thay thế vào bài toán của chúng ta:

$$\delta^2(\overline{I_{ij}}, \overline{P_{ij}^*}) \leq \frac{\delta^2(P_{ij}^*, \overline{P_{ij}^*})}{|P_{ij}^*|}\left(\frac{1}{C_2} + \frac{1}{C_1}\right)$$

Tính toán các nghịch đảo của hệ số:

$$\frac{1}{C_2} = \frac{\alpha}{1-\alpha}$$

$$\frac{1}{C_1} = \frac{\alpha(4+\varepsilon)}{(1-2\alpha)(1-\alpha)}$$

Cộng lại ta được kết quả cuối cùng :

$$\delta^2(\overline{I_{ij}}, \overline{P^*_{ij}}) \leq \left(\frac{\alpha}{1-\alpha} + \frac{\alpha(4+\varepsilon)}{(1-2\alpha)(1-\alpha)} \right) \frac{\delta^2(P^*_{ij}, \overline{P^*_{ij}})}{|P^*_{ij}|}$$

□

Theorem 1. *Tồn tại một thuật toán k-means có hỗ trợ học (Fast-Sampling) trả về một giải pháp xấp xỉ* $(1+O(\alpha))$ *trong thời gian* $O(\varepsilon^{-1} md \log(kd))$ *với xác suất hằng số, trong đó tỷ lệ lỗi nhãn thỏa mãn* $\alpha \in [0, 1/2)$.

Chứng minh.

1. Chi phí phân cụm

Giả sử $C = \{\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_k\}$ là tập hợp các tâm được thuật toán trả về. Mỗi tâm $\hat{c}_i$ được cấu thành từ các tọa độ xấp xỉ trên từng chiều j, ký hiệu là c_{ij} (trong thuật toán được xác định là $\overline{I_{ij}}$).

Tổng chi phí phân cụm $\delta^2(P,C)$ được chặn trên bởi tổng chi phí của từng cụm tối ưu đối với tâm tương ứng được gán:

$$\delta^2(P,C) \leq \sum_{i=1}^{k} \sum_{j=1}^{d} \delta^2(P^*_{ij}, c_{ij})$$

Áp dụng Bổ đề 1:

$$\delta^2(P^*_{ij}, c_{ij}) = \delta^2(P^*_{ij}, \overline{P^*_{ij}}) + |P^*_{ij}| \delta^2(\overline{P^*_{ij}}, c_{ij})$$

Sử dụng kết quả từ Bổ đề 7, ta có chặn trên cho khoảng cách giữa tâm tối ưu $\overline{P^*_{ij}}$ và tâm thuật toán c_{ij}:

$$\delta^2(\overline{P^*_{ij}}, c_{ij}) \leq \left(\frac{\alpha}{1-\alpha} + \frac{\alpha(4+\varepsilon)}{(1-2\alpha)(1-\alpha)} \right) \frac{\delta^2(P^*_{ij}, \overline{P^*_{ij}})}{|P^*_{ij}|}$$

Thay thế phương trình bổ đề 1 vào bất đẳng thức trên:

$$\begin{aligned}
\delta^2(P^*_{ij}, c_{ij}) &\leq \delta^2(P^*_{ij}, \overline{P^*_{ij}}) + |P^*_{ij}| \left[\left(\frac{\alpha}{1-\alpha} + \frac{\alpha(4+\varepsilon)}{(1-2\alpha)(1-\alpha)} \right) \frac{\delta^2(P^*_{ij}, \overline{P^*_{ij}})}{|P^*_{ij}|} \right] \\
&= \left(1 + \frac{\alpha}{1-\alpha} + \frac{\alpha(4+\varepsilon)}{(1-2\alpha)(1-\alpha)} \right) \delta^2(P^*_{ij}, \overline{P^*_{ij}})
\end{aligned}$$

Lấy tổng trên tất cả các cụm i và các chiều j, ta thu được chặn trên cho toàn bộ dữ liệu. Đặt $\mathscr{K}(\alpha) = \frac{\alpha}{1-\alpha} + \frac{4\alpha+\alpha\varepsilon}{(1-2\alpha)(1-\alpha)}$, ta có:

$$\delta^2(P,C) \leq (1+\mathscr{K}(\alpha))\delta^2(P,C^*)$$

Vì $\alpha < 1/2$, hệ số $\mathscr{K}(\alpha)$ là một hằng số phụ thuộc tuyến tính vào α (ký hiệu là $O(\alpha)$). Do đó, thuật toán đạt tỷ lệ xấp xỉ $(1+O(\alpha))$.

2. Xác suất thành công

Thuật toán thành công nếu trên mỗi chiều j của mỗi cụm i, ta tìm được ít nhất một "tọa độ tốt".

1. **Xác suất thất bại trên một mẫu:** Theo Bổ đề 4, tập hợp các tọa độ tốt G_{ij}^2 chiếm ít nhất một nửa số lượng các tọa độ trong tập tối ưu con Q'_{ij}. Do đó, tỷ lệ phần tử tốt trong toàn bộ P_{ij} là:

$$\frac{|G_{ij}^2|}{|P_{ij}|} \geq \frac{1}{m_i} \cdot \frac{(1-\alpha)m_i}{2} = \frac{1-\alpha}{2}$$

   Khi lấy ngẫu nhiên một mẫu, xác suất *không* chọn được tọa độ tốt là $1 - \frac{|G_{ij}^2|}{m_i}$.

2. **Xác suất thất bại trên tập mẫu U_{ij}:** Thuật toán lấy tập mẫu U_{ij} với kích thước $|U_{ij}| = \frac{2}{1-\alpha}\ln(\frac{kd}{\eta})$. Xác suất để *tất cả* các điểm trong U_{ij} đều không phải là tọa độ tốt là:

$$\Pr(\text{Thất bại tại } i,j) = \left(1 - \frac{|G_{ij}^2|}{m_i}\right)^{|U_{ij}|}$$

   Sử dụng bất đẳng thức $1-x \leq e^{-x}$ (suy ra từ Bernoulli), ta có:

$$\Pr(\text{Thất bại tại } i,j) \leq e^{-\frac{|G_{ij}^2|}{m_i}|U_{ij}|} \leq e^{-\frac{1-\alpha}{2}\cdot\frac{2}{1-\alpha}\ln(\frac{kd}{\eta})} = e^{-\ln(\frac{kd}{\eta})} = \frac{\eta}{kd}$$

3. **Áp dụng chặn hợp:** Để đảm bảo thành công toàn cục, ta cần thuật toán thành công trên tất cả k cụm và d chiều. Xác suất thất bại toàn cục không vượt quá tổng xác suất thất bại của từng thành phần:

$$\Pr(\text{Thất bại toàn cục}) \leq \sum_{i=1}^{k}\sum_{j=1}^{d}\Pr(\text{Thất bại tại } i,j) \leq k\cdot d\cdot\frac{\eta}{kd} = \eta$$

   Do đó, thuật toán thành công với xác suất ít nhất $1-\eta$ (xác suất hằng số).

3. Thời gian Chạy

Thời gian chạy được tính tổng trên k cụm và d chiều:

- **Lấy mẫu:** Bước 3 thực hiện lấy mẫu U_{ij} mất thời gian $O(|U_{ij}|) = O(\log(kd))$.
- **Tìm lân cận:** Bước 5 tìm $(1-\alpha)m_i$ tọa độ gần nhất. Sử dụng thuật toán lựa chọn trong thời gian

$O(m_i)$ (Linear Selection - Blum, Floyd, Pratt, Rivest, and Tarjan 1973), bước này tốn

- **Xây dựng khoảng và ứng viên:** Bước 6 và 7 chia khoảng ước lượng thành các đoạn nhỏ với tham số ε'. Số lượng ứng viên được tạo ra là $O(\varepsilon^{-1})$ cho mỗi mẫu trong U_{ij}. Tổng số ứng viên là $O(\varepsilon^{-1}\log(kd))$.
- **Chọn lọc tối ưu:** Bước 8-10 duyệt qua tất cả ứng viên để tìm tập có chi phí nhỏ nhất. Mỗi ứng viên cần tính toán trên m_i điểm, tốn $O(m_i)$. Tổng thời gian là $O(\varepsilon^{-1}m_i\log(kd))$.

Tổng hợp lại trên toàn bộ dữ liệu:

$$T = \sum_{i=1}^{k}\sum_{j=1}^{d} O(\varepsilon^{-1}m_i\log(kd)) = O(\varepsilon^{-1}\log(kd))\sum_{j=1}^{d}\sum_{i=1}^{k} m_i$$

Lưu ý rằng $\sum_{i=1}^{k} m_i = m$ (tổng số điểm dữ liệu). Do đó:

$$T = O(\varepsilon^{-1}md\log(kd))$$

□

9.3 FAST-ESTIMATION

Lemma 8. *Giả sử S_{ij} là một mẫu được lấy ngẫu nhiên từ cụm dự đoán P_{ij} với kích thước mẫu $|S_{ij}| = \tilde{O}(1/\alpha\varepsilon_1^4)$. Với xác suất ít nhất $1 - \frac{\varepsilon_1}{m^3 d\log^2(m\Delta_{\max}^2)}$, các bất đẳng thức sau đây đồng thời xảy ra cho mọi khối lớn $\mathscr{B}_u^l \in \mathscr{L}(u)$ và tập các điểm xa nhất $\mathscr{O}(u)$:*

$$(1-\varepsilon_1)\mathbb{E}[|\mathscr{B}_u^l \cap S_{ij}|] \le |\mathscr{B}_u^l \cap S_{ij}| \le (1+\varepsilon_1)\mathbb{E}[|\mathscr{B}_u^l \cap S_{ij}|]$$

$$(1-\varepsilon_1)\mathbb{E}[|\mathscr{O}(u) \cap S_{ij}|] \le |\mathscr{O}(u) \cap S_{ij}| \le (1+\varepsilon_1)\mathbb{E}[|\mathscr{O}(u) \cap S_{ij}|]$$

Chứng minh. Chúng ta sẽ phân tích chi tiết cho một khối lớn bất kỳ $\mathscr{B}_u^l \in \mathscr{L}(u)$. Quy trình tương tự cũng áp dụng cho tập $\mathscr{O}(u)$.

1. **Kỳ vọng**

   Các tọa độ trong P_{ij} được lấy mẫu độc lập và phân phối đều. Xác suất để một mẫu đơn lẻ rơi vào khối $\mathscr{B}_u^l$ là tỷ lệ kích thước $|\mathscr{B}_u^l|/|P_{ij}|$. Với tập mẫu kích thước $|S_{ij}|$, giá trị kỳ vọng số điểm rơi vào khối là:

   $$\mathbb{E}[|\mathscr{B}_u^l \cap S_{ij}|] = |S_{ij}| \cdot \frac{|\mathscr{B}_u^l|}{m_i}$$

   do tuyến tính của kỳ vọng.

Theo định nghĩa của thuật toán, kích thước mẫu $|S_{ij}|$ được là:

$$|S_{ij}| = \frac{c\log(m^3 d\log^3(m\Delta_{\max}^2)/\varepsilon_1^2)\log(m\Delta_{\max}^2)}{\alpha\varepsilon_1^4}$$

trong đó c là một hằng số đủ lớn. Theo định nghĩa của tập hợp các khối lớn $\mathscr{L}(u)$, kích thước của khối $\mathscr{B}_u^l$ phải thỏa mãn chặn dưới:

$$|\mathscr{B}_u^l| \geq \frac{\varepsilon_1^2\alpha m_i}{(1+\varepsilon_1)\log(m_i\Delta_{\max}^2)}$$

$$\begin{aligned}\mathbb{E}[|\mathscr{B}_u^l \cap S_{ij}|] &= \left(\frac{c\log(m^3 d\ldots)\log(m\Delta_{\max}^2)}{\alpha\varepsilon_1^4}\right)\cdot\left(\frac{|\mathscr{B}_u^l|}{m_i}\right)\\ &\geq \left(\frac{c\log(m^3 d\ldots)\log(m\Delta_{\max}^2)}{\alpha\varepsilon_1^4}\right)\cdot\left(\frac{\varepsilon_1^2\alpha m_i}{(1+\varepsilon_1)\log(m_i\Delta_{\max}^2)m_i}\right)\end{aligned}$$

Ta thu được:

$$\mathbb{E}[|\mathscr{B}_u^l \cap S_{ij}|] \geq \frac{c\log(m^3 d\log^3(m\Delta_{\max}^2)/\varepsilon_1^2)}{(1+\varepsilon_1)\varepsilon_1^2} \tag{21}$$

2. **Áp dụng bất đẳng thức Chernoff**

   (a) **Bất đẳng thức:** Gọi X là tổng các biến ngẫu nhiên Bernoulli $X_1,\ldots,X_{m_i}$, $X_i = 1$ nếu điểm i thuộc $\mathscr{B}_u^l \cap S_{ij}$. Áp dụng bất đẳng thức Chernoff dạng nhân cho tổng các biến Bernoulli độc lập với độ lệch tương đối $\varepsilon_1 \in (0,1)$:

$$\Pr(|X - \mathbb{E}[X]| \geq \varepsilon_1\mathbb{E}[X]) \leq 2e^{-\frac{\varepsilon_1^2\mathbb{E}[X]}{3}}$$

   (b) **Thay thế cận dưới của kỳ vọng:**

$$\begin{aligned}\text{Số mũ} &= -\frac{\varepsilon_1^2}{3}\cdot\mathbb{E}[X]\\ &\leq -\frac{\varepsilon_1^2}{3}\cdot\frac{c\ln\left(\frac{m^3 d\log^3(m\Delta_{\max}^2)}{\varepsilon_1^2}\right)}{(1+\varepsilon_1)\varepsilon_1^2}\\ &= -\frac{c}{3(1+\varepsilon_1)}\ln\left(\frac{m^3 d\log^3(m\Delta_{\max}^2)}{\varepsilon_1^2}\right)\end{aligned}$$

(c) **Biến đổi:** Đặt $\Lambda = \frac{m^3 d \log^3(m\Delta_{\max}^2)}{\varepsilon_1^2}$. Khi đó, vế phải Chernoff:

$$2e^{-\frac{c}{3(1+\varepsilon_1)}\ln(\Lambda)} = 2\Lambda^{-\frac{c}{3(1+\varepsilon_1)}}$$

Để đảm bảo xác suất thất bại đủ nhỏ, ta chọn hằng số c đủ lớn sao cho số mũ $\frac{c}{3(1+\varepsilon_1)} \geq 1$. Khi đó:

$$\begin{aligned}
\Pr(\text{Thất bại tại } \mathscr{B}_u^l) &\leq 2\Lambda^{-\frac{c}{3(1+\varepsilon_1)}} \\
&\leq 2\Lambda^{-1} \\
&= 2\left(\frac{m^3 d \log^3(m\Delta_{\max}^2)}{\varepsilon_1^2}\right)^{-1} \\
&= \frac{2\varepsilon_1^2}{m^3 d \log^3(m\Delta_{\max}^2)}
\end{aligned}$$

$$\Pr(|X - \mathbb{E}[X]| \geq \varepsilon_1 \mathbb{E}[X]) \leq O\left(\frac{\varepsilon_1^2}{m^3 d \log^3(m\Delta_{\max}^2)}\right)$$

3. **Chặn hợp**

Bổ đề yêu cầu bất đẳng thức đúng cho *tất cả* các khối lớn. Số lượng khối lớn γ bị chặn bởi $O(\log(m\Delta_{\max}^2)/\varepsilon_1)$. Áp dụng chặn hợp để tính tổng xác suất thất bại:

$$\begin{aligned}
\Pr(\exists \mathscr{B}_u^l \text{ vi phạm}) &\leq \sum_{l=1}^{\gamma} \Pr(\text{Thất bại tại } \mathscr{B}_u^l) \\
&\leq \gamma \cdot O\left(\frac{\varepsilon_1^2}{m^3 d \log^3(m\Delta_{\max}^2)}\right) \\
&\leq \frac{\varepsilon_1}{m^3 d \log^2(m\Delta_{\max}^2)}
\end{aligned}$$

Đối với tập ngoại lai $\mathscr{O}(u)$, vì kích thước $|\mathscr{O}(u)| = \alpha m_i$ lớn hơn kích thước tối thiểu của khối lớn, kết quả tương tự cũng được áp dụng.

□

Lemma 9. *Gọi $\mathscr{J}(u)$ là tập hợp các tọa độ nằm trong các khối nhỏ đối với một tọa độ ứng viên u. Với xác suất ít nhất $1 - \frac{\varepsilon_1}{m^3 d \log^2(m\Delta_{\max}^2)}$, giao của tập mẫu S_{ij} và $\mathscr{J}(u)$ bị chặn như sau:*

$$|\mathscr{J}(u) \cap S_{ij}| \leq 2\varepsilon_1 \alpha |S_{ij}|$$

Chứng minh. Chứng minh này dựa trên việc áp dụng Bất đẳng thức Chernoff để giới hạn độ lệch của biến ngẫu nhiên so với kỳ vọng của nó.

1. Gọi biến ngẫu nhiên $X = |\mathscr{J}(u) \cap S_{ij}|$. Vì S_{ij} được lấy mẫu ngẫu nhiên đều từ P_{ij}, giá trị kỳ vọng của X được tính bằng tỷ lệ kích thước:

$$\mathbb{E}[X] = |S_{ij}| \cdot \frac{|\mathscr{J}(u)|}{m_i}$$

2. **Chuẩn bị áp dụng bất đẳng thức Chernoff** Chúng ta muốn chứng minh rằng X không vượt quá ngưỡng $2\varepsilon_1\alpha|S_{ij}|$. Để làm điều này, ta biểu diễn ngưỡng này dưới dạng độ lệch so với kỳ vọng $(1+\lambda')\mathbb{E}[X]$. Ta cần tìm λ' sao cho:

$$(1+\lambda')\mathbb{E}[X] = 2\varepsilon_1\alpha|S_{ij}|$$

Thay thế $\mathbb{E}[X]$ vào phương trình trên:

$$(1+\lambda')\left(|S_{ij}|\frac{|\mathscr{J}(u)|}{m_i}\right) = 2\varepsilon_1\alpha|S_{ij}|$$

Giải phương trình tìm λ':

$$1+\lambda' = \frac{2\varepsilon_1\alpha m_i}{|\mathscr{J}(u)|} \quad \Rightarrow \quad \lambda' = \frac{2\varepsilon_1\alpha m_i}{|\mathscr{J}(u)|} - 1$$

vì $|\mathscr{J}(u)| \leq \varepsilon_1\alpha m_i$, ta có tỷ số $\frac{\varepsilon_1\alpha m_i}{|\mathscr{J}(u)|} \geq 1$, suy ra $\frac{2\varepsilon_1\alpha m_i}{|\mathscr{J}(u)|} \geq 2$, do đó $\lambda' \geq 1$.

3. **Bất đẳng thức**

Đặt biến ngẫu nhiên $X = |\mathscr{J}(u) \cap S_{ij}|$. Ta muốn chặn trên xác suất X vượt quá ngưỡng $2\varepsilon_1\alpha|S_{ij}|$. Đặt độ lệch $\lambda' = \frac{2\varepsilon_1\alpha m_i}{|\mathscr{J}(u)|} - 1$. Khi đó, ngưỡng cần chặn chính là $(1+\lambda')\mathbb{E}[X]$.

Áp dụng bất đẳng thức Chernoff dạng nhân:

$$\Pr(X \geq (1+\lambda')\mathbb{E}[X]) \leq e^{-\frac{\mathbb{E}[X](\lambda')^2}{3}}$$

Ta xét số mũ $\mathscr{E} = \frac{\mathbb{E}[X](\lambda')^2}{3}$. Thay thế $\mathbb{E}[X] = \frac{|S_{ij}||\mathscr{J}(u)|}{m_i}$ và giá trị của λ':

$$\mathscr{E} = \frac{|S_{ij}||\mathscr{J}(u)|}{3m_i}\left(\frac{2\varepsilon_1\alpha m_i}{|\mathscr{J}(u)|} - 1\right)^2$$

Để tìm chặn dưới cho số mũ $\mathscr{E}$, ta thực hiện biến đổi đại số sau. Đặt $A = \frac{\varepsilon_1\alpha m_i}{|\mathscr{J}(u)|}$. Theo định nghĩa khối nhỏ, $|\mathscr{J}(u)| \leq \varepsilon_1\alpha m_i$, suy ra $A \geq 1$. Ta có: $(2A-1)^2 \geq A^2 \Leftrightarrow A \geq 1$.

Áp dụng vào biểu thức của $\mathscr{E}$:

$$\begin{aligned}\mathscr{E} &\geq \frac{|S_{ij}||\mathscr{J}(u)|}{3m_i}\left(\frac{\varepsilon_1\alpha m_i}{|\mathscr{J}(u)|}\right)^2 \\ &= \frac{|S_{ij}||\mathscr{J}(u)|}{3m_i}\cdot\frac{\varepsilon_1^2\alpha^2 m_i^2}{|\mathscr{J}(u)|^2} \\ &= \frac{\varepsilon_1^2\alpha^2 m_i|S_{ij}|}{3|\mathscr{J}(u)|}\end{aligned}$$

Để $\mathscr{E}$ nhỏ nhất, ta thay $|\mathscr{J}(u)|$ bằng giá trị lớn nhất:

$$\mathscr{E} \geq \frac{\varepsilon_1^2\alpha^2 m_i|S_{ij}|}{3(\varepsilon_1\alpha m_i)} = \frac{\varepsilon_1\alpha|S_{ij}|}{3}$$

4. Theo thuật toán, kích thước mẫu $|S_{ij}|$ được chọn là:

$$|S_{ij}| = \Omega\left(\frac{\log(m^3 d\log^3(m\Delta_{\max}^2)/\varepsilon_1^2)\log(m\Delta_{\max}^2)}{\alpha\varepsilon_1^4}\right)$$

Thay thế $|S_{ij}|$ vào chặn dưới của số mũ $\mathscr{E}$ tìm được ở Bước 3:

$$\mathscr{E} \geq \frac{\varepsilon_1\alpha}{3}\cdot\frac{C\cdot\ln(\dots)}{\alpha\varepsilon_1^4} = \frac{C\cdot\ln(\dots)}{3\varepsilon_1^3}$$

Vì $\varepsilon_1 < 1$ và C là hằng số đủ lớn, ta có:

$$e^{-\mathscr{E}} \leq \frac{\varepsilon_1}{m^3 d\log^2(m\Delta_{\max}^2)}$$

Do đó:

$$\Pr(|\mathscr{J}(u)\cap S_{ij}| \geq 2\varepsilon_1\alpha|S_{ij}|) \leq \frac{\varepsilon_1}{m^3 d\log^2(m\Delta_{\max}^2)}$$

Lấy phần bù, ta có điều phải chứng minh.

□

Lemma 10. *Cho một tọa độ ứng viên bất kỳ $u \in U'_{ij}$. Với xác suất cao (xác suất hằng số), ước lượng $\omega(u)$ thỏa mãn các chặn sau:*

$$\frac{\delta^2(P_{ij}\setminus\mathscr{F}^\dagger(u),u)}{1+7\varepsilon_1} \leq \omega(u) \leq (1+\varepsilon_1)^2\delta^2(\mathscr{N}_{ij}(u),u)$$

trong đó:

- $\mathscr{F}^{\dagger}(u)$ *là tập hợp gồm* $(2+20\varepsilon_1)\alpha m_i$ *tọa độ xa nhất từ* P_{ij} *đến* u.
- $\mathscr{N}_{ij}(u)$ *là tập hợp gồm* $(1-\alpha)m_i$ *tọa độ gần nhất trong* P_{ij} *đến* u.

Chứng minh. Theo Bổ đề 8 và 9, với xác suất ít nhất $1-\frac{\varepsilon_1}{m^2 d\log^2(m\Delta^2_{\max})}$, các điều kiện sau đây đồng thời xảy ra đối với tập mẫu ngẫu nhiên S_{ij}:

1. Số lượng phần tử thuộc các khối nhỏ trong mẫu: $|\mathscr{J}(u)\cap S_{ij}| \le 2\varepsilon_1\alpha|S_{ij}|$.
2. Số lượng phần tử ngoại lai trong mẫu: $|\mathscr{O}(u)\cap S_{ij}| \le (1+\varepsilon_1)\alpha|S_{ij}|$.
3. Với mọi khối lớn $\mathscr{B}^l_u \in \mathscr{L}(u)$, số lượng phần tử trong mẫu xấp xỉ giá trị kỳ vọng:

$$(1-\varepsilon_1)\frac{|S_{ij}|}{m_i}|\mathscr{B}^l_u| \le |\mathscr{B}^l_u\cap S_{ij}| \le (1+\varepsilon_1)\frac{|S_{ij}|}{m_i}|\mathscr{B}^l_u|$$

Chúng ta áp dụng chặn hợp để đảm bảo các điều kiện này đúng cho mọi $u \in U'_{ij}$ với xác suất hằng số.

1. Chặn trên

Mục tiêu là chứng minh $\omega(u) \le (1+\varepsilon_1)^2\delta^2(\mathscr{N}_{ij}(u),u)$.

Gọi $\mathscr{F}'(u) = (\mathscr{J}(u)\cup\mathscr{O}(u))\cap S_{ij}$ là tập hợp các điểm thuộc khối nhỏ và các điểm ngoại lai nằm trong mẫu. Kích thước của tập này bị chặn bởi:

$$|\mathscr{F}'(u)| = |\mathscr{J}(u)\cap S_{ij}| + |\mathscr{O}(u)\cap S_{ij}| \le 2\varepsilon_1\alpha|S_{ij}| + (1+\varepsilon_1)\alpha|S_{ij}| = (1+3\varepsilon_1)\alpha|S_{ij}|$$

Theo định nghĩa trong thuật toán, $\mathscr{F}(u)$ là tập hợp gồm $(1+3\varepsilon_1)\alpha|S_{ij}|$ điểm *xa nhất* từ S_{ij} đến u. Do đó, $|\mathscr{F}(u)| \ge |\mathscr{F}'(u)|$. Vì $\omega(u)$ tính tổng chi phí sau khi loại bỏ những điểm xa nhất ($\mathscr{F}(u)$), giá trị này sẽ nhỏ hơn hoặc bằng chi phí khi loại bỏ tập $\mathscr{F}'(u)$:

$$\omega(u) = \frac{m_i}{|S_{ij}|}\delta^2(S_{ij}\setminus\mathscr{F}(u),u) \le \frac{m_i}{|S_{ij}|}\delta^2(S_{ij}\setminus\mathscr{F}'(u),u)$$

Khi loại bỏ $\mathscr{F}'(u)$, phần còn lại của mẫu S_{ij} chỉ chứa các điểm thuộc các khối lớn $\mathscr{L}(u)$. Ta có:

$$\delta^2(S_{ij}\setminus\mathscr{F}'(u),u) = \sum_{\mathscr{B}^l_u\in\mathscr{L}(u)}\delta^2(\mathscr{B}^l_u\cap S_{ij},u)$$

$$\begin{aligned}\delta^2(\mathscr{B}_u^l \cap S_{ij}, u) &< |\mathscr{B}_u^l \cap S_{ij}| \cdot (1+\varepsilon_1)^{l+1} \\ &\leq \left((1+\varepsilon_1)\frac{|S_{ij}|}{m_i}|\mathscr{B}_u^l|\right) \cdot (1+\varepsilon_1)^{l+1} \quad \text{(từ Bổ đề 8)} \\ &= \frac{|S_{ij}|}{m_i}(1+\varepsilon_1)^2\left(|\mathscr{B}_u^l|(1+\varepsilon_1)^l\right) \\ &\leq \frac{|S_{ij}|}{m_i}(1+\varepsilon_1)^2\delta^2(\mathscr{B}_u^l, u)\end{aligned}$$

$$\omega(u) \leq \frac{m_i}{|S_{ij}|}\sum_{\mathscr{B}_u^l \in \mathscr{L}(u)} \frac{|S_{ij}|}{m_i}(1+\varepsilon_1)^2\delta^2(\mathscr{B}_u^l, u) \leq (1+\varepsilon_1)^2\delta^2(\mathscr{N}_{ij}(u), u)$$

2. Chặn dưới

Với mỗi khối lớn $\mathscr{B}_u^l \in \mathscr{L}(u)$, gọi $\mathscr{Z}_u^l = \mathscr{F}(u) \cap \mathscr{B}_u^l$ là các điểm thuộc khối này bị loại bỏ trong mẫu. Gọi $\mathscr{H}_u^l$ là tập con (tùy ý) trong tập $\mathscr{B}_u^l$ sao cho:

$$|\mathscr{H}_u^l| = \left\lceil (1+3\varepsilon_1)\frac{m_i}{|S_{ij}|}|\mathscr{Z}_u^l| \right\rceil$$

Đặt $\mathscr{F}''(u)$ là tập hợp các điểm "bị loại bỏ"trên toàn bộ dữ liệu, bao gồm các điểm ngoại lai, các khối nhỏ và các phần tỉ lệ từ khối lớn:

$$\mathscr{F}''(u) = \mathscr{O}(u) \cup \mathscr{J}(u) \cup \left(\bigcup_{\mathscr{B}_u^l \in \mathscr{L}(u)} \mathscr{H}_u^l\right)$$

Ta ước tính kích thước của $\mathscr{F}''(u)$:

$$\begin{aligned}|\mathscr{F}''(u)| &\leq |\mathscr{O}(u)| + |\mathscr{J}(u)| + \sum_{\mathscr{B}_u^l} |\mathscr{H}_u^l| \\ &\leq \alpha m_i + \varepsilon_1 \alpha m_i + (1+3\varepsilon_1)\frac{m_i}{|S_{ij}|}\sum_{\mathscr{B}_u^l} |\mathscr{Z}_u^l|\end{aligned}$$

$\sum |\mathscr{Z}_u^l| \leq |\mathscr{F}(u)| \leq (1+3\varepsilon_1)\alpha|S_{ij}|$. Do đó:

$$\begin{aligned}|\mathscr{F}''(u)| &\leq \alpha m_i(1+\varepsilon_1) + (1+3\varepsilon_1)^2\alpha m_i \\ &\leq \alpha m_i(2+20\varepsilon_1)\end{aligned}$$

Theo định nghĩa, $\mathscr{F}^\dagger(u)$ là tập hợp gồm $(2+20\varepsilon_1)\alpha m_i$ điểm xa nhất trong P_{ij}. Do đó, việc loại bỏ $\mathscr{F}^\dagger(u)$ sẽ làm giảm chi phí nhiều hơn hoặc bằng việc loại bỏ $\mathscr{F}''(u)$:

$$\delta^2(P_{ij} \setminus \mathscr{F}^\dagger(u), u) \leq \delta^2(P_{ij} \setminus \mathscr{F}''(u), u)$$

Chi phí còn lại sau khi loại bỏ $\mathscr{F}''(u)$ là tổng chi phí của các khối lớn sau khi trừ đi $\mathscr{H}_u^l$. Sử dụng chặn trên khoảng cách $(1+\varepsilon_1)^{l+1}$ trong khối $\mathscr{B}_u^l$:

$$\delta^2(P_{ij} \setminus \mathscr{F}''(u), u) = \sum_{\mathscr{B}_u^l} \delta^2(\mathscr{B}_u^l \setminus \mathscr{H}_u^l, u) \leq \sum_{\mathscr{B}_u^l} (1+\varepsilon_1)^{l+1} (|\mathscr{B}_u^l| - |\mathscr{H}_u^l|)$$

Từ Bổ đề 8, ta có $|\mathscr{B}_u^l| \leq \frac{m_i}{|S_{ij}|(1-\varepsilon_1)} |\mathscr{B}_u^l \cap S_{ij}|$. Thay thế vào bất đẳng thức:

$$\begin{aligned} |\mathscr{B}_u^l| - |\mathscr{H}_u^l| &\leq \frac{m_i}{|S_{ij}|(1-\varepsilon_1)} |\mathscr{B}_u^l \cap S_{ij}| - (1+3\varepsilon_1) \frac{m_i}{|S_{ij}|} |\mathscr{Z}_u^l| \\ &= \frac{m_i}{|S_{ij}|} \left(\frac{1}{1-\varepsilon_1} |\mathscr{B}_u^l \cap S_{ij}| - (1+3\varepsilon_1) |\mathscr{Z}_u^l| \right) \end{aligned}$$

Với $\varepsilon_1 < 0.5$, ta có $\frac{1}{1-\varepsilon_1} \leq 1+3\varepsilon_1$.

$$|\mathscr{B}_u^l| - |\mathscr{H}_u^l| \leq (1+3\varepsilon_1) \frac{m_i}{|S_{ij}|} (|\mathscr{B}_u^l \cap S_{ij}| - |\mathscr{Z}_u^l|)$$

Thay thế trở lại công thức tổng chi phí:

$$\begin{aligned} \delta^2(P_{ij} \setminus \mathscr{F}^\dagger(u), u) &\leq \sum_{\mathscr{B}_u^l} (1+\varepsilon_1)^{l+1} (1+3\varepsilon_1) \frac{m_i}{|S_{ij}|} (|\mathscr{B}_u^l \cap S_{ij}| - |\mathscr{Z}_u^l|) \\ &= (1+\varepsilon_1)(1+3\varepsilon_1) \frac{m_i}{|S_{ij}|} \sum_{\mathscr{B}_u^l} (1+\varepsilon_1)^l (|\mathscr{B}_u^l \cap S_{ij}| - |\mathscr{Z}_u^l|) \\ &\leq (1+7\varepsilon_1) \frac{m_i}{|S_{ij}|} \sum_{\mathscr{B}_u^l} \delta^2((\mathscr{B}_u^l \cap S_{ij}) \setminus \mathscr{Z}_u^l, u) \end{aligned}$$

Do đó:

$$\delta^2(P_{ij} \setminus \mathscr{F}^\dagger(u), u) \leq (1+7\varepsilon_1)\omega(u)$$

□

Lemma 11. *Với tập hợp các tọa độ I_{ij} được xác định bởi thuật toán Fast-Estimation, chặn sau đây luôn thỏa mãn:*

$$\delta^2(\overline{I_{ij}}, \overline{Q_{ij}}) \leq \frac{13\alpha - 15\alpha^2}{(1-3\alpha-\varepsilon)(1-2\alpha-\varepsilon)} \frac{\delta^2(Q_{ij}, \overline{Q_{ij}})}{|Q_{ij}|}$$

Chứng minh. Chứng minh được chia thành ba giai đoạn chính: xác định sự tồn tại của ứng viên tốt, giới hạn chi phí của ứng viên được chọn, và sử dụng kỹ thuật cầu nối để giới hạn khoảng cách giữa các tâm.

1. Tọa độ ứng viên tốt

Theo Bổ đề 4 và Bổ đề 5, với xác suất hằng số, tồn tại ít nhất một tọa độ $u_1 \in U'_{ij}$ nằm rất gần trọng tâm

của tập Q'_{ij} (tập con của Q_{ij} có chi phí nhỏ nhất với kích thước $(1-\alpha)m_i$). Cụ thể:

$$\delta^2(u_1, \overline{Q'_{ij}}) \leq \frac{\varepsilon_1 \delta^2(Q'_{ij}, \overline{Q'_{ij}})}{|Q'_{ij}|}$$

Sử dụng Bổ đề 1 , ta liên hệ chi phí của tập các điểm lân cận $\mathscr{N}_{ij}(u_1)$ với chi phí tối ưu:

$$\delta^2(\mathscr{N}_{ij}(u_1), u_1) \leq \delta^2(Q'_{ij}, u_1) = \delta^2(Q'_{ij}, \overline{Q'_{ij}}) + |Q'_{ij}|\delta^2(u_1, \overline{Q'_{ij}})$$

Thay thế chặn của u_1 vào, ta có:

$$\delta^2(\mathscr{N}_{ij}(u_1), u_1) \leq (1+\varepsilon_1)\delta^2(Q'_{ij}, \overline{Q'_{ij}})$$

2. Giới hạn chi phí của tập được chọn I_{ij}

Gọi c_{ij} là tọa độ được bộ ước lượng ω chọn ở Bước 11 của Thuật toán 2. Do c_{ij} tối thiểu hóa ω trên U'_{ij}, ta có $\omega(c_{ij}) \leq \omega(u_1)$. Kết hợp với các chặn của bộ ước lượng từ Bổ đề 10:

$$\frac{\delta^2(P_{ij} \setminus \mathscr{F}^\dagger(c_{ij}), c_{ij})}{1+7\varepsilon_1} \leq \omega(c_{ij}) \leq \omega(u_1) \leq (1+\varepsilon_1)^2 \delta^2(\mathscr{N}_{ij}(u_1), u_1)$$

Từ đó suy ra chặn trên cho chi phí thực tế của c_{ij}:

$$\delta^2(I_{ij}, c_{ij}) \leq (1+7\varepsilon_1)\omega(c_{ij}) \leq (1+\varepsilon_1)^3(1+7\varepsilon_1)\delta^2(Q'_{ij}, \overline{Q'_{ij}})$$

Bằng cách chọn $\varepsilon_1 = \varepsilon/126$, và $\delta^2(I_{ij}, \overline{I_{ij}}) \leq \delta^2(I_{ij}, c_{ij})$, ta thu được:

$$\delta^2(I_{ij}, \overline{I_{ij}}) \leq (1+\varepsilon/2)\delta^2(Q'_{ij}, \overline{Q'_{ij}})$$

3. Giới hạn khoảng cách tâm bằng bắc cầu

Để giới hạn $\delta^2(\overline{I_{ij}}, \overline{Q_{ij}})$, ta sử dụng giao tập hợp $S = I_{ij} \cap Q_{ij}$ làm cầu nối và áp dụng Bất đẳng thức tam giác cho khoảng cách Euclid:

$$\delta(\overline{I_{ij}}, \overline{Q_{ij}}) \leq \delta(\overline{I_{ij}}, \overline{S}) + \delta(\overline{S}, \overline{Q_{ij}})$$

Áp dụng Bổ đề 6 (đã được chứng minh cho Fast-Sampling và mở rộng cho Fast-Estimation), ta có các chặn sau cho từng thành phần khoảng cách:

$$\delta^2(\overline{I_{ij}}, \overline{S}) \leq \frac{(2\alpha+\alpha\varepsilon)(1+\varepsilon)}{(1-3\alpha-\varepsilon)} \frac{|Q'_{ij}|}{|I_{ij}||Q_{ij}|} \delta^2(Q_{ij}, \overline{Q_{ij}})$$

Do $|I_{ij}| = |Q'_{ij}|$:

$$\delta^2(\overline{I_{ij}}, \overline{S}) \leq \frac{(2\alpha + \alpha\varepsilon)(1+\varepsilon)(1-\alpha)}{(1-3\alpha-\varepsilon)(1-2\alpha-\varepsilon)} \frac{\delta^2(Q_{ij}, \overline{Q_{ij}})}{|Q_{ij}|}$$

Tương tự:

$$\delta^2(\overline{Q_{ij}}, \overline{S}) \leq \frac{2\alpha + \alpha\varepsilon}{1-3\alpha-\varepsilon} \frac{\delta^2(Q_{ij}, \overline{Q_{ij}})}{|Q_{ij}|}$$

Kết hợp lại, bình phương tổng các khoảng cách và thực hiện các phép biến đổi đại số với điều kiện $\varepsilon < 0.5$ và $\alpha < 1/3$, ta thu được chặn cuối cùng:

$$\begin{aligned}\delta^2(\overline{I_{ij}}, \overline{Q_{ij}}) &\leq \left(\sqrt{\delta^2(\overline{I_{ij}}, \overline{S})} + \sqrt{\delta^2(\overline{S}, \overline{Q_{ij}})}\right)^2 \\ &\leq \frac{13\alpha - 15\alpha^2}{(1-3\alpha-\varepsilon)(1-2\alpha-\varepsilon)} \frac{\delta^2(Q_{ij}, \overline{Q_{ij}})}{|Q_{ij}|}\end{aligned}$$

□

Theorem 2. *Thuật toán Fast-Estimation xấp xỉ* $(1+O(\alpha))$ *cho bài toán k-means có hỗ trợ học (learning-augmented) trong thời gian* $O(md) + \tilde{O}(\varepsilon^{-5}kd/\alpha)$ *với xác suất hằng số, với tỷ lệ lỗi nhãn* $\alpha \in (0, 1/3-\varepsilon)$.

Chứng minh.

1. Chất lượng Phân cụm

Giả sử $C = \{\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_k\}$ là tập hợp các tâm được thuật toán trả về, trong đó mỗi tâm $\hat{c}_i$ được cấu thành từ các tọa độ trên từng chiều j, ký hiệu là c_{ij} (trong thuật toán được xác định là $\overline{I_{ij}}$).

Tổng chi phí phân cụm $\delta^2(P,C)$ có thể được phân rã theo từng cụm tối ưu P_i^* và từng chiều j:

$$\delta^2(P,C) \leq \sum_{i=1}^{k} \sum_{j=1}^{d} \delta^2(P_{ij}^*, c_{ij})$$

$$\delta^2(P_{ij}^*, c_{ij}) = \delta^2(P_{ij}^*, \overline{P_{ij}^*}) + |P_{ij}^*| \delta^2(\overline{P_{ij}^*}, c_{ij})$$

Dựa vào Bổ đề 7 (được chứng minh dựa trên kết quả của Bổ đề 11 về khoảng cách giữa $\overline{I_{ij}}$ và $\overline{Q_{ij}}$), ta có chặn trên cho khoảng cách giữa các tâm:

$$\delta^2(\overline{P_{ij}^*}, c_{ij}) \leq \left(\frac{\alpha}{1-\alpha} + \frac{13\alpha - 15\alpha^2}{(1-3\alpha-\varepsilon)(1-2\alpha-\varepsilon)}\right) \frac{\delta^2(P_{ij}^*, \overline{P_{ij}^*})}{|P_{ij}^*|}$$

Dùng với 1

$$\delta^2(P_{ij}^*, c_{ij}) \le \delta^2(P_{ij}^*, \overline{P_{ij}^*}) + |P_{ij}^*| \left[\left(\frac{\alpha}{1-\alpha} + \frac{13\alpha - 15\alpha^2}{(1-3\alpha-\varepsilon)(1-2\alpha-\varepsilon)} \right) \frac{\delta^2(P_{ij}^*, \overline{P_{ij}^*})}{|P_{ij}^*|} \right]$$
$$= \left(1 + \frac{\alpha}{1-\alpha} + \frac{13\alpha - 15\alpha^2}{(1-3\alpha-\varepsilon)(1-2\alpha-\varepsilon)} \right) \delta^2(P_{ij}^*, \overline{P_{ij}^*})$$

Đặt $\mathscr{K}(\alpha) = \frac{\alpha}{1-\alpha} + \frac{13\alpha-15\alpha^2}{(1-3\alpha-\varepsilon)(1-2\alpha-\varepsilon)}$. Vì $\alpha < 1/3$, $\mathscr{K}(\alpha) = O(\alpha)$. Lấy tổng trên tất cả các cụm i và các chiều j:

$$\delta^2(P,C) \le (1 + \mathscr{K}(\alpha)) \sum_{i,j} \delta^2(P_{ij}^*, \overline{P_{ij}^*}) = (1 + O(\alpha))\delta^2(P, C^*)$$

2. Xác suất thành công

Sự thành công của Fast-Estimation phụ thuộc vào độ chính xác của bộ ước lượng $\omega(u)$. Điều này được đảm bảo bởi Bổ đề 8 và Bổ đề 9 thông qua việc lấy mẫu ngẫu nhiên.

1. **Biến ngẫu nhiên:** Xét quá trình lấy mẫu S_{ij} từ P_{ij}. Gọi biến ngẫu nhiên X_p, bằng 1 nếu điểm $p \in P_{ij}$ được chọn vào S_{ij} và 0 nếu ngược lại. Tổng số điểm thuộc một tập con bất kỳ $A \subseteq P_{ij}$ rơi vào mẫu là $X = \sum_{p \in A} X_p$.
2. **Kỳ vọng:** $\mathbb{E}[X] = \frac{|S_{ij}|}{|P_{ij}|}|A|$. Thuật toán kích thước mẫu $|S_{ij}|$ đủ lớn sao cho kỳ vọng số điểm trong các "khối lớn"thỏa mãn $\mathbb{E}[X] \ge \Omega(\frac{\log m}{\varepsilon^2})$.
3. **Chặn chernoff:** Để chứng minh độ tập trung của giá trị ước lượng quanh giá trị kỳ vọng, ta sử dụng Bất đẳng thức Chernoff dạng nhân:

   $$\Pr(|X - \mathbb{E}[X]| \ge \varepsilon_1 \mathbb{E}[X]) \le 2e^{-\frac{\varepsilon_1^2 \mathbb{E}[X]}{3}}$$

   Với $\mathbb{E}[X] \approx \log m$, số mũ trở thành $-\Omega(\log m)$, dẫn đến xác suất sai lệch là nghịch đảo đa thức của m (xấp xỉ $1/m^3$).
4. **Chặn hợp:** Để đảm bảo thuật toán hoạt động đúng trên toàn cục, ta lấy tổng xác suất thất bại trên tất cả các chiều d, tất cả các cụm k, và tất cả các ứng viên u. Do xác suất thất bại cá lẻ rất nhỏ, tổng xác suất thất bại vẫn được giữ ở mức hằng số nhỏ.

3. Thời gian chạy

Thời gian chạy của thuật toán được phân tích theo từng giai đoạn xử lý:

- **Ứng viên (Bước 3-7):** Việc lấy mẫu U_{ij} mất thời gian $O(\log kd)$. Thuật toán lặp qua $O(\log(m\Delta_{\max}))$ giá trị độ dài khoảng, mỗi lần tạo ra tập ứng viên. Tổng số lượng ứng viên được tạo ra là $O(\varepsilon^{-1}\log(m\Delta_{\max})\log(ka$
- **Ước lượng chi phí (Bước 10):** Kích thước của bộ ước lượng (số lượng mẫu S_{ij}) là $\tilde{O}\left(\frac{1}{\alpha\varepsilon^4}\right)$. Việc tính toán $\omega(u)$ cho tất cả các ứng viên đòi hỏi thời gian tỷ lệ thuận với số lượng ứng viên nhân với

kích thước bộ ước lượng. Tổng thời gian cho bước này là:

$$O(\text{số ứng viên}) \times O(\text{kích thước ước lượng}) = \tilde{O}\left(\frac{1}{\alpha\varepsilon^5}\right)$$

bước này độc lập với kích thước dữ liệu m (sublinear).

- **Lựa chọn (Bước 12):** Sau khi chọn được tâm xấp xỉ c_{ij}, thuật toán cần tìm $(1-2\alpha-\alpha\varepsilon)m_i$ điểm lân cận nhất trong P_{ij}. Sử dụng thuật toán lựa chọn tuyến tính (linear selection algorithm - Blum et al., 1973), bước này mất thời gian $O(m_i)$ cho mỗi chiều của mỗi cụm.
- Cộng gộp thời gian trên tất cả k cụm và d chiều:

$$\sum_{i=1}^{k}\sum_{j=1}^{d} O(m_i) + k\cdot d\cdot\tilde{O}\left(\frac{1}{\alpha\varepsilon^5}\right) = O(md) + \tilde{O}\left(\frac{kd}{\alpha\varepsilon^5}\right)$$

□

9.4 FAST-FILTERING

Theorem 3. *Cho* $R_1 = O\left(\frac{\log k}{1-2\alpha}\right)$ *và* $R_2 = O\left(\frac{\log(m^3 d\log^3(m\Delta^2)/\varepsilon^2)\log(m\Delta^2)}{\alpha\varepsilon^4}\right)$*, trong đó* Δ *là tỷ lệ chiều của tập dữ liệu. Với xác suất hằng số, Thuật toán 4 (Fast-Filtering) trả về nghiệm xấp xỉ* $(1+O(\sqrt{\alpha}))$ *cho bài toán k-means có hỗ trợ học trong thời gian* $O(md)+\tilde{O}\left(\frac{kd}{\varepsilon^4(1-2\alpha)\alpha}\right)$ *với* $\alpha\in(0,1/3-\varepsilon)$.

Chứng minh. Chứng minh được chia thành ba giai đoạn chính: phân tích thành công của việc lấy mẫu ứng viên, độ tin cậy của bộ ước lượng chi phí, và tổng hợp chi phí phân cụm cuối cùng.

1. Xác suất lấy mẫu thành công

Mục tiêu là đảm bảo tập ứng viên U_i chứa ít nhất một điểm "tốt"nằm gần tâm tối ưu thực sự c_i^* (ký hiệu là $\overline{P_i^*}$ trong các phần trước, ở đây ta dùng c_i^* để đồng nhất với ký hiệu trong bài báo cho Fast-Filtering).

Gọi tập $G_2(P_i^*) = \{x\in P_i^* : \delta^2(x,c_i^*) \le 2\delta^2(P_i^*,c_i^*)/|P_i^*|\}$. Theo Bổ đề 4 , $|P_i\cap P_i^*| \ge (1-\alpha)\max(|P_i|,|P_i^*|)$, ta suy ra:

$$|P_i\cap G_2(P_i^*)| \ge |P_i\cap P_i^*| - |P_i^*\setminus G_2(P_i^*)| \ge (1-\alpha)|P_i^*| - \frac{|P_i^*|}{2} = \left(\frac{1}{2}-\alpha\right)|P_i^*|$$

Tỷ lệ điểm tốt trong P_i là $\zeta_i = \frac{|P_i\cap G_2(P_i^*)|}{|P_i|} \ge (1-\alpha)(\frac{1}{2}-\alpha)$. Với kích thước mẫu $R_1 = \Theta\left(\frac{1}{1-2\alpha}\log\left(\frac{k}{\eta}\right)\right)$, xác suất để tập U_i chứa ít nhất một điểm tốt $u_i\in G_2(P_i^*)$ là rất cao.

2. Độ tin cậy của bộ ước lượng

Tiếp theo, ta cần đảm bảo bộ ước lượng $\omega(u)$ chọn ra được tâm c_i tốt từ tập U_i. Do tồn tại $u_i\in G_2(P_i^*)$

trong tập ứng viên, chi phí của nó bị chặn bởi:

$$\delta^2(H_i(u_i), u_i) \leq \delta^2(Q_i, u_i) \leq 3\delta^2(P_i^*, c_i^*)$$

Vì thuật toán chọn c_i để tối thiểu hóa ω, ta có kết quả quan trọng:

$$\delta^2(P_i \setminus \mathscr{Z}^\dagger(c_i), c_i) \leq 4\delta^2(P_i^*, c_i^*)$$

Điều này đảm bảo rằng tâm được chọn c_i (và tập hợp sau lọc I_i) có chất lượng tốt, làm tiền đề cho Bổ đề 12.

3. Tổng chi phí

Ta đánh giá tổng chi phí của giải pháp cuối cùng $C = \{\overline{I_1}, \ldots, \overline{I_k}\}$. Tổng chi phí là tổng chi phí của từng cụm tối ưu P_i^* được gán cho tâm tương ứng $\overline{I_i}$:

$$\delta^2(P, C) \leq \sum_{i=1}^{k} \delta^2(P_i^*, \overline{I_i})$$

Sử dụng kết quả trực tiếp từ Bổ đề 14 (Lemma 14), ta có chặn trên cho từng cụm:

$$\delta^2(P_i^*, \overline{I_i}) \leq \left(1 + \frac{O(\sqrt{\alpha})}{(1-\alpha)(1-(3+\varepsilon)\alpha)}\right) \delta^2(P_i^*, c_i^*)$$

Lấy tổng trên tất cả k cụm:

$$\delta^2(P, C) \leq \left(1 + \frac{O(\sqrt{\alpha})}{(1-\alpha)(1-(3+\varepsilon)\alpha)}\right) \sum_{i=1}^{k} \delta^2(P_i^*, c_i^*)$$

Biểu thức trong ngoặc có thể được đơn giản hóa thành $(1 + O(\sqrt{\alpha}))$ khi α nhỏ và ε là hằng số. Vậy thuật toán đạt tỷ lệ xấp xỉ $(1 + O(\sqrt{\alpha}))$.

4. Thời gian chạy

- **Lấy mẫu (Bước 2 & 3):** Việc lấy mẫu U_i và S_i mất thời gian $O(1)$ cho mỗi cụm (hoặc phụ thuộc kích thước mẫu nhưng độc lập với m).
- **Ước lượng (Bước 4 & 5):** Tính toán $\omega(u)$ cho tất cả $u \in U_i$ đòi hỏi tính khoảng cách giữa các cặp điểm trong U_i và S_i. Thời gian cho mỗi cụm là $O(R_1 \cdot R_2 \cdot d)$. Tổng thời gian ước lượng là:

$$k \cdot O\left(\frac{\log k}{1-2\alpha} \cdot \frac{\text{polylog}(m)}{\alpha\varepsilon^4} \cdot d\right) = \tilde{O}\left(\frac{kd}{\varepsilon^4(1-2\alpha)\alpha}\right)$$

- **Lọc và tính tâm (Bước 6 & 7):** Tìm $(1-\alpha)m_i$ lân cận gần nhất cho tâm c_i đã chọn đòi hỏi quét qua P_i. Sử dụng thuật toán chọn tuyến tính (Linear Selection), bước này mất $O(m_i d)$. Tổng thời gian cho

k cụm là $\sum O(m_i d) = O(md)$.

Tổng hợp lại, độ phức tạp thời gian là $O(md) + \tilde{O}\left(\frac{kd}{\varepsilon^4(1-2\alpha)\alpha}\right)$. □

Corollary 3.1. *Cho kích thước mẫu $R_1 = \Theta\left(\frac{\log k}{1-2\alpha}\right)$. Với mỗi cụm dự đoán $i \in [k]$, với xác suất hằng số, tồn tại ít nhất một điểm dữ liệu u trong tập mẫu U_i sao cho $u \in G_2(P_i^*)$, trong đó $G_2(P_i^*)$ là tập hợp các điểm nằm gần tâm tối ưu.*

Chứng minh. Hệ quả này tương tự Corollary 1, chứng minh tương tự.

Gọi ζ_i là xác suất chọn được một điểm thuộc $G_2(P_i^*)$ khi lấy mẫu ngẫu nhiên đều từ P_i:

$$\begin{aligned}\zeta_i &= \frac{|P_i \cap G_2(P_i^*)|}{|P_i|} \\ &\geq \frac{(\frac{1}{2}-\alpha)|P_i^*|}{|P_i|}\end{aligned}$$

Ta cần chặn trên cho $|P_i|$. Từ giả thiết $|Q_i| \geq (1-\alpha)|P_i|$ và $Q_i \subseteq P_i^*$, ta suy ra $|P_i| \leq \frac{|P_i^*|}{1-\alpha}$.

$$\begin{aligned}\zeta_i &\geq \left(\frac{1}{2}-\alpha\right) \frac{|P_i^*|}{\frac{|P_i^*|}{1-\alpha}} \\ &= \left(\frac{1}{2}-\alpha\right)(1-\alpha) \\ &= \frac{(1-2\alpha)(1-\alpha)}{2}\end{aligned}$$

Với $\alpha < 1/2$, giá trị ζ_i luôn dương.

Giả sử ta lấy mẫu độc lập. Xác suất để *tất cả* các mẫu đều không thuộc tập điểm tốt là:

$$\Pr(\text{Thất bại tại cụm } i) = (1-\zeta_i)^{R_1} \leq e^{-\zeta_i R_1}$$

Để xác suất này nhỏ hơn $\frac{\eta}{k}$:

$$e^{-\zeta_i R_1} \leq \frac{\eta}{k} \iff -\zeta_i R_1 \leq \ln\left(\frac{\eta}{k}\right) \iff R_1 \geq \frac{1}{\zeta_i}\ln\left(\frac{k}{\eta}\right)$$

Thay thế chặn dưới của ζ_i:

$$\begin{aligned} R_1 &\geq \frac{2}{(1-2\alpha)(1-\alpha)} \ln\left(\frac{k}{\eta}\right) \\ &= \left(\frac{4}{1-2\alpha} - \frac{2}{1-\alpha}\right) \ln\left(\frac{k}{\eta}\right) \end{aligned}$$

Ta chỉ cần lấy mẫu vừa đủ, như vậy phù hợp với $R_1 = \Theta\left(\frac{\log k}{1-2\alpha}\right)$.

Để đảm bảo thành công trên tất cả k cụm:

$$\Pr(\exists i \in [k] : U_i \cap G_2(P_i^*) = \emptyset) \leq \sum_{i=1}^{k} \frac{\eta}{k} = \eta$$

Như vậy, với xác suất ít nhất $1-\eta$ (xác suất hằng số), thuật toán tìm được ít nhất một ứng viên tốt cho mọi cụm $i \in [k]$. □

Corollary 3.2. *Cho*

$$R_2 = O\left(\frac{\log(m^3 d \log^3(m\Delta^2)/\varepsilon_1^2)\log(m\Delta^2)}{\alpha\varepsilon_1^4}\right)$$

Với một điểm dữ liệu bất kỳ $u \in U_i$, với xác suất cao, bộ ước lượng $\omega(u)$ thỏa mãn:

$$\frac{\delta^2(P_i \setminus \mathscr{L}^\dagger(u), u)}{1+7\varepsilon_1} \leq \omega(u) \leq (1+\varepsilon_1)^2 \delta^2(H_i(u), u)$$

trong đó $H_i(u)$ là tập hợp $(1-\alpha)m_i$ điểm gần u nhất trong P_i, và $\mathscr{L}^\dagger(u)$ là tập hợp $(2+20\varepsilon_1)\alpha m_i$ điểm xa u nhất.

Chứng minh. Hệ quả này tương tự Lemma 10, chứng minh tương tự.

Gọi $\mathscr{F}'(u)$ là tập hợp các điểm thuộc mẫu S_i nằm trong các khối nhỏ hoặc là điểm ngoại lai. Ta đã biết $|\mathscr{F}'(u)| \leq (1+3\varepsilon_1)\alpha|S_i|$. Khi tính $\omega(u)$, thuật toán loại bỏ một lượng điểm tương ứng, do đó chi phí chỉ còn phụ thuộc vào các khối lớn. Xét tổng chi phí trên mẫu:

$$\omega(u) \leq \frac{m_i}{|S_i|} \sum_{\mathscr{B}_u^l \in \mathscr{L}(u)} \delta^2(\mathscr{B}_u^l \cap S_i, u)$$

Sử dụng tính chất khoảng cách trong khối $\delta^2(x,u) \leq (1+\varepsilon_1)^{l+1}$ và chặn trên của số lượng mẫu :

$$\begin{aligned}\delta^2(\mathscr{B}_u^l \cap S_i, u) &\leq (1+\varepsilon_1)^{l+1}|\mathscr{B}_u^l \cap S_i| \\ &\leq (1+\varepsilon_1)^{l+1}(1+\varepsilon_1)\frac{|S_i|}{m_i}|\mathscr{B}_u^l| \\ &= \frac{|S_i|}{m_i}(1+\varepsilon_1)^2\left((1+\varepsilon_1)^l|\mathscr{B}_u^l|\right)\end{aligned}$$

Lưu ý rằng $(1+\varepsilon_1)^l|\mathscr{B}_u^l| \approx \delta^2(\mathscr{B}_u^l, u)$. Tổng hợp lại trên các khối lớn (là tập con của $H_i(u)$):

$$\omega(u) \leq (1+\varepsilon_1)^2\delta^2(H_i(u), u)$$

Ta sử dụng bất đẳng thức đại số: với ε_1 nhỏ, $\frac{1}{1-\varepsilon_1} \leq 1+3\varepsilon_1$. Từ kết quả tập trung ở Bước 1, ta suy ra kích thước thực tế của khối lớn trong P_i:

$$|\mathscr{B}_u^l| \leq \frac{m_i}{|S_i|(1-\varepsilon_1)}|\mathscr{B}_u^l \cap S_i| \leq (1+3\varepsilon_1)\frac{m_i}{|S_i|}|\mathscr{B}_u^l \cap S_i|$$

Nhân cả hai vế với bình phương khoảng cách (xấp xỉ $(1+\varepsilon_1)^l$) và lấy tổng trên các khối lớn (lưu ý rằng việc loại bỏ $\mathscr{Z}^\dagger(u)$ tương ứng với việc giữ lại các khối này):

$$\begin{aligned}\delta^2(P_i \setminus \mathscr{Z}^\dagger(u), u) &\leq \sum(1+\varepsilon_1)^{l+1}(1+3\varepsilon_1)\frac{m_i}{|S_i|}|\mathscr{B}_u^l \cap S_i| \\ &\leq (1+3\varepsilon_1)(1+\varepsilon_1)\frac{m_i}{|S_i|}\sum(1+\varepsilon_1)^l|\mathscr{B}_u^l \cap S_i| \\ &\approx (1+4\varepsilon_1)\omega(u)\end{aligned}$$

Để đảm bảo tính chặt chẽ cho mọi số hạng bậc cao, bài báo sử dụng hệ số an toàn là $1+7\varepsilon_1$:

$$\delta^2(P_i \setminus \mathscr{Z}^\dagger(u), u) \leq (1+7\varepsilon_1)\omega(u)$$

Sắp xếp lại bất đẳng thức ta thu được chặn dưới cần chứng minh . □

Lemma 12. *Khoảng cách giữa trọng tâm của tập hợp đã lọc $\overline{I}_i$ và tâm tối ưu c_i^* bị chặn như sau:*

$$\delta^2(\overline{I}_i, c_i^*) \leq \frac{9\delta^2(P_i^*, c_i^*)}{(1-(3+\varepsilon)\alpha)m_i}$$

Chứng minh.

1. Kích thước các tập hợp điểm

Theo định nghĩa của quy trình lọc trong thuật toán 3, tập I_i được tạo thành bằng cách loại bỏ tập $\mathscr{Z}^\dagger(c_i)$ gồm các điểm xa nhất từ tâm ứng viên c_i. Kích thước của phần bị loại bỏ là $|\mathscr{Z}^\dagger(c_i)| = (2+20\varepsilon_1)\alpha m_i$. Do

đó, kích thước của tập hợp giữ lại là:

$$|I_i| = m_i - (2+20\varepsilon_1)\alpha m_i = (1-(2+20\varepsilon_1)\alpha)m_i$$

Tiếp theo, ta xét phần giao giữa tập đã lọc I_i và cụm tối ưu P_i^*. Ta biết rằng số lượng điểm "sai nhãn"(nhiễu) trong cụm dự đoán P_i tối đa là $|P_i \setminus P_i^*| \leq \alpha m_i$. Trong trường hợp xấu nhất, toàn bộ các điểm nhiễu này vẫn nằm trong I_i. Do đó, số lượng điểm thuộc cụm tối ưu thực sự nằm trong I_i bị chặn dưới bởi:

$$|I_i \cap P_i^*| \geq |I_i| - |P_i \setminus P_i^*|$$

Thay thế kích thước của $|I_i|$ vào:

$$|I_i \cap P_i^*| \geq (1-(2+20\varepsilon_1)\alpha)m_i - \alpha m_i = (1-(3+20\varepsilon_1)\alpha)m_i$$

2. Kẹp tập đã lọc

Dựa trên Hệ quả 4 (Corollary 4) trong bài báo , tâm c_i được chọn bởi bộ ước lượng thỏa mãn điều kiện về chi phí với xác suất cao:

$$\delta^2(I_i, c_i) \leq 4\delta^2(P_i^*, c_i^*)$$

Theo tính chất của trọng tâm, tổng bình phương khoảng cách từ các điểm trong một tập hợp đến trọng tâm của nó ($\overline{I}_i$) luôn nhỏ hơn hoặc bằng tổng bình phương khoảng cách đến bất kỳ điểm nào khác (c_i). Do đó:

$$\delta^2(I_i, \overline{I}_i) \leq \delta^2(I_i, c_i) \leq 4\delta^2(P_i^*, c_i^*)$$

3. Áp dụng Bất đẳng thức tam giác nới lỏng

Để chặn khoảng cách $\delta^2(\overline{I}_i, c_i^*)$, ta xét tổng khoảng cách trên các điểm trung gian p thuộc giao tập $I_i \cap P_i^*$. Ta có đẳng thức trung bình:

$$\delta^2(\overline{I}_i, c_i^*) = \frac{1}{|I_i \cap P_i^*|} \sum_{p \in I_i \cap P_i^*} \delta^2(\overline{I}_i, c_i^*)$$

Áp dụng bất đẳng thức tam giác nới lỏng (relaxed triangle inequality) dạng $(a+b)^2 \leq (1+\frac{1}{\lambda})a^2 + (1+\lambda)b^2$. Ở đây ta chọn $\lambda = 2$ để tối ưu hóa các hệ số theo bài báo:

$$\delta^2(\overline{I}_i, c_i^*) \leq (1+0.5)\delta^2(\overline{I}_i, p) + (1+2)\delta^2(p, c_i^*)$$

Thay thế vào công thức tổng:

$$\delta^2(\overline{I}_i, c_i^*) \leq \frac{1}{|I_i \cap P_i^*|} \sum_{p \in I_i \cap P_i^*} \left[1.5\delta^2(\overline{I}_i, p) + 3\delta^2(p, c_i^*)\right]$$

Ta thực hiện chặn trên cho từng thành phần của tử số:

- Tổng khoảng cách từ p đến $\overline{I}_i$: Vì $p \in I_i$, tổng này nhỏ hơn tổng trên toàn bộ tập I_i:

$$\sum_{p \in I_i \cap P_i^*} \delta^2(\overline{I}_i, p) \leq \delta^2(I_i, \overline{I}_i)$$

- Tổng khoảng cách từ p đến c_i^*: Vì $p \in P_i^*$, tổng này nhỏ hơn tổng chi phí của cụm tối ưu:

$$\sum_{p \in I_i \cap P_i^*} \delta^2(p, c_i^*) \leq \delta^2(P_i^*, c_i^*)$$

Thay thế các bất đẳng thức này vào biểu thức chính:

$$\delta^2(\overline{I}_i, c_i^*) \leq \frac{1.5\delta^2(I_i, \overline{I}_i) + 3\delta^2(P_i^*, c_i^*)}{|I_i \cap P_i^*|}$$

Sử dụng kết quả từ Bước 2 ($\delta^2(I_i, \overline{I}_i) \leq 4\delta^2(P_i^*, c_i^*)$) và Bước 1 cho mẫu số:

$$\begin{aligned}
\delta^2(\overline{I}_i, c_i^*) &\leq \frac{1.5(4\delta^2(P_i^*, c_i^*)) + 3\delta^2(P_i^*, c_i^*)}{(1-(3+20\varepsilon_1)\alpha)m_i} \\
&= \frac{(6+3)\delta^2(P_i^*, c_i^*)}{(1-(3+20\varepsilon_1)\alpha)m_i} \\
&= \frac{9\delta^2(P_i^*, c_i^*)}{(1-(3+20\varepsilon_1)\alpha)m_i}
\end{aligned}$$

Cuối cùng, dựa vào điều kiện thiết lập tham số trong thuật toán là $20\varepsilon_1 \leq \varepsilon$ (với $\varepsilon_1 = \varepsilon/126$), ta có chặn cuối cùng:

$$\delta^2(\overline{I}_i, c_i^*) \leq \frac{9\delta^2(P_i^*, c_i^*)}{(1-(3+\varepsilon)\alpha)m_i}$$

□

Lemma 13. *Chi phí phân cụm của tập Q_i đối với tâm của tập hợp đã lọc $\overline{I}_i$ thỏa mãn chặn cụ thể sau:*

$$\delta^2(Q_i, \overline{I}_i) \leq \delta^2(Q_i, c_i^*) + \left(5\sqrt{\alpha} + \frac{36(\sqrt{\alpha}+\alpha)}{1-(3+\varepsilon)\alpha}\right)\delta^2(P_i^*, c_i^*)$$

Chứng minh.

1. Phân rã tập hợp và chi phí Gọi $A_i = Q_i \setminus I_i$ là tập các điểm thuộc Q_i bị loại bỏ (âm tính giả của bộ lọc). Gọi $B_i = I_i \setminus Q_i$ là tập các điểm nhiễu được giữ lại (dương tính giả của bộ lọc).

Ta có đẳng thức phân rã chi phí như sau:

$$\delta^2(Q_i,\overline{I}_i) - \delta^2(Q_i,c_i^*) = \underbrace{[\delta^2(I_i,\overline{I}_i) - \delta^2(I_i,c_i^*)]}_{\leq 0} + [\delta^2(A_i,\overline{I}_i) - \delta^2(A_i,c_i^*)] + [\delta^2(B_i,c_i^*) - \delta^2(B_i,\overline{I}_i)]$$

Vì $\overline{I}_i$ là trọng tâm của I_i, số hạng đầu tiên luôn ≤ 0. Ta tập trung chặn hai số hạng còn lại.

2. Áp dụng bất đẳng thức tam giác nới lỏng Sử dụng bất đẳng thức $\delta^2(x,y) \leq (1+\lambda)\delta^2(x,z) + (1+\frac{1}{\lambda})\delta^2(z,y)$ với $\lambda = \sqrt{\alpha}$.

Đối với tập A_i (tương tự cho B_i):

$$\begin{aligned}\delta^2(A_i,\overline{I}_i) - \delta^2(A_i,c_i^*) &\leq \sum_{a\in A_i}\left((1+\sqrt{\alpha})\delta^2(a,c_i^*) + (1+\frac{1}{\sqrt{\alpha}})\delta^2(c_i^*,\overline{I}_i) - \delta^2(a,c_i^*)\right)\\ &= \sqrt{\alpha}\delta^2(A_i,c_i^*) + |A_i|\left(1+\frac{1}{\sqrt{\alpha}}\right)\delta^2(c_i^*,\overline{I}_i)\end{aligned}$$

Tương tự cho B_i:

$$\delta^2(B_i,c_i^*) - \delta^2(B_i,\overline{I}_i) \leq \sqrt{\alpha}\delta^2(B_i,\overline{I}_i) + |B_i|\left(1+\frac{1}{\sqrt{\alpha}}\right)\delta^2(c_i^*,\overline{I}_i)$$

Tổng hợp lại:

$$\delta^2(Q_i,\overline{I}_i) - \delta^2(Q_i,c_i^*) \leq \sqrt{\alpha}[\delta^2(A_i,c_i^*) + \delta^2(B_i,\overline{I}_i)] \tag{22}$$

$$+\left(1+\frac{1}{\sqrt{\alpha}}\right)(|A_i|+|B_i|)\delta^2(\overline{I}_i,c_i^*) \tag{23}$$

3. Tổng hợp

Theo giả thiết bài toán và Bổ đề 12:

- Tổng kích thước sai số: $|A_i| + |B_i| \leq 3\alpha m_i + \alpha m_i = 4\alpha m_i$.
- Hệ số khoảng cách tâm:

$$\left(1+\frac{1}{\sqrt{\alpha}}\right)(|A_i|+|B_i|) \leq \frac{\sqrt{\alpha}+1}{\sqrt{\alpha}} \cdot 4\alpha m_i = 4\sqrt{\alpha}(1+\sqrt{\alpha})m_i$$

- Khoảng cách giữa các tâm (từ Lemma 12): $\delta^2(\overline{I}_i,c_i^*) \leq \frac{9\delta^2(P_i^*,c_i^*)}{(1-(3+\varepsilon)\alpha)m_i}$.
- Chi phí trong cụm: $\delta^2(A_i,c_i^*) \leq \delta^2(P_i^*,c_i^*)$ và $\delta^2(B_i,\overline{I}_i) \leq 4\delta^2(P_i^*,c_i^*)$.

Thay thế các giá trị này vào phương trình Lemma 13:

$$\delta^2(Q_i,\overline{I}_i) - \delta^2(Q_i,c_i^*) \le \sqrt{\alpha}[\delta^2(P_i^*,c_i^*) + 4\delta^2(P_i^*,c_i^*)] + 4\sqrt{\alpha}(1+\sqrt{\alpha})m_i \cdot \frac{9\delta^2(P_i^*,c_i^*)}{(1-(3+\varepsilon)\alpha)m_i}$$
$$= 5\sqrt{\alpha}\delta^2(P_i^*,c_i^*) + \frac{36(\sqrt{\alpha}+\alpha)}{1-(3+\varepsilon)\alpha}\delta^2(P_i^*,c_i^*)$$

Kết luận:

$$\delta^2(Q_i,\overline{I}_i) \le \delta^2(Q_i,c_i^*) + \left(5\sqrt{\alpha} + \frac{36(\sqrt{\alpha}+\alpha)}{1-(3+\varepsilon)\alpha}\right)\delta^2(P_i^*,c_i^*)$$

□

Lemma 14. *Tổng chi phí phân cụm của cụm tối ưu P_i^* đối với trọng tâm $\overline{I}_i$ bị chặn bởi:*

$$\delta^2(P_i^*,\overline{I}_i) \le \left(1 + 6\sqrt{\alpha} + \frac{36(\sqrt{\alpha}+\alpha)}{1-(3+\varepsilon)\alpha} + \frac{9(\sqrt{\alpha}+\alpha)}{(1-\alpha)(1-(3+\varepsilon)\alpha)}\right)\delta^2(P_i^*,c_i^*)$$

Chứng minh.

Ta có (theo định nghĩa):

$$\delta^2(P_i^*,\overline{I}_i) = \delta^2(Q_i,\overline{I}_i) + \delta^2(R_i,\overline{I}_i)$$

1. **Chặn trên cho Q_i** Sử dụng kết quả từ Bổ đề 13:

$$\delta^2(Q_i,\overline{I}_i) \le \delta^2(Q_i,c_i^*) + \left(5\sqrt{\alpha} + \frac{36(\sqrt{\alpha}+\alpha)}{1-(3+\varepsilon)\alpha}\right)\delta^2(P_i^*,c_i^*)$$

2. **Chặn trên cho R_i (âm tính giả của dự đoán)** Áp dụng bất đẳng thức tam giác nới lỏng với $\lambda = \sqrt{\alpha}$ cho mỗi $p \in R_i$:

$$\delta^2(R_i,\overline{I}_i) \le (1+\sqrt{\alpha})\delta^2(R_i,c_i^*) + |R_i|\left(1 + \frac{1}{\sqrt{\alpha}}\right)\delta^2(c_i^*,\overline{I}_i)$$
$$= \delta^2(R_i,c_i^*) + \sqrt{\alpha}\delta^2(R_i,c_i^*) + |R_i|\frac{\sqrt{\alpha}+1}{\sqrt{\alpha}}\delta^2(c_i^*,\overline{I}_i)$$

   Ta có các chặn kích thước và chi phí:

   - $|R_i| \le \frac{\alpha m_i}{1-\alpha}$ (do điều kiện P_i chứa ít nhất $(1-\alpha)$ phần tử của P_i^*).
   - $\delta^2(R_i,c_i^*) \le \delta^2(P_i^*,c_i^*)$.

Thay thế vào biểu thức của R_i:

$$\begin{aligned}\delta^2(R_i,\overline{I}_i) &\le (1+\sqrt{\alpha})\delta^2(R_i,c_i^*) + \left(\frac{\alpha m_i}{1-\alpha}\right)\frac{\sqrt{\alpha}+1}{\sqrt{\alpha}}\cdot\frac{9\delta^2(P_i^*,c_i^*)}{(1-(3+\varepsilon)\alpha)m_i} \\ &\le (1+\sqrt{\alpha})\delta^2(R_i,c_i^*) + \frac{9(\sqrt{\alpha}+\alpha)}{(1-\alpha)(1-(3+\varepsilon)\alpha)}\delta^2(P_i^*,c_i^*)\end{aligned}$$

3. **Tổng hợp** Cộng bước 1 + 2, và $\delta^2(Q_i,c_i^*)+\delta^2(R_i,c_i^*) = \delta^2(P_i^*,c_i^*)$.

$$\begin{aligned}\delta^2(P_i^*,\overline{I}_i) \le{}& \underbrace{\delta^2(Q_i,c_i^*)+\delta^2(R_i,c_i^*)}_{\delta^2(P_i^*,c_i^*)} + \underbrace{\sqrt{\alpha}\delta^2(R_i,c_i^*)}_{\le\sqrt{\alpha}\delta^2(P_i^*,c_i^*)} \\ &+ \left(5\sqrt{\alpha}+\frac{36(\sqrt{\alpha}+\alpha)}{1-(3+\varepsilon)\alpha}\right)\delta^2(P_i^*,c_i^*) \\ &+ \frac{9(\sqrt{\alpha}+\alpha)}{(1-\alpha)(1-(3+\varepsilon)\alpha)}\delta^2(P_i^*,c_i^*)\end{aligned}$$

Gộp các hệ số chứa $\sqrt{\alpha}$: $1\sqrt{\alpha}+5\sqrt{\alpha}=6\sqrt{\alpha}$. Ta thu được bất đẳng thức cuối cùng:

$$\delta^2(P_i^*,\overline{I}_i) \le \left(1+6\sqrt{\alpha}+\frac{36(\sqrt{\alpha}+\alpha)}{1-(3+\varepsilon)\alpha}+\frac{9(\sqrt{\alpha}+\alpha)}{(1-\alpha)(1-(3+\varepsilon)\alpha)}\right)\delta^2(P_i^*,c_i^*)$$

Vậy với $\alpha \in [0,1)$:

$$\delta^2(P_i^*,\overline{I}_i) \le \left(1+\frac{O(\sqrt{\alpha})}{(1-\alpha)(1-(3+\varepsilon)\alpha)}\right)\delta^2(P_i^*,c_i^*)$$

□

9.5 Mở rộng cho k-median - FAST-SAMPLING (k-median)

Corollary 3.3. *Đối với một cụm dự đoán P_i, với xác suất ít nhất $1-1/k$, tâm u_i được chọn bởi thuật toán Fast-Sampling cho mục tiêu k-median thỏa mãn:*

$$\delta(\mathcal{N}_i(u_i),u_i) \le \delta(\mathcal{N}_i(u_i),c_i^*) + \alpha\varepsilon\delta(P_i^*,c_i^*)$$

trong đó $\mathcal{N}_i(u_i)$ là tập hợp $(1-\alpha)m_i$ điểm gần nhất trong P_i đến u_i, và c_i^* là tâm phân cụm tối ưu cho cụm thứ i.

Chứng minh.

1. Sự tồn tại của ứng viên tốt trong lưới Theo Bước 6 và 7 của Thuật toán 5, không gian xung quanh các mẫu được phân rã thành các lưới con. Với xác suất cao, tồn tại một điểm ứng viên $u' \in U_i'$ nằm rất gần tâm tối ưu c_i^*. Cụ thể, dựa trên kích thước lưới con $(1-\alpha)\alpha\varepsilon_1 l_i/\sqrt{d}$, khoảng cách này được chặn bởi:

$$\delta(u', c_i^*) \leq \frac{\alpha\varepsilon\delta(P_i^*, c_i^*)}{m_i}$$

2. Tính tối ưu của tâm được chọn (u_i) Trong Bước 9 của thuật toán, u_i được chọn để tối thiểu hóa chi phí k-median đối với $(1-\alpha)m_i$ điểm lân cận nhất của nó. Do đó, với mọi ứng viên khác $u' \in U_i'$, ta có:

$$\delta(\mathcal{N}_i(u_i), u_i) \leq \delta(\mathcal{N}_i(u'), u')$$

3. Áp dụng bất đẳng thức tam giác Ta cần chặn trên vế phải $\delta(\mathcal{N}_i(u'), u')$.

Chọn tập so sánh là $\mathcal{N}_i(u_i)$ (tập lân cận của u_i), ta có:

$$\delta(\mathcal{N}_i(u'), u') \leq \delta(\mathcal{N}_i(u_i), u')$$

$$\delta(x, u') \leq \delta(x, c_i^*) + \delta(c_i^*, u')$$

Lấy tổng trên toàn bộ tập $\mathcal{N}_i(u_i)$:

$$\begin{aligned}
\delta(\mathcal{N}_i(u_i), u') &= \sum_{x \in \mathcal{N}_i(u_i)} \delta(x, u') \\
&\leq \sum_{x \in \mathcal{N}_i(u_i)} \delta(x, c_i^*) + \sum_{x \in \mathcal{N}_i(u_i)} \delta(c_i^*, u') \\
&= \delta(\mathcal{N}_i(u_i), c_i^*) + |\mathcal{N}_i(u_i)| \cdot \delta(u', c_i^*)
\end{aligned}$$

4. Tổng hợp

Kết hợp các bất đẳng thức từ Bước 2 và Bước 3:

$$\delta(\mathcal{N}_i(u_i), u_i) \leq \delta(\mathcal{N}_i(u_i), c_i^*) + |\mathcal{N}_i(u_i)| \cdot \delta(u', c_i^*)$$

$$\begin{aligned}
\delta(\mathcal{N}_i(u_i), u_i) &\leq \delta(\mathcal{N}_i(u_i), c_i^*) + m_i \left(\frac{\alpha\varepsilon\delta(P_i^*, c_i^*)}{m_i} \right) \\
&= \delta(\mathcal{N}_i(u_i), c_i^*) + \alpha\varepsilon\delta(P_i^*, c_i^*)
\end{aligned}$$

□

Lemma 15. *Đối với hàm mục tiêu k-median, khoảng cách giữa tâm thuật toán u_i và tâm phân cụm tối ưu*

c_i^* thỏa mãn:

$$\delta(u_i, c_i^*) \leq \frac{(2+\alpha\varepsilon)\delta(P_i^*, c_i^*)}{(1-2\alpha)m_i}$$

Chứng minh.

1. **Kích thước tập giao**

   Thuật toán chọn tập $\mathcal{N}_i(u_i)$ gồm $(1-\alpha)m_i$ điểm gần u_i nhất trong cụm dự đoán P_i. Theo giả thiết của mô hình, số lượng điểm trong P_i không thuộc cụm tối ưu P_i^* (dương tính giả) tối đa là αm_i (vì $|P_i \setminus P_i^*| \leq \alpha m_i$). Do đó, số lượng điểm thuộc P_i^* nằm trong tập $\mathcal{N}_i(u_i)$ bị chặn dưới bởi:

   $$|\mathcal{N}_i(u_i) \cap P_i^*| \geq |\mathcal{N}_i(u_i)| - |P_i \setminus P_i^*|$$

   Thay thế các giá trị kích thước vào:

   $$|\mathcal{N}_i(u_i) \cap P_i^*| \geq (1-\alpha)m_i - \alpha m_i = (1-2\alpha)m_i$$

   Điều này đảm bảo rằng phần giao chứa một lượng điểm đáng kể .

2. **Chặn trên cho chi phí**

   Dựa vào tính chất tối ưu trong Bước 9 của Thuật toán 5, u_i là điểm tối thiểu hóa chi phí k-median đối với tập lân cận của nó. Theo Hệ quả 5 (Corollary 5) đã chứng minh trước đó, với xác suất cao, chi phí này bị chặn bởi:

   $$\delta(\mathcal{N}_i(u_i), u_i) \leq (1+\alpha\varepsilon)\delta(P_i^*, c_i^*)$$

3. **Bất đẳng thức tam giác**

   Áp dụng bất đẳng thức tam giác cho mọi p:

   $$\delta(u_i, c_i^*) \leq \delta(p, u_i) + \delta(p, c_i^*)$$

   Lấy tổng trên tất cả các điểm p thuộc phần giao $\mathcal{N}_i(u_i) \cap P_i^*$ và chia cho kích thước của phần giao này:

   $$\delta(u_i, c_i^*) \leq \frac{\sum_{p \in \mathcal{N}_i(u_i) \cap P_i^*} \delta(p, u_i) + \sum_{p \in \mathcal{N}_i(u_i) \cap P_i^*} \delta(p, c_i^*)}{|\mathcal{N}_i(u_i) \cap P_i^*|}$$

   Ta thực hiện chặn trên cho tử số:

   - Tổng thứ nhất $\sum \delta(p, u_i)$ nhỏ hơn hoặc bằng tổng chi phí của toàn bộ tập $\mathcal{N}_i(u_i)$ đối với u_i:

     $$\sum_{p \in \mathcal{N}_i(u_i) \cap P_i^*} \delta(p, u_i) \leq \delta(\mathcal{N}_i(u_i), u_i) \leq (1+\alpha\varepsilon)\delta(P_i^*, c_i^*)$$

- Tổng thứ hai $\sum \delta(p, c_i^*)$ nhỏ hơn hoặc bằng tổng chi phí của toàn bộ cụm tối ưu P_i^*:

$$\sum_{p \in \mathcal{N}_i(u_i) \cap P_i^*} \delta(p, c_i^*) \leq \delta(P_i^*, c_i^*)$$

Thay thế các chặn trên vào tử số và sử dụng chặn dưới của mẫu số từ Bước 1:

$$\begin{aligned}\delta(u_i, c_i^*) &\leq \frac{(1+\alpha\varepsilon)\delta(P_i^*, c_i^*) + \delta(P_i^*, c_i^*)}{(1-2\alpha)m_i} \\ &= \frac{(2+\alpha\varepsilon)\delta(P_i^*, c_i^*)}{(1-2\alpha)m_i}\end{aligned}$$

□

Lemma 16. *Đối với hàm mục tiêu k-median, chi phí phân cụm của tập Q_i đối với tâm được chọn u_i thỏa mãn chặn sau:*

$$\delta(Q_i, u_i) \leq \delta(Q_i, c_i^*) + \frac{\alpha(4+3\varepsilon)}{1-2\alpha}\delta(P_i^*, c_i^*)$$

trong đó c_i^* là tâm tối ưu của cụm thứ i.

Chứng minh.

1. Phân rã tập hợp và chênh lệch chi phí

Ta có $Q_i = P_i \cap P_i^*$. Dựa trên tập lân cận $\mathcal{N}_i(u_i)$ được thuật toán chọn, ta định nghĩa các tập sai số:

- $A_i = Q_i \cap \mathscr{L}^\dagger(u_i)$: Tập các điểm thuộc Q_i nhưng bị loại bỏ (âm tính giả).
- $B_i = P_i \setminus (Q_i \cup \mathscr{L}^\dagger(u_i))$: Tập các điểm không thuộc Q_i nhưng được chọn (dương tính giả).

Khi đó $Q_i = (\mathcal{N}_i(u_i) \setminus B_i) \cup A_i$. Hiệu chi phí phân cụm được viết lại như sau:

$$\begin{aligned}\delta(Q_i, u_i) - \delta(Q_i, c_i^*) = &[\delta(\mathcal{N}_i(u_i), u_i) - \delta(\mathcal{N}_i(u_i), c_i^*)] \\ &+ [\delta(B_i, c_i^*) - \delta(B_i, u_i)] \\ &+ [\delta(A_i, u_i) - \delta(A_i, c_i^*)]\end{aligned}$$

2. Từng thành phần

Thành phần 1 (Tập được chọn): Theo Hệ quả 5 (Corollary 5), với xác suất cao, chi phí của tập được chọn thỏa mãn:

$$\delta(\mathcal{N}_i(u_i), u_i) - \delta(\mathcal{N}_i(u_i), c_i^*) \leq \alpha\varepsilon\delta(P_i^*, c_i^*)$$

Thành phần 2 (Dương tính giả B_i): Sử dụng bất đẳng thức tam giác $\delta(b, c_i^*) \leq \delta(b, u_i) + \delta(u_i, c_i^*)$. Suy ra

$\delta(b,c_i^*) - \delta(b,u_i) \leq \delta(u_i,c_i^*)$. Lấy tổng trên B_i:

$$\delta(B_i,c_i^*) - \delta(B_i,u_i) \leq |B_i|\delta(u_i,c_i^*)$$

Thành phần 3 (Âm tính giả A_i): Tương tự, với $a \in A_i$, ta có $\delta(a,u_i) \leq \delta(a,c_i^*) + \delta(c_i^*,u_i)$. Suy ra $\delta(a,u_i) - \delta(a,c_i^*) \leq \delta(u_i,c_i^*)$. Lấy tổng trên A_i:

$$\delta(A_i,u_i) - \delta(A_i,c_i^*) \leq |A_i|\delta(u_i,c_i^*)$$

3. Tập sai số

Theo phân tích kích thước trong bài báo :

$$|Q_i| = |\mathscr{N}_i(u_i)| + |A_i| - |B_i| = (1-\alpha)m_i + |A_i| - |B_i|$$

Mặt khác $|Q_i| \geq (1-\alpha)m_i$ (theo định nghĩa). Suy ra $|A_i| \geq |B_i|$. Đồng thời, $A_i \subseteq \mathscr{L}^\dagger(u_i)$ (tập các điểm bị loại bỏ), nên $|A_i| \leq \alpha m_i$. Do đó, tổng kích thước sai số bị chặn bởi:

$$|A_i| + |B_i| \leq 2|A_i| \leq 2\alpha m_i$$

4. Tổng hợp

Thay thế các chặn trên vào phương trình hiệu chi phí:

$$\delta(Q_i,u_i) - \delta(Q_i,c_i^*) \leq \alpha\varepsilon\delta(P_i^*,c_i^*) + (|A_i| + |B_i|)\delta(u_i,c_i^*)$$

Thay thế $|A_i| + |B_i| \leq 2\alpha m_i$ và khoảng cách tâm $\delta(u_i,c_i^*)$ từ Bổ đề 15:

$$\delta(u_i,c_i^*) \leq \frac{(2+\alpha\varepsilon)\delta(P_i^*,c_i^*)}{(1-2\alpha)m_i}$$

Ta có:

$$\begin{aligned}\Delta_{cost} &\leq \alpha\varepsilon\delta(P_i^*,c_i^*) + 2\alpha m_i\left(\frac{(2+\alpha\varepsilon)\delta(P_i^*,c_i^*)}{(1-2\alpha)m_i}\right) \\ &= \delta(P_i^*,c_i^*)\left[\alpha\varepsilon + \frac{2\alpha(2+\alpha\varepsilon)}{1-2\alpha}\right]\end{aligned}$$

Quy đồng mẫu số cho biểu thức trong ngoặc vuông:

$$\begin{aligned}\alpha\varepsilon + \frac{4\alpha + 2\alpha^2\varepsilon}{1-2\alpha} &= \frac{\alpha\varepsilon(1-2\alpha) + 4\alpha + 2\alpha^2\varepsilon}{1-2\alpha} \\ &= \frac{\alpha\varepsilon - 2\alpha^2\varepsilon + 4\alpha + 2\alpha^2\varepsilon}{1-2\alpha} \\ &= \frac{4\alpha + \alpha\varepsilon}{1-2\alpha} \\ &= \frac{\alpha(4+\varepsilon)}{1-2\alpha}\end{aligned}$$

Nhận thấy rằng $\frac{\alpha(4+\varepsilon)}{1-2\alpha} < \frac{\alpha(4+3\varepsilon)}{1-2\alpha}$ (với $\varepsilon > 0$). Để đảm bảo tính tổng quát và khớp với chặn đã công bố trong Bổ đề, ta sử dụng chặn rộng hơn:

$$\delta(Q_i, u_i) \leq \delta(Q_i, c_i^*) + \frac{\alpha(4+3\varepsilon)}{1-2\alpha}\delta(P_i^*, c_i^*)$$

□

Lemma 17. *Với mỗi cụm $i \in [k]$, với xác suất ít nhất $1 - 1/k$, chi phí phân cụm k-median của cụm tối ưu P_i^* đối với tâm thuật toán u_i bị chặn bởi:*

$$\delta(P_i^*, u_i) \leq \left(1 + \frac{6\alpha + 4\alpha\varepsilon - 4\alpha^2 - 3\varepsilon\alpha^2}{(1-\alpha)(1-2\alpha)}\right)\delta(P_i^*, c_i^*)$$

trong đó c_i^* là tâm tối ưu của cụm thứ i.

Chứng minh. Chúng ta phân tích tổng chi phí bằng cách chia cụm tối ưu P_i^* thành hai phần rời nhau dựa trên kết quả dự đoán của mô hình: phần giao với cụm dự đoán ($P_i^* \cap P_i$) và phần bị dự đoán sai ($P_i^* \setminus P_i$).

$$\delta(P_i^*, u_i) = \delta(P_i^* \cap P_i, u_i) + \delta(P_i^* \setminus P_i, u_i)$$

1. **Giới hạn chi phí của phần giao (dương tính thật)**

   Xét tập hợp $Q_i = P_i^* \cap P_i$. Áp dụng trực tiếp kết quả từ Bổ đề 16, ta có chặn trên cho chi phí của tập này đối với tâm u_i:

   $$\delta(P_i^* \cap P_i, u_i) \leq \delta(P_i^* \cap P_i, c_i^*) + \frac{\alpha(4+3\varepsilon)}{1-2\alpha}\delta(P_i^*, c_i^*)$$

2. **Giới hạn chi phí của phần sai lệch (âm tính giả)**

   Với các điểm $p \in P_i^* \setminus P_i$ (các điểm thuộc cụm tối ưu nhưng không nằm trong cụm dự đoán P_i), ta sử

dụng bất đẳng thức tam giác qua tâm tối ưu c_i^*:

$$\delta(p,u_i) \le \delta(p,c_i^*) + \delta(c_i^*,u_i)$$

Lấy tổng trên toàn bộ tập $P_i^* \setminus P_i$:

$$\delta(P_i^* \setminus P_i, u_i) \le \delta(P_i^* \setminus P_i, c_i^*) + |P_i^* \setminus P_i|\delta(c_i^*,u_i)$$

3. **Khoảng cách tâm**

Ta có

$$|P_i^*| \ge |P_i \cap P_i^*| \ge (1-\alpha)|P_i^*|$$
$$\Rightarrow |P_i^*| \le \frac{|P_i|}{1-\alpha}$$

$$|P_i^* \setminus P_i| \le \alpha|P_i^*| \quad \text{hoặc theo } m_i: \quad |P_i^* \setminus P_i| \le \frac{\alpha m_i}{1-\alpha}$$

Sử dụng kết quả từ Bổ đề 15 cho khoảng cách giữa hai tâm $\delta(u_i,c_i^*) \le \frac{(2+\alpha\varepsilon)\delta(P_i^*,c_i^*)}{(1-2\alpha)m_i}$. Thay thế vào biểu thức ở Bước 2:

$$\begin{aligned}\text{Sai số dịch chuyển} &= |P_i^* \setminus P_i|\delta(c_i^*,u_i) \\ &\le \left(\frac{\alpha m_i}{1-\alpha}\right)\left(\frac{(2+\alpha\varepsilon)\delta(P_i^*,c_i^*)}{(1-2\alpha)m_i}\right) \\ &= \frac{\alpha(2+\alpha\varepsilon)}{(1-\alpha)(1-2\alpha)}\delta(P_i^*,c_i^*)\end{aligned}$$

4. **Tổng hợp**

Cộng gộp kết quả từ Bước 1 và Bước 3:

$$\begin{aligned}\delta(P_i^*,u_i) \le{}& \underbrace{\delta(P_i^* \cap P_i, c_i^*) + \delta(P_i^* \setminus P_i, c_i^*)}_{\delta(P_i^*,c_i^*)} \\ &+ \left[\frac{\alpha(4+3\varepsilon)}{1-2\alpha} + \frac{\alpha(2+\alpha\varepsilon)}{(1-\alpha)(1-2\alpha)}\right]\delta(P_i^*,c_i^*)\end{aligned}$$

$$\begin{aligned}\text{Hệ số vế phải} &= \frac{\alpha(4+3\varepsilon)(1-\alpha)+\alpha(2+\alpha\varepsilon)}{(1-\alpha)(1-2\alpha)} \\ &= \frac{(4\alpha+3\alpha\varepsilon-4\alpha^2-3\alpha^2\varepsilon)+(2\alpha+\alpha^2\varepsilon)}{(1-\alpha)(1-2\alpha)} \\ &= \frac{6\alpha-4\alpha^2+3\alpha\varepsilon-2\alpha^2\varepsilon}{(1-\alpha)(1-2\alpha)}\end{aligned}$$

Ta nhận thấy rằng $3\alpha\varepsilon - 2\alpha^2\varepsilon = \alpha\varepsilon(3-2\alpha)$. Biểu thức trong Bổ đề yêu cầu là $4\alpha\varepsilon - 3\alpha^2\varepsilon = \alpha\varepsilon(4-3\alpha)$. Vì $\alpha < 1$, ta có $3\alpha\varepsilon - 2\alpha^2\varepsilon \leq 4\alpha\varepsilon - 3\alpha^2\varepsilon$ (do hiệu số là $\alpha\varepsilon(1-\alpha) > 0$). Do đó, ta có thể nới lỏng chặn trên để khớp với công thức tổng quát của bổ đề:

$$\delta(P_i^*, u_i) \leq \left(1 + \frac{6\alpha+4\alpha\varepsilon-4\alpha^2-3\varepsilon\alpha^2}{(1-\alpha)(1-2\alpha)}\right)\delta(P_i^*, c_i^*)$$

□

Tài liệu

Arthur, D. and Vassilvitskii, S. (2007). *k*-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035.

Ashtiani, H., Kushagra, S., and Ben-David, S. (2016). Clustering with same-cluster queries. In *Proceedings of the 29th International Conference on Neural Information Processing Systems*, pages 3216–3224.

Awasthi, P., Balcan, M.-F., and Voevodski, K. (2014). Local algorithms for interactive clustering. In *International Conference on Machine Learning*, pages 550–558. PMLR.

Balcan, M.-F. and Blum, A. (2008). Clustering with interactive feedback. In *International Conference on Algorithmic Learning Theory*, pages 316–328. Springer.

Beretta, L., Cohen-Addad, V., Lattanzi, S., and Parotsidis, N. (2023). Multi-swap *k*-means++. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 26069–26091.

Blum, M., Floyd, R. W., Pratt, V. R., Rivest, R. L., and Tarjan, R. E. (1973). Time bounds for selection. *Journal of Computer and System Science*, 7(4):448–461.

Choo, D., Grunau, C., Portmann, J., and Rozhon, V. (2020). *k*-means++: Few more steps yield constant approximation. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1909–1917.

Cohen-Addad, V., Esfandiari, H., Mirrokni, V., and Narayanan, S. (2022). Improved approximations for euclidean *k*-means and *k*-median, via nested quasi-independent sets. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1621–1628.

Cohen-Addad, V. and Karthik, C. S. (2019). Inapproximability of clustering in lp metrics. In *Proceedings of the 60th IEEE Annual Symposium on Foundations of Computer Science*, pages 519–539.

Dasgupta, S. (2008). The hardness of *k*-means clustering. In *Technical Report*.

Ergun, J. C., Feng, Z., Silwal, S., Woodruff, D., and Zhou, S. (2021). Learning-augmented *k*-means clustering. In *Proceedings of the 9th International Conference on Learning Representations*.

Fan, C., Li, P., and Li, X. (2023). Lsds++: Dual sampling for accelerated *k*-means++. In *Proceedings of the 40th International Conference on Machine Learning*, pages 9640–9649.

Friggstad, Z., Rezapour, M., and Salavatipour, M. R. (2019). Local search yields a ptas for *k*-means in doubling metrics. *SIAM Journal on Computing*, 48(2):452–480.

Gamlath, B., Lattanzi, S., Norouzi-Fard, A., and Svensson, O. (2022). Approximate cluster recovery from noisy labels. In *Proceedings of the 35th Conference on Learning Theory*, pages 1463–1509.

Huang, J., Feng, Q., Huang, Z., Zhang, Z., Xu, J., and Wang, J. (2025). New algorithms for the learning-

augmented k-means problem. In *The Thirteenth International Conference on Learning Representations*.

Jaiswal, R., Kumar, A., and Sen, S. (2014). A simple d^2-sampling based ptas for k-means and other clustering problems. *Algorithmica*, 70(1):22–46.

Kraska, T., Beutel, A., Chi, E. H., Dean, J., and Polyzotis, N. (2018). The case for learned index structures. In *Proceedings of the 2018 International Conference on Management of Data*, pages 489–504.

Lattanzi, S. and Sohler, C. (2019). A better k-means++ algorithm via local search. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3662–3671.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137.

Matsui, Y., Ogaki, K., Yamasaki, T., and Aizawa, K. (2017). Pqk-means: Billion-scale clustering for product-quantized codes. In *Proceedings of the 25th ACM International Conference on Multimedia*, pages 1725–1733.

Mitzenmacher, M. (2018). A model for learned bloom filters and optimizing by sandwiching. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 462–471.

Mitzenmacher, M. and Vassilvitskii, S. (2022). Algorithms with predictions. *Communications of the ACM*, 65(7):33–35.

Nguyen, T. D., Chaturvedi, A., and Nguyen, H. (2022). Improved learning-augmented algorithms for k-means and k-medians clustering. In *Proceedings of the 10th International Conference on Learning Representations*.

Vikram, S. and Dasgupta, S. (2016). Interactive bayesian hierarchical clustering. In *International Conference on Machine Learning*, pages 2081–2090. PMLR.

Phụ lục

Nơi tải các tập dữ liệu đã xử lý

- Tập dữ liệu PHY: https://drive.google.com/file/d/1ldCuGcioN_nR4wl6JYHMeXIbBGntHxGe/view?usp=sharing
- Tập dữ liệu MNIST, USPS: Có thể sử dụng thư viện Scikit-Learn để tải hai tập dữ liệu này.