BỘ GIÁO DỤC VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ

Vũ Việt Hoàng

PHƯƠNG PHÁP TỐI THIỂU LUÂN PHIÊN VÀ ỨNG DỤNG

LUẬN VĂN THẠC SĨ TOÁN HỌC

Hà Nội – 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ

Vũ Việt Hoàng

PHƯƠNG PHÁP TỐI THIỂU LUÂN PHIÊN VÀ ỨNG DỤNG

LUẬN VĂN THẠC SĨ TOÁN HỌC

Ngành: Toán ứng dụng

Mã số: 8 46 01 12

NGƯỜI HƯỚNG DẪN KHOA HỌC:

TS. Lê Hải Yến

Hà Nội – 2024

Lời cam đoan

Tôi xin cam kết rằng nội dung trong luận văn này là kết quả của quá trình tìm hiểu, học hỏi và trau dồi kiến thức cá nhân, được thực hiện dưới sự hướng dẫn tận tâm của TS Lê Hải Yến. Tất cả các kết quả nghiên cứu và ý tưởng từ các tác giả khác, nếu có sử dụng, đều được trích dẫn rõ ràng trong luận văn. Đề tài này chưa từng được bảo vệ trước bất kỳ hội đồng bảo vệ luận văn thạc sĩ nào. Tôi xin hoàn toàn chịu trách nhiệm về những cam kết nêu trên.

Hà Nội, tháng 10 năm 2024

Tác giả luận văn

Vũ Việt Hoàng

Lời cảm ơn

Trước tiên, tôi xin bày tỏ lòng biết ơn sâu sắc nhất đến TS Lê Hải Yến, người đã không chỉ tận tình hướng dẫn mà còn truyền cảm hứng cho tôi trong suốt quá trình xác định đề tài và định hướng nghiên cứu cho Luận văn này. Dưới sự hướng dẫn đầy tâm huyết và quan tâm của cô, tôi đã có cơ hội tiếp cận một cách sâu sắc và chân thực với con đường nghiên cứu khoa học. Cô đã luôn quan tâm, giúp đỡ, động viên tôi rất nhiều trong suốt quá trình học tập và nghiên cứu để tôi có thể hoàn thành Luận văn.

Tôi cũng xin gửi lời tri ân chân thành đến tất cả các thầy cô đã giảng dạy và trang bị cho tôi những kiến thức quý báu trong quá trình học tập và nghiên cứu. Đặc biệt, tôi xin cảm ơn Viện Toán học và cơ sở đào tạo Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam đã tạo điều kiện thuận lợi, môi trường học tập chuyên nghiệp và đầy cảm hứng cho tôi.

Cuối cùng, tôi xin chân thành cảm ơn Quỹ VINIF đã hỗ trợ tài chính trong năm đầu tiên của chương trình Cao học, mang lại cho tôi không chỉ nguồn lực mà còn là động lực lớn để tiếp tục theo đuổi đam mê nghiên cứu khoa học. Mọi sự hỗ trợ từ Quỹ VINIF là động lực giúp tôi tiến xa hơn trên con đường học tập và phát triển bản thân.

Hà Nội, tháng 10 năm 2024

Tác giả luận văn

Vũ Việt Hoàng

Mục lục

Danh mục các kí hiệu

$\mathbb{E}$	Không gian Euclide
$\|\mathbf{x}\|$	Chuẩn Euclide của $\mathbf{x} \in \mathbb{E}$
$\langle \mathbf{x}, \mathbf{y} \rangle$	Tích vô hướng của $\mathbf{x}, \mathbf{y} \in \mathbb{E}$
$\lim$	Giới hạn
$\inf$	Cận dưới
$\mathbb{R}^n$	Không gian vector thực n chiều
$\mathbb{R}^{m \times n}$	Không gian ma trận thực $m \times n$ chiều
$\|A\|_F$	Chuẩn Frobenius của ma trận A
$\|A\|_*$	Chuẩn nguyên tử của ma trận A
sign(.)	Hàm dấu
$\|x\|$	Giá trị tuyệt đối của x

Danh mục các hình vẽ, đồ thị

Mở đầu

Phương pháp tối thiểu luân phiên (alternating minimization method) là một trong những phương pháp cổ điển và quan trọng của Lý thuyết Tối ưu. Được đề xuất ban đầu để giải quyết bài toán cực tiểu của hàm hai biến, phương pháp này đã được mở rộng và phát triển để áp dụng cho bài toán tối thiểu với nhiều biến số. Ý tưởng chính của thuật toán là tại mỗi bước lặp, ta thực hiện tối ưu hóa hàm mục tiêu theo một tập hợp con của các biến trong khi giữ cố định các biến còn lại, nhằm giảm dần giá trị của hàm mục tiêu. Quá trình này lặp đi lặp lại cho đến khi đạt được một tiêu chí dừng nào đó, thường là khi thay đổi giá trị hàm mục tiêu giữa hai bước liên tiếp là nhỏ hơn một ngưỡng nhất định. Phương pháp tối thiểu luân phiên đã chứng tỏ tính linh hoạt và hiệu quả trong nhiều bài toán tối ưu khác nhau, như bài toán phân cụm K-Trung Bình, khôi phục ma trận, phân tích ma trận, và nhiều lĩnh vực khác của khoa học và kỹ thuật.

Phương pháp này không chỉ có ý nghĩa về mặt lý thuyết mà còn có nhiều ứng dụng thực tiễn. Năm 1973, Powell đã nghiên cứu sâu hơn về phương pháp này và đưa ra các ví dụ minh họa rằng dãy lặp sinh bởi phương pháp có thể không hội tụ đến điểm cực tiểu toàn cục khi chỉ xét theo từng tọa độ ([1]). Tuy nhiên, những phát hiện này cũng mở ra cơ hội để cải tiến phương pháp

bằng cách áp dụng các điều kiện hội tụ bổ sung hoặc sử dụng các chiến lược lựa chọn biến linh hoạt hơn. Vào những năm 1980, Csiszar và Tusnady đã chứng minh được sự hội tụ của phương pháp đối với hàm hai biến trong một số trường hợp nhất định, từ đó mở đường cho việc nghiên cứu tính hội tụ và ổn định của phương pháp trong các bài toán nhiều biến phức tạp hơn (xem trong [2] và các tài liệu tham khảo trong đó).

Trong những năm gần đây, các nhà nghiên cứu đã tập trung vào việc phát triển và cải tiến phương pháp tối thiểu luân phiên để đối phó với các bài toán có quy mô lớn và có cấu trúc phức tạp hơn. Trong bài toán phân cụm K-Trung Bình, một trong những ứng dụng phổ biến nhất của phương pháp này, các phiên bản cải tiến đã được phát triển để tăng tốc độ hội tụ và nâng cao độ chính xác, như được trình bày trong [3]. Đối với bài toán khôi phục ma trận, phương pháp này đã được ứng dụng thành công trong việc xây dựng các thuật toán hiệu quả để khôi phục ma trận từ các dữ liệu quan sát không đầy đủ, trong đó nghiên cứu của Hastie và các đồng tác giả [4] đã mang lại những bước tiến đáng kể.

Phương pháp tối thiểu luân phiên cũng đóng vai trò quan trọng trong các bài toán phân tích ma trận, với nhiều ứng dụng đa dạng như xử lý ảnh, học máy, và khai phá dữ liệu. Những nghiên cứu mới nhất tiếp tục tập trung vào việc cải tiến các tiêu chí hội tụ, tối ưu hóa thuật toán để giảm thiểu số bước lặp cần thiết, và tích hợp các kỹ thuật học sâu để xử lý các bài toán lớn với dữ liệu có cấu trúc phức tạp. Do đó, việc nghiên cứu sâu hơn về phương pháp tối thiểu luân phiên và ứng dụng của nó không chỉ có ý nghĩa lý thuyết mà còn mang lại nhiều giá trị thực tiễn trong việc giải quyết các vấn đề tối ưu hóa

hiện đại.

Trong luận văn này, chúng tôi sẽ tìm hiểu về phương pháp này và nghiên cứu sự hội tụ của nó dưới một số điều kiện. Đồng thời, chúng tôi cũng sẽ xem xét phương pháp này trong một vài mô hình cụ thể. Luận văn được chia thành các chương như sau:

- *Chương 1*: Trình bày các khái niệm và ký hiệu cơ bản trong lý thuyết tối ưu, giới thiệu chi tiết về phương pháp tối thiểu luân phiên và các tính chất hội tụ của nó.

- *Chương 2*: Nghiên cứu ứng dụng của phương pháp tối thiểu luân phiên trong bài toán phân cụm K-Trung Bình, bao gồm việc so sánh với thuật toán Lloyd truyền thống.

- *Chương 3*: Giới thiệu và áp dụng thuật toán bài toán khôi phục ma trận, trình bày một phiên bản nhanh của thuật toán luân phiên phân tích sự hội tụ và độ phức tạp tính toán của thuật toán. Cuối cùng, chúng tôi so sánh bằng thực nghiệm giữa hai phương pháp tối thiểu luân và phiên bản cải tiến của nó cho bài toán khôi phục ma trận.

  Trong luận văn này, chúng tôi đóng góp việc tiến hành các thí nghiệm trên ngôn ngữ lập trình Python.

Chương 1

Phương pháp tối thiểu luân phiên

Chương này chúng tôi trình bày về phương pháp tối thiểu luân phiên, một kỹ thuật phổ biến để giải quyết các bài toán tối ưu phức tạp thông qua việc chia nhỏ và tối ưu từng phần. Cấu trúc chương được trình bày theo thứ tự sau:

- Mục 1.1: Trình bày một số khái niệm và ký hiệu quan trọng trong tối ưu.
- Mục 1.2: Giới thiệu về phương pháp tối thiểu luân phiên và cách thức hoạt động.
- Mục 1.3: Trình bày khái niệm cực tiểu từng tọa độ và nghiên cứu sự hội tụ của dãy lặp đến điểm cực tiểu từng tọa độ.
- Mục 1.4: Trình bày cách áp dụng tối thiểu luân phiên cho các mô hình hỗn hợp. Phân tích sự hội tụ của phương pháp cho mô hình này trong điều kiện lồi.

Nội dung chương này được tham khảo từ tài liệu [2].

1.1 Các khái niệm và một số kí hiệu

Trong phần này, chúng tôi tập trung vào các hàm giá trị thực mở rộng, trình bày một số khái niệm quan trọng và kết quả cơ bản trong lý thuyết Tối ưu. Ta kí hiệu không gian vector $\mathbb{E}$ là không gian Euclid có chuẩn được định nghĩa $\|x\| = \sqrt{\langle x, x\rangle}$, trong đó $\langle \cdot, \cdot \rangle$ là tích vô hướng được trang bị cho $\mathbb{E}$.

Định nghĩa 1.1. Cho hàm giá trị thực mở rộng $f : \mathbb{E} \to [-\infty, \infty]$, miền xác định được định nghĩa bởi tập:

$$\mathrm{dom}(f) = \{\mathbf{x} \in \mathbb{E} : f(\mathbf{x}) < \infty\}.$$

Định nghĩa 1.2. Tập trên đồ thị (epigraph) của một hàm giá trị thực mở rộng $f : \mathbb{E} \to [-\infty, \infty]$ được định nghĩa bởi:

$$\mathrm{epi}(f) = \{(\mathbf{x}, y) : f(\mathbf{x}) \leq y, \mathbf{x} \in \mathbb{E}, y \in \mathbb{R}\}.$$

Định nghĩa 1.3. Một hàm $f : \mathbb{E} \to [-\infty, \infty]$ được gọi là hàm chính thường (proper) nếu nó không đạt giá trị $-\infty$ và tồn tại ít nhất một điểm $\mathbf{x} \in \mathbb{E}$ sao cho $f(\mathbf{x}) < \infty$.

Định nghĩa 1.4. Một hàm $f : \mathbb{E} \to [-\infty, \infty]$ được gọi là đóng nếu phần trên đồ thị của nó là tập đóng.

Ví dụ 1.5. Với tập con bất kỳ $D \subseteq \mathbb{E}$, xét hàm f được định nghĩa là hàm mở rộng giá trị thực như sau:

$$f(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in D, \\ \infty, & \mathbf{x} \notin D \end{cases}.$$

Khi đó, ta có:

$$\mathrm{dom}\,(f) = D$$

Tập trên đồ thị của f là

$$\mathrm{epi}(f) = \{(\mathbf{x}, y) : \mathbf{x} \in D \text{ và } y \geq 0\}.$$

Hàm f là hàm chính thường nếu D khác rỗng. Hàm f là hàm đóng nếu $\mathrm{epi}(f)$ đóng, nghĩa là D là tập đóng.

Định nghĩa 1.6. Một hàm $f : \mathbb{E} \to [-\infty, \infty]$ được gọi là nửa liên tục dưới tại $\mathbf{x} \in \mathbb{E}$ nếu

$$f(\mathbf{x}) \leq \liminf_{n\to\infty} f\left(\mathbf{x}_n\right).$$

với mọi dãy $\{\mathbf{x}_n\}_{n\geq 1} \subseteq \mathbb{E}$ sao cho $\mathbf{x}_n \to \mathbf{x}$ khi $n \to \infty$. Một hàm $f : \mathbb{E} \to [-\infty, \infty]$ được gọi là nửa liên tục dưới nếu nó nửa liên tục dưới tại mọi điểm trong $\mathbb{E}$.

Định nghĩa 1.7. Với mọi $\alpha \in \mathbb{R}$, tập mức α của một hàm $f : \mathbb{E} \to [-\infty, \infty]$ là tập

$$\mathrm{Lev}(f, \alpha) = \{\mathbf{x} \in \mathbb{E} : f(\mathbf{x}) \leq \alpha\}.$$

Định nghĩa 1.8. Một hàm giá trị thực mở rộng $f : \mathbb{E} \to [-\infty, \infty]$ được gọi là hàm lồi nếu phần trên đồ thị $\mathrm{epi}(f)$ là một tập lồi.

Định nghĩa 1.9. Một hàm giá trị thực mở rộng: $f : \mathbb{E} \to [-\infty, \infty]$ được gọi là hàm lồi khi và chỉ khi:

$$f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}), \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{E}, \lambda \in [0, 1].$$

Định lý dưới đây chỉ ra sử tương đương tính đóng, nửa liên tục dưới và tính đóng của tập mức.

Định lý 1.10. Cho $f : \mathbb{E} \to [-\infty, \infty]$. Khi đó, ba mệnh đề sau đây là tương đương:

(i) f là nửa liên tục dưới.

(ii) f là hàm đóng.

(iii) Với mọi $\alpha \in \mathbb{R}$, tập mức

$$\mathrm{Lev}(f, \alpha) = \{\mathbf{x} \in \mathbb{E} : f(\mathbf{x}) \leq \alpha\},$$

là tập đóng.

Định nghĩa 1.11. Giả sử $f : \mathbb{E} \to (-\infty, \infty]$ là một hàm chính thường và $\mathbf{x} \in \mathrm{dom}(f)$. Một vector $\mathbf{g} \in \mathbb{E}$ được gọi là dưới đạo hàm của f tại $\mathbf{x}$ nếu $f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{g}, \mathbf{y} - \mathbf{x} \rangle$ với mọi $\mathbf{y} \in \mathbb{E}$.

Định nghĩa 1.12. Tập hợp tất cả các dưới đạo hàm của f tại $\mathbf{x}$ được gọi là dưới vi phân của f tại $\mathbf{x}$ và được ký hiệu là $\partial f(\mathbf{x})$:

$$\partial f(\mathbf{x}) \equiv \{\mathbf{g} \in \mathbb{E}^* : f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{g}, \mathbf{y} - \mathbf{x} \rangle \text{ với mọi } \mathbf{y} \in \mathbb{E}\}.$$

Định lý 1.13. Giả sử $f : \mathbb{E} \to (-\infty, \infty]$ là một hàm chính thường và đóng, và C là một tập compact thỏa mãn $C \cap \mathrm{dom}(f) \neq \emptyset$. Khi đó:

(a) f bị chặn dưới trên C.

(b) f đạt giá trị nhỏ nhất của nó trên C.

Chứng minh. (a) Ta sẽ chứng minh bằng phản chứng. Giả sử rằng hàm f không bị chặn dưới trên tập C. Điều này có nghĩa là tồn tại một dãy $\{\mathbf{x}_n\}_{n \geq 1} \subseteq C$ sao cho

$$\lim_{n \to \infty} f(\mathbf{x}_n) = -\infty.$$

Vì C là tập compact, ta có thể lấy được một dãy con $\{\mathbf{x}_{n_k}\}_{k\geq 1}$ hội tụ đến một điểm $\overline{\mathbf{x}} \in C$.

Theo định nghĩa 1.6, hàm f là nửa liên tục dưới, do đó

$$f(\overline{\mathbf{x}}) \leq \liminf_{k\to\infty} f(\mathbf{x}_{n_k}),$$

điều này mâu thuẫn với phát biểu trên.

(b) Ta ký hiệu f_{opt} là giá trị cận dưới của hàm f trên tập C. Khi đó tồn tại một dãy $\{\mathbf{x}_n\}_{n\geq 1}$ sao cho $f(\mathbf{x}_n) \to f_{\text{opt}}$ khi $n \to \infty$. Tương tự như trên, lấy một dãy con $\{\mathbf{x}_{n_k}\}_{k\geq 1}$ hội tụ đến một điểm $\overline{\mathbf{x}} \in C$. Do tính nửa liên tục dưới của f, suy ra rằng

$$f(\overline{\mathbf{x}}) \leq \lim_{k\to\infty} f(\mathbf{x}_{n_k}) = f_{\text{opt}},$$

chứng tỏ rằng $\overline{\mathbf{x}}$ là một điểm cực tiểu của hàm f trên tập C. □

1.2 Giới thiệu phương pháp

Cho $\mathbb{E}_1, \mathbb{E}_2, \cdots, \mathbb{E}_p$ là các không gian Euclide n chiều. Xét bài toán

$$\min_{\mathbf{x}_1\in\mathbb{E}_1, \mathbf{x}_2\in\mathbb{E}_2, \ldots, \mathbf{x}_p\in\mathbb{E}_p} F\left(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p\right). \tag{1.1}$$

Ta ký hiệu không gian tích $\mathbb{E} = \mathbb{E}_1 \times \mathbb{E}_2 \times \cdots \times \mathbb{E}_p$. Để đơn giản, ta ký hiệu

$$\mathbf{x} = \left(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p\right),$$

viết gọn lại là $\mathbf{x} = (\mathbf{x}_i)_{i=1}^p$.

Trong phần này, ta xem xét phương pháp tối thiểu luân phiên trong đó lần lượt chọn một khối theo chu kỳ và cập nhật giá trị mới của khối đó bằng một giá trị tối thiểu của hàm mục tiêu đối với khối được chọn. Phương pháp tối thiểu luân phiên để tối thiểu F được mô tả như trong thuật toán 1. Ta ký hiệu

giá trị lần lặp thứ k bởi $\mathbf{x}^k = \left(\mathbf{x}_1^k, \mathbf{x}_2^k, \ldots, \mathbf{x}_p^k\right)$. Mỗi lần lặp của phương pháp tối thiểu luân phiên bao gồm p các lần lặp con và các dãy con của các lần lặp này sẽ được ký hiệu bởi các dãy:

$$\begin{aligned}
\mathbf{x}^{k,0} &= \mathbf{x}^k = \left(\mathbf{x}_1^k, \mathbf{x}_2^k, \ldots, \mathbf{x}_p^k\right), \\
\mathbf{x}^{k,1} &= \left(\mathbf{x}_1^{k+1}, \mathbf{x}_2^k, \ldots, \mathbf{x}_p^k\right), \\
\mathbf{x}^{k,2} &= \left(\mathbf{x}_1^{k+1}, \mathbf{x}_2^{k+1}, \mathbf{x}_3^k, \ldots, \mathbf{x}_p^k\right), \\
&\vdots \\
\mathbf{x}^{k,p} &= \mathbf{x}^{k+1} = \left(\mathbf{x}_1^{k+1}, \mathbf{x}_2^{k+1}, \ldots, \mathbf{x}_p^{k+1}\right).
\end{aligned} \tag{1.2}$$

Thuật toán 1 Phương pháp tối thiểu luân phiên

Khởi tạo $\mathbf{x}^0 = \left(\mathbf{x}_1^0, \mathbf{x}_2^0, \ldots, \mathbf{x}_p^0\right) \in \mathrm{dom}(F)$

for $k = 1, 2, \ldots$ **do**

  for $i = 1, 2, \ldots$ **do**

$$\mathbf{x}_i^{k+1} \in \mathrm{argmin}_{\mathbf{x}_i \in \mathbb{E}_i} F\left(\mathbf{x}_1^{k+1}, \ldots, \mathbf{x}_{i-1}^{k+1}, \mathbf{x}_i, \mathbf{x}_{i+1}^k, \ldots, \mathbf{x}_p^k\right). \tag{1.3}$$

  end for

end for

Với mọi $i \in \{1, 2, \ldots, p\}$, ta định nghĩa $\mathcal{U}_i : \mathbb{E}_i \to \mathbb{E}$ là phép biến đổi tuyến tính được cho bởi

$$\mathcal{U}_i(\mathbf{d}) = (\mathbf{0}, \ldots, \mathbf{0}, \underbrace{\mathbf{d}}_{\text{khối thứ } i}, \mathbf{0}, \ldots, \mathbf{0}), \mathbf{d} \in \mathbb{E}_i.$$

Sử dụng kí hiệu trên, ta có thể viết lại bước tổng quát của phương pháp tối thiểu luân phiên như sau:

- đặt $\mathbf{x}^{k,0} = \mathbf{x}^k$;
- với $i = 1, 2, \ldots, p$, tính $\mathbf{x}^{k,i} = \mathbf{x}^{k,i-1} + \mathcal{U}_i\left(\tilde{\mathbf{y}} - \mathbf{x}_i^k\right)$, trong đó

$$\tilde{\mathbf{y}} \in \mathrm{argmin}_{\mathbf{y} \in \mathbb{E}_i} F\left(\mathbf{x}^{k,i-1} + \mathcal{U}_i\left(\mathbf{y} - \mathbf{x}_i^k\right)\right); \tag{1.4}$$

- đặt $\mathbf{x}^{k+1} = \mathbf{x}^{k,p}$.

Định lý dưới đấy chỉ ra nếu F là hàm chính thường và đóng và có các tập mức bị chặn, thì bài toán (1.1) có nghiệm tối ưu. Khi đó phương pháp tối thiểu luân phiên là xác định tốt, nghĩa là các bài toán tối ưu (1.3) có nghiệm tối ưu.

Bổ đề 1.14. Giả sử rằng $F : \mathbb{E} \to (-\infty, \infty]$ là một hàm chính thường và đóng. Giả sử thêm rằng F có các tập mức bị chặn; nghĩa là,

$$\mathrm{Lev}(F, \alpha) = \{\mathbf{x} \in \mathbb{E} : F(\mathbf{x}) \leq \alpha\},$$

là bị chặn với mọi $\alpha \in \mathbb{R}$. Khi đó hàm F có ít nhất một điểm cực tiểu, và với mọi $\overline{\mathbf{x}} \in \mathrm{dom}(F)$ và $i \in \{1, 2, \ldots, p\}$, bài toán

$$\min_{\mathbf{y} \in \mathbb{E}_i} F\left(\overline{\mathbf{x}} + \mathcal{U}_i\left(\mathbf{y} - \overline{\mathbf{x}}_i\right)\right), \tag{1.5}$$

có nghiệm tối ưu.

Chứng minh. Lấy $\tilde{\mathbf{x}} \in \mathrm{dom}(F)$. Khi đó

$$\mathrm{argmin}_{\mathbf{x} \in \mathbb{E}} F(\mathbf{x}) = \mathrm{argmin}_{\mathbf{x} \in \mathbb{E}} \{F(\mathbf{x}) : \mathbf{x} \in \mathrm{Lev}(F, F(\tilde{\mathbf{x}}))\}.$$

Vì F là hàm đóng với các tập mức bị chặn, nên $\mathrm{Lev}(F, F(\tilde{\mathbf{x}}))$ là compact. Do đó, theo Định lý 1.13, tối ưu F trên toàn bộ không gian tồn tại một điểm tối ưu.

Vì hàm $F^i_{\overline{\mathbf{x}}} : \mathbf{y} \mapsto F\left(\overline{\mathbf{x}} + \mathcal{U}_i\left(\mathbf{y} - \overline{\mathbf{x}}_i\right)\right)$ là chính quy và đóng với các tập mức bị chặn, lập luận tương tự cho thấy rằng bài toán (1.5) cũng có tồn tại một điểm tối ưu. □

Chú ý rằng chỉ cần tồn tại tập mức α của F khác rỗng bị chặn thì hàm F có điểm cực tiểu. Tuy nhiên, điều này không đảm bảo với $\overline{\mathbf{x}} \in \mathrm{dom}(F)$ thì

tồn tại một tập mức của $F^i_{\overline{\mathbf{x}}}$ bị chặn khác rỗng, tức không đảm bảo $F^i_{\overline{\mathbf{x}}}$ có điểm cực tiểu. Xét ví du:

$$F(\mathrm{x}_1, \mathrm{x}_2) = \begin{cases} -1, & x_1 = x_2 = 0, \\ e^{x_1+x_2}, & \text{còn lại.} \end{cases}$$

Ta thấy rằng F là hàm chính thường đóng, có tập mức $\alpha = -0.5$ bị chặn khác rỗng nên có nghiệm tối ưu. Tuy nhiên với mọi $\overline{\mathbf{x}} \neq \mathbf{0}$ thì đều không tồn tại một điểm cực tiểu cho các hàm $F^i_{\overline{\mathbf{x}}}$.

1.3 Khái niệm cực tiểu từng tọa độ

Trong phần này, chúng ta xem khái niệm cực tiểu từng tọa độ và một số điều kiện thuật toán tối thiểu luân phiên hội tụ tới chúng.

Định nghĩa 1.15. Một vector $\mathbf{x}^* \in \mathbb{E}$ là cực tiểu từng tọa độ của một hàm $F : \mathbb{E}_1 \times \mathbb{E}_2 \times \cdots \times \mathbb{E}_p \to (-\infty, \infty]$ nếu $\mathbf{x}^* \in \mathrm{dom}(F)$ và

$$F(\mathbf{x}^*) \leq F(\mathbf{x}^* + \mathcal{U}_i(\mathbf{y})) \text{ với mọi } i = 1, 2, \ldots, p, \mathbf{y} \in \mathbb{E}_i.$$

Định lý tiếp theo đưa ra các giả thiết về hàm mục tiêu đề đảm bảo sự hội tụ về điểm cực tiểu của phương pháp.

Định lý 1.16. Giả sử rằng $F : \mathbb{E} \to (-\infty, \infty]$ là một hàm chính thường và đóng, liên tục trên miền xác định. Giả sử rằng

(A) với mỗi $\overline{\mathbf{x}} \in \mathrm{dom}(F)$ và $i \in \{1, 2, \ldots, p\}$, $\min_{\mathbf{y} \in \mathbb{E}_i} F(\overline{\mathbf{x}} + \mathcal{U}_i(\mathbf{y} - \overline{\mathbf{x}}_i)$ có nghiệm tối ưu duy nhất;

(B) các tập mức của F bị chặn, nghĩa là với mọi $\alpha \in \mathbb{R}$, tức là tập

$$\mathrm{Lev}(F, \alpha) = \{\mathbf{x} \in \mathbb{E} : F(\mathbf{x}) \leq \alpha\}$$

là bị chặn.

Gọi $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ là dãy được sinh ra bởi phương pháp tối thiểu luân phiên (Thuật toán 1) để tối thiểu F. Khi đó $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ là bị chặn, và bất kỳ điểm giới hạn nào của dãy này là một điểm cực tiểu từng tọa độ.

Chứng minh. Theo cách xây dựng của thuật toán, dãy $\left\{F\left(\mathbf{x}^k\right)\right\}_{k\geq 0}$ là không tăng, suy ra $\left\{\mathbf{x}^k\right\}_{k\geq 0} \subseteq \mathrm{Lev}\left(F, F\left(\mathbf{x}^0\right)\right)$. Theo điều kiện (B), dãy $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ bị chặn, với tính đóng của F ta có $\left\{F\left(\mathbf{x}^k\right)\right\}_{k\geq 0}$ bị chặn dưới. Khi đó $\left\{F\left(\mathbf{x}^k\right)\right\}_{k\geq 0}$ hội tụ tới một số thực $\bar{F}$ nào đó. Vì $F\left(\mathbf{x}^k\right) \geq F\left(\mathbf{x}^{k,1}\right) \geq F\left(\mathbf{x}^{k+1}\right)$, suy ra $\left\{F\left(\mathbf{x}^{k,1}\right)\right\}_{k\geq 0}$ cũng hội tụ tới cùng giá trị $\bar{F}$.

Ta giả sử $\overline{\mathbf{x}} = (\overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2, \ldots, \overline{\mathbf{x}}_p)$ là một điểm giới hạn của $\left\{\mathbf{x}^k\right\}_{k\geq 0}$. Khi đó, tồn tại một dãy con $\left\{\mathbf{x}^{k_j}\right\}_{j\geq 0}$ hội tụ tới $\overline{\mathbf{x}}$. Vì dãy $\left\{\mathbf{x}^{k_j,1}\right\}_{j\geq 0}$ bị chặn, ta lấy được dãy con $\left\{\mathbf{x}^{k'_j,1}\right\}_{j\geq 0}$ hội tụ, để đơn giản kí hiệu ta giả sử rằng $\left\{\mathbf{x}^{k_j,1}\right\}_{j\geq 0}$ hội tụ và hội tụ tới một vector $(\mathbf{v}, \overline{\mathbf{x}}_2, \ldots, \overline{\mathbf{x}}_p)$ $(\mathbf{v} \in \mathbb{E}_1)$. Theo cách xây dựng của phương pháp,

$$F\left(\mathbf{x}_1^{k_j+1}, \mathbf{x}_2^{k_j}, \ldots, \mathbf{x}_p^{k_j}\right) \leq F\left(\mathbf{x}_1, \mathbf{x}_2^{k_j}, \ldots, \mathbf{x}_p^{k_j}\right) \text{ với bất kỳ } \mathbf{x}_1 \in \mathbb{E}_1.$$

Lấy giới hạn $j \to \infty$ và sử dụng tính đóng của F, cũng như tính liên tục của F trên miền xác định, ta có

$$F\left(\mathbf{v}, \overline{\mathbf{x}}_2, \ldots, \overline{\mathbf{x}}_p\right) \leq F\left(\mathbf{x}_1, \overline{\mathbf{x}}_2, \ldots, \overline{\mathbf{x}}_p\right) \text{ với bất kỳ } \mathbf{x}_1 \in \mathbb{E}_1.$$

Vì $\left\{F\left(\mathbf{x}^k\right)\right\}_{k\geq 0}$ và $\left\{F\left(\mathbf{x}^{k,1}\right)\right\}_{k\geq 0}$ hội tụ tới cùng một giá trị, ta có

$$F\left(\mathbf{v}, \overline{\mathbf{x}}_2, \ldots, \overline{\mathbf{x}}_p\right) = F\left(\overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2, \ldots, \overline{\mathbf{x}}_p\right),$$

Theo tính duy nhất của nghiệm tối ưu đối với khối đầu tiên (điều kiện (A)) ta suy ra được $\mathbf{v} = \overline{\mathbf{x}}_1$. Do đó,

$$F\left(\overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2, \ldots, \overline{\mathbf{x}}_p\right) \leq F\left(\mathbf{x}_1, \overline{\mathbf{x}}_2, \ldots, \overline{\mathbf{x}}_p\right) \text{ với bất kỳ } \mathbf{x}_1 \in \mathbb{E}_1,$$

điều này là điều kiện đầu tiên cho cực tiểu từng tọa độ. Chúng ta đã chứng minh rằng $\mathbf{x}^{k_j,1} \to \overline{\mathbf{x}}$ khi $j \to \infty$. Làm tương tự ta thay thế $\mathbf{x}^{k_j,1}$ bởi $\mathbf{x}^{k_j}$ và tập trung vào tọa độ thứ hai để thu được

$$F\left(\overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2, \ldots, \overline{\mathbf{x}}_p\right) \leq F\left(\overline{\mathbf{x}}_1, \mathbf{x}_2, \overline{\mathbf{x}}_3, \ldots, \overline{\mathbf{x}}_p\right) \text{ với bất kỳ } \mathbf{x}_2 \in \mathbb{E}_2,$$

điều này là điều kiện thứ hai cho cực tiểu từng tọa độ. Làm tương tự các khối còn lại, chúng ta chứng minh rằng $\overline{\mathbf{x}}$ thỏa mãn tất cả các điều kiện cho cực tiểu từng tọa độ. □

Powell đã đưa ra một ví dụ về hàm dưới đây vi phạm cả hai điều kiện (A) và (B) trong Định lý 1.16. Khi đó chuỗi sinh ra bởi thuật toán không hội tụ tới điểm cực tiểu theo từng tọa độ:

$$\begin{aligned}\varphi(x, y, z) = &-xy - yz - zx + [x-1]_+^2 + [-x-1]_+^2 \\ &+ [y-1]_+^2 + [-y-1]_+^2 + [z-1]_+^2 + [-z-1]_+^2,\end{aligned}$$

trong đó

$$[a]_+ = \begin{cases} a & \text{nếu } a > 0, \\ 0 & \text{còn lại.} \end{cases}$$

Tuy nhiên, bằng cách giới hạn miền xác định, ta chỉ ra việc vi phạm một trong hai điều kiện cũng có thể dẫn đến chuỗi sinh ra bởi thuật toán không hội tụ tới điểm cực tiểu theo từng tọa độ.

Ví dụ 1.17. Xét hàm

$$\begin{aligned}\varphi(x,y,z) = -xy - yz - zx + [x-1]_+^2 + [-x-1]_+^2 \\ + [y-1]_+^2 + [-y-1]_+^2 + [z-1]_+^2 + [-z-1]_+^2,\end{aligned}$$

nếu $(x,y,z) \in [-2,2]^3$ và $\varphi(x,y,z) = +\infty$ trong trường hợp còn lại. Ta thấy rằng $\mathrm{dom}(F) = [-2,2]^3$ và F có tập mức bị chặn.

Lưu ý rằng hàm φ là khả vi. Giữ cố định y và z, dễ thấy

$$\mathrm{argmin}_x\, \varphi(x,y,z) = \begin{cases} \mathrm{sgn}(y+z)\left(1 + \frac{1}{2}|y+z|\right), & y+z \neq 0, \\ [-1,1], & y+z = 0 \end{cases}$$

Và tương tự nhờ tính đối xứng

$$\mathrm{argmin}_y\, \varphi(x,y,z) = \begin{cases} \mathrm{sgn}(x+z)\left(1 + \frac{1}{2}|x+z|\right), & x+z \neq 0, \\ [-1,1], & x+z = 0, \end{cases}$$

$$\mathrm{argmin}_z\, \varphi(x,y,z) = \begin{cases} \mathrm{sgn}(x+y)\left(1 + \frac{1}{2}|x+y|\right), & x+y \neq 0, \\ [-1,1], & x+y = 0. \end{cases}$$

Giả sử rằng $\varepsilon > 0$ và chúng ta khởi tạo phương pháp tối thiểu luân phiên với điểm $\left(-1-\varepsilon, 1+\frac{1}{2}\varepsilon, -1-\frac{1}{4}\varepsilon\right)$. Thì sáu lần lặp đầu tiên là

$$\left(1+\frac{1}{8}\varepsilon, 1+\frac{1}{2}\varepsilon, -1-\frac{1}{4}\varepsilon\right)$$
$$\left(1+\frac{1}{8}\varepsilon, -1-\frac{1}{16}\varepsilon, -1-\frac{1}{4}\varepsilon\right)$$
$$\left(1+\frac{1}{8}\varepsilon, -1-\frac{1}{16}\varepsilon, 1+\frac{1}{32}\varepsilon\right)$$
$$\left(-1-\frac{1}{64}\varepsilon, -1-\frac{1}{16}\varepsilon, 1+\frac{1}{32}\varepsilon\right)$$
$$\left(-1-\frac{1}{64}\varepsilon, 1+\frac{1}{128}\varepsilon, 1+\frac{1}{32}\varepsilon\right)$$
$$\left(-1-\frac{1}{64}\varepsilon, 1+\frac{1}{128}\varepsilon, -1-\frac{1}{256}\varepsilon\right).$$

Chúng ta cơ bản quay trở lại điểm đầu tiên, nhưng với $\frac{1}{64}\varepsilon$ thay thế cho ε. Quá trình tiếp tục bằng cách xoay quanh sáu điểm

$$(1,1,-1),(1,-1,-1),(1,-1,1),(-1,-1,1),(-1,1,1),(-1,1,-1).$$

Không điểm nào trong số các điểm này là điểm dừng của φ. Thực vậy,

$$\nabla\varphi(1,1,-1) = (0,0,-2),$$
$$\nabla\varphi(-1,1,1) = (-2,0,0),$$
$$\nabla\varphi(1,-1,1) = (0,-2,0),$$
$$\nabla\varphi(-1,-1,1) = (0,0,2),$$
$$\nabla\varphi(1,-1,-1) = (2,0,0),$$
$$\nabla\varphi(-1,1,-1) = (0,2,0).$$

Như vậy, tất cả các điểm này đều có đạo hàm không bằng $\mathbf{0}$, và do đó, không điểm nào trong số đó là điểm dừng của hàm φ. Vì chuỗi trong ví dụ này là bị chặn nên việc không hội tụ về điểm cực tiểu theo từng tọa độ là do sự không duy nhất của các nghiệm tối ưu của các bài toán con.

1.4 Phương pháp tối tiểu luân phiên cho mô hình hỗn hợp

Trong phần này, chúng ta áp dụng phương pháp tối thiểu luân phiên để giải mô hình hỗn hợp $F = f + g$ trong trường hợp g có cấu trúc tách biệt với các hàm thành phần $g_i : \mathbb{E}_i \to (-\infty, \infty]$. Bây giờ, ta xét một mô hình hỗn hợp như sau

$$\min_{\mathbf{x}_1 \in \mathbb{E}_1, \mathbf{x}_2 \in \mathbb{E}_2, \ldots, \mathbf{x}_p \in \mathbb{E}_p} \left\{ F\left(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p\right) = f\left(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p\right) + \sum_{j=1}^{p} g_j\left(\mathbf{x}_j\right) \right\}. \tag{1.6}$$

Hàm $g : \mathbb{E} \to (-\infty, \infty]$ được định nghĩa bởi

$$g\left(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p\right) \equiv \sum_{i=1}^{p} g_i\left(\mathbf{x}_i\right).$$

Gradient đối với khối thứ i $(i \in \{1, 2, \ldots, p\})$ được ký hiệu là $\nabla_i f$, và

$$\nabla f(\mathbf{x}) = \left(\nabla_1 f(\mathbf{x}), \nabla_2 f(\mathbf{x}), \ldots, \nabla_p f(\mathbf{x})\right).$$

Lưu ý rằng bằng cách ký hiệu trên, mô hình (1.6) có thể được viết đơn giản là

$$\min_{\mathbf{x} \in \mathbb{E}} \{F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x})\}.$$

Trước khi đi vào phân tích sự hội tụ thuật toán tối thiểu luân phiên cho mô hình hỗn hợp, chúng ta xem xét một số định nghĩa và Định lý liên quan đến mô hình hỗn hợp.

Định nghĩa 1.18. Giả sử $f : \mathbb{E} \to (-\infty, \infty]$ là một hàm chính thường và $g : \mathbb{E} \to (-\infty, \infty]$ là một hàm lồi chính thường sao cho $\mathrm{dom}(g) \subseteq \mathrm{int}(\mathrm{dom}(f))$. Xét bài toán (P): $\min_{\mathbf{x} \in \mathbb{E}} f(\mathbf{x}) + g(\mathbf{x})$.

Một điểm $\mathbf{x}^*$ mà tại đó f khả vi được gọi là điểm dừng của (P) nếu

$$-\nabla f\left(\mathbf{x}^*\right) \in \partial g\left(\mathbf{x}^*\right).$$

Định lý 1.19. Giả sử $f : \mathbb{E} \to (-\infty, \infty]$ là một hàm chính thường, và $g : \mathbb{E} \to (-\infty, \infty]$ là một hàm lồi chính thường sao cho $\mathrm{dom}(g) \subseteq \mathrm{int}(\mathrm{dom}(f))$. Xét bài toán (P)

$$\min_{\mathbf{x}\in\mathbb{E}} f(\mathbf{x}) + g(\mathbf{x}).$$

(a) (điều kiện cần) Nếu $\mathbf{x}^* \in \mathrm{dom}(g)$ là một nghiệm tối ưu địa phương của (P) và f khả vi tại $\mathbf{x}^*$, thì

$$-\nabla f(\mathbf{x}^*) \in \partial g(\mathbf{x}^*). \tag{1.7}$$

(b) (điều kiện cần và đủ cho các bài toán lồi) Giả sử f là hàm lồi. Nếu f khả vi tại $\mathbf{x}^* \in \mathrm{dom}(g)$, thì $\mathbf{x}^*$ là nghiệm tối ưu toàn cục của (P) nếu và chỉ nếu (1.7) được thỏa mãn.

Định lý 1.20. Giả sử $f : \mathbb{E} \to (-\infty, \infty]$ là một hàm chính thường và đóng, và $\mathrm{dom}(f)$ là tập lồi và f khả vi trên $\mathrm{int}(\mathrm{dom}(f))$ và các $g_i : \mathbb{E}_i \to (-\infty, \infty]$ là các hàm lồi đóng chính thường sao cho $\mathrm{dom}(g) \subseteq \mathrm{int}(\mathrm{dom}(f))$. Khi đó:

$\mathbf{x}^* \in \mathrm{dom}(g)$ là một điểm dừng của bài toán (P) nếu và chỉ nếu

$$-\nabla_i f(\mathbf{x}^*) \in \partial g_i(\mathbf{x}_i^*), \quad i = 1, 2, \ldots, p.$$

Giả thiết 1.21. (A) $g_i : \mathbb{E}_i \to (-\infty, \infty]$ là hàm chính thường lồi đóng đối với mọi $i \in \{1, 2, \ldots, p\}$. Ngoài ra, g_i liên tục trên $\mathrm{dom}(g_i)$.

(B) $f : \mathbb{E} \to (-\infty, \infty]$ là hàm đóng; $\mathrm{dom}(f)$ là tập lồi; f khả vi trên $\mathrm{int}(\mathrm{dom}(f))$ và $\mathrm{dom}(g) \subseteq \mathrm{int}(\mathrm{dom}(f))$.

Với cấu trúc trên của hàm F, ta có thể viết lại mỗi bước trong vòng lặp con của phương pháp tối thiểu luân phiên (1.3) có thể được viết ngắn gọn như sau

$$\mathbf{x}_i^{k+1} \in \mathrm{argmin}_{\mathbf{x}_i\in\mathbb{E}_i} \left\{ f\left(\mathbf{x}_1^{k+1}, \ldots, \mathbf{x}_{i-1}^{k+1}, \mathbf{x}_i, \mathbf{x}_{i+1}^{k}, \ldots, \mathbf{x}_p^{k}\right) + g_i(\mathbf{x}_i) \right\}.$$

Ta có điểm $\mathbf{x}^* \in \mathrm{dom}(g)$ là một điểm dừng của bài toán (1.6) nếu nó thỏa mãn $-\nabla f(\mathbf{x}^*) \in \partial g(\mathbf{x}^*)$ (Định nghĩa 1.18) và theo Định lý 1.20, điều kiện này có thể được viết tương đương là $-\nabla_i f(\mathbf{x}^*) \in \partial g_i(\mathbf{x}^*), i = 1, 2, \ldots, p$. Bổ đề sau sẽ chỉ ra rằng các điểm cực tiểu từng tọa độ của F là các điểm dừng của bài toán (1.6).

Bổ đề 1.22. Giả sử rằng Giả thiết 1.21 được thỏa mãn và rằng $\mathbf{x}^* \in \mathrm{dom}(g)$ là một điểm cực tiểu từng tọa độ của $F = f + g$. Khi đó $\mathbf{x}^*$ là một điểm dừng của bài toán (1.6).

Chứng minh. Vì $\mathbf{x}^*$ là một điểm cực tiểu từng tọa độ của F, nên với mọi $i \in \{1, 2, \ldots, p\}$,

$$\mathbf{x}_i^* \in \mathrm{argmin}_{\mathbf{y} \in \mathbb{E}_i} \left\{ \tilde{f}_i(\mathbf{y}) + g_i(\mathbf{y}) \right\},$$

trong đó

$$\tilde{f}_i(\mathbf{y}) \equiv f\left(\mathbf{x}^* + \mathcal{U}_i\left(\mathbf{y} - \mathbf{x}_i^*\right)\right) = f\left(\mathbf{x}_1^*, \ldots, \mathbf{x}_{i-1}^*, \mathbf{y}, \mathbf{x}_{i+1}^*, \ldots, \mathbf{x}_p^*\right).$$

Do đó, theo Định lý 1.19, $-\nabla \tilde{f}_i(\mathbf{x}_i^*) \in \partial g_i(\mathbf{x}^*)$. Vì $\nabla \tilde{f}_i(\mathbf{x}_i^*) = \nabla_i f(\mathbf{x}^*)$, khi đó ta có với mọi i, $-\nabla_i f(\mathbf{x}^*) \in \partial g_i(\mathbf{x}^*)$. Do đó, áp dụng Định lý 1.20, chúng ta có $-\nabla f(\mathbf{x}^*) \in \partial g(\mathbf{x}^*)$, nghĩa là $\mathbf{x}^*$ là một điểm dừng của bài toán (1.6). □

Ta nhắc lại rằng Định lý 1.16 đã chỉ ra dưới các giả thiết thích hợp rằng các điểm giới hạn của dãy được tạo ra bởi phương pháp tối thiểu luân phiên là các điểm cực tiểu từng tọa độ. Kết hợp kết quả này với Bổ đề 1.22, chúng ta có được hệ quả sau đây.

Hệ quả 1.23. Giả sử rằng Giả thiết 1.21 được thỏa mãn, và giả sử thêm rằng $F = f + g$ thỏa mãn các điều kiện sau:

- Đối với mỗi $\overline{\mathbf{x}} \in \mathrm{dom}(F)$ và $i \in \{1, 2, \ldots, p\}$, $\min_{\mathbf{y} \in \mathbb{E}_i} F\left(\overline{\mathbf{x}} + \mathcal{U}_i(\mathbf{y} - \overline{\mathbf{x}}_i)\right.$ có nghiệm duy nhất;

- Các tập mức của F bị chặn, với mọi $\alpha \in \mathbb{R}$, tập $\{\mathbf{x} \in \mathbb{E} : F(\mathbf{x}) \leq \alpha\}$ bị chặn.

Gọi $\left\{\mathbf{x}^k\right\}_{k \geq 0}$ là dãy được tạo ra bởi phương pháp tối thiểu luân phiên để giải (1.6). Khi đó $\left\{\mathbf{x}^k\right\}_{k \geq 0}$ bị chặn, và bất kỳ điểm giới hạn nào của dãy này là một điểm dừng của bài toán (1.6).

Kết quả hội tụ trong Hệ quả đó yêu cầu một giả thiết khá mạnh về tính duy nhất của nghiệm tối ưu cho mỗi bài toán con trong mỗi bước của phương pháp tối thiểu luân phiên. Khi hàm mục tiêu là hàm lồi, giả thiết này có thể được bỏ qua, mà vẫn đảm bảo nghiệm tìm được theo thuật toán là nghiệm tối ưu, như Định lý dưới đây sẽ chỉ ra.

Định lý 1.24. Giả sử rằng Giả thiết 1.21 được thỏa mãn và thêm vào đó:

- f là hàm lồi;

- f liên tục khả vi trên $\mathrm{int}(\mathrm{dom}(f))$;

- hàm $F = f + g$ thỏa mãn rằng các tập mức của F bị chặn, nghĩa là đối với bất kỳ $\alpha \in \mathbb{R}$, $\mathrm{Lev}(F, \alpha) = \{\mathbf{x} \in \mathbb{E} : F(\mathbf{x}) \leq \alpha\}$ bị chặn.

Khi đó dãy được tạo ra bởi phương pháp tối thiểu luân phiên để giải bài toán (1.6) bị chặn, và bất kỳ điểm giới hạn nào của dãy này là một nghiệm tối ưu của bài toán.

Chứng minh. Gọi $\left\{\mathbf{x}^k\right\}_{k \geq 0}$ sinh ra bởi phương pháp tối thiểu luân phiên, và

dãy $\left\{\mathbf{x}^{k,i}\right\}_{k\geq 0}\,(i=0,1,\ldots,p)$ là các dãy trung gian được định nghĩa trong (1.2). Ta chứng minh ra rằng dãy $\left\{\mathbf{x}^{k}\right\}_{k\geq 0}$ bị chặn. Thật vậy, theo cách xây dựng lặp của phương pháp, dãy các giá trị hàm là không tăng, và do đó $\left\{\mathbf{x}^{k}\right\}_{k\geq 0}\subseteq \mathrm{Lev}\left(F,F\left(\mathbf{x}^{0}\right)\right)$. Vì $\mathrm{Lev}\left(F,F\left(\mathbf{x}^{0}\right)\right)$ bị chặn theo giả thuyết, ta suy ra rằng $\left\{\mathbf{x}^{k}\right\}_{k\geq 0}$ bị chặn.

Giả sử $\overline{\mathbf{x}}\in \mathrm{dom}(g)$ là một điểm giới hạn của dãy $\left\{\mathbf{x}^{k}\right\}_{k\geq 0}$. Chúng ta sẽ chứng minh rằng $\overline{\mathbf{x}}$ là nghiệm tối ưu của bài toán (1.6). Vì $\overline{\mathbf{x}}$ là điểm giới hạn của dãy, tồn tại một dãy con $\left\{\mathbf{x}^{k_j}\right\}_{j\geq 0}$ sao cho $\mathbf{x}^{k_j}\to\overline{\mathbf{x}}$. Bằng cách chọn dãy con thích hợp, các dãy $\left\{\mathbf{x}^{k_j,i}\right\}_{j\geq 0}\,(i=1,2,\ldots,p)$ cũng hội tụ và $\mathbf{x}^{k_j,i}\to\overline{\mathbf{x}}^{i}\in\mathrm{dom}(g)$ khi $j\to\infty$ với mọi $i\in\{0,1,2,\ldots,p\}$. Khi đó, ba tính chất sau đây được thỏa mãn:

[P1] $\overline{\mathbf{x}}=\overline{\mathbf{x}}^{0}$.

[P2] Với mọi $i,\overline{\mathbf{x}}^{i}$ chỉ khác $\overline{\mathbf{x}}^{i-1}$ ở khối thứ i (nếu có khác).

[P3] $F(\overline{\mathbf{x}})=F\left(\overline{\mathbf{x}}^{i}\right)$ với mọi $i\in\{0,1,2,\ldots,p\}$ (dễ dàng chứng minh bằng cách lấy giới hạn $j\to\infty$ trong bất đẳng thức

$$F\left(\mathbf{x}^{k_j}\right)\geq F\left(\mathbf{x}^{k_j,i}\right)\geq F\left(\mathbf{x}^{k_j+1}\right),$$

và sử dụng tính liên tục của F trên miền xác định).

Theo cách xây dựng thuật toán, với mọi $j\geq 0$ và $i\in\{1,2,\ldots,p\}$,

$$\mathbf{x}_i^{k_j,i}\in \mathrm{argmin}_{\mathbf{x}_i\in\mathbb{E}_i}\,F\left(\mathbf{x}_1^{k_j+1},\ldots,\mathbf{x}_{i-1}^{k_j+1},\mathbf{x}_i,\mathbf{x}_{i+1}^{k_j},\ldots,\mathbf{x}_p^{k_j}\right).$$

Do đó $\mathbf{x}_i^{k_j,i}$ là một điểm dừng của bài toán tối ưu nên dựa trên Định lý 1.19,

$$-\nabla_i f\left(\mathbf{x}^{k_j,i}\right)\in\partial g_i\left(\mathbf{x}_i^{k_j,i}\right).$$

Từ điều này, với mọi $\mathbf{x} \in \mathbb{E}$, theo định nghĩa dưới vi phân ta có:

$$\left\langle -\nabla_i f\left(\mathbf{x}^{k_j,i}\right), \mathbf{x} - \mathbf{x}^{k_j,i} \right\rangle + g(\mathbf{x}^{k_j,i}) \leq g(\mathbf{x}).$$

Lấy giới hạn $j \to \infty$ và sử dụng tính liên tục của ∇f và tính đóng (tương đương nửa liên tục dưới như trong Định lý 1.10) của g, chúng ta thu được

$$\left\langle -\nabla_i f\left(\overline{\mathbf{x}}^i\right), \mathbf{x} - \overline{\mathbf{x}}^i \right\rangle + g(\overline{\mathbf{x}}^i) \leq g(\mathbf{x}),$$

hay ta có thể viết gọn lại

$$-\nabla_i f\left(\overline{\mathbf{x}}^i\right) \in \partial g_i\left(\overline{\mathbf{x}}_i^i\right). \tag{1.8}$$

Lưu ý rằng với mọi $\mathbf{x}_{i+1} \in \operatorname{dom}(g_{i+1})$,

$$F\left(\mathbf{x}^{k_j,i+1}\right) \leq F\left(\mathbf{x}_1^{k_j+1}, \ldots, \mathbf{x}_i^{k_j+1}, \mathbf{x}_{i+1}, \mathbf{x}_{i+2}^{k_j}, \ldots, \mathbf{x}_p^{k_j}\right).$$

Lấy giới hạn $j \to \infty$ và sử dụng [P3], chúng ta suy ra rằng với mọi $\mathbf{x}_{i+1} \in \operatorname{dom}(g_{i+1})$,

$$F\left(\overline{\mathbf{x}}^i\right) = F\left(\overline{\mathbf{x}}^{i+1}\right) \leq F\left(\overline{\mathbf{x}}_1^i, \ldots, \overline{\mathbf{x}}_i^i, \mathbf{x}_{i+1}, \overline{\mathbf{x}}_{i+2}^i, \ldots, \overline{\mathbf{x}}_p^i\right),$$

sử dụng Định lý 1.19, ta có với mọi $i \in \{0, 1, \ldots, p-1\}$,

$$-\nabla_{i+1} f\left(\overline{\mathbf{x}}^i\right) \in \partial g_{i+1}\left(\overline{\mathbf{x}}_{i+1}^i\right). \tag{1.9}$$

Ta cần chứng minh điều kiện sau đúng với mọi $i \in \{2, 3, \ldots, p\}$, $l \in \{1, 2, \ldots, p-1\}$ sao cho $l < i$:

$$-\nabla_i f\left(\overline{\mathbf{x}}^l\right) \in \partial g_i\left(\overline{\mathbf{x}}_i^l\right) \Rightarrow -\nabla_i f\left(\overline{\mathbf{x}}^{l-1}\right) \in \partial g_i\left(\overline{\mathbf{x}}_i^{l-1}\right). \tag{1.10}$$

Để chứng minh điều kiện trên, giả sử rằng $-\nabla_i f\left(\overline{\mathbf{x}}^l\right) \in \partial g_i\left(\overline{\mathbf{x}}_i^l\right)$ và cho

$\boldsymbol{\eta} \in \mathbb{E}_i$. Ta có

$$\begin{aligned}
&\left\langle \nabla f\left(\overline{\mathbf{x}}^l\right), \overline{\mathbf{x}}^{l-1} + \mathcal{U}_i(\boldsymbol{\eta}) - \overline{\mathbf{x}}^l \right\rangle \\
&\overset{(*)}{=} \left\langle \nabla_l f\left(\overline{\mathbf{x}}^l\right), \overline{\mathbf{x}}_l^{l-1} - \overline{\mathbf{x}}_l^l \right\rangle + \left\langle \nabla_i f\left(\overline{\mathbf{x}}^l\right), \boldsymbol{\eta} \right\rangle \\
&\overset{(**)}{\geq} g_l\left(\overline{\mathbf{x}}_l^l\right) - g_l\left(\overline{\mathbf{x}}_l^{l-1}\right) + \left\langle \nabla_i f\left(\overline{\mathbf{x}}^l\right), \boldsymbol{\eta} \right\rangle \\
&\overset{(**)}{=} g_l\left(\overline{\mathbf{x}}_l^l\right) - g_l\left(\overline{\mathbf{x}}_l^{l-1}\right) + \left\langle \nabla_i f\left(\overline{\mathbf{x}}^l\right), \left(\overline{\mathbf{x}}_i^{l-1} + \boldsymbol{\eta}\right) - \overline{\mathbf{x}}_i^l \right\rangle \\
&\overset{(****)}{\geq} g_l\left(\overline{\mathbf{x}}_l^l\right) - g_l\left(\overline{\mathbf{x}}_l^{l-1}\right) + g_i\left(\overline{\mathbf{x}}_i^l\right) - g_i\left(\overline{\mathbf{x}}_i^{l-1} + \boldsymbol{\eta}\right) \\
&= g\left(\overline{\mathbf{x}}^l\right) - g\left(\overline{\mathbf{x}}^{l-1} + \mathcal{U}_i(\boldsymbol{\eta})\right),
\end{aligned} \tag{1.11}$$

trong đó $(*)$ được suy ra từ [P2], $(**)$ là hệ quả từ (1.8) với $i = l$, $(***)$ được suy ra từ việc với mọi $l < i$ thì $\overline{\mathbf{x}}_i^l = \overline{\mathbf{x}}_i^{l-1}$, và $(****)$ là do giả thiết $-\nabla_i f\left(\overline{\mathbf{x}}^l\right) \in \partial g_i\left(\overline{\mathbf{x}}_i^l\right)$. Sử dụng bất đẳng thức (1.11) chúng ta có

$$\begin{aligned}
&F\left(\overline{\mathbf{x}}^{l-1} + \mathcal{U}_i(\boldsymbol{\eta})\right) \\
&= f\left(\overline{\mathbf{x}}^{l-1} + \mathcal{U}_i(\boldsymbol{\eta})\right) + g\left(\overline{\mathbf{x}}^{l-1} + \mathcal{U}_i(\boldsymbol{\eta})\right) \\
&\geq f\left(\overline{\mathbf{x}}^l\right) + \left\langle \nabla f\left(\overline{\mathbf{x}}^l\right), \overline{\mathbf{x}}^{l-1} + \mathcal{U}_i(\boldsymbol{\eta}) - \overline{\mathbf{x}}^l \right\rangle + g\left(\overline{\mathbf{x}}^{l-1} + \mathcal{U}_i(\boldsymbol{\eta})\right) \\
&\geq F\left(\overline{\mathbf{x}}^l\right) \\
&\overset{[P3]}{=} F\left(\overline{\mathbf{x}}^{l-1}\right).
\end{aligned}$$

Do đó, chúng ta thu được

$$\overline{\mathbf{x}}_i^{l-1} \in \operatorname{argmin}_{\mathbf{x}_i \in \mathbb{E}_i} F\left(\overline{\mathbf{x}}_1^{l-1}, \ldots, \overline{\mathbf{x}}_{i-1}^{l-1}, \mathbf{x}_i, \overline{\mathbf{x}}_{i+1}^{l-1}, \ldots, \overline{\mathbf{x}}_p^{l-1}\right),$$

suy ra $-\nabla_i f\left(\overline{\mathbf{x}}^{l-1}\right) \in \partial g_i\left(\overline{\mathbf{x}}_i^{l-1}\right)$, ta đã chứng minh được (1.10). Bây giờ chúng ta cần chứng minh rằng $\overline{\mathbf{x}} = \overline{\mathbf{x}}^0$ là một nghiệm tối ưu của bài toán (1.6). Để chứng minh được điều này ta sẽ chỉ ra rằng mọi $m \in \{1, 2, \ldots, p\}$ thì

$$-\nabla_m f(\overline{\mathbf{x}}) \in \partial g_m\left(\overline{\mathbf{x}}_m\right). \tag{1.12}$$

Theo Định lý 1.20 ta có $\overline{\mathbf{x}}$ là một điểm dừng và sử dụng Định lý 1.19 và tính lồi của f, chúng ta có thể suy ra rằng $\overline{\mathbf{x}}$ là một nghiệm tối ưu của bài toán (1.6).

Thật vậy, với $m = 1$, tính chất 1.12 được suy ra bằng cách thay $i = 0$ vào (1.9) và sử dụng $\overline{\mathbf{x}} = \overline{\mathbf{x}}^0$ (tính chất [P1]). Giả sử $m > 1$, khi đó từ (1.9) chúng ta có $-\nabla_m f\left(\overline{\mathbf{x}}^{m-1}\right) \in \partial g_m\left(\overline{\mathbf{x}}_m^{m-1}\right)$. Sử dụng (1.10) thu được

$$
\begin{array}{c}
-\nabla_m f\left(\overline{\mathbf{x}}^{m-1}\right) \in \partial g_m\left(\overline{\mathbf{x}}_m^{m-1}\right) \\
\Downarrow \\
-\nabla_m f\left(\overline{\mathbf{x}}^{m-2}\right) \in \partial g_m\left(\overline{\mathbf{x}}_m^{m-2}\right) \\
\Downarrow \\
\vdots \\
\Downarrow \\
-\nabla_m f\left(\overline{\mathbf{x}}^{0}\right) \in \partial g_m\left(\overline{\mathbf{x}}_m^{0}\right),
\end{array}
$$

và do đó, vì $\overline{\mathbf{x}} = \overline{\mathbf{x}}^0$ (tính chất $[\mathrm{P1}]$), chúng ta kết luận rằng $-\nabla_m f(\overline{\mathbf{x}}) \in \partial g_m\left(\overline{\mathbf{x}}_m\right)$ cho mọi m, suy ra $\overline{\mathbf{x}}$ là một nghiệm tối ưu của bài toán (1.6). □

Chương 2

Ứng dụng của phương pháp tối thiểu luân phiên trong bài toán phân cụm

Trong chương này, chúng tôi tập trung vào việc áp dụng phương pháp tối thiểu luân phiên trong các bài toán phân cụm K-Trung Bình. Nội dung chính của chương này gồm có:

2.1 Trình bày vấn đề phân cụm dữ liệu và mục tiêu của bài toán K-Trung Bình và thuật toán Lloyd, một phương pháp phổ biến để giải quyết vấn đề này.

2.2 Chúng tôi trình bày ý tưởng áp dùng dụng thuật toán tối thiểu luân phiên cho bài toán K-Trung Bình, đưa ra các cải tiến và so sánh với thuật toán Lloyd.

2.3 Tiến hành thí nghiệm các thuật toán trên các dữ liệu thực tế.

Nội dung tìm hiểu trong chương này tham khảo chính bài báo của Feiping Nie và các đồng nghiệp [3].

2.1 Giới thiệu bài toán

Cho tập dữ liệu $\mathbf{X} = \{x_1, x_2, \cdots, x_n\} \in \mathbb{R}^{d \times n}$ bao gồm n điểm dữ liệu trong không gian $\mathbb{R}^d$, trong đó x_i là cột thứ i của $\mathbf{X}$, đại diện cho điểm thứ i. Bài toán phân cụm k-means là tìm c cụm $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_c\}$ thỏa mãn $\mathbf{X} = \bigcup_{j=1}^{c} \mathcal{C}_j$ và $\bigcap_{j=1}^{c} \mathcal{C}_j = \emptyset$ sao cho tổng bình phương sai số (SSE) của mỗi điểm x_i với trung bình của cụm tương ứng m_j được tối thiểu

$$\min_{\mathbf{M}} \text{SSE} = \min_{\mathcal{C}} \sum_{j=1}^{c} \sum_{x_i \in \mathcal{C}_j} \|x_i - m_j\|_2^2 \tag{2.1}$$

trong đó, $\mathcal{C}_j$ là tập thứ j của $\mathcal{C}$, m_j là cột thứ j của ma trận $\mathbf{M} \in \mathbb{R}^{d \times c}$, đại diện cho trung bình (tâm) của tập $\mathcal{C}_j$ với $j = 1, 2, \cdots, c$.

Bài toán trong (2.1) là một bài toán phân cụm với tổng bình phương tối thiểu, và được chỉ ra là một bài toán NP-khó do tính chất không lồi của nó. Với những khó khăn này, ta có thể xem xét các phương pháp xấp xỉ trong thời gian đa thức (cung cấp đảm bảo về chất lượng) hoặc phương pháp heuristic (không đảm bảo chất lượng). Một trong những phương pháp heuristic phổ biến nhất cho bài toán k-means là thuật toán Lloyd, thường được gọi là thuật toán k-means(xem trong Thuật toán 2).

Trong thuật toán 2, ban đầu, thuật toán khởi tạo ngẫu nhiên c tâm cụm. Sau đó, thuật toán thực hiện hai bước chính lặp đi lặp lại: bước gán nhãn và bước cập nhật tâm. Trong bước gán nhãn, mỗi điểm dữ liệu được gán vào cụm có tâm gần nhất như dòng 5. Tiếp theo, bước cập nhật tâm tính lại tâm của mỗi cụm bằng trung bình các điểm hiện có trong cụm đó trong dòng số 9. Quá trình này lặp lại cho đến khi các tâm cụm không thay đổi, đảm bảo các cụm

được phân chia một cách tối ưu.

Thuật toán 2 Lloyd cho bài toán K-Trung Bình

Require: Tập dữ liệu $X = \{x_1, x_2, \ldots, x_n\}$, số cụm c

Ensure: Tập các tâm cụm $M = \{m_1, m_2, \ldots, m_c\}$ và nhãn cụm cho mỗi điểm dữ liệu

1: Khởi tạo ngẫu nhiên c tâm cụm ban đầu $M = \{m_1, m_2, \ldots, m_c\}$ từ tập dữ liệu X
2: **repeat**
3: **Bước gán nhãn:**
4: **for** mỗi điểm dữ liệu $x_i \in X$ **do**
5: Gán x_i vào cụm có tâm gần nhất:

$$\mathcal{C}_j \leftarrow \{x_i : \|x_i - m_j\|^2 \leq \|x_i - m_l\|^2, \forall l = 1, \ldots, c\}$$

6: **end for**
7: **Bước cập nhật tâm:**
8: **for** mỗi cụm $j = 1$ đến c **do**
9: Cập nhật tâm cụm m_j bằng trung bình các điểm trong cụm:

$$m_j \leftarrow \frac{1}{|S_j|} \sum_{x_i \in \mathcal{C}_j} x_i$$

10: **end for**
11: **until** Các $\mathcal{C}_j$ không đổi so với vòng lặp trước

Độ phức tạp của thuật toán Lloyd bao gồm hai bước chính: gán nhãn và cập nhật tậm. Bước gán nhãn dữ liệu ứng với tâm gần nhất có độ phức tạp $O(ncd)$, do phải tính khoảng cách giữa mỗi điểm và tất cả các tâm. Bước cập nhật tâm, tính trung bình các điểm trong cụm, có độ phức tạp $O(nd)$. Tổng độ phức tạp của thuật toán, với t vòng lặp là $O(tncd)$.

Mặc dù được sử dụng rộng rãi, phương pháp này vẫn tồn tại một số hạn chế đáng kể. Cụ thể, hiệu quả của nó phụ thuộc vào việc chọn các điểm khởi tạo, thuật toán dễ bị mắc kẹt tại các cực tiểu cục bộ không tốt. Hơn nữa, mỗi lần lặp lại yêu cầu tính toán khoảng cách giữa tất cả các điểm dữ liệu với các

trung tâm mới, gây tốn kém về mặt chi phí tính toán.

2.2 Thuật toán tối thiểu luân phiên cho K-Trung Bình

2.2.1. Ý tưởng thuật toán

Để dễ dàng, chúng ta phát biểu lại bài toán phân cụm k-means sử dụng một ma trận để biểu diễn sự phân loại điểm vào các cụm:

$$\begin{aligned} &\min_{\mathbf{F}\in\text{Ind},\mathbf{M}} \sum_{j=1}^{c}\sum_{i=1}^{n} \|x_i - m_j\|_2^2 f_{ij} \\ =&\min_{\mathbf{F}\in\text{Ind},\mathbf{M}} \left\|\mathbf{X} - \mathbf{M}\mathbf{F}^{\mathrm{T}}\right\|_F^2 \end{aligned} \tag{2.2}$$

trong đó, $\mathbf{M} \in \mathbb{R}^{d\times c}$ là các tâm của các cụm, $\mathbf{F} \in \mathbb{R}^{n\times c}$ là ma trận gán nhãn, trong đó x_i được phân vào cụm thứ k thì $f_{ik} = 1, f_{ij} = 0, \forall j \neq k$ với $f_{ij} \in \{0, 1\}$ là phần tử (i, j) của $\mathbf{F}$ và Ind tập tất cả các ma trận $\mathbf{F}$ thỏa mãn tính chất trên.

Khi cố định $\mathbf{F}$ và cực tiểu hàm mục tiêu của bài toán (2.2) theo $\mathbf{M}$, ta thu được bài toán sau:

$$\min_{\mathbf{M}} \mathrm{Tr}\left(\left(\mathbf{X} - \mathbf{M}\mathbf{F}^{\mathrm{T}}\right)\left(\mathbf{X} - \mathbf{M}\mathbf{F}^{\mathrm{T}}\right)^{\mathrm{T}}\right) \tag{2.3}$$

trong đó, $\mathrm{Tr}(\cdot)$ là vết của một ma trận. Vì chỉ có một phần tử khác không bằng không trong mỗi hàng của ma trận $\mathbf{F}$, nên ma trận $\mathbf{F}^{\mathrm{T}}\mathbf{F}$ là một ma trận chéo với phần tử (l, l) bằng $f_l^{\mathrm{T}} f_l$, trong đó f_l là cột thứ l của ma trận $\mathbf{F}$. Vì hàm mục tiêu là hàm lồi, lấy đạo hàm theo $\mathbf{M}$ và cho nó bằng 0, chúng ta thu được nghiệm của bài toán (2.3) được cho bởi:

$$\mathbf{M} = \mathbf{X}\mathbf{F}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\right)^{-1} \tag{2.4}$$

trong đó $\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\right)^{-1}$ là ma trận đường chéo với:

$$\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\right)^{-1}_{ll} = \begin{cases} \frac{1}{f_l^{\mathrm{T}} f_l} & f_l^{\mathrm{T}} f_l \neq 0 \\ 0 & f_l^{\mathrm{T}} f_l = 0 \end{cases}$$

Thay (2.4) vào (2.3), ta có

$$\begin{aligned}
&\left(\mathbf{X} - \mathbf{M}\mathbf{F}^{\mathrm{T}}\right)\left(\mathbf{X} - \mathbf{M}\mathbf{F}^{\mathrm{T}}\right)^{\mathrm{T}} \\
&= \mathbf{X}\mathbf{X}^{\mathrm{T}} - \mathbf{X}\left(\mathbf{M}\mathbf{F}^{\mathrm{T}}\right)^{\mathrm{T}} - \left(\mathbf{M}\mathbf{F}^{\mathrm{T}}\right)\mathbf{X}^{\mathrm{T}} + \left(\mathbf{M}\mathbf{F}^{\mathrm{T}}\right)\left(\mathbf{M}\mathbf{F}^{\mathrm{T}}\right)^{\mathrm{T}} \\
&= \mathbf{X}\mathbf{X}^{\mathrm{T}} - \mathbf{X}\mathbf{F}\mathbf{M}^{\mathrm{T}} - \mathbf{M}\mathbf{F}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}} + \mathbf{M}\mathbf{F}^{\mathrm{T}}\mathbf{F}\mathbf{M}\mathrm{T} \\
&= \mathbf{X}\mathbf{X}^{\mathrm{T}} - \mathbf{X}\mathbf{F}\left(\mathbf{X}\mathbf{F}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\right)^{-1}\right)^{\mathrm{T}} - \left(\mathbf{X}\mathbf{F}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\right)^{-1}\right)\mathbf{F}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}} \\
&+ \left(\mathbf{X}\mathbf{F}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\right)^{-1}\right)\mathbf{F}^{\mathrm{T}}\mathbf{F}\left(\mathbf{X}\mathbf{F}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\right)^{-1}\right)^{\mathrm{T}} \\
&= \mathbf{X}\mathbf{X}^{\mathrm{T}} - \mathbf{X}\mathbf{F}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\right)^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}} - \mathbf{X}\mathbf{F}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\right)^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}} \\
&+ \mathbf{X}\mathbf{F}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\right)^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{F}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\right)^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}} \\
&= \mathbf{X}\mathbf{X}^{\mathrm{T}} - \mathbf{X}\mathbf{F}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\right)^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}
\end{aligned}$$

Vì $\mathrm{Tr}(\mathbf{X}\mathbf{X}^{T})$ là hằng số nên ta có bài toán mới tương đương với bài toán (2.2) như sau:

$$\max_{\mathbf{F}\in\ \mathrm{Ind}} \mathrm{Tr}\left(\mathbf{X}\mathbf{F}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\right)^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\right). \tag{2.5}$$

Hay bài toán tối ưu (2.5) có thể được viết lại thành:

$$\max_{\mathbf{F}\in\mathrm{Ind}} \mathrm{obj}(\mathbf{F}) = \sum_{l=1}^{c} \frac{f_l^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_l}{f_l^{\mathrm{T}} f_l}, \tag{2.6}$$

trong đó ta luôn coi $\frac{f_l^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_l}{f_l^{\mathrm{T}} f_l} = 0$ với mọi $f_l^{\mathrm{T}} f_l = 0$ hay $f_l = \mathbf{0}$. Ta áp dụng phương pháp tối thiểu luân phiên để cập nhật ma trận $\mathbf{F}$, và ở mỗi vòng lặp, bài toán tối ưu để giải $\mathbf{F}$ có thể được phân ra thành n bài toán con đơn giản cho mỗi i ($i = 1, 2, \cdots, n$).

Khi cập nhật hàng i của ma trận $\mathbf{F}$, có c trường hợp $\left\{\mathbf{F}^{(1)}, \mathbf{F}^{(2)}, \cdots, \mathbf{F}^{(c)}\right\}$, trong đó, $\mathbf{F}^{(l)}$ và $\mathbf{F}^{(j)}$ $(l, j = 1, 2, \cdots, c$ và $l \neq j)$ có các phần tử giống nhau ngoại trừ hàng thứ i. Đối với hàng thứ i của ma trận $\mathbf{F}^{(l)}$, phần tử thứ l là 1, các phần tử còn lại là 0. Tương tự, đối với hàng thứ i của ma trận $\mathbf{F}^{(j)}$, phần tử thứ j là 1, các phần tử còn lại là 0, như được thể hiện dưới đây.

$$F^{(j)} = \begin{bmatrix} & & & f_{ij} & & & \\ & \vdots & & \downarrow & \vdots & & \\ \cdots & 0 & \cdots & 1 & 0 & \cdots & 0 \\ \cdots & \vdots & \cdots & & \vdots & \cdots & 0 \end{bmatrix}. \tag{2.7}$$

Để cập nhật hàng thứ i của ma trận $\mathbf{F}$, ta giải bài toán tối ưu

$$\max_{k \in \{1,2,\cdots,c\}} \operatorname{obj}\left(\mathbf{F}^{(k)}\right) = \sum_{l=1}^{c} \frac{\left(f_l^{(k)}\right)^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \mathbf{X} f_l^{(k)}}{\left(f_l^{(k)}\right)^{\mathrm{T}} f_l^{(k)}}. \tag{2.8}$$

trong đó, $f_l^{(k)}$ là cột thứ l của ma trận $\mathbf{F}^{(k)}$. Khi đó ta có hàng thứ i mới như sau

$$f_{iq} = \begin{cases} 1 & \text{nếu } q = \arg\max_k \operatorname{obj}\left(\mathbf{F}^{(k)}\right), \\ 0 & \text{còn lại.} \end{cases} \tag{2.9}$$

trong đó, $\operatorname{obj}\left(\mathbf{F}^{(k)}\right)$ được nêu trong bài toán (2.8), q là vị trí được cập nhật của phần tử khác không trong hàng thứ i của ma trận $\mathbf{F}$.

Trước mỗi lần cập nhật hàng i, ta lưu lại vị trí của phần tử 1 trong hàng thứ i là $p(p = 1, 2, \ldots, c)$, trong đó f_l là cột thứ l của $\mathbf{F}$. Sau bước lặp thứ i của $\mathbf{F}$, $f_l(l = 1, 2, \cdots, c)$ được cập nhật lại như sau: nếu $q = p$, điều này có nghĩa là hàng thứ i của $\mathbf{F}$ không thay đổi, và do đó $f_l(l = 1, 2, \cdots, c)$ cũng không thay đổi; nếu $q \neq p$, chúng ta chỉ cần cập nhật $f_l(l = p, q)$ bằng cách hoán

đổi vị trí thứ i của nhau. Thuật toán chi tiết áp dụng trực tiếp phương pháp tối thiểu luân phiên để giải quyết bài toán (2.6) được tóm tắt trong Thuật toán 3.

Chú ý: Thuật toán 3 có một số nhược điểm khi số lượng tính toán không được tối ưu. Cụ thể, thuật toán cần thực hiện $O\left(n^2dct\right)$ phép nhân để tính toán $obj\left(\mathbf{F}^{(k)}\right)$ trong bước cập nhật (2.8), trong đó t là số lần lặp của phương pháp, c là số lượng cụm, d là chiều dữ liệu, và n là số lượng điểm dữ liệu. Khi đó, Thuật toán 3 yêu cầu số lượng tính toán nhiều hơn so với thuật toán Lloyd.

Thuật toán 3 Thuật toán tối thiểu luân phiên cho bài toán K-Trung Bình (ALS-kmeans)

1: **Đầu vào:** ma trận dữ liệu $\mathbf{X} \in \mathbb{R}^{d\times n}$, số lượng cụm c.
2: **Khởi tạo** c tâm cụm bằng một chiến lược khởi tạo và nhận ma trận $\mathbf{F} \in \mathbb{R}^{n\times c}$ ban đầu.
3: **repeat**
4: **for** $i = 1$ **to** n **do**
5: Tính $\mathrm{obj}(\mathbf{F}^{(k)})$ theo (2.8);
6: Cập nhật hàng thứ i của ma trận $\mathbf{F}$ theo (2.9);
7: **end for**
8: **until** hội tụ
9: **Đầu ra:** ma trận $\mathbf{F} \in \mathbb{R}^{n\times c}$.

2.2.2. Thuật toán tối thiểu luân phiên cải tiến

Phần này, chúng tôi sẽ trình bày cách giải quyết vấn đề về tính toán đã nêu ở phần trước và đề xuất một phương pháp tối thiểu luân phiên cải tiến cho bài toán k-means. Các ma trận $\mathbf{F}^{(l)}$ và $\mathbf{F}^{(j)}$ (với $l, j = 1, 2, \cdots, c$ và $l \neq j$) được mô tả trong (2.7) có các phần tử giống nhau, chỉ khác nhau ở cột thứ l và cột thứ j.

Để đơn giản hóa bài toán, chúng tôi giới thiệu ma trận $\mathbf{F}^{(0)}$, được định

nghĩa tương tự như ma trận $\mathbf{F}^{(j)}$ $(j = 1, 2, \cdots, c)$. Sự khác biệt duy nhất là tất cả các phần tử trong hàng thứ i của ma trận $\mathbf{F}^{(0)}$ đều bằng 0 ở bước cập nhật hàng thứ i của ma trận $\mathbf{F}$. Vì vậy, bài toán (2.8) tương đương với các bài toán sau:

$$\begin{aligned}
&\max_{k\in\{1,\cdots,c\}} \psi(k) = \max_{k\in\{1,\cdots,c\}} \left(\text{obj}\left(\mathbf{F}^{(k)}\right) - \text{obj}\left(\mathbf{F}^{(0)}\right)\right) \\
&= \max_{k\in\{1,\cdots,c\}} \left(\sum_{l=1}^{c} \frac{\left(f_l^{(k)}\right)^{\mathrm{T}} \mathbf{X}^{\mathrm{T}}\mathbf{X} f_l^{(k)}}{\left(f_l^{(k)}\right)^{\mathrm{T}} f_l^{(k)}} - \sum_{l=1}^{c} \frac{\left(f_l^{(0)}\right)^{\mathrm{T}} \mathbf{X}^{\mathrm{T}}\mathbf{X} f_l^{(0)}}{\left(f_l^{(0)}\right)^{\mathrm{T}} f_l^{(0)}} \right) \\
&= \max_{k\in\{1,\cdots,c\}} \left(\frac{\left(f_k^{(k)}\right)^{\mathrm{T}} \mathbf{X}^{\mathrm{T}}\mathbf{X} f_k^{(k)}}{\left(f_k^{(k)}\right)^{\mathrm{T}} f_k^{(k)}} - \frac{\left(f_k^{(0)}\right)^{\mathrm{T}} \mathbf{X}^{\mathrm{T}}\mathbf{X} f_k^{(0)}}{\left(f_k^{(0)}\right)^{\mathrm{T}} f_k^{(0)}} \right),
\end{aligned} \tag{2.10}$$

Xét bước cập nhật hàng thứ i của ma trận $\mathbf{F}$, khi tìm giá trị cực đại của $\psi(k)$ trong bài toán (2.10), chúng ta lưu lại vị trí của phần tử 1 trong hàng thứ i là $p(p = 1, 2, \ldots, c)$. $f_k^{(k)}$ là cột thứ k của $\mathbf{F}^{(k)}$, $f_k^{(0)}$ là cột thứ k của $\mathbf{F}^{(0)}$ và f_k là cột thứ k của $\mathbf{F}$ hiện tại. Khi đó, chúng ta tính $\psi(k)$ trong bài toán (2.10) với hai trường hợp:

Trường hợp 1: Khi $k = p$, $f_k^{(k)} = f_k$, tức phần tử thứ i của f_k và $f_k^{(k)}$ là 1. Chúng ta đặt $\delta_k = f_k - f_k^{(0)}$, có nghĩa là phần tử thứ i của δ_k là 1, các phần tử còn lại đều là 0. Do đó, chúng ta tính $\mathbf{X}f_k = \mathbf{X}(f_k^{(0)} + \delta_k) = \mathbf{X}f_k^{(0)} + x_i$, nghĩa là $\mathbf{X}f_k^{(0)} = \mathbf{X}f_k - x_i$, trong đó x_i là cột thứ i của $\mathbf{X}$. Mặt khác, chúng ta có $\left(f_k^{(0)}\right)^{\mathrm{T}} f_k^{(0)} = \left(f_k - \delta_k\right)^{\mathrm{T}} \left(f_k - \delta_k\right) = f_k^{\mathrm{T}} f_k - 2f_k^{\mathrm{T}}\delta_k + \delta_k^{\mathrm{T}}\delta_k = f_k^{\mathrm{T}} f_k - 1$. Tiếp đến thay thế $f_k^{(k)} = f_k$, $\mathbf{X}f_k^{(0)} = \mathbf{X}f_k - x_i$, và $\left(f_k^{(0)}\right)^{\mathrm{T}} f_k^{(0)} = f_k^{\mathrm{T}} f_k - 1$ vào bài toán (2.10) sẽ cho kết quả trong (2.11).

Trường hợp 2: Khi $k \neq p$, $f_k^{(0)} = f_k$, tức là phần tử thứ i của cả f_k và

$f_k^{(0)}$ là 0. Chúng ta đặt $\delta_k = f_k^{(k)} - f_k$, có nghĩa là phần tử thứ i của δ_k là 1, các phần tử còn lại đều là 0. Do đó, chúng ta tính $\mathbf{X}f_k^{(k)} = \mathbf{X}(f_k + \delta_k) = \mathbf{X}f_k + x_i$. Mặt khác, chúng ta có $\left(f_k^{(k)}\right)^{\mathrm{T}} f_k^{(k)} = (f_k + \delta_k)^{\mathrm{T}} (f_k + \delta_k) = f_k^{\mathrm{T}} f_k + 2f_k^{\mathrm{T}}\delta_k + \delta_k^{\mathrm{T}}\delta_k = f_k^{\mathrm{T}} f_k + 1$. Khi đó, thay thế $f_k^{(0)} = f_k$, $\mathbf{X}f_k^{(k)} = \mathbf{X}f_k + x_i$, và $\left(f_k^{(k)}\right)^{\mathrm{T}} f_k^{(k)} = f_k^{\mathrm{T}} f_k + 1$ vào bài toán (2.10) sẽ cho kết quả dưới đây:

$$\psi(k) = \begin{cases} \frac{f_k^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_k}{f_k^{\mathrm{T}} f_k} - \frac{f_k^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_k - 2x_i^{\mathrm{T}}\mathbf{X}f_k + x_i^{\mathrm{T}}x_i}{f_k^{\mathrm{T}} f_k - 1} & k = p, \\ \frac{f_k^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_k + 2x_i^{\mathrm{T}}\mathbf{X}f_k + x_i^{\mathrm{T}}x_i}{f_k^{\mathrm{T}} f_k + 1} - \frac{f_k^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_k}{f_k^{\mathrm{T}} f_k} & k \neq p. \end{cases} \tag{2.11}$$

Chú ý, $k = p$ ta luôn coi $\frac{f_k^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_k - 2x_i^{\mathrm{T}}\mathbf{X}f_k + x_i^{\mathrm{T}}x_i}{f_k^{\mathrm{T}} f_k - 1} = \frac{\left(f_k^{(0)}\right)^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_k^{(0)}}{\left(f_k^{(0)}\right)^{\mathrm{T}} f_k^{(0)}} = 0$ nếu $\left(f_k^{(0)}\right)^{\mathrm{T}} f_k^{(0)} = 0$. Khi đó, việc cập nhật hàng thứ i của ma trận $\mathbf{F}$ được xác định bởi:

$$f_{iq} = \begin{cases} 1 & q = \arg\max_k \psi(k), \\ 0 & \text{còn lại.} \end{cases} \tag{2.12}$$

trong đó, $\psi(k)$ được tính như trong (2.11), q là vị trí cập nhật của phần tử khác 0 trong hàng thứ i của ma trận $\mathbf{F}$.

Mặc dù chúng ta có thể thu được $\mathbf{F}$ trực tiếp từ (2.12), nhưng nó vẫn tốn rất nhiều chi phí dừ thừa. Do đó, chúng ta sẽ phân tích và đề xuất một số chiến lược giảm chi phí tính toán hơn nữa.

Đối với hàng thứ i của ma trận $\mathbf{F}$, để tính $\max_k \psi(k)$ trong (2.11), chúng ta phải tính c lần $\psi(k)(k = 1, 2, \cdots, c)$ với phức tạp tính toán là $O(ndc)$, trong đó, tính $\mathbf{X}f_k$ và $f_k^{\mathrm{T}} f_k$ với c lần cần $O(ndc)$ và $O(nc)$ tương ứng, tính cả $x_i^{\mathrm{T}}x_i$ và $f_k^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_k$ với c lần cần $O(dc)$. Để khối lượng tính toán, chúng ta sử dụng chiến lược lưu trữ các phép tính lặp lại để giảm từ $O(ndc)$ xuống $O(dc)$. Cụ

thể, đầu tiên, chúng ta tính toán và lưu trữ $a_k = \mathbf{X}f_k, b_k = f_k^{\mathrm{T}}f_k, a_k^{\mathrm{T}}a_k = f_k^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_k (k = 1, 2, \cdots, c)$ và $x_i^{\mathrm{T}}x_i (i = 1, 2, \cdots, n)$ trước một lần, và sau đó đối với $\max_k \psi(k) (k = 1, 2, \cdots, c)$, chúng ta chỉ cần tính $x_i^{\mathrm{T}}a_k$ cho c lần, mà phức tạp tính toán là $O(dc)$.

Hơn nữa, $x_i^{\mathrm{T}}x_i (i = 1, 2, \cdots, n)$ được lưu lại nên chỉ yêu cầu tính 1 lần. Còn đối với việc cập nhật các giá trị đã lưu của $\mathbf{X}f_k$ và $f_k^{\mathrm{T}}f_k (k = 1, 2, \cdots, c)$, có hai tình huống: khi $q = p$, nghĩa là hàng thứ i của $\mathbf{F}$ không thay đổi, do đó $\mathbf{X}f_k$ và $f_k^{\mathrm{T}}f_k (k = 1, 2, \cdots, c)$ không cần phải cập nhật cho hàng thứ $(i + 1)$ của $\mathbf{F}$; khi $q \neq p$, chỉ có các phần tử thứ p và q của hàng thứ i của $\mathbf{F}$ bị thay đổi, nghĩa là chỉ có f_p và f_q bị thay đổi, chúng ta chỉ cần cập nhật $\mathbf{X}f_k$ và $f_k^{\mathrm{T}}f_k (k = p, q)$: gán giá trị của $f_p^{(0)}$ và $f_q^{(q)}$ cho f_p và f_q tương ứng, nghĩa là $\mathbf{X}f_p = \mathbf{X}f_p^{(0)}, \mathbf{X}f_q = \mathbf{X}f_q^{(q)}, f_p^{\mathrm{T}}f_p = \left(f_p^{(0)}\right)^{\mathrm{T}} f_p^{(0)}$, và $f_q^{\mathrm{T}}f_q = \left(f_q^{(q)}\right)^{\mathrm{T}} f_q^{(q)}$, trong đó, $f_p^{(0)}$ là cột thứ p của $\mathbf{F}^{(0)}$, và $f_q^{(q)}$ là cột thứ q của $\mathbf{F}^{(q)}$. Để giảm bớt độ phức tạp tính toán hơn nữa, chúng ta có các bước sau để cập nhật $\mathbf{X}f_k$ và $f_k^{\mathrm{T}}f_k (k = p, q)$:

$$\begin{aligned} &\mathbf{X}f_p \leftarrow \mathbf{X}f_p - x_i; \mathbf{X}f_q \leftarrow \mathbf{X}f_q + x_i, \\ &f_p^{\mathrm{T}}f_p \leftarrow f_p^{\mathrm{T}}f_p - 1; f_q^{\mathrm{T}}f_q \leftarrow f_q^{\mathrm{T}}f_q + 1. \end{aligned} \tag{2.13}$$

Hơn nữa, việc cập nhật giá trị đã lưu $a_k^{\mathrm{T}}a_k = f_k^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_k (k = 1, 2, \cdots, c)$ phụ thuộc vào $\mathbf{X}f_k (k = 1, 2, \cdots, c)$. Cụ thể, khi $q = p$, $\mathbf{X}f_k (k = 1, 2, \cdots, c)$ ta có $f_k^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_k (k = 1, 2, \cdots, c)$ cũng không thay đổi; khi $q \neq p$, chỉ có $\mathbf{X}f_k (k = p, q)$ cần được cập nhật theo (2.13), vì vậy chúng ta chỉ cần cập nhật $a_k^{\mathrm{T}}a_k (k = p, q)$ theo (2.14):

$$\begin{aligned} f_p^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_p &\leftarrow f_p^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_p - 2x_i^{\mathrm{T}}\mathbf{X}f_p + x_i^{\mathrm{T}}x_i, \\ f_q^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_q &\leftarrow f_q^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_q + 2x_i^{\mathrm{T}}\mathbf{X}f_q + x_i^{\mathrm{T}}x_i. \end{aligned} \tag{2.14}$$

Lưu ý rằng $f_p^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_p - 2x_i^{\mathrm{T}}\mathbf{X}f_p + x_i^{\mathrm{T}}x_i$ và $f_q^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_q + 2x_i^{\mathrm{T}}\mathbf{X}f_q + x_i^{\mathrm{T}}x_i$ trong $\psi(p)$ và $\psi(q)$ đã được tính toán trước đó, chúng ta không cần tính toán lại, chỉ cần gán chúng cho $f_p^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_p$ và $f_q^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_q$ để cập nhật $a_k^{\mathrm{T}}a_k(k = p, q)$.

Thuật toán chi tiết để giải bài toán (2.6) được tóm tắt trong Thuật toán 4. Chú ý rằng thuật toán cái tiến này chỉ làm giảm chi phí toán, tuy nhiên, không làm thay cách phân cụm tìm được so với thuật toán tối thiểu luân phiên trong thuật toán 3.

Thuật toán 4 Phiên bản cải tiến thuật toán tối ưu luân phiên cho bài toán K-Trung Bình

Đầu vào: Dữ liệu đầu vào ma trận $\mathbf{X} \in \mathbb{R}^{d\times n}$, số lượng cụm c.
Khởi tạo c tâm cụm bằng một chiến lược khởi tạo và lấy $\mathbf{F} \in \mathbb{R}^{n\times c}$ ban đầu.
Tính toán và lưu trữ $\mathbf{X}f_k, f_k^{\mathrm{T}}f_k, f_k^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_k$ $(k = 1, 2, \cdots, c)$, và $x_i^{\mathrm{T}}x_i$ $(i = 1, 2, \cdots, n)$.
repeat
    for $i = 1$ to n **do**
        Tính toán $\psi(k)$ $(k = 1, 2, \cdots, c)$ theo (2.11);
        Cập nhật hàng thứ i của $\mathbf{F}$ theo (2.12);
        if $p \neq q$ **then**
            Cập nhật $\mathbf{X}f_k$ và $f_k^{\mathrm{T}}f_k$ $(k = p, q)$ theo (2.13);
            Cập nhật $f_k^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_k$ $(k = p, q)$ theo (2.14).
        end if
    end for
until Ma trận F không đổi với vòng lặp trước
Đầu ra: Ma trận nhãn $\mathbf{F} \in \mathbb{R}^{n\times c}$.

2.2.3. So sánh với thuật toán Lloyd cho bài toán K-Trung Bình

Độ phức tạp tính toán là một chỉ số quan trọng của một thuật toán. Trong phần này, chúng tôi phân tích độ phức tạp tính toán của thuật toán tối thiểu

luân phiên cải tiến.

Đối với các phép tính lưu trữ trước, việc tính toán $\mathbf{X}f_k$ và $f_k^{\mathrm{T}}f_k(k = 1, 2, \cdots, c)$ cần nd phép cộng và n phép cộng tương ứng, việc tính $x_i^{\mathrm{T}}x_i(i = 1, 2, \cdots, n)$ cần nd phép nhân, và việc tính $f_k^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_k(k = 1, 2, \cdots, c)$ cần dc phép nhân. Ở bước 6, việc tính $\psi(k)(k = 1, 2, \cdots, c)$ theo (2.11), nghĩa là tính $x_i^{\mathrm{T}}a_k$ cần dc phép nhân. Việc cập nhật hàng thứ i của ma trận $\mathbf{F}$ theo phương trình (2.12) có độ phức tạp thời gian nhỏ và có thể bỏ qua khi so sánh với các bước khác. Ở bước 9, cập nhật $\mathbf{X}f_k$ và $f_k^{\mathrm{T}}f_k$ (với $k = p, q$) theo phương trình (2.13) cần $2d + 2$ phép cộng. Ở bước 10, các biểu thức $f_p^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_p - 2x_i^{\mathrm{T}}\mathbf{X}f_p + x_i^{\mathrm{T}}x_i$ và $f_q^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}f_q + 2x_i^{\mathrm{T}}\mathbf{X}f_q + x_i^{\mathrm{T}}x_i$ đã được tính trước nên không cần phải tính lại. Thuật toán có n vòng lặp từ bước 6 đến bước 11, vì vậy mỗi vòng lặp đòi hỏi $ndc+nd+dc$ phép nhân và $nd+n+m(2d+2)$ phép cộng, với $0 \leq m \leq n$. Vì phép nhân tốn thời gian hơn phép cộng, nên độ phức tạp được tính dựa trên số phép nhân. Do đó, độ phức tạp tổng thể là $O(ndct + nd + dc)$, trong đó t là số lần lặp của thuật toán, c là số cụm, d là chiều dữ liệu, và n là số điểm dữ liệu. Kết quả là độ phức tạp tính toán của phương pháp là $O(ndct)$, tương đương với phương pháp heuristic của Lloyd.

Tiếp theo, chúng tôi chỉ ra kết quả phân cụm của thuật toán tối thiểu luân phiên tốt hơn so với thuật toán Lloyd. Định lý 2.2 chỉ ra thuật toán tối thiểu luân phiên khi giải quyết bài toán k-means không tạo ra cụm rỗng, Định lý 2.3 nói rằng thuật toán Lloyd không thể thoát khỏi cực tiểu cục bộ mà tối thiểu luân phiên đạt được và để giá trị của hàm mục tiêu tiếp tục giảm.

Bổ đề 2.1. Việc thêm một điểm dữ liệu mới không làm giảm tổng khoảng cách từ tất cả các điểm dữ liệu đến tâm.

Chứng minh. Giả sử tập D_1 có n điểm dữ liệu $x_1, x_2, \cdots, x_n$, trung bình là $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, tổng khoảng cách từ tất cả n điểm dữ liệu đến trung bình $\bar{x}$ là $\sum_{i=1}^{n} \|x_i - \bar{x}\|_2^2$. Khi một điểm dữ liệu mới x_{n+1} được thêm vào tập D_1, chúng ta có một tập mới được gọi là D_2, có $n+1$ điểm dữ liệu $x_1, x_2, \cdots, x_{n+1}$, trong đó trung bình là $\hat{x} = \frac{1}{n+1}\sum_{i=1}^{n+1} x_i$, và tổng khoảng cách từ tất cả $n+1$ điểm dữ liệu đến trung bình $\hat{x}$ là $\sum_{i=1}^{n+1} \|x_i - \hat{x}\|_2^2$.

Với cách định nghĩa $\bar{x}$ ta có $\bar{x} = \arg\min_x \sum_{i=1}^{n} \|x_i - x\|^2$. Từ đó ta suy ra được $\sum_{i=1}^{n} \|x_i - \bar{x}\|_2^2 \leq \sum_{i=1}^{n} \|x_i - \hat{x}\|_2^2$ và dấu bằng xảy ra khi và chỉ khi $\bar{x} = \hat{x}$. Hơn nữa, bất đẳng thức $\sum_{i=1}^{n} \|x_i - \hat{x}\|_2^2 \leq \sum_{i=1}^{n+1} \|x_i - \hat{x}\|_2^2$ đúng vì có điểm dữ liệu x_{n+1}, và đẳng thức xảy ra khi và chỉ khi $x_{n+1} = \hat{x}$. Tóm lại, chúng ta có kết luận rằng $\sum_{i=1}^{n} \|x_i - \bar{x}\|_2^2 \leq \sum_{i=1}^{n+1} \|x_i - \hat{x}\|_2^2$, trong đó đẳng thức xảy ra khi và chỉ khi $x_{n+1} = \bar{x}$. □

Định lý 2.2. Thuật toán tối thiểu luân phiên giải bài toán phân cụm K-Trung Bình không chứa cụm rỗng.

Chứng minh. Giả sử rằng có n điểm dữ liệu thuộc về c cụm, số lượng điểm dữ liệu trong mỗi cụm lần lượt là $n_1, n_2, \cdots, n_c$, $\sum_{j=1}^{c} n_j = n$, và tâm của mỗi cụm lần lượt là $m_1, m_2, \cdots, m_c$, x_i^j biểu diễn điểm dữ liệu thứ i thuộc về cụm thứ j, $m_j = \frac{1}{n_j}\sum_{i=1}^{n_j} x_i^j$ là trung bình của n_j điểm dữ liệu thuộc về cụm thứ j. Bây giờ chúng ta chứng minh rằng nếu cụm thứ p chứa duy nhất dữ liệu i thì điểm dữ liệu thứ i vẫn thuộc cụm thứ p sau khi cập nhật ở bước thứ i. Để chứng minh điều này, ta cần chỉ ra khi điểm dữ liệu thứ i không thuộc về cụm thứ p, giá trị hàm mục tiêu của bài toán (2.1) luôn lớn hơn hoặc bằng khi điểm dữ liệu thứ i thuộc về cụm thứ p.

Ta xét điểm dữ liệu thứ i thuộc về cụm thứ p, cụm này chỉ có một điểm dữ

liệu, thì hàm mục tiêu của bài toán (2.1) là

$$SSE = \sum_{i=1}^{n_p} \|x_i^p - m_p\|_2^2 + \sum_{i=1}^{n_j} \left\|x_i^j - m_j\right\|_2^2 + \sum_{k\neq p,j} \left(\sum_{i=1}^{n_k} \left\|x_i^k - m_k\right\|_2^2 \right).$$

trong đó $m_j = \frac{1}{n_j}\sum_{i=1}^{n_j} x_i^j$, $\sum_{i=1}^{n_p} \|x_i^p - m_p\|_2^2 = 0, n_p = 1$. Vì vậy, chúng ta có

$$SSE_1 = 0 + \sum_{i=1}^{n_j} \left\|x_i^j - m_j\right\|_2^2 + \sum_{k\neq p,j} \left(\sum_{i=1}^{n_k} \left\|x_i^k - m_k\right\|_2^2 \right).$$

Khi điểm dữ liệu trong cụm p được chuyển sang cụm j (p và j là hai cụm bất kỳ từ 1 đến c), hàm mục tiêu của bài toán (2.1) trở thành

$$SSE_2 = 0 + \sum_{i=1}^{n_j+1} \left\|x_i^j - \tilde{m}_j\right\|_2^2 + \sum_{k\neq p,j} \left(\sum_{i=1}^{n_k} \left\|x_i^k - m_k\right\|_2^2 \right).$$

trong đó $\hat{m}_j = \frac{1}{n_j+1}\sum_{i=1}^{n_j+1} x_i^j$. Bây giờ chúng ta chứng minh $SSE_1 \leq SSE_2$, nghĩa là, chúng ta chứng minh rằng

$$\sum_{i=1}^{n_j} \left\|x_i^j - m_j\right\|_2^2 \leq \sum_{i=1}^{n_j+1} \left\|x_i^j - \tilde{m}_j\right\|_2^2.$$

Bất đẳng thức trên đúng theo Bổ đề 2.1. Do đó, $SSE_1 \leq SSE_2$ được chứng minh, và đẳng thức chỉ xảy ra khi điểm dữ liệu thuộc về cụm p là m_j. Vì vậy, chúng ta đã chứng minh rằng thuật toán tối thiểu luân phiên không tạo ra cụm rỗng. □

Định lý 2.3. Giá trị hàm mục tiêu của K-Trung Bình hội tụ bởi thuật toán tối thiểu luân phiên không thể bị giảm bởi thuật tóan Lloyd.

Chứng minh. Giả sử trong thuật toán tối thiểu luân phiên, dãy giá trị hàm mục tiêu hội tụ tới giá trị

$$SSE_3 = \sum_{i=1}^{n_p} \|x_i^p - m_p\|_2^2 + \sum_{i=1}^{n_j} \left\|x_i^j - m_j\right\|_2^2 + \sum_{k\neq p,j} \left(\sum_{i=1}^{n_k} \left\|x_i^k - m_k\right\|_2^2 \right).$$

trong đó, $m_p = \frac{1}{n_p}\sum_{i=1}^{n_p} x_i^p$, $m_j = \frac{1}{n_j}\sum_{i=1}^{n_j} x_i^j$, và $m_k = \frac{1}{n_k}\sum_{i=1}^{n_k} x_i^k$. Nếu sự hội tụ thay đổi bởi Lloyd, chúng ta giả sử điểm dữ liệu x_i^p trong cụm p sẽ chuyển sang cụm j (p và j là hai cụm bất kỳ từ 1 đến c), nghĩa là điểm dữ liệu x_i^p gần tâm m_j hơn tâm m_p, ta có

$$SSE_4 = \sum_{i=1}^{n_p-1} \|x_i^p - m_p\|_2^2 + \sum_{i=1}^{n_j+1} \left\|x_i^j - m_j\right\|_2^2 + \sum_{k\neq p,j} \left(\sum_{i=1}^{n_k} \left\|x_i^k - m_k\right\|_2^2 \right).$$

Chúng ta sẽ chứng minh $SSE_4 \geq SSE_3$ bằng cách phản chứng, trước tiên chúng ta giả sử $SSE_4 < SSE_3$ và định nghĩa SSE_5 như sau

$$SSE_5 = \sum_{i=1}^{n_p-1} \|x_i^p - \hat{m}_p\|_2^2 + \sum_{i=1}^{n_j+1} \left\|x_i^j - \hat{m}_j\right\|_2^2 + \sum_{k\neq p,j} \left(\sum_{i=1}^{n_k} \left\|x_i^k - \hat{m}_k\right\|_2^2 \right)$$

trong đó, $\hat{m}_p = \frac{1}{n_p-1}\sum_{i=1}^{n_p-1} x_i^p$, $\hat{m}_j = \frac{1}{n_j+1}\sum_{i=1}^{n_j+1} x_i^j$ và $\hat{m}_k = \frac{1}{n_k}\sum_{i=1}^{n_k} x_i^k$. Giả sử $SSE_4 < SSE_3$, do sự hội tụ của thuật toán tối thiểu luân phiên, chúng ta có $SSE_3 \leq SSE_5$ (SSE_3 là giá trị hàm mục tiêu sau khi thuật toán luân phiên hội tụ, điều này có nghĩa là điểm dữ liệu x_i^p trong cụm p sẽ làm giảm giá trị mục tiêu so với cụm j), vì vậy chúng ta có $SSE_4 < SSE_5$.

Ta lại có

$$\sum_{i=1}^{n_p-1} \|x_i^p - \hat{m}_p\|_2^2 \leq \sum_{i=1}^{n_p-1} \|x_i^p - m_p\|_2^2 ,$$
$$\sum_{i=1}^{n_j+1} \left\|x_i^j - \hat{m}_j\right\|_2^2 \leq \sum_{i=1}^{n_j+1} \left\|x_i^j - m_j\right\|_2^2 .$$

, ta suy ra được $SSE_5 \leq SSE_4$, điều này mâu thuẫn với kết luận $SSE_4 < SSE_5$ được suy ra từ giả thuyết. Do đó, giả thuyết $SSE_4 < SSE_3$ là không đúng, ta suy ra $SSE_3 \leq SSE_4$, hay Định lý đúng. □

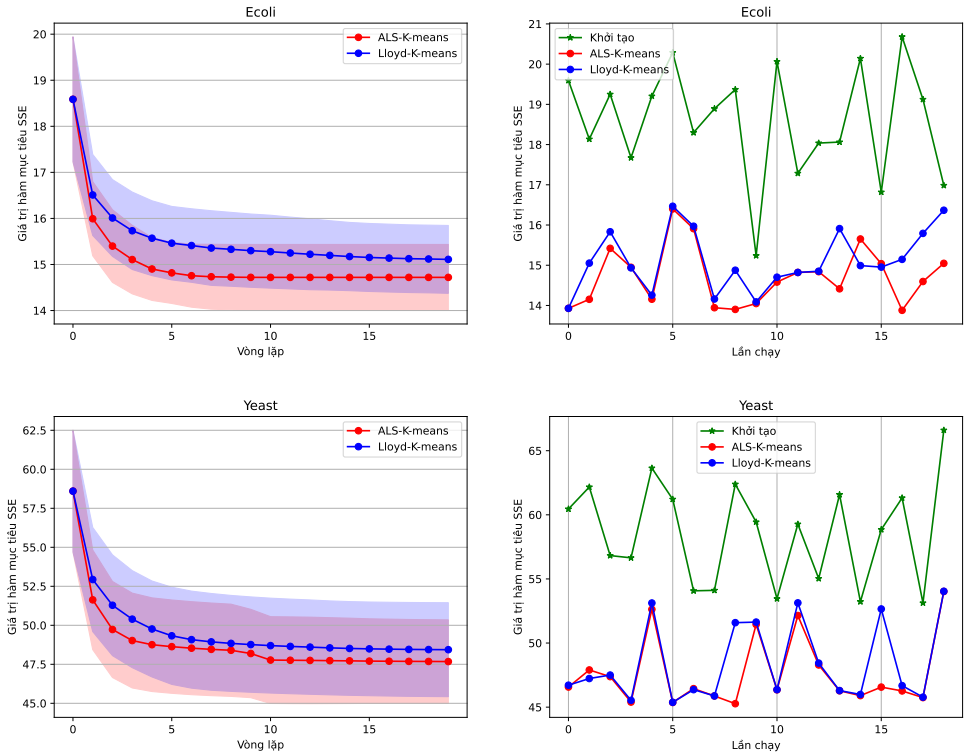
2.3 Thử nghiệm số

Tên dữ liệu	Số lượng điểm dữ liệu	Chiều điểm dữ liệu	Số nhóm
Yeast	1484	8	10
Ecoli	336	7	8

Bảng 2.1: Tập dữ liệu được sử dụng làm thí nghiệm cho bài toán K-Trung Bình

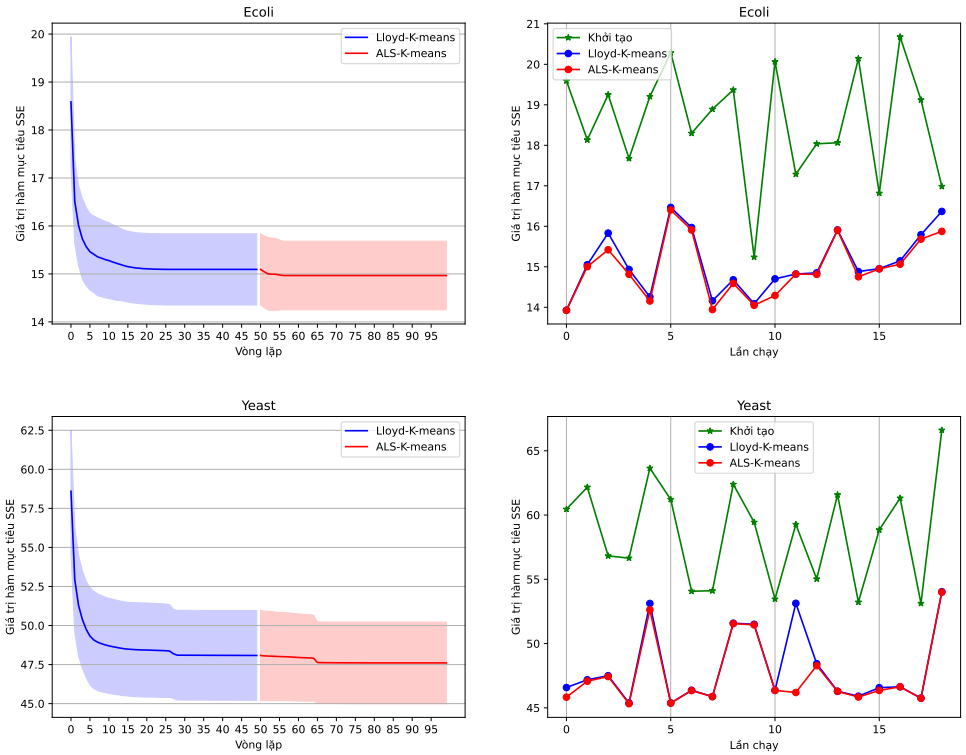
Trong phần này, chúng tôi thực hiện các thí nghiệm trên một số bộ dữ liệu tiêu chuẩn thực tế để kiểm chứng ưu điểm của phương pháp tối thiểu luân phiên cải tiến với thuật toán Lloyd. Các thí nghiệm cụ thể được chia thành ba phần:

- So sánh giá trị của hàm mục tiêu của hai phương pháp với cùng một khởi tạo ban đầu.
- Kiểm tra xem thuật toán tối thiểu luân phiên có thể thoát ra khỏi điểm cực tiểu cục bộ mà Lloyd đạt được và làm cho giá trị hàm mục tiêu tiếp tục giảm hay không.



Hình 2.1: Kết quả so sánh giữa thuật toán là tối thiểu luân phiên và Lloyd cho bài toán K-Trung Bình.

Dữ liệu được lấy để tiến hành thí nghiệm từ [5]. Chi tiết dữ liệu được mô tả như bảng (2.1). Việc khởi tạo tâm cụm ban đầu cho mọi thuật toán chúng tôi sử dụng thuật toán K-Trung Bình++ [6] và sử dụng kết quả 20 lần chạy độc lập để so sánh. Đầu tiên, chúng tôi so sánh kết quả phân cụm của 2 thuật toán là tối thiểu luân phiên và Lloyd. Kết quả trung bình và phương sai các lần chạy của các thuật toán được thể hiện như ở trong hình 2.1, trong đó trục x thể hiện số vòng lặp, trục y biểu diễn giá trị hàm mục tiêu của các thuật toán. Từ các hình bên trái, ta quan sát thấy rằng giá trị trung bình của hàm mục tiêu ở các vòng trong thuật toán tối thiểu luân phiên tốt hơn so với thuật toán Lloyd. Khi quan sát kết quả ở tất cả lần lặp khác nhau (bên phải), ta thấy rằng số lần giá trị hàm mục tiêu tìm được của thuật toán tối thiểu luân phiên nhỏ hơn so với của thuật toán Lloyd nhiều hơn so với trường hợp ngược lại. Điều



Hình 2.2: Kết quả việc phân cụm khi kết hợp thuật toán tối thiểu luân phiên sau thuật toán Lloyd.

này cho thấy sự hiệu quả của thuật toán tối thiểu luân phiên.

Tiếp theo, chúng ta kiểm tra liệu rằng cách phân cụm tìm được ở thuật toán Lloyd liệu rằng có thể cải thiện được khi tiếp tục sử dụng thuật toán tối thiểu Lloyd hay không? Từ hình (2.2) minh họa kết quả việc kết hợp hai thuật toán này với các trục tương tự như hình 2.1. Kết quả chi ra rằng, trong một vài trường hợp, thuật toán tối thiểu luân phiên vẫn có thể cải thiện hàm mục tiêu khi thuật toán Lloyd dừng.

Chương 3

Ứng dụng của phương pháp tối thiểu luân phiên trong bài toán khôi phục ma trận

Trong chương này, chúng tôi tập trung trình bày thuật toán tối thiểu luân phiên và một phiên bản cải tiến của thuật toán tối thiểu luân phiên SoftImpute-ALS để giải bài toán khôi phục ma trân. Nội dung chính của chương này gồm có:

3.1 Trình bày bài toán khôi phục ma trận trong thực tiễn và giới thiệu một số cách tiếp cận để giải quyết bài toán.

3.2 Chúng tôi trình bày thuật toán SoftImpute-ALS, phiên bản nhanh của thuật toán tối thiểu luân phiên cho bài toán khôi phục ma trận.

3.3 Phân tích lý thuyết về sự hội tụ của thuật toán SoftImpute-ALS.

3.4 Một số cải thiện cho thuật toán SoftImpute-ALS.

3.5 Tiến hành thử nghiệm các thuật toán trên các dữ liệu mô phỏng.

Nội dung tìm hiểu trong chương này tham khảo chính bài báo của Feiping Nie và các đồng nghiệp [7].

3.1 Giới thiệu bài toán khôi phục ma trận

Bài toán khôi phục ma trận (matrix completion) có thể được phát biểu như sau: Ta cần khôi phục ma trận X với m hàng và n cột nhưng chỉ quan sát được một số phần tử trong mn phần tử của ma trận X.

Bài toán này xuất hiện trong nhiều ứng dụng thực tế đặc biệt là trong việc thiết kế hệ thống gợi ý. Ở đó, người dùng gửi các đánh giá về một tập hợp các sản phẩm trong cơ sở dữ liệu và nhà cung cấp sản phẩm sẽ đưa ra các đề xuất dựa trên sự ưa thích của người dùng. Bởi vì người dùng chỉ đánh giá được một số các sản phẩm, vì vậy người ta muốn suy ra sở thích của họ đối với các sản phẩm chưa được đánh giá.

Một ví dụ đặc biệt của vấn đề này là vấn đề Netflix nổi tiếng hiện nay. Người dùng (các hàng của ma trận dữ liệu) có cơ hội xếp hạng phim (các cột của ma trận dữ liệu) nhưng người dùng thường chỉ xếp hạng rất ít phim do đó có rất ít các phần tử được quan sát và nằm rải rác trong ma trận dữ liệu này. Tuy nhiên, ta muốn khôi phục ma trận này để nhà cung cấp (ví dụ như Netflix) có thể giới thiệu các tựa phim mà bất kỳ người dùng cụ thể nào đó cũng ưa thích và có thể sẵn sàng mua. Trong trường hợp này, ma trận dữ liệu của tất cả đánh giá của người dùng có thể xấp xỉ hạng thấp vì người ta thường tin rằng chỉ một số yếu tố góp phần vào thị hiếu hoặc sở thích của một cá nhân.

Ta kí hiệu tập chỉ số của các phần tử quan sát được bởi Ω, nghĩa là

$$\Omega = \{(i, j) : X_{ij} \text{ được quan sát}\}.$$

Khi đó, ta có thể định nghĩa hình chiếu của X lên Ω, kí hiệu bởi $P_\Omega(X)$ là ma

trận $m \times n$ với các phần tử quan sát của X được bảo toàn trong khi các phần tử không được quan sát được thay bằng 0. Tương tự như vậy, ta kí hiệu $P_\Omega^\perp$ là phép chiếu lên phần bù của tập Ω.

Bài toán khôi phục ma trận có thể được mô hình hóa thành bài toán tối ưu sau đây:

$$\begin{aligned} &\text{minimize} && \operatorname{rank}(M) \\ &\text{v.d.k} && P_\Omega(M) = P_\Omega(X), \end{aligned} \tag{3.1}$$

Bài toán (3.1) là bài toán NP-khó, vì vậy thay vì giải trực tiếp bài toán này, người ta thường giải bài toán với hàm mục tiêu là hàm chuẩn nguyên tử của ma trận (xem trong [8, 9]):

$$\begin{aligned} &\text{minimize} && \|M\|_* \\ &\text{v.d.k} && P_\Omega(M) = P_\Omega(X). \end{aligned} \tag{3.2}$$

Ngoài ra, dựa trên mô hình của bài toán (3.2), Mazumder và các đồng tác giả ([10]) đã đưa ra bài toán quy hoạch lồi sau để khôi phục ma trận X:

$$\underset{M}{\text{minimize}}\ H(M) = \frac{1}{2}\|P_\Omega(X - M)\|_F^2 + \lambda\|M\|_*. \tag{3.3}$$

Bài toán (3.2) là bài toán tối ưu lồi và do đó chúng ta có thể giải nó bằng các phương pháp tối ưu cổ điển: phương pháp chiếu gradient, phương pháp điểm trong, ... Tuy vậy, khi bài toán có số chiều lớn, các phương pháp này tỏ ra không hiệu quả về mặt thời gian tính toán.

Bài toán khôi phục ma trận cũng được xem xét như một bài toán phân tích ma trận thỏa mãn ràng buộc bởi Rennies và Srebro ([11]). Cụ thể hơn, họ xem xét bài toán sau:

$$\underset{A,B}{\text{minimize}}\ F(A,B) = \frac{1}{2}\left\|P_\Omega\left(X - AB^T\right)\right\|_F^2 + \frac{\lambda}{2}\left(\|A\|_F^2 + \|B\|_F^2\right), \tag{3.4}$$

trong đó A là ma trận cỡ $m \times r$ và B là ma trận cỡ $n \times r$ với r cho trước. Để giải bài toán này ta thường sử dụng các thuật toán tối thiểu luân phiên (ALS) để giải quyết (3.4). Để tìm lời giải cho B, ta cố định A và giải (3.4). Cụ thể, ta cần đi giải bài toán:

$$\begin{aligned} B &= \arg\min_{\widetilde{B}} \frac{1}{2} \left\| P_\Omega \left(X - A\widetilde{B}^T \right) \right\|_F^2 + \frac{\lambda}{2} \|\widetilde{B}\|_F^2 \\ &= \arg\min_{\widetilde{B}} \sum_{j=1}^{n} \left(\frac{1}{2} \sum_{i \in \Omega_{\circ j}} \left(A_i^T \widetilde{B}_j - X_{ij} \right)^2 + \frac{\lambda}{2} \left\| \widetilde{B}_j \right\|_2^2 \right). \end{aligned} \tag{3.5}$$

Trong đó $\Omega_{\circ j} = \{i : (i,j) \in \Omega\}$. Quan sát thấy rằng, mỗi hàm thành phần thứ j trong hàm mục tiêu dạng tổng trong bài toán (3.5) chỉ phụ thuộc vào biến B_j nên ta có thể tối ưu độc lập từng hàm thành phần j để tìm B_j. Dễ thấy các hàm thành phần này là hàm lồi nên ta có các hàng B_j của ma trận B lần lượt thỏa mãn:

$$\begin{aligned} B_j &= \arg\min_{\widetilde{B}_j} \frac{1}{2} \sum_{i \in \Omega_{\circ j}} \left(A_i^T \widetilde{B}_j - X_{ij} \right)^2 + \frac{\lambda}{2} \left\| \widetilde{B}_j \right\|_2^2 \\ &= \left(\sum_{i \in \Omega_{\circ j}} A_i A_i^T + \lambda I \right)^{-1} \left(\sum_{i \in \Omega_{\circ j}} X_{ij} A_i \right). \end{aligned}$$

Tương tự, nếu B được cố định, việc giải cho A sẽ tương ứng với m bài toán con riêng biệt. Thuật toán tối thiểu luân phiên như mô tả ở trên được trình bày trong Thuật toán (5). Để dừng thuật toán, chúng tôi sử dụng sai số giữa hai lần lặp liên tiếp:

$$\delta_t = \|A_t - A_{t-1}\|_F + \|B_t - B_{t-1}\|_F. \tag{3.6}$$

Trong phần tiếp theo, chúng tôi sẽ trình thuật toán softImputeALS [7], một

Thuật toán 5 Thuật toán tối thiểu luân phiên

1: **Đầu vào:** $P_\Omega(X)$, giá trị khởi tạo A, B và ϵ
2: **Lặp lai đến khi nào** $\delta_t < \epsilon$
3: **for** i=1 to m **do**
4: $\quad A_i \leftarrow \left(\sum_{j\in\Omega_i} B_j B_j^T + \lambda I\right)^{-1} \left(\sum_{j\in\Omega_i} X_{ij} B_j\right).$
5: **end for**
6: **for** j=1 to n **do**
7: $\quad B_j \leftarrow \left(\sum_{i\in\Omega_j} A_i A_i^T + \lambda I\right)^{-1} \left(\sum_{i\in\Omega_j} X_{ij} A_i\right).$
8: **end for**

phiên bản nhanh của thuật toán tối thiểu luân phiên với chi phi tính toán mỗi vòng lặp sẽ được giảm đi.

3.2 SoftImpute-ALS: Phiên bản nhanh của thuật toán tối thiểu luân phiên

Giả sử $X \in \mathbb{R}^{m\times n}$ là ma trận có các giá trị không quan sát được và tập các chỉ số phần tử được quan sát là tập Ω. Xét bài toán (3.4):

$$\underset{A,B}{\text{minimize}} F(A,B) = \left\| P_\Omega\left(X - AB^T\right)\right\|_F^2 + \lambda\left(\|A\|_F^2 + \|B\|_F^2\right).$$

trong đó $A \in \mathbb{R}^{m\times r}$ và $B \in \mathbb{R}^{n\times r}$ với $r \leq \min(m,n)$ cho trước.

Thuật toán softImpute-ALS được xem như là một thuật toán kiểu tối thiểu "majorization minimization"(MM). Giả sử cần tối thiểu hóa hàm $f(x)$, thuật toán MM giải quyết bài toán thông qua hai bước chính: Majorization và Minimization. Ở bước Majorization, tại mỗi điểm hiện tại $x^{(t)}$, thuật toán xây dựng một hàm chặn trên $g(x \mid x^{(t)})$ sao cho $g(x^{(t)} \mid x^{(t)}) = f(x^{(t)})$ và $g(x \mid x^{(t)}) \geq f(x)$ với mọi x. Tiếp theo, ở bước Minimization, hàm chặn trên được tối thiểu hóa để tìm điểm mới $x^{(t+1)} = \arg\min_x g(x \mid x^{(t)})$. Quá

trình này lặp lại đến khi hội tụ, giúp tối ưu hóa hàm mục tiêu một cách hiệu quả thông qua các bài toán đơn giản hơn. Cụ thể, quay lại bài toán bài toán (3.4) ta có hàm mục tiêu:

$$F(A,B) = \frac{1}{2}\left\|P_\Omega\left(X - AB^T\right)\right\|_F^2 + \frac{\lambda}{2}\|A\|_F^2 + \frac{\lambda}{2}\|B\|_F^2.$$

Ta định nghĩa các hàm thay thế

$$\begin{aligned} Q_A\left(Z_1 \mid A,B\right) =& \frac{1}{2}\left\|P_\Omega\left(X - Z_1B^T\right) + P_\Omega^\perp\left(AB^T - Z_1B^T\right)\right\|_F^2 \\ &+ \frac{\lambda}{2}\left\|Z_1\right\|_F^2 + \frac{\lambda}{2}\|B\|_F^2 \end{aligned} \tag{3.7}$$

$$\begin{aligned} Q_B\left(Z_2 \mid A,B\right) =& \frac{1}{2}\left\|P_\Omega\left(X - AZ_2^T\right) + P_\Omega^\perp\left(AB^T - AZ_2^T\right)\right\|_F^2 \\ &+ \frac{\lambda}{2}\|A\|_F^2 + \frac{\lambda}{2}\left\|Z_2\right\|_F^2. \end{aligned} \tag{3.8}$$

Ta xem xét hàm $g\left(AB^T\right) = \frac{1}{2}\left\|P_\Omega\left(X - AB^T\right)\right\|_F^2$. Nếu đặt $Z = AB^T$, ta quan sát thấy rằng với mọi Z, bất kỳ $\bar{Z}$, ta có

$$\begin{aligned} g(Z) &\leq \frac{1}{2}\left\|P_\Omega(X - Z) + P_\Omega^\perp(\bar{Z} - Z)\right\|_F^2 \\ &= \frac{1}{2}\left\|\left(P_\Omega(X) + P_\Omega^\perp(\bar{Z})\right) - Z\right\|_F^2. \end{aligned} \tag{3.9}$$

trong đó đẳng thức xảy ra khi $Z = \bar{Z}$. Điều này dẫn đến điều đơn giản nhưng quan trọng sau:

$$Q_A\left(Z_1 \mid A,B\right) \geq F\left(Z_1,B\right), \quad Q_B\left(Z_2 \mid A,B\right) \geq F\left(A,Z_2\right) \tag{3.10}$$

gợi ý rằng $Q_A\left(Z_1 \mid A,B\right)$ là một chặn trên của $F(Z_1,B)$, tương tự ta có $Q_B\left(Z_2 \mid A,B\right)$ là một chặn trên của $F(A,Z_2)$. Hơn nữa, dấu bằng xảy ra khi $Z_1 = A, Z_2 = B$, tức là

$$Q_A(A \mid A,B) = F(A,B) = Q_B(B \mid A,B), \tag{3.11}$$

Ta thấy rằng thuật toán softImpute-ALS có thể miêu tả như thủ tục lặp sau:

$$A_{k+1} \in \arg\min_{Z_1} Q_A\left(Z_1 \mid A_k, B_k\right) \tag{3.12}$$

$$B_{k+1} \in \arg\min_{Z_2} Q_B\left(Z_2 \mid A_{k+1}, B_k\right). \tag{3.13}$$

Thuật toán 6 softImputeALS ([7])

Đầu vào: Ma trận X, khởi tạo A_0 và B_0, và $k = 0$.

Đầu ra: $(A^\star, B^\star)$ là ước lượng cực tiểu của bài toán 3.4.

Lặp lai đến khi nào $\delta_t < \epsilon$

1. $k \leftarrow k + 1$
2. $X^* \leftarrow P_\Omega(X) + P_\Omega^\perp\left(AB^T\right) = P_\Omega\left(X - AB^T\right) + AB^T$
3. $A \leftarrow X^* B\left(B^T B + \lambda I\right)^{-1} = \arg\min_{Z_1} Q_A\left(Z_1 \mid A, B\right)$
4. $X^* \leftarrow P_\Omega(X) + P_\Omega^\perp\left(AB^T\right)$
5. $B \leftarrow X^{*T} A\left(A^T A + \lambda I\right)^{-1} = \arg\min_{Z_2} Q_B\left(Z_2 \mid A, B\right)$

Thuật toán khai thác được sự phân tách

$$P_\Omega\left(X - AB^T\right) = P_\Omega(X) + P_\Omega^\perp\left(AB^T\right) - AB^T. \tag{3.14}$$

Giả sử rằng ta có ước lượng ở vòng lặp hiện tại cho cả A và B, ta mong muốn tính giá trị $\widetilde{B}$ mới. Đầu tiên, thuật toán sẽ thay sự xuất hiện lần đầu của AB^T trong về phải của (3.14) với ước lượng hiện tại, dẫn đến một ma trận đầy đủ $X^* = P_\Omega(X) + P_\Omega^\perp\left(AB^T\right)$, sau đó giải theo $\widetilde{B}$ trong bài toán

$$\underset{\widetilde{B}}{\text{minimize}} \left\| X^* - A\widetilde{B} \right\|_F^2 + \lambda \|\widetilde{B}\|_F^2. \tag{3.15}$$

Ta có thể viết lại X^* thành

$$X^* = P_\Omega(X) + P_\Omega^\perp\left(AB^T\right) = \left(P_\Omega(X) - P_\Omega\left(AB^T\right)\right) + AB^T; \tag{3.16}$$

Đây là cách biểu diễn *thưa với hạng thấp* hiệu quả cho bài toán nhiều chiều: hiệu quả trong việc lưu trữ và phép tính tích ma trận.

3.3 Phân tích sự hội tụ của thuật toán SoftImpute-ALS

Trong phần này, chúng ta sẽ nghiên cứu các tính chất của dãy lặp sinh bởi thuật toán SoftImpute-ALS và đánh giá tốc độ hội của toán. Trước tiên, ta sẽ chỉ ra rằng hàm mục tiêu F của bài toán (3.4) không bao giờ tăng sau mỗi vòng lặp của thuật toán softImpute-ALS.

Định lý 3.1. ([7]) Cho (A_k, B_k) là các lần lặp được tạo bởi softImpute-ALS. Giá trị hàm mục tiêu là đơn điệu giảm theo quy tắc cập nhập này, nghĩa là

$$F(A_k, B_k) \geq F(A_{k+1}, B_k) \geq F(A_{k+1}, B_{k+1}), \quad k \geq 1$$

Chứng minh. Theo quy tắc cập nhật A_{k+1}, khi đó ta có

$$\begin{aligned} Q_A(A_{k+1} \mid A_k, B_k) &= \min_{Z_1} Q_A(Z_1 \mid A_k, B_k) \\ &\leq Q_A(A_k \mid A_k, B_k) = F(A_k, B_k). \end{aligned}$$

Ta lại có $Q_A(A_{k+1} \mid A_k, B_k) \geq F(A_{k+1}, B_k)$. Từ những điều trên ta thu được $F(A_k, B_k) \geq F(A_{k+1}, B_k)$. Tương tự như trên đối với việc cập nhật B ta có:

$$\begin{aligned} F(A_{k+1}, B_k) &= Q_B(B_k \mid A_{k+1}, B_k) \\ &\geq Q_B(B_{k+1} \mid A_{k+1}, B_k) \geq F(A_{k+1}, B_{k+1}). \end{aligned}$$

Điều này chỉ ra rằng $F(A_k, B_k) \geq F(A_{k+1}, B_{k+1})$ với mọi k dẫn tới điều phải chứng minh. □

Trong phần tiếp theo, ta sẽ đưa ra một vài kết quả để nói lên tốc độ hội tụ của thuật toán softImpute-ALS tiến đến một điểm dừng.

Bổ đề 3.2. ([7]) Cho (A_k, B_k) là ước lượng tìm được của thuật toán ở lần lặp thứ k. Khi đó ta có:

$$\begin{aligned} &F\left(A_k, B_k\right) - F\left(A_{k+1}, B_{k+1}\right) \geq \\ &\frac{1}{2}\left(\left\|\left(A_k - A_{k+1}\right) B_k^T\right\|_F^2 + \left\|A_{k+1}\left(B_{k+1} - B_k\right)^T\right\|_F^2\right) \\ &+ \frac{\lambda}{2}\left(\left\|A_k - A_{k+1}\right\|_F^2 + \left\|B_{k+1} - B_k\right\|_F^2\right), \quad \forall k \geq 1. \end{aligned} \tag{3.17}$$

Để chứng minh bổ đề này ta cần sử dụng một kết quả cơ bản liên quan đến bài toán hồi quy ridge như sau:

Bổ đề 3.3. ([7]) Xét bài toán hồi quy ridge

$$H(\beta) = \frac{1}{2}\|y - M\beta\|_2^2 + \frac{\lambda}{2}\|\beta\|_2^2 \tag{3.18}$$

với $\beta^* \in \arg\min_\beta H(\beta)$. Khi đó đẳng thức sau đúng:

$$\begin{aligned} H(\beta) - H\left(\beta^*\right) &= \frac{1}{2}\left(\beta - \beta^*\right)^T \left(M^T M + \lambda \mathbf{I}\right)\left(\beta - \beta^*\right) \\ &= \frac{1}{2}\left\|M\left(\beta - \beta^*\right)\right\|_2^2 + \frac{\lambda}{2}\left\|\beta - \beta^*\right\|_2^2. \end{aligned}$$

Chứng minh. Quay lại bổ đề (3.2), ta có:

$$F\left(A_k, B_k\right) = g\left(A_k B_k^T\right) + \frac{\lambda}{2}\left(\|A_k\|_F^2 + \|B_k\|_F^2\right) \tag{3.19}$$

$$= Q_A\left(A_k \mid A_k, B_k\right) \tag{3.20}$$

$$\geq \min_{Z_1} Q_A\left(Z_1 \mid A_k, B_k\right) \tag{3.21}$$

$$= Q_A\left(A_{k+1} \mid A_k, B_k\right) \tag{3.22}$$

$$\geq g\left(A_{k+1} B_k^T\right) + \frac{\lambda}{2}\left(\|A_{k+1}\|_F^2 + \|B_k\|_F^2\right) \tag{3.23}$$

$$= F\left(A_{k+1}, B_k\right). \tag{3.24}$$

Từ (3.24) và (3.19) ta có:

$$F\left(A_k, B_k\right) - F\left(A_{k+1}, B_k\right) \geq Q_A\left(A_k \mid A_k, B_k\right) - Q_A\left(A_{k+1} \mid A_k, B_k\right) \tag{3.25}$$

$$= \frac{1}{2}\left\|\left(A_{k+1} - A_k\right) B_k^T\right\|_2^2 + \frac{\lambda}{2}\left\|A_{k+1} - A_k\right\|_2^2, \tag{3.26}$$

trong đó (3.26) được suy ra từ (3.25) sử dụng bổ đề 3.3. Tương từ bước trên với việc cập nhật B ta có:

$$F\left(A_k, B_k\right) - F\left(A_{k+1}, B_{k+1}\right) \geq \frac{1}{2}\left\|A_{k+1}\left(B_{k+1} - B_k\right)^T\right\|_2^2 + \frac{\lambda}{2}\left\|B_{k+1} - B_k\right\|_2^2. \tag{3.27}$$

Từ (3.26) và (3.27) ta suy ra được điều chứng minh. □

Với ma trận A và B bất kỳ lần lượt định nghĩa A^+, B^+ như sau:

$$A^+ \in \arg\min_{Z_1} Q_A\left(Z_1 \mid A, B\right), \quad B^+ \in \arg\min_{Z_2} Q_B\left(Z_2 \mid A^+, B\right) \tag{3.28}$$

Từ đó ta đặt:

$$\Delta\left((A, B), (A^+, B^+)\right) = \tfrac{1}{2}\left(\left\|\left(A - A^+\right) B^T\right\|_F^2 + \left\|A^+\left(B - B^+\right)^T\right\|_F^2\right) + \tfrac{\lambda}{2}\left(\left\|A - A^+\right\|_F^2 + \left\|B - B^+\right\|_F^2\right). \tag{3.29}$$

Định lý 3.4. ([7]) $\Delta\left((A, B), (A^+, B^+)\right) = 0$ khi và chỉ khi (A, B) là điểm bất động của softImpute-ALS.

Chứng minh. Đầu tiên, dễ thấy nếu A, B là điểm bất động thì

$$\Delta\left((A, B), \left(A^+, B^+\right)\right) = 0$$

Ta xét trường hợp ngược lại nghĩa là $\Delta = 0$. Chú ý rằng nếu $\Delta = 0$ thì mỗi thành phần của tổng xuất hiện trong định nghĩa của Δ là phải bằng 0. Ta có:

$$Q_A(A \mid A, B) - Q_A\left(A^+ \mid A, B\right) = \frac{1}{2}\left\|\left(A^+ - A\right)B^T\right\|_2^2 + \frac{\lambda}{2}\left\|A^+ - A\right\|_2^2$$

Vì vế phải bằng 0 nên ta suy ra được $Q_A(A \mid A, B) = Q_A\left(A^+ \mid A, B\right)$. Từ đó ta có $F(A, B) = F\left(A_+, B_+\right)$, hay (A, B) là điểm bất động. □

Ta sử dụng kí hiệu sau:

$$\eta_k = \Delta\left(\left(A_k, B_k\right), \left(A_{k+1}, B_{k+1}\right)\right). \tag{3.30}$$

Vì vậy η_k có thể sử dụng để xác định khi nào (A_k, B_k) gần đến điểm dừng.

Như là hệ quả của bổ đề 3.2 và tính đơn điệu giảm của chuỗi giá trị hàm mục tiêu $F(A_k, B_k)$, ta có $\eta_k \to 0$ khi $k \to \infty$. Định lý sau mô tả tốc độ mà η_k hội tụ về 0.

Định lý 3.5. ([7]) Cho $(A_k, B_k), k \geq 0$ là chuỗi được tạo ra bởi thuật toán softImpute-ALS. Khi đó chuỗi giá trị mục tiêu $F(A_k, B_k)$ giảm và hội tụ về $F^\infty \geq 0$ và $\eta_k \to 0$.

Hơn nữa, ta có tốc độ hội tụ của thuật toán softImpute-ALS như sau:

$$\min_{1 \leq k \leq K} \eta_k \leq \frac{\left(F\left(A_1, B_1\right) - F^\infty\right)}{K} \tag{3.31}$$

Chứng minh. Từ bổ đề 3.2 ta có:

$$\sum_{i=1}^{K}\left(F\left(A_k, B_k\right) - F\left(A_{k+1}, B_{k+1}\right)\right) \geq \sum_{k=1}^{K}\eta_k \geq K\left(\min_{K \geq k \geq 1}\eta_k\right) \tag{3.32}$$

Vì vậy $F\left(A_k, B_k\right)$ là chuỗi giảm bị chặn dưới, từ đó nó hội tụ về F^{∞}. Ta có thể suy ra rằng:

$$\begin{aligned}\sum_{i=1}^{K}\left(F\left(A_k, B_k\right)-F\left(A_{k+1}, B_{k+1}\right)\right) &= F\left(A^1, B^1\right)-F\left(A^{K+1}, B^{K+1}\right) \\ &\leq F\left(A^1, B^1\right)-F^{\infty}.\end{aligned} \tag{3.33}$$

Sử dụng (3.33) cùng với (3.32) ta có tốc độ hội tụ sau:

$$\min_{1\leq k\leq K}\eta_k \leq \left(F\left(A^1, B^1\right)-F\left(A^{\infty}, B^{\infty}\right)\right)/K, \tag{3.34}$$

từ đó ta có điều phải chứng minh. □

Định lý trên chỉ ra rằng tốc độ hội tụ của thuật toán softImpute-ALS $O(\frac{1}{K})$; nói cách khác, với một số $\epsilon > 0$. ta chỉ cần nhiều nhất $K = O(\frac{1}{\epsilon})$ để đạt được điểm (A_{k*}, B_{k*}) sao cho $\eta_{k^*} \leq \epsilon$, trong đó $1 \leq k^* \leq K$.

Chú ý rằng Định lý 3.5 chỉ ra tốc độ hội tụ của thuật toán với mọi giá trị $\lambda \geq 0$. Trong hệ quả sau, ta nêu ra vai trò cụ thể của λ anh hưởng tới tốc độ hội tụ của thuật toán.

Hệ quả 3.6. ([7]) Cho $(A_k, B_k), k \geq 1$ được định nghĩa như trong Định lý 3.5. Giả sử rằng với mọi $k \geq 1$ thì:

$$\ell^U\mathbf{I} \succeq B_k^T B_k \succeq \ell^L\mathbf{I}, \quad \ell^U\mathbf{I} \succeq A_k^T A_k \succeq \ell^L\mathbf{I}, \tag{3.35}$$

trong đó ℓ^U, ℓ^L là các hằng số độc lập với k. Khi đó ta có:

$$\begin{aligned}&\min_{1\leq k\leq K}\left(\|(A_k - A_{k+1})\|_F^2 + \|B_k - B_{k+1}\|_F^2\right) \\ &\leq \frac{2}{(\ell^L+\lambda)}\left(\frac{F\left(A_1, B_1\right)-F^{\infty}}{K}\right)\end{aligned} \tag{3.36}$$

$$\begin{aligned}
&\min_{1\le k\le K}\left(\left\|\left(A_k-A_{k+1}\right)B_k^T\right\|_F^2+\left\|A_{k+1}\left(B_k-B_{k+1}\right)^T\right\|_F^2\right)\\
&\le\frac{2\ell^U}{\lambda+\ell_U}\left(\frac{F\left(A_1,B_1\right)-F^\infty}{K}\right)
\end{aligned}\tag{3.37}$$

$$\begin{aligned}
&\min_{1\le k\le K}\left(\left\|\nabla_A f\left(A_k,B_k\right)\right\|^2+\left\|\nabla_B f\left(A_{k+1},B_k\right)\right\|^2\right)\\
&\le\frac{2\left(\ell^U\right)^2}{\left(\ell^L+\lambda\right)}\left(\frac{F\left(A_1,B_1\right)-F^\infty}{K}\right)
\end{aligned}\tag{3.38}$$

trong đó $\nabla_A f(A,B)$ (tương ứng $\nabla_B f(A,B)$) kí hiệu là đạo hàm riêng của $F(A,B)$ theo biến A(tương ứng B).

Chứng minh. Nhắc lại định nghĩa của η_k

$$\begin{aligned}
\eta_k=\frac{1}{2}\left(\left\|\left(A_k-A_{k+1}\right)B_k^T\right\|_F^2+\left\|A_{k+1}\left(B_k-B_{k+1}\right)^T\right\|_F^2\right)\\
+\frac{\lambda}{2}\left(\left\|A_k-A_{k+1}\right\|_F^2+\left\|B_k-B_{k+1}\right\|_F^2\right)
\end{aligned}$$

Theo giả thiết thì

$$\ell^U\mathbf{I}\succeq B_k^TB_k\succeq\ell^L\mathbf{I},\quad \ell^U\mathbf{I}\succeq A_k^TA_k\succeq\ell^L\mathbf{I},\quad \forall k$$

Ta lại có

$$\eta_k\ge\left(\frac{\ell^L}{2}+\frac{\lambda}{2}\right)\left\|A_k-A_{k+1}\right\|_F^2+\left(\frac{\ell^L}{2}+\frac{\lambda}{2}\right)\left\|B_k-B_{k+1}\right\|_F^2.$$

Áp dụng điều này vào (3.32) và giả sử $\ell>0$, ta có:

$$\begin{aligned}
&\min_{1\le k\le K}\left(\left\|\left(A_k-A_{k+1}\right)\right\|_F^2+\left\|B_k-B_{k+1}\right\|_F^2\right)\\
&\le\frac{2}{\left(\ell^L+\lambda\right)}\left(\frac{F\left(A^1,B^1\right)-F^\infty}{K}\right)
\end{aligned}\tag{3.39}$$

Thay vì sử dụng phép khoảng cách:

$$\left(\left\|\left(A_k-A_{k+1}\right)\right\|_F^2+\left\|B_k-B_{k+1}\right\|_F^2\right),$$

ta sẽ sự dụng phép đó khoảng cách mới:

$$\left(\left\|\left(A_k - A_{k+1}\right) B_k^T\right\|_F^2 + \left\|A_{k+1}\left(B_k - B_{k+1}\right)\right\|_F^2\right).$$

Quan sát thấy rằng:

$$\ell^U \left\|\left(A_k - A_{k+1}\right)\right\|_F^2 \geq \left\|\left(A_k - A_{k+1}\right) B_k^T\right\|_F^2,$$
$$\ell^U \left\|B_k - B_{k+1}\right\|_F^2 \geq \left\|A_{k+1}\left(B_k - B_{k+1}\right)^T\right\|_F^2.$$

Ta có:

$$\eta_k \geq \left(\frac{\lambda}{2\ell^U} + \frac{1}{2}\right)\left(\left\|\left(A_k - A_{k+1}\right) B_k^T\right\|_F^2 + \left\|A_{k+1}\left(B_k - B_{k+1}\right)\right\|_F^2\right).$$

Kết hợp với (3.32) ta thu được :

$$\begin{aligned}&\min_{1\leq k\leq K}\left(\left\|\left(A_k - A_{k+1}\right) B_k^T\right\|_F^2 + \left\|A_{k+1}\left(B_k - B_{k+1}\right)\right\|_F^2\right)\\ &\leq \frac{2\ell^U}{\lambda + \ell_U}\left(\frac{F\left(A^1, B^1\right) - F^\infty}{K}\right).\end{aligned} \tag{3.40}$$

Với giả thiết $\ell^U \mathbf{I} \succeq B_k^T B_k$ và $\ell^U \mathbf{I} \succeq A_k^T A_k$ có thể hiểu như cận trên của hệ số Lipschitz của các gradient của $Q_A\left(Z \mid A_k, B_k\right)$ và $Q_B\left(Z \mid A_{k+1}, B_k\right)$ với mọi k :

$$\begin{aligned}\left\|\nabla Q_A\left(A_{k+1} \mid A_k, B_k\right) - \nabla Q_A\left(A_k \mid A_k, B_k\right)\right\| &\leq \ell^U \left\|A_{k+1} - A_k\right\|,\\ \left\|\nabla Q_B\left(B_k \mid A_{k+1}, B_k\right) - \nabla Q_B\left(B_{k+1} \mid A_{k+1}, B_k\right)\right\| &\leq \ell^U \left\|B_{k+1} - B_k\right\|.\end{aligned} \tag{3.41}$$

Tự đó dẫn tới chặn trên tốc độ hội tụ trên các gradient của hàm $F(A, B)$, nghĩa là

$$\begin{aligned}&\min_{1\leq k\leq K}\left(\left\|\nabla_A f\left(A_k, B_k\right)\right\|^2 + \left\|\nabla_B f\left(A_{k+1}, B_k\right)\right\|^2\right)\\ &\leq \frac{2\left(\ell^U\right)^2}{\left(\ell^L + \lambda\right)}\left(\frac{F\left(A^1, B^1\right) - F^\infty}{K}\right).\end{aligned}$$

□

3.4 Thay đổi softImpute-ALS cho bài toán (3.3)

Chuỗi (A_k, B_k) được tạo bởi thuật toán (6) hướng tới việc tối ưu bài toán (3.4). Tiếp theo, ta sẽ xem xét mối liên hệ của chuỗi đó vơi (3.3). Ta biết rằng chuỗi $F(A_k, B_k)$ là chuỗi giảm, vậy liệu nó có suy ra được chuỗi $H(A_k B_k^T)$ là chuỗi giảm? Trong phần này, ta sẽ chỉ ra rằng có thể đạt được một chuỗi giảm $H(A_k B_k^T)$ với một sự thay đổi nhỏ. Thuật toán mới được miêu tả như trong thuật toán (7) với sự thay đổi ở bước 3 trong thuật toán này. Cụ thể, thay vì lấy trực tiếp $\widetilde{B}$ là giá trị cần tìm của B như phiên bản trước đó, ta sẽ cập nhật B như trong dòng bước số 3. Ý tưởng này dựa trên bổ đề sau:

Bổ đề 3.7. ([10]) Cho trước ma trận Z cỡ $m \times n$ và $rank(Z) \leq r \leq \min(m, n)$. Khi đó, giá trị tối ưu của bài toán

$$\begin{aligned} \underset{A,B}{\text{minimize}} \quad & \tfrac{1}{2}(\|A\|_F^2 + \|B\|_F^2), \\ \text{thoản mãn} \quad & A \in \mathbb{R}^{m\times r}, B \in \mathbb{R}^{n\times r} \\ & AB^T = Z. \end{aligned} \tag{3.47}$$

chính là $\|Z\|_*$ với nghiệm có dạng là:

$$A = UD, \quad B = VD \tag{3.48}$$

trong đó $U_{m\times r}, V_{n\times r}$ là ma trận trực chuẩn, $D_{r\times r}$ là ma trận đường chéo.

Từ bổ đề (3.7) ta thấy rằng

$$F\left(A_k, B_k\right) \geq H\left(A_k B_k^T\right).$$

Việc điều chỉnh tham số trong bước 3 của thuật toán (7) không làm thay đổi phần sai số của hàm mục tiêu $F\left(A_k, B_k\right)$ nhưng làm giảm phần chỉnh

Thuật toán 7 softImpute-ALS thay đổi [7]

1. Khởi tạo $A = UD$ trong đó $U_{m\times r}$ là ma trận trực chuẩn được chọn ngẫu nhiên and $D = I_r$, I_r là ma trận đơn vị kích thước $r \times r$, và $B = VD$ với $V = 0$.

2. Cho $A = UD$ và $B = VD$, ta giải quyết bài toán

$$\underset{\widetilde{B}}{\text{minimize}} \frac{1}{2} \left\| P_\Omega \left(X - A\widetilde{B}^T \right) \right\|_F^T + \frac{\lambda}{2} \|\widetilde{B}\|_F^2 \tag{3.42}$$

   để cập nhật B. Ta đạt được điều này với các bước sau:

   (a) Đặt $X^* = \left(P_\Omega(X) - P_\Omega \left(AB^T \right) \right) + AB^T$, được lưu trữ dưới dạng *thưa và hạng thấp*.

   (b) Giải quyết bài toán

$$\underset{\widetilde{B}}{\text{minimize}} \frac{1}{2} \left\| X^* - A\widetilde{B}^T \right\|_F^2 + \frac{\lambda}{2} \|\widetilde{B}\|_F^2, \tag{3.43}$$

   có nghiệm

$$\widetilde{B}^T = \left(D^2 + \lambda I \right)^{-1} DU^T X^* \tag{3.44}$$

$$= \left(D^2 + \lambda I \right)^{-1} DU^T \left(P_\Omega(X) - P_\Omega \left(AB^T \right) \right) \tag{3.45}$$

$$+ \left(D^2 + \lambda I \right)^{-1} D^2 B^T. \tag{3.46}$$

   (c) Cập nhật V và D :

      i. Tính SVD $\tilde{B}D = \tilde{U}\tilde{D}^2\tilde{V}^T$;

      ii. $V \leftarrow \tilde{U}$, và $D \leftarrow \tilde{D}$.

3. Cho $B = VD$, Giải theo A, bởi tính đối xứng, thay X^T bằng X, và B với A được đổi chỗ cho nhau.

4. Lặp lại bước 2 và 3 đến khi $\delta_t < \epsilon$.

5. Tính $M = X^*V$, và sau đó tính SVD: $M = UD_\sigma R^T$. Ta đưa ra $U, V \leftarrow VR$ và $D_{\sigma,\lambda} = \text{diag}\left[(\sigma_1 - \lambda)_+, \ldots, (\sigma_r - \lambda)_+ \right]$

hóa, vì vậy dẫn đến toàn bộ hàm mục tiêu giảm khi so sánh với thuật toán softImpute-ALS được mô tả trong (6). Vì vậy ta có bổ đề sau:

Bổ đề 3.8. ([7]) Cho chuỗi (A_k, B_k) được sinh ra bởi thuật toán softImpute-ALS thay đổi. Điều này dẫn đến một chuỗi giảm giá trị hàm mục tiêu của bài toán (3.3):

$$H\left(A_k B_k^T\right) \geq H\left(A_{k+1} B_{k+1}^T\right),$$

với $H\left(A_k B_k^T\right) = F\left(A_k, B_k\right)$ với mọi k. Vì vậy chuỗi $H\left(A_k B_k^T\right)$ hội tụ về F^∞.

Chú ý rằng, F^∞ không cần là giá trị cực tiểu của bài toán (3.3), có thể thấy dễ thấy điều này bằng việc lấy r nhỏ hơn hạng của nghiệm tối ưu bài toán lồi (3.3).

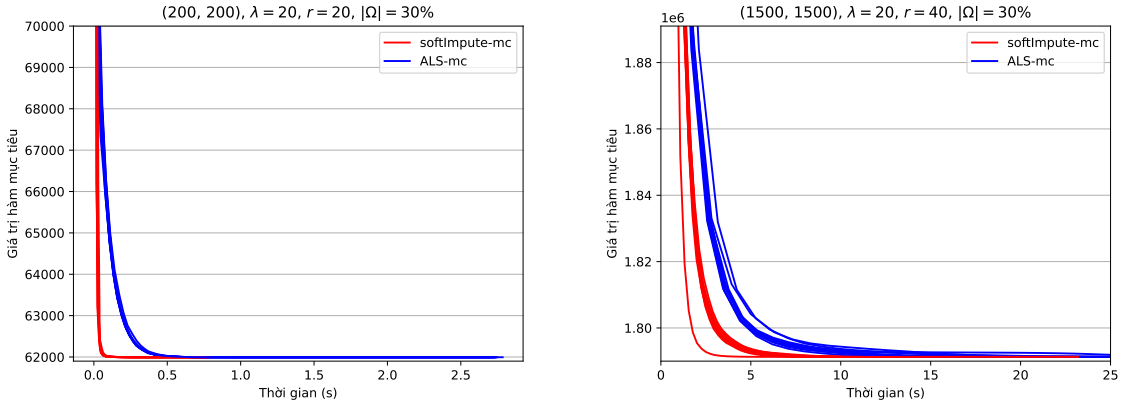
3.5 Độ phức tạp tính toán

Chi phí tính toán của thuật toán softImpute-ALS có thể được chia thành 3 bước. Đầu tiên xét chi phí cập nhật A. Bước đầu tiên ta hình thành ma trận $X^* = P_\Omega\left(X - AB^T\right) + AB^T$ yêu cầu $O(r|\Omega|)$ phép tính cho phần $P_\Omega(AB^T)$, trong khi phần còn lại không cần tính toán cụ thể. Ma trận $B\left(B^T B + \lambda I\right)^{-1}$ yêu cầu $O(2nr^2 + r^3)$ phép tính. Phép nhân ma trận $X^* B\left(B^T B + \lambda I\right)^{-1}$ và $AB^T B\left(B^T B + \lambda I\right)^{-1}$ yêu cầu $O\left(r|\Omega| + mr^2 + nr^2\right)$ phép tính bằng việc sử dụng câu trúc thưa và hạng thấp của X^*. Tổng chí phí của một vòng lặp $O\left(2r|\Omega| + 4mr^2 + 4nr^2 + 2r^3\right)$.

So sánh với thuật toán ALS với công thức cập nhật được cho thuật toán 5: mỗi hàng của A và B được cập nhật một cách riêng biệt thông qua các bài

toán con. Chi phí tính toán để giải mỗi bài toán con này là $O\left(|\Omega_j|\, r^2 + r^3\right)$, vì vậy chi phí cho một vòng lặp của ALS cần $O\left(2|\Omega|r^2 + mr^3 + nr^3\right)$ nhiều phép tính gấp r lần một vòng lặp của softImpute-ALS.

3.6 Thử nghiệm số



Hình 3.1: Kết quả so sánh hai thuật toán trong bài toán khôi phục ma trận.

Trong phần này, chúng tôi thực hiện so sánh về thời gian của thuật toán tối thiểu luân phiên và phiên bản nhanh SoftImpute-ALS trên dữ liệu mô phỏng. Hình (3.1) thể hiện kết quả thực nghiệm của hai thuật toán trong bài toán khôi phục ma trận khi chạy ở 10 lần khởi tạo khác nhau với mỗi một đường là một lần chạy. Mỗi hình sẽ được gán nhãn với kích thước ma trận (m, n), tham số λ, số chiều r được sử dụng và số lượng phần tử quan sát được $|\Omega|$ của ma trận cần khôi phục. Ma trận cần khôi phục M ở mỗi thí nghiệm tạo ra bằng cách lấy $M = AA^{\mathrm{T}} + E$ trong đó A là ma trận kích thước $m \times r$, E có kích thước $m \times m$, các phần tử trong các ma trận được rút độc lập và ngẫu nhiên từ phân phối đều $\mathcal{U}[0, 1]$. Quan sát thấy rằng, ở mọi lần khởi tạo, thuật toán SoftImpute-ALS ở các lần chạy trong thí nghiệm đều hội tụ nhanh hơn so với

các lần trong thuật toán tối thiểu luân phiên.

Kết luận và kiến nghị

Trong luận văn này, chúng tôi nghiên cứu và trình bày phương pháp tối thiểu luân phiên, một kỹ thuật phổ biến trong lý thuyết tối ưu với nhiều ứng dụng thực tiễn. Chúng tôi đưa ra các khái niệm cơ bản và phân tích cách hoạt động của thuật toán trong trường hợp tổng quát. Chúng tôi cũng đã nghiên cứu và trình bày ứng ứng của thuật toán trong các bài toán thực tế vào gồm bài toán phân cụm K-Trung Bình và bài toán khôi phục ma trận. Việc áp dụng phương pháp vào các bài toán thực tiễn đã chứng minh tính hiệu quả và tiềm năng của nó, mở ra những hướng nghiên cứu mới nhằm cải tiến thuật toán và mở rộng phạm vi ứng dụng trong các lĩnh vực.

Tài liệu tham khảo

[1] Michael JD Powell. On search directions for minimization algorithms. *Mathematical Programming*, 4:193–201, 1973.

[2] Amir Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.

[3] Feiping Nie, Jingjing Xue, Danyang Wu, Rong Wang, Hui Li, and Xuelong Li. Coordinate descent method for k k-means. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2371–2385, 2021.

[4] Trevor J. Hastie, Rahul Mazumder, J. Lee, and Reza Bosagh Zadeh. Matrix completion and low-rank svd via fast alternating least squares. *The Journal of Machine Learning Research*, 16:3367–3402, 2014.

[5] Dheeru Dua, Casey Graff, et al. Uci machine learning repository, 2017. *URL http://archive. ics. uci. edu/ml*, 7(1):62, 2017.

[6] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[7] Trevor Hastie, Rahul Mazumder, Jason D Lee, and Reza Zadeh. Matrix completion and low-rank svd via fast alternating least squares. *The Journal of Machine Learning Research*, 16(1):3367–3402, 2015.

[8] Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[9] Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[10] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322, 2010.

[11] Nathan Srebro, Jason Rennie, and Tommi Jaakkola. Maximum-margin matrix factorization. *Advances in neural information processing systems*, 17, 2004.