**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

Ảnh có chứa biểu tượng, Nhãn hiệu, Phông chữ, vòng tròn

Mô tả được tạo tự động

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN**

**NĂM HỌC 2024-2025**

**TÌM HIỂU BÀI TOÁN KHÔI PHỤC DỮ LIỆU**

**VÀ ỨNG DỤNG**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sinh viên thực hiện** : | | | |
| 1. | **Nguyễn Mạnh Khang** | Lớp : TUD63 | Khoa : KHCB |
| 2. | Đoàn Tiến Hiệp | Lớp : TUD63 | Khoa : KHCB |
| 3. | Nguyễn Ngọc Hiếu | Lớp : TUD63 | Khoa : KHCB |
| 4. | Hoàng Quang Tuyến | Lớp : TUD63 | Khoa : KHCB |

|  |  |
| --- | --- |
| **Người hướng dẫn** | : **PGS. TS Trần Văn Long** |

*Hà Nội, 2025*

**MỤC LỤC**

[**1. Mở đầu** 4](#_Toc196002026)

[**2. Tổng quan nghiên cứu về khôi phục dữ liệu** 4](#_Toc196002027)

[**3. Lý do chọn đề tài** 5](#_Toc196002028)

[**4. Mục đích, nội dung và phương pháp nghiên cứu** 6](#_Toc196002029)

[4.1 Mục đích nghiên cứu 6](#_Toc196002030)

[4.2 Nội dung nghiên cứu 6](#_Toc196002031)

[4.3 Phương pháp nghiên cứu 6](#_Toc196002032)

[**5. Đối tượng và phạm vi nghiên cứu** 6](#_Toc196002033)

[5.1 Đối tượng nghiên cứu 6](#_Toc196002034)

[5.2. Phạm vi nghiên cứu 7](#_Toc196002035)

[**CHƯƠNG I : HOÀN THÀNH MA TRẬN VÀ ỨNG DỤNG** 8](#_Toc196002036)

[**1.1 Giới thiệu** 8](#_Toc196002037)

[**1.2 Bài toán hoàn thành ma trận** 9](#_Toc196002038)

[1.2.1 Tối thiểu hóa chuẩn hạt nhân (Nuclear Norm Minimization) 9](#_Toc196002039)

[1.2.2 Xấp xỉ hạng tối thiểu 11](#_Toc196002040)

[1.2.3 Phân tích ma trận 12](#_Toc196002041)

[1.2.4 Tối thiểu hóa ℓp – norm 13](#_Toc196002042)

[1.2.5 Theo đuổi ngoại lai thích ứng 15](#_Toc196002043)

[**CHƯƠNG II : THUẬT TOÁN SVT HOÀN THÀNH MA TRẬN** 16](#_Toc196002044)

[**2.1 Giới thiệu** 16](#_Toc196002045)

[2.1.1 Tác nhân 16](#_Toc196002046)

[2.1.2 Mô tả thuật toán 16](#_Toc196002047)

[2.1.3 Công thức tổng quát 17](#_Toc196002048)

[2.1.4 Mối quan hệ và các công trình khác 17](#_Toc196002049)

[**2.2 Thuật toán ngưỡng giá trị kỳ dị (SVT)** 18](#_Toc196002050)

[2.2.1 Toán tử thu hẹp giá trị đơn (kỳ dị) 18](#_Toc196002051)

[2.2.2 Bước lặp thu nhỏ 20](#_Toc196002052)

[2.2.3 Mối quan hệ với các nghiên cứu khác 21](#_Toc196002053)

[2.2.4 Diễn giải theo phương pháp nhân tử Lagrange 22](#_Toc196002054)

[2.2.5 Phân tích hội tụ 23](#_Toc196002055)

[**2.3 Kết quả mô phỏng** 28](#_Toc196002056)

[2.3.1 Triển khai thuật toán 28](#_Toc196002057)

[2.3.2 Áp dụng vào thuật toán 32](#_Toc196002058)

[**KẾT LUẬN** 38](#_Toc196002059)

[**Kiến nghị** 39](#_Toc196002060)

[**Tài liệu tham khảo** 40](#_Toc196002061)

# **1. Mở đầu**

Trong thời đại dữ liệu lớn và trí tuệ nhân tạo đang phát triển mạnh mẽ, việc xử lý và khai thác thông tin từ dữ liệu không đầy đủ đang trở thành một thách thức quan trọng trong nhiều lĩnh vực như y tế, giáo dục, giải trí và mạng xã hội. Trong bối cảnh đó, kỹ thuật khôi phục ma trận (Matrix Completion) đã nổi lên như một giải pháp hiệu quả để tái tạo các phần dữ liệu bị thiếu thông qua việc khai thác các tính chất tiềm ẩn bên trong cấu trúc của dữ liệu.

Một minh chứng tiêu biểu cho tiềm năng ứng dụng của khôi phục ma trận là cuộc thi Netflix Prize (2006–2009). Trong đó, các nhóm nghiên cứu đã sử dụng phương pháp phân rã ma trận để dự đoán đánh giá phim của người dùng, trên một tập dữ liệu có mật độ chỉ khoảng 1,2%. Nhóm chiến thắng đã đạt mức cải thiện 10,06% so với hệ thống gốc, và giành được giải thưởng 1 triệu USD. Trong y học, kỹ thuật này giúp rút ngắn thời gian chụp ảnh cộng hưởng từ (MRI) từ 30 phút xuống còn 6 phút mà vẫn giữ nguyên chất lượng hình ảnh, theo nghiên cứu của Lustig et al. (2007). Ngoài ra, trong lĩnh vực cảm biến môi trường, mạng xã hội và giáo dục, khôi phục ma trận cũng cho thấy hiệu quả vượt trội trong việc tái tạo dữ liệu thiếu, với độ chính xác phục hồi lên tới 95% và sai số thấp (RMSE < 0.2).

Từ những dẫn chứng trên có thể thấy, khôi phục ma trận không chỉ là một bài toán lý thuyết mà còn có ứng dụng sâu rộng và thực tiễn, góp phần quan trọng vào việc đảm bảo tính toàn vẹn của dữ liệu trong nhiều hệ thống thông minh. Do đó, nghiên cứu này hướng đến việc tìm hiểu và ứng dụng các phương pháp khôi phục ma trận hiện đại để giải quyết vấn đề mất dữ liệu, qua đó làm nổi bật vai trò và tiềm năng của kỹ thuật này trong thời đại công nghệ số.

# **2. Tổng quan nghiên cứu về khôi phục dữ liệu**

Trong kỷ nguyên dữ liệu, ma trận là một cấu trúc phổ biến để biểu diễn thông tin trong nhiều lĩnh vực khoa học và kỹ thuật. Tuy nhiên, việc thu thập đầy đủ tất cả các phần tử của ma trận thường gặp khó khăn do nhiều yếu tố như lỗi thiết bị, hạn chế khảo sát, vấn đề riêng tư hay chi phí cao, dẫn đến tình trạng dữ liệu bị thiếu hoặc không hoàn chỉnh. Để khắc phục vấn đề này, bài toán khôi phục dữ liệu bị mất, hay còn gọi là Hoàn thành Ma trận (Matrix Completion), đã trở thành một lĩnh vực nghiên cứu quan trọng.

Bài toán Hoàn thành Ma trận tập trung vào việc tái tạo lại các giá trị còn thiếu của một ma trận dựa trên thông tin từ một tập hợp con các phần tử đã quan sát được. Về mặt toán học, đây là một bài toán không xác định nếu không có thêm giả định nào. Tuy nhiên, trong nhiều bài toán thực tế, ma trận dữ liệu thường có một cấu trúc tiềm ẩn đơn giản, có thể được biểu diễn hoặc xấp xỉ bằng một ma trận có hạng thấp (low-rank). Giả định hạng thấp này là nền tảng then chốt, cho phép khôi phục ma trận một cách hiệu quả từ một lượng dữ liệu thiếu đáng kể.

Bài toán Hoàn thành Ma trận có ý nghĩa thực tiễn sâu sắc và được ứng dụng rộng rãi, nổi bật nhất là trong:

* Hệ thống gợi ý (Recommendation Systems): Dự đoán sở thích (xếp hạng) của người dùng đối với các mặt hàng mà họ chưa tương tác, dựa trên dữ liệu xếp hạng đã có. Đây là ứng dụng nổi tiếng từ cuộc thi Netflix Prize.
* Xử lý tín hiệu và ảnh: Khôi phục các phần bị thiếu hoặc bị nhiễu trong tín hiệu thu được hoặc hoàn thành các vùng bị khuyết trong ảnh.
* Phân tích dữ liệu: Áp dụng trong các lĩnh vực như genomics, phân tích mạng xã hội, và các bài toán khác liên quan đến ma trận dữ liệu không đầy đủ.

Để giải quyết bài toán Hoàn thành Ma trận, nhiều phương pháp tối ưu hóa đã được phát triển. Một hướng tiếp cận truyền thống và có nền tảng lý thuyết vững chắc là dựa trên tối ưu lồi. Thay vì tối ưu trực tiếp hàm hạng (vốn là phi lồi), các phương pháp này thường tối thiểu hóa chuẩn hạt nhân (nuclear norm) - một hàm lồi là bao lồi của hàm hạng. Trong khuôn khổ này, Thuật toán Ngưỡng mềm Giá trị kỳ dị (Singular Value Thresholding - SVT) nổi lên như một thuật toán tiêu biểu và hiệu quả. SVT hoạt động dựa trên quá trình lặp, sử dụng phép phân tích giá trị kỳ dị (SVD) và áp dụng ngưỡng mềm lên các giá trị kỳ dị để ước lượng ma trận hạng thấp tại mỗi bước, đồng thời đảm bảo ma trận ước lượng khớp với các phần tử đã quan sát.

Bài toán Hoàn thành Ma trận là một thách thức quan trọng trong xử lý dữ liệu thiếu, được giải quyết hiệu quả nhờ vào giả định hạng thấp. Các phương pháp tối ưu, đặc biệt là hướng tiếp cận tối ưu lồi với các thuật toán như SVT, đã cung cấp những công cụ mạnh mẽ cho phép khôi phục dữ liệu và ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau.

# **3. Lý do chọn đề tài**

Nghiên cứu về khôi phục và phục hồi dữ liệu bị mất là một lĩnh vực mang tính chiến lược trong kỷ nguyên số hiện nay. Việc đã chọn đề tài này vì ba lý do chính:

+ Thứ nhất, tình trạng mất dữ liệu đang gia tăng đáng báo động trên phạm vi toàn cầu, với ước tính thiệt hại lên đến hàng tỷ đô la mỗi năm.

+Thứ hai, mặc dù tầm quan trọng hiển nhiên, nhưng các nghiên cứu chuyên sâu về kỹ thuật khôi phục dữ liệu trong bối cảnh công nghệ mới nổi như điện toán đám mây, trí tuệ nhân tạo và Internet vạn vật (IoT) còn khá hạn chế.

+Thứ ba, đây là lĩnh vực liên ngành, kết hợp giữa khoa học máy tính, an ninh mạng và quản trị hệ thống thông tin, tạo cơ hội để phát triển các giải pháp đột phá, đáp ứng nhu cầu cấp thiết của xã hội số.

# **4. Mục đích, nội dung và phương pháp nghiên cứu**

## 4.1 Mục đích nghiên cứu

Đề tài hướng đến các mục đích cụ thể sau:

**-** Hệ thống hóa cơ sở lý thuyết của bài toán khôi phục ma trận (Matrix Completion)

**-** Phát triển thuật toán khôi phục dữ liệu hiệu quả để tái tạo ma trận bị khuyết thiếu

**-** Triển khai và đánh giá các thuật toán được xây dựng trên các tập dữ liệu thực tế, nhằm kiểm chứng khả năng khôi phục ma trận trong các ứng dụng cụ thể như xử lý ảnh

## 4.2 Nội dung nghiên cứu

**Tìm hiểu bài toán khôi phục ma trận**, **khám phá ứng dụng của bài toán khôi phục ma trận**, **xây dựng và triển khai các thuật toán khôi phục dữ liệu**, phát triển một số thuật toán khôi phục ma trận, **đánh giá và đề xuất cải tiến**

## 4.3 Phương pháp nghiên cứu

Triển khai các thuật toán khôi phục ma trận bằng Python, sử dụng các thư viện như NumPy, Pandas, SciPy

Thử nghiệm trên các tập dữ liệu chuẩn (như MovieLens) và dữ liệu tự tạo (như ảnh số bị mất pixel ngẫu nhiên).

Đánh giá hiệu suất thuật toán thông qua các chỉ số định lượng và trực quan hóa kết quả ( hình ảnh phục hồi).

Mô phỏng các kịch bản khuyết thiếu dữ liệu, bao gồm mất ngẫu nhiên và mất có cấu trúc, để kiểm tra khả năng của các thuật toán.Sử dụng các công cụ Python để mô phỏng và phân tích kết quả.

# **5. Đối tượng và phạm vi nghiên cứu**

## 5.1 Đối tượng nghiên cứu

- Đối tượng nghiên cứu của đề tài bao gồm:

**- Bài toán khôi phục ma trận (Matrix Completion)**

- Các khái niệm, giả thuyết, và điều kiện lý thuyết của bài toán, đặc biệt là ma trận hạng thấp bị khuyết thiếu.

- Các thuật toán khôi phục ma trận, từ phương pháp tối ưu hóa truyền thống đến các kỹ thuật học sâu.

**-** Các dạng dữ liệu không đầy đủ, bao gồm dữ liệu mất ngẫu nhiên và một phần dữ liệu mất có cấu trúc (như dữ liệu)

**- Ứng dụng của khôi phục ma trận**:

+ Các lĩnh vực ứng dụng chính, bao gồm hệ thống đề xuất, khôi phục ảnh số, và xử lý dữ liệu lớn.

**+ Thuật toán và công cụ tính toán**:

+ Các thuật toán được xây dựng và triển khai để khôi phục ma trận.

+ Các công cụ lập trình và thư viện phần mềm (như Python, NumPy, TensorFlow) dùng để phát triển và đánh giá thuật toán.

## 5.2. Phạm vi nghiên cứu

- **Phạm vi nội dung**: Tập trung vào bài toán khôi phục ma trận hạng thấp (low-rank Matrix Completion), với trọng tâm là các thuật toán khôi phục dữ liệu bị khuyết thiếu.

**- Phạm vi phương pháp**: Xây dựng và triển khai thuật toán tiêu biểu SVT

**- Phạm vi ứng dụng**:

+ Chủ yếu áp dụng trong hệ thống đề xuất (ví dụ: dự đoán đánh giá người dùng) và xử lý ảnh số (khôi phục ảnh bị mất pixel).

+ Sử dụng các tập dữ liệu chuẩn như MovieLens (cho hệ thống đề xuất) và dữ liệu ảnh số tự tạo (cho xử lý ảnh).

+ Tập trung vào dữ liệu mất ngẫu nhiên

# **CHƯƠNG I : HOÀN THÀNH MA TRẬN VÀ ỨNG DỤNG**

## **1.1 Giới thiệu**

Trong vài năm gần đây, Matrix Completion (MC) đã thu hút được sự quan tâm rộng rãi trên toàn thế giới nhờ tính độc đáo và nhiều ứng dụng trong các lĩnh vực như cảm biến giao thông , tích hợp radar và thông tin liên lạc , phục hồi ảnh , nhận dạng hệ thống , học đa nhiệm và hiều lĩnh vực khác.

Tiếp nối với Compressed Sensing (CS), MC là một công nghệ quan trọng khác sử dụng tính chất thưa (sparse) để xử lý dữ liệu. Trong CS, "thưa" có nghĩa là tín hiệu chứa nhiều phần tử bằng 0 trong một miền xác định. Tuy nhiên, trong MC, nó biểu thị rằng các vector giá trị suy biến (singular value) của ma trận gốc là thưa – nói cách khác, ma trận có hạng thấp.  
 MC có khả năng khôi phục tín hiệu gốc X từ một tín hiệu phân mảnh (còn gọi là tín hiệu thiếu hoặc chưa lấy mẫu đầy đủ), trong đó Ω là một tập con chứa tọa độ 2D của các phần tử đã được lấy mẫu. Tín hiệu thiếu có thể được biểu diễn như sau:

(1.1)  
  
trong đó :

tất cả các biến đều thuộc

⊙ là phép nhân phần tử (element-wise),

và N lần lượt là ma trận lấy mẫu và ma trận nhiễu.

Lưu ý rằng là một ma trận nhị phân được lấy ngẫu nhiên sao cho mỗi hàng và cột có ít nhất một phần tử bằng 1 . Hơn nữa, giả định rằng tín hiệu gốc X có tính chất hạng thấp hoặc xấp xỉ hạng thấp.

Tín hiệu hạng thấp phổ biến trong các ứng dụng thực tế. Ví dụ, trong hệ thống radar MIMO, tín hiệu có hạng thấp vì các mục tiêu và nhiễu trong khu vực thử (CUT) thưa trong miền không gian. Số lượng mục tiêu và nhiễu trong tín hiệu phản xạ tương ứng với hạng của tín hiệu gốc, vốn thường nhỏ hơn nhiều so với số lượng ăng-ten truyền và thu. Một ví dụ khác là ma trận dữ liệu ảnh – thông tin chính được chi phối bởi một số giá trị suy biến lớn nhất, còn các giá trị nhỏ hơn có thể xem như bằng 0 mà không làm mất thông tin chính. Do đó, dữ liệu ảnh có cấu trúc xấp xỉ hạng thấp.

Trong công trình tiên phong của Candès và Recht , người ta đề xuất sử dụng bài toán cực tiểu hóa hạng để khôi phục tín hiệu gốc X. Bài toán MC trong môi trường không nhiễu được phát biểu như sau:

trong đó : M ∈ , , và là phép chiếu của M lên Ω.

Khi tín hiệu thu được bị nhiễu, cần ràng buộc mức độ nhiễu trong một giới hạn cho phép. Khi đó, bài toán MC có thể được phát biểu lại như sau:

(1.3)

trong đó :

được định nghĩa trong (1),

là chuẩn Frobenius và δ > 0 là tham số dung sai sai số.

Đáng tiếc, việc cực tiểu hóa hạng là bài toán NP-khó, vì tất cả thuật toán giải chính xác cho (2) và (1.3) đều có độ phức tạp bùng nổ theo kích thước max(m,n) cả về lý thuyết và thực hành.  
 Vì lý do đó, tất cả các thuật toán hiện đại đều cố gắng giải bài toán xấp xỉ bằng cách cực tiểu hóa chuẩn hạt nhân (nuclear norm) – là một dạng lồi tương đương với hạng. Fazel chứng minh rằng chuẩn hạt nhân là dạng lồi tương đương với hạng, tương tự như việc nới lỏng từ chuẩn ℓ₀ sang ℓ₁ trong CS . Candès và Recht sau đó đề xuất thay thế (1.2) bằng:

(1.4)  
  
trong đó là chuẩn hạt nhân của ma trận.

Quan trọng hơn, đã chứng minh rằng nếu ma trận gốc X thỏa mãn tính chất phân tán mạnh (strong incoherence), thì có thể khôi phục chính xác với xác suất cao bằng cách giải bài toán (1.4).

Báo cáo này nhằm cung cấp một cái nhìn tổng quan về các phương pháp MC, bao gồm các bài toán tối ưu khác nhau và thuật toán giải chúng, nhấn mạnh vào nguyên lý và điểm khác biệt. Đồng thời, bài viết trình bày nhiều ví dụ ứng dụng cụ thể của MC.

## **1.2 Bài toán hoàn thành ma trận**

### 1.2.1 Tối thiểu hóa chuẩn hạt nhân (Nuclear Norm Minimization)

Bài toán tối thiểu hóa chuẩn hạt nhân (1.4)( Nuclear Norm Minimization)

có thể được biểu diễn lại dưới dạng bài toán lập trình bán xác định (semidefinite programming – SDP) như sau :

Trong đó :

, là các ma trận bán xác định dương (positive semidefinite).

là tổng trên phần tử của

là ma trận chuyển vị của M

Bài toán MC như là một trường hợp đặc biệt của bài toán PCA vững và xây dựng như sau:

Trong đó : là ma trận thưa.

Phương pháp nhân tử Lagrange mở rộng không chính xác (IALM) giải phiên bản nhân tử Lagrange mở rộng của bài toán (1.6) để thu được kết quả . Tuy nhiên, phương pháp trong (1.6) không xem xét môi trường có nhiễu do , vì vậy hạn chế khả năng ứng dụng.

Để cải thiện độ chính xác của bài toán MC, phương pháp tối thiểu hóa chuẩn hạt nhân có trọng số (WNNM) đưa ra các trọng số khác nhau cho các giá trị kỳ dị, tránh việc thu hẹp tất cả các giá trị kỳ dị theo cùng một cách. WNNM linh hoạt hơn TNNR vì trọng số càng lớn thì giá trị kỳ dị càng giảm. Trong tình huống đặc biệt, WNNM có thể giữ lại các giá trị kỳ dị lớn nhất với trọng số bằng 0. Chuẩn hạt nhân có trọng số của ma trận được định nghĩa như sau:

trong đó là trọng số không âm gán cho .

Dựa trên chuẩn hạt nhân có trọng số , một biến thể của PCA vững cho bài toán MC đã được xây dựng trong , được mô tả như sau :

Cần lưu ý rằng, mặc dù PCA vững tiêu chuẩn dành cho khôi phục ma trận hạng thấp có thể xử lý nhiễu xung, PCA vững cho MC trong (1.6) và (1.8) không đủ mạnh trước nhiễu xung. PCA vững tiêu chuẩn được mô tả như sau:

trong đó:

là ma trận mục tiêu có tính chất hạng thấp,

Điều thú vị là trong ràng buộc của (9) có thể xem là nhiễu xung được thêm vào . Do đó, tính chất thưa của có thể đặc trưng qua chuẩn ​. Vì vậy, PCA vững tiêu chuẩn có khả năng chống nhiễu xung, trong khi biến thể của nó cho bài toán MC không giữ được tính bền vững này. Thực tế, nếu các phần tử được lấy mẫu trong (6) bị nhiễu cộng thêm, thành phần nhiễu không thể bị loại bỏ do . Đây là lý do tại sao PCA vững cho MC hoạt động kém hiệu quả trong trường hợp có nhiễu, đặc biệt là nhiễu xung.

### 1.2.2 Xấp xỉ hạng tối thiểu

Các phương pháp đã đề cập để giải quyết các vấn đề hoàn thiện mà trận (MC) được thiết kế dựa trên giả định rằng các mẫu dữ liệu không nhiễu hoặc có nhiễu. Trên thực tế, chúng ta không thể biết trước liệu dữ liệu có bị nhiễu hay không. Để giải quyết vấn đề này, phương pháp phân rã nguyên tử cho xấp xỉ bậc tối thiểu (ADMiRA) đã được đề xuất để giải quyết vấn đề MC thông qua công thức thay thể của bài toán tối thiểu hoá hạng, được gọi là bài toán xấp xỉ hạng tối thiểu, đó là:

trong đó : r là giới hạn của hạng.

Ưu điểm của bài toán tối ưu này là nó xem xét cả trường hợp không có nhiễu và có nhiễu. Nó cũng phù hợp hơn cho tình huống mà ma trận gốc không hoàn toàn có hạng thấp nhưng có thể xấp xỉ có hạng thấp

ADMiRA được phát triển trong khuôn khổ của phương pháp truy tìm khớp trực giao và trong mỗi lần lặp, nó đầu tiên tìm kiếm 2r thành phần và sau đó thu được ma trận hạng r bằng cách thực hiện SVD. Kếu quả là, nó thể hiện hiệu quả tính toán thấp đối với các ma trận có kích thước lớn. Để giải quyết vấn đề này , phương pháp chiếu giá trị kỳ dị (SVP) đã đề xuất để giải quyết (1.10). Đồng thời, nó cũng sử dụng bước kiểu Newton để cải thiện độ chính xác và hội tụ. Ngoài ra các biến thể của ADMiRA đã được đề xuất để giải quyết vấn đề xấp xỉ hạng tối thiểu

### 1.2.3 Phân tích ma trận

Mặc dù các phương pháp Hoàn Thành Ma trận có khả năng cải thiện hiệu suất bằng cách điều chỉnh việc giảm thiểu định mức hạt nhân tiêu chuẩn, chúng gặp phải một số vấn đề như hiệu quả tính toán thấp và khả năng mở rộng hạn chế khi xử lý dữ liệu lớn. Để khắc phục những vấn đề này, phương pháp Hệ số hóa Ma trận (MF) đã được đề xuất để giải quyết vấn đề Hoàn Thành Ma trận mà không cần SVD (Phân tích giá trị kỳ dị).

Ý tưởng cơ bản của phương pháp MF là sử dụng hai ma trận cấp thấp để biểu diễn ma trận mục tiêu, với giả định rằng thứ hạng của ma trận gốc là đã biết. Thuật toán LMaFit là thuật toán đầu tiên ứng dụng kỹ thuật MF để giải quyết vấn đề Hoàn Thành Ma trận.

trong phương pháp LMaFit, các ma trận ,, được sử dụng, với r là hạng dự đoán của ma trận mục tiêu. Phương pháp này áp dụng Lagrange của bài toán tối ưu. Mặc dù LMaFit có khả năng đạt được giải pháp chính xác, nhưng nó không thể đảm bảo tính tối ưu toàn cục do hàm không lồi.

Phương pháp tối ưu hóa là một biến thể của LMaFit, trong đó bài toán tối ưu được điều chỉnh để cải thiện khả năng Hoàn Thành Ma Trận.

Để tăng cường hội tụ của quy trình tối ưu hóa, phương pháp giảm bậc đã được điều chỉnh để giải quyết bài toán tối ưu. Đây là công trình đầu tiên nghiên cứu lý thuyết về tính tối ưu toàn cục trong phương pháp Hoàn thành Ma trận dựa trên Hệ số hóa Ma trận.

Để tối ưu hiệu suất của phương pháp Hoàn thành Ma trận dựa trên Hệ số hóa Ma trận, OptSpace đã phân tích ma trận mục tiêu thành và giải quyết bài toán tối ưu hóa bằng không gian Grassmann.

Kí hiệu:

: Ma trận chứa các vector hàng

: Ma trận chứa các vector cột

: Ma trận đường chéo

Điều kiện:

: Điều kiện ràng buộc cho ma trận U (với I là ma trận đơn vị)

: Điều kiện ràng buộc cho ma trận V

Vấn đề tối ưu:

+ Phương pháp OptSpace cần đồng thời không gian hàng và không gian cột để có hàm mục tiêu.

+ Tuy nhiên, việc này không đảm bảo giải pháp tốido có thể xuất hiện các rào cản trong quá trình tìm kiếm.

Giải pháp:

+ SET: Đề xuất phân rã ma trận MMM thành hai ma trận có hạng thấp dưới dạng .

+ Xây dựng một bài toán tối ưu hóa để tìm ra các ma trận U và V nhằm tối thiểu hóa hàm mục tiêu, đồng thời đảm bảo các ràng buộc.

Trong phương pháp SET, ma trận là ma trận trực chuẩn, và ma trận với r nhỏ hơn nhiều so với giá trị nhỏ hơn giữa m và n. So với OptSpace, SET chỉ tìm kiếm không gian cột (hoặc hàng).

### 1.2.4 Tối thiểu hóa ℓp – norm

Phép đo khoảng cách euclidian (ℓ2 – norm) có thể mô tả chính xác phương sai của nhiễu Gauss độc lập và phân phối giống nhau. Tuy nhiên , đối với nhiễu xung ℓ2 – norm thường làm hỏng dữ liệu nhận được trong các ứng dụng thực tế vì nó khuếch đại đáng kể sức mạnh của các giá trị ngoại lai, làm cho ảnh hưởng của nhiễu xung lớn hơn nhiều so với nhiễu Gauss.Do đó , diều này thúc đẩy ngưởi ta khai thác các phép đo khác cho kịch bản nhiễu xung. Đối với ma trận R ℓ2 – norm được định nghĩa :

(1.15)

trong đó :

là phần tử trong ma trận R

ℓp – norm với 0 < p < 2 có thể chống lại giá trị ngoại lai, do đó nó được áp dụng rộng rãi để xử lí nhiễu xung.

Tuy nhiên ,có ít bài viết giải thích tại sao nó có thể chống lại nhiễu xung. Dưới đây là 1 ví dụ giải thích giúp người đọc hiểu được tính chất này.

Xét bài toán tối thiểu hóa:

R là ma trận dư giữa M và X ,

,

là số hạng phạt dư, và tổng của chúng biểu thị cho tổng hình phạt.

Các lựa chọn khác nhau của M dẫn đến các số dư khác nhau, và cuối cùng có thể đưa ra nhiều các tiếp cận khác nhau.

Một cách đại khái, đo lường mức độ không thích của chúng ta đối với . Nếu rất nhỏ, nó không ảnh hưởng đến hiệu xuất phục hồi . Tuy nhiên nếu trở nên lớn, điều đó cho thấy chung ta phải xử lí không thích mạnh đối với các giá trị dư lớn này. Sự không thích này tương ứng với giá trị chung ta cần.

Thuật toán hồi quy ℓp-reg kết hợp kỹ thuật phân rã ma trận (MF) và chuẩn ℓp để giải quyết bài toán hoàn thiện ma trận (MC), được định dạng như sau:

Để xử lý hiệu quả dữ liệu lớn và khung phân tán, chiến lược tối ưu hóa xen kẽ (alternating minimization) để giải bài toán (1.17).

Là một biến thể của phương pháp tối ưu hóa xen kẽ dựa trên chuẩn ℓp, thuật toán chiếu xen kẽ (AP) . Khác với phương pháp tối ưu hóa xen kẽ tiêu chuẩn, phương pháp AP định dạng bài toán MC như một bài toán khả thi (feasibility problem). Cụ thể, nó định nghĩa hai tập hợp sau:

trong đó (1.18) là tập hợp các ma trận có hạng thấp và (1.19) là tập hợp ràng buộc độ trung thực (fidelity constraint). Hằng số r là hạng ước lượng của ma trận M, và là một tham số dung sai nhỏ được xác định bởi ma trận nhiễu. Thuật toán AP tìm ma trận M nằm trong giao của và thông qua phương pháp chiếu xen kẽ.

Cần lưu ý rằng, mặc dù thuật toán AP và ℓp-reg có khả năng cung cấp hiệu suất khôi phục vượt trội, cả hai đều yêu cầu biết trước hạng của ma trận M, điều này có thể không khả thi trong các ứng dụng thực tế. Ngoài ra, tham số nhiễu trong thuật toán AP được tính từ ma trận chỉ chứa nhiễu, điều này gây thêm chi phí trong hệ thống thực tế.

# **CHƯƠNG II : THUẬT TOÁN SVT HOÀN THÀNH MA TRẬN**

## **2.1 Giới thiệu**

### 2.1.1 Tác nhân

Hiện nay có sự quan tâm ngày càng tăng về việc khôi phục ma trận hạng thấp hoặc xấp xỉ hạng thấp không xác định từ thông tin rất hạn chế. Vấn đề này xuất hiện trong nhiều lĩnh vực kỹ thuật và khoa học ứng dụng như học máy, điều khiển và thị giác máy tính.

Một ví dụ điển hình là bài toán khôi phục ma trận dữ liệu từ một tập mẫu các phần tử của nó. Điều này thường xảy ra khi thu thập các khảo sát được điền một phần và muốn suy ra nhiều mục còn thiếu. Trong lĩnh vực hệ thống đề xuất như Netflix, người dùng đánh giá một tập con các mục trong cơ sở dữ liệu, và cần suy ra sở thích của họ cho các mục chưa được đánh giá.

Việc khôi phục ma trận từ một tập mẫu các phần tử được gọi là bài toán hoàn thiện ma trận (matrix completion). Vấn đề là bài toán này rất thiếu xác định, vì với số lượng mẫu ít hơn tổng số phần tử, có vô số cách hoàn thiện khả thi.

Tuy nhiên, trong nhiều trường hợp, ma trận cần khôi phục có hạng thấp hoặc xấp xỉ hạng thấp. Giả định này thay đổi căn bản bài toán, làm cho việc tìm kiếm giải pháp trở nên khả thi vì giải pháp có hạng thấp nhất thường là giải pháp đúng.

Candès và Recht đã chứng minh rằng hầu hết các ma trận hạng thấp có thể được khôi phục chính xác từ hầu hết các tập hợp phần tử được lấy mẫu, dù số lượng mẫu có thể rất nhỏ. Quan trọng hơn, điều này có thể thực hiện bằng cách giải một bài toán tối ưu hóa lồi đơn giản:

*Tối thiểu hóa*  *với điều kiện*

trong đó là chuẩn hạt nhân (nuclear norm) của ma trận X, tức là tổng các giá trị suy biến của nó.

Bài toán tối ưu hóa này là lồi và có thể được viết lại dưới dạng một chương trình nửa xác định.

### 2.1.2 Mô tả thuật toán

Việc tối thiểu hóa chuẩn hạt nhân có thể khôi phục ma trận có hạng thấp nhất và thường cho kết quả thực nghiệm tốt. Các phương pháp hiện có như SDPT3 sử dụng kỹ thuật điểm nội nhưng gặp vấn đề với ma trận kích thước lớn do phải giải các hệ phương trình tuyến tính khổng lồ.

Báo cáo này phát triển thuật toán ngưỡng giá trị suy biến (Singular Value Thresholding - SVT) để giải xấp xỉ bài toán tối thiểu hóa chuẩn hạt nhân cho bài toán dạng:

*Tối thiểu hóa với điều kiện*

Trong bối cảnh hoàn thiện ma trận, bài toán trở thành:

*Tối thiểu hóa* với điều kiện

Thuật toán SVT hoạt động như sau:

* Bắt đầu với = 0
* Lặp lại:
  + = shrink(, τ)
  + = + (M -)
* Cho đến khi đạt tiêu chí dừng

Trong đó, shrink(Y, τ) là hàm phi tuyến áp dụng ngưỡng mềm cho các giá trị suy biến của ma trận đầu vào. Thuật toán có hai ưu điểm quan trọng:

1. Tính thưa thớt: bằng không bên ngoài Ω nên rất thưa thớt, giúp đánh giá hàm shrink nhanh chóng.
2. Tính chất hạng thấp: Ma trận có hạng thấp, giúp giảm yêu cầu lưu trữ.

Thực nghiệm cho thấy thuật toán có thể giải các bài toán với ma trận kích thước 30.000×30.000 (gần một tỷ ẩn số) trong 17 phút trên máy tính thông thường.

### 2.1.3 Công thức tổng quát

Thuật toán SVT có thể được điều chỉnh cho các loại ràng buộc lồi khác, như các bài toán dạng:

*Tối thiểu hóa*  *với điều kiện* (X) ≤ 0, i = 1,...,m

Khi là các hàm affine, thuật toán tổng quát sẽ có dạng tương tự cho phép phát triển các thuật toán số hiệu quả để khôi phục ma trận từ dữ liệu có nhiễu.

### 2.1.4 Mối quan hệ và các công trình khác

Thuật toán của lấy cảm hứng từ các nghiên cứu gần đây trong tối thiểu hóa chuẩn ℓ1, đặc biệt là các vòng lặp Bregman tuyến tính hóa được áp dụng trong cảm biến nén. Ban đầu, các vòng lặp này được giới thiệu như một công cụ tính toán trong khoa học hình ảnh và sau đó được chứng minh là hiệu quả trong việc tìm nghiệm thưa thớt cho các hệ phương trình tuyến tính thiếu định. Nhiều cải tiến đã được đề xuất, chẳng hạn như kỹ thuật tăng tốc "kicking" hay vòng lặp Bregman phân tách, cùng với các phân tích lý thuyết về hội tụ và ứng dụng thực tiễn trong khử mờ và phục hồi ảnh.

Trong bối cảnh tối thiểu hóa chuẩn ℓ1, các vòng lặp Bregman tuyến tính hóa được thực hiện thông qua các bước ngưỡng mềm lặp lại, vốn là kỹ thuật quen thuộc trong xử lý tín hiệu và hình ảnh. Thuật toán SVT kế thừa và mở rộng cách tiếp cận này: thay vì tìm nghiệm thưa thớt trong một miền cố định, SVT tìm các giá trị suy biến thưa thớt đồng thời khám phá cơ sở biểu diễn thích hợp, mang tính thích ứng cao hơn. Theo nghĩa này, SVT là một phần mở rộng tự nhiên của các phương pháp ngưỡng mềm lặp lại truyền thống.

Cuối cùng, so sánh vòng lặp SVT với thuật toán ngưỡng mềm lặp lại phổ biến trong xử lý hình ảnh – điển hình là Phương pháp Tách Tiến-Lùi Gần đúng (PFBS). Bài toán tối thiểu hóa có ràng buộc ban đầu cũng có thể được nới lỏng thành một dạng rút gọn tương đương, giúp đơn giản hóa việc giải bài toán trong thực tế.

*Tối thiểu hóa* λ+

Cho một số λ > 0. Với là toán tử gần đúng của λ bởi phương trình điểm cố định cho mỗi δ > 0 để có được các vòng lặp có dạng

+))

Giới thiệu một ma trận trung gian Y^k, thuật toán này có thể được biểu diễn như :

Sự khác biệt ban đầu sẽ thay thế bằng và đặt δₖ = δvới τ = λδ nhưng có hệ quả to lớn vì điều này tạo ra các thuật toán hoàn toàn khác nhau. Thứ nhất, chúng có các giới hạn khác nhau. Thứ hai, chọn một λ lớn (hoặc một giá trị lớn của τ = λδ) trong cho một chuỗi các lặp hạng thấp và một giới hạn với chuẩn hạt nhân nhỏ. Tuy nhiên, giới hạn này không phù hợp với dữ liệu và đây là lý do tại sao người ta phải chọn một giá trị nhỏ hoặc vừa phải của λ (hoặc của τ = λδ). Tuy nhiên, khi λ không đủ lớn, có thể không có hạng thấp ngay cả khi nghiệm có hạng thấp (và người ta có thể cần tính toán nhiều vector suy biến), và không đủ thưa thớt để làm cho thuật toán hấp dẫn về mặt tính toán.

## **2.2 Thuật toán ngưỡng giá trị kỳ dị (SVT)**

### 2.2.1 Toán tử thu hẹp giá trị đơn (kỳ dị)

Xét phân tích phân rã giá trị đơn (SVD) của ma trận X ∈ có hạng là r

\*, ∑ = diag({σi}1 ≤ i ≤ r), (2.1)

Trong đó, U và V lần lượt là các ma trận có kích thước n1 × r và n2 × r với các cột trực chuẩn, và các giá trị đơn σi là dương (trừ khi được chỉ định khác, chúng ta sẽ luôn giả định rằng SVD của một ma trận được đưa ra dưới dạng rút gọn như trên). Đối với mỗi τ ≥ 0, chúng ta giới thiệu toán tử ngưỡng mềm Dτ được định nghĩa như sau:

Dτ(X)V\* ,

Dτ(∑) = diag({σi − τ )+}), (2.2)

Trong đó, t+là phần dương của t, cụ thể t+ = max(0,t). Nói cách khác, toán tử này chỉ đơn giản áp dụng quy tắc ngưỡng mềm lên các giá trị đơn của X, thực chất giảm chúng về không. Đó là lý do tại sao chugns ta sẽ gọi sự biến đổi này là toán tử thu hẹp giá trị đơn. Mặc dù SVD có thể không duy nhát, nhưng dễ thấy rằng toán tử thu hẹp giá trị đơn được định nghĩa tốt và chúng ta sẽ không giải thích thêm về vấn đề này. Theo một cách khác, toán tử thu hẹp này là một mở rộng đơn giản của quy tắc ngưỡng mềm cho các số vô hướng và vector. Đặc biệt, lưu ý rằng nếu nhiều giá trị đơn của X nằm dưới ngưỡng τ, hạng của Dτ(X) có thể thấp hơn đáng kể so với hạng của X, tương tự như quy tắc ngưỡng mềm áp dụng cho các vectơ sẽ dẫn đến các đầu ra thưa hơn khi một số phần tử của đầu vào nằm dưới ngưỡng.

Toán tử ngưỡng giá trị đơn là toán tử cận (proximity operator) liên kết với chuẩn hạt nhân (nuclear norm).

Định lý 2.1: Với mỗi τ ≥ 0 và Y ∈ , toán tử thu hẹp giá trị đơn được áp dụng như sau:

(2.3)

Minh chứng: Vì hàm

là hàm lồi nghiêm ngặt, dễ thấy rằng tồn tại một giá trị cực tiểu duy nhất, và chúng ta cần chứng minh nó bằng với Dτ(Y). Để làm điều này, hãy nhớ lại định nghĩa của một đạo hàm dưới của hàm lồi : → . Chúng ta nói rằng Z là đạo hàm dưới của f tại ,được ký hiệu Z ∈ ∂, nếu

Với mọi X, giờ đây tối thiểu hoá h0 nếu và chỉ nếu 0 là một đạo hàm dưới của hàm h0 tại điểm

0 ∈ − Y + τ∂ (2.5)

Khi đó, ∂ là tập hợp các đạo hàm dưới của chuẩn hạt nhân. Giả sử X ∈ là một ma trận bất kỳ và UΣV\* là phân tích giá trị đơn (SVD). Đã được biết như sau:

∂ (2.6)

Giả sử := Dτ(Y) để ngắn gọn. Để chứng minh rằng tuân theo (2.5), phân tích SVD của Y như sau:

Y =U0Σ0 + U1Σ1

Trong đó U0, V0 (tương ứng U1, V1) là các vector kỳ dị liên kết với các giá trị kỳ dị lớn hơn τ (tương ứng là nhỏ hơn hoặc bằng τ), Với các ký hiệu này, ta có:

= U0(Σ0 – τI)

Và do đó:

Y – = τ(*U0*, W = τ-1 U1Σ1

Theo định nghĩa, = 0, WV0 = 0 và do các phần tử đường chéo của Σ1 có độ lớn bị giới hạn bởi τ, chứng ta cũng có ||W||2 ≤ 1. Do đó Y – ∈ τ∂, điều này đã được chứng minh

### 2.2.2 Bước lặp thu nhỏ

Chúng ta hiện đã sẵn sàng giới thiệu thuật toán ngưỡng giá trị kỳ dị. Đặt τ > 0 và một dãy {δk} của các bước kích thước dương. Bắt đầu với Y0, lầnlượt xác định cho k = 1, 2, …

(2.7)

Cho đến khi một điều kiện dừng được thoả mãn. Quá trình lặp rút gọn này rất dễ triển khai. Ở mỗi bước, chúng ta chỉ cần tính SVD và thực hiện các phép toán ma trận cơ bản. Với sự trợ giúp của đại số tuyến tính tiêu chuẩn, toàn bộ thuật toán có thể được mã hóa chỉ trong vài dòng.

Trước khi giải quyết các vấn đề tính toán sâu hơn, chúng ta sẽ làm rõ mối quan hệ giữa vòng lặp này và bài toán ban đầu (1.1). Trong Mục 4, sẽ chứng minh rằng chuỗi {Xk} hội tụ đến nghiệm duy nhất của một bài toán tối ưu hóa có liên quan chặt chẽ đến (1.1), cụ thể là

Có thể trực tiếp thấy rằng nghiệm của bài toán đã được sửa đổi này hội tụ về nghiệm của (1.5) khi τ. Do đó, bằng cách chọn một giá trị lớn cho tham số τ, chuỗi lặp hội tụ về một ma trận gần như tối thiểu hóa bài toán (1.1).

Có hai tính chất quan trọng khiến thuật toán này đặc biệt phù hợp cho bài toán hoàn tất ma trận:

1. Tính chất hạng thấp: Một thực tế thực nghiệm đáng chú ý là các ma trận trong chuỗi {Xk} có hạng thấp (dĩ nhiên, với điều kiện rằng nghiệm của phương trình (2.8) cũng có hạng thấp). Sử dụng từ “thực nghiệm” bởi vì tất cả các thử nghiệm số đều cho ra các chuỗi có hạng thấp, nhưng không thể chứng minh chặt chẽ rằng điều này luôn đúng trong mọi trường hợp.

Khi hạng của nghiệm nhỏ hơn đáng kể so với bất kỳ chiều nào của ma trận, yêu cầu lưu trữ cũng thấp, vì ta có thể lưu trữ từng Xk dưới dạng phân tích giá trị kỳ dị (SVD). Lưu ý rằng chúng ta chỉ cần giữ lại giá trị hiện tại của vòng lặp và có thể loại bỏ các giá trị trước đó.

2. Tính chất thưa (Sparsity): Một đặc điểm quan trọng khác của thuật toán SVT là ma trận vòng lặp Yk có tính chất thưa. Vì Y0 = 0, theo nguyên lý quy nạp, ta có Yk triệt tiêu bên ngoài tập Ω. Số lượng phần tử khả dụng càng ít thì Yk càng thưa. Do mô hình thưa Ω được cố định trong suốt quá trình, ta có thể áp dụng các kỹ thuật ma trận thưa để tiết kiệm bộ nhớ.

Ngoài ra, nếu ∣Ω∣=m, chi phí tính toán để cập nhật Yk là bậc m. Hơn nữa, ta có thể gọi các thủ tục con hỗ trợ tính toán ma trận thưa, giúp giảm thêm chi phí tính toán.

Một thủ tục con quan trọng là phân tích giá trị kỳ dị (SVD). Tuy nhiên, ta không cần tính toàn bộ SVD của Yk để áp dụng toán tử ngưỡng giá trị kỳ dị. Chỉ cần tính phần liên quan đến các giá trị kỳ dị lớn hơn τ. Do đó, một chiến lược hiệu quả là sử dụng thuật toán lặp Lanczos để tính toán một vài giá trị kỳ dị và vector kỳ dị đầu tiên. Vì Yk là ma trận thưa, ta có thể áp dụng nó vào các vector tùy ý một cách nhanh chóng, và quá trình này mang lại tốc độ xử lý đáng kể so với các phương pháp thông thường.

### 2.2.3 Mối quan hệ với các nghiên cứu khác

Thuật toán lấy cảm hứng từ các nghiên cứu gần đây trong tối thiểu hóa chuẩn ℓ1, đặc biệt là các vòng lặp Bregman tuyến tính hóa được áp dụng trong cảm biến nén. Ban đầu, các vòng lặp này được giới thiệu như một công cụ tính toán trong khoa học hình ảnh và sau đó được chứng minh là hiệu quả trong việc tìm nghiệm thưa thớt cho các hệ phương trình tuyến tính thiếu định. Nhiều cải tiến đã được đề xuất, chẳng hạn như kỹ thuật tăng tốc "kicking" hay vòng lặp Bregman phân tách, cùng với các phân tích lý thuyết về hội tụ và ứng dụng thực tiễn trong khử mờ và phục hồi ảnh.

Trong bối cảnh tối thiểu hóa chuẩn ℓ1, các vòng lặp Bregman tuyến tính hóa được thực hiện thông qua các bước ngưỡng mềm lặp lại, vốn là kỹ thuật quen thuộc trong xử lý tín hiệu và hình ảnh. Thuật toán SVT kế thừa và mở rộng cách tiếp cận này: thay vì tìm nghiệm thưa thớt trong một miền cố định, SVT tìm các giá trị suy biến thưa thớt đồng thời khám phá cơ sở biểu diễn thích hợp, mang tính thích ứng cao hơn. Theo nghĩa này, SVT là một phần mở rộng tự nhiên của các phương pháp ngưỡng mềm lặp lại truyền thống.

Cuối cùng, so sánh vòng lặp SVT với thuật toán ngưỡng mềm lặp lại phổ biến trong xử lý hình ảnh – điển hình là Phương pháp Tách Tiến-Lùi Gần đúng (PFBS). Bài toán tối thiểu hóa có ràng buộc ban đầu cũng có thể được nới lỏng thành một dạng rút gọn tương đương, giúp đơn giản hóa việc giải bài toán trong thực tế.

*minimize* λ+ (2.9)

Cho một số λ > 0. Với là toán tử gần đúng của λ bởi phương trình điểm cố định cho mỗi δ > 0 để có được các vòng lặp có dạng

+))

Giới thiệu một ma trận trung gian Yk, thuật toán này có thể được biểu diễn như

Sự khác biệt ban đầu sẽ thay thế bằng và đặt δₖ = δ với τ = λδ nhưng có hệ quả to lớn vì điều này tạo ra các thuật toán hoàn toàn khác nhau. Thứ nhất, chúng có các giới hạn khác nhau. Thứ hai, chọn một λ lớn (hoặc một giá trị lớn của τ = λδ) trong cho một chuỗi các lặp hạng thấp và một giới hạn với chuẩn hạt nhân nhỏ. Tuy nhiên, giới hạn này không phù hợp với dữ liệu và đây là lý do tại sao người ta phải chọn một giá trị nhỏ hoặc vừa phải của λ (hoặc của τ = λδ). Tuy nhiên, khi λ không đủ lớn, có thể không có hạng thấp ngay cả khi nghiệm có hạng thấp (và người ta có thể cần tính toán nhiều vector suy biến), và không đủ thưa thớt để làm cho thuật toán hấp dẫn về mặt tính toán.

### 2.2.4 Diễn giải theo phương pháp nhân tử Lagrange

Thuật toán nhân tử Lagrange hay còn được gọi là thuật toán Uzawa. Một hệ quả quan trọng của cách tiếp cận này là nó cho phép mở rộng thuật toán SVT sang các bài toán khác liên quan đến việc tối thiểu hóa chuẩn hạt nhân dưới các ràng buộc lồi.

Tiếp theo, đặt fτ(X) = τ + với τ > 0

Minimize fτ(X)

subject to Ω(X) = Ω(M)

Phương trình Lagrange trong bài toán này được cho bởi

Khi đó với Y ∈ n1 x n2. Tính chất đối ngẫu mạnh được đảm bảo, và X\* cùng Y\* là nghiệm tối ưu sơ - đối ngẫu nếu (X\*,Y)là điểm yên của hàm Lagrange , tức là một cặp thỏa mãn điều kiện:

(2.11)

Thuật toán tiếp cận Uzawa vấn đề tìm điểm yên bằng một quy trình lặp. Bắt đầu từ Y0 = 0, sau đó, định nghĩa theo cách quy nạp

(2.12)

Trong đó {δk}k ≥ 1 là một chuỗi dương tăng. Thuật toán Uzawa thực chất là một phương pháp dưới gradient được áp dụng cho bài toán đối ngẫu, trong đó mỗi bước di chuyển điểm lặp hiện tại theo hướng của gradient hoặc của một dưới-gradient. Ta thấy

(2.13)

Trong đó, là điểm cực tiểu của hàm Lagrange tại giá trị Y, do đó, một bước cập nhật theo phương pháp hạ gradient cho Y có dạng

Tiếp tục tính toán điểm cực tiểu của hàm Lagrange (2.12), lưu ý

(2.14)

Tuy nhiên, chúng ta biết rằng điểm cực tiểu được cho bởi Dτ((Y)) và vì

Yk = (Yk) với mọi k ≥ 0, thuật toán Uzawa có dạng:

Đây chính là công thức (2.7). Quan điểm này tận dụng nhiều công cụ toán học khác nhau để chứng minh sự hội tụ của các bước lặp ngưỡng giá trị kỳ dị. Đối với một ứng dụng sớm của thuật toán Uzawa trong việc tối thiểu hóa một hàm ℓ1, cụ thể là chuẩn biến thiên toàn phần, dưới các ràng buộc bất đẳng thức tuyến tính.

### 2.2.5 Phân tích hội tụ

a) Hội tụ trong bài toán hoàn thiện ma trận

Chúng ta bắt đầu bằng việc ghi nhận một bồ đề thiết lập tính lồi mạnh của hàm mục tiêu *f*τ

Bồ đề 1: Cho Z ∈ ∂f(X) và Z’ ∈ ∂fτ (X’). Khi đó

⟨Z − Z , X − X ⟩ ≥

Chứng Minh. Một phần tử Z của ∂f(X) có dạng Z = Z0 + X, trong đó Z0 ∈ ∂ ||X||\*, tương tự với Z’, điều này dẫn đến

⟨Z – Z’ , X – X’ ⟩ = τ⟨Z0 – Z0’, X – X’⟩ +

Do đó, chỉ cần chứng minh rằng số hạng đầu tiên ở vế phải là không âm. Từ 2.2.2, ta có rằng bất kỳ đạo hàm dưới nào của chuẩn hạt nhân tại X đều tuân theo ||Z0||2 1 và

⟨*Z*0*, X*⟩ =  *X *∗. Đặc biệt, điều này dẫn đến  
 |⟨𝐙₀, 𝐗′⟩| ≤ ‖𝐙₀‖₂‖𝐗′‖\* ≤ ‖𝐗′‖\*, |⟨𝐙′₀, 𝐗⟩| ≤ ‖𝐙′₀‖₂‖𝐗‖\* ≤ ‖𝐗‖\*

Từ đó

⟨Z₀ − Z′₀, X − X′⟩ = ⟨Z₀, X⟩ + ⟨Z′₀, X′⟩ − ⟨Z₀, X′⟩ − ⟨Z′₀, X⟩

= ‖X‖\* + ‖X′‖\* − ⟨Z₀, X′⟩ − ⟨Z′₀, X⟩ ≥ 0

Điều này chứng minh bồ đề và bồ đề này là chìa khoá quan trọng trong việc chỉ ra rằng thuật toán SVT này hội tụ

Định lý 4.2: Giả sử rằng chuỗi các bước kích thước tuân theo điều kiện

0 < inf δk ≤ sup δk < 2. Khi đó, chuỗi {Xk} được xác định sẽ hội tụ với nghiệm duy nhất

Chứng minh: Giả sử (X\*, Y\*) là cặp tối ưu nguyên-thứ đối với bài toán, điều kiện tối ưu cho ta:

0 = Zk − PΩ (Yk−1)

0 = Z\*− PΩ(Y\*),

Với một số Zk ∈ ∂fτ(Xk) và Z\* ∈ ∂fτ(X\*). Từ đây suy ra

(Zk −Z\*) − PΩ(Yk−1 − Y\*) =0

Do đó, theo bồ đề 4.1:

Tiếp tục phân tích, ta thấy rằng vì PΩX\* = PΩM, ta có:

Đặt rk = || PΩ(Yk −Y\*) ||F,

Do với bất kỳ ma trận X, ta có ||PΩ(X)||F ≤ ||X||F. Dựa trên giả thiết về kích thước của δk, ta có 2δk − ≥ β ∀ k≥1 với một số β > 0. Do đó

Từ đây ta rút ra 2 thuộc tính chất:

1. Dãy số {||PΩ(Yk − Y\*)||F} không tăng và do đó hội tụ đến một giới hạn.
2. Hệ quả là và k

b) Định lí hội tụ tổng quát

Tổng quát hơn, kết quả thứ hai thiết lập sự hội tụ của thuật toán SVT đến nghiệm của bài toán (3.4) khi có các ràng buộc lồi tổng quát. Giải thiết được sử dụng là hàm (X) Lipschitz theo nghĩa:

Với một hằng số không âm L(F). Lưu ý rằng nếu F là ánh xạ affine,

tức là (X) = b − A(X), ta có L(F)=||A||2,

trong đó A2 là chuẩn phổ của ánh xạ tuyến tính A được xác định bởi:

||A||2 := sup{||A(X)||ℓ2 : ||X||F = 1}

Ngoài ra, ta cũng nhớ rằng

(X) =(f1(X),...,fm(X)), trong đó mỗi fi là hàm lồi, và Lagrangian cho bài toán được xác định bởi

Giả định để đơn giản hóa rằng tính đối ngẫu mạnh được thỏa mãn, điều này tự động đúng nếu các ràng buộc tuân theo các điều kiện đủ. Trước tiên, thiết lập bổ đề chuẩn bị sau.

Bồ đề 4.3 : Cho (X\*,y\*) là một cặp tối ưu nguyên-đôi cho phương trình (3.4). Khi đó với mỗi δ > 0, y\* thảo mãn

y\* =[y\* +δF(X\*)]+.

Chứng minh. Nhớ lại rằng phép chiếu x0 của một điểm x lên một tập hợp lồi C được đặc trưng bởi

Trong trường hợp C = = {x ∈ m : x ≥ 0} điều kiện này trở thành x0 ≥ 0 và

(y – x0,x−x0) ≤ 0, ∀ y ≥ 0

Vì y\* là tối ưu kép nên ta có

(X\*,y\*) ≥ (X\*,y), ∀ y ≥ 0

Thay thế biểu thức của Lagrangian, điều này tương đương với:

Tương đương:

Từ đó ta có thể suy ra rằng phải là phép chiếu của lên góc phần tư không âm . Vì phép chiếu của một vector bất kỳ x lên được cho bởi x+, vì thế kết luận của chúng ta đã được chứng minh

Đinh lý 4.4 Giả sử dãy kích thước bước thoả mãn điều kiện

0 < inf δk sup δk < 2/||*L*()||2

Trong đó *L*( là hằng số Lipschitz trong (4.5). Khi giả định có tính đỗi ngẫu mạnh, thì dãy {Xk}thu được từ (3.5) hội tụ về nghiệm duy nhất của (3.4)

Chứng minh:

Gọi (X\*, y\*) là nghiệm tối ưu sơ – đối ngẫu của bài toán (3.4). Chúng ta khẳng định rằng các điều kiện tối ưu cho thấy rằng với mọi X

Với một số Zk ∈ ∂fτ(Xk) và một số Z\* ∂fτ(X\*). Chúng ta chứng minh khẳng định này bằng cách chứng minh một trong hai bất đẳng thức, vì bất đẳng thức còn lại có cấu trúc tương tự. Đầu tiên Xk tối thiểu hoá (X,yk-1) đối với mọi X, do đó tồn tại Zk ∈ ∂fτ(Xk) và ∈ ∂fi(Xk), 1 i m, sao cho

Bởi vì fi là một hàm lồi

fi(X) – fi(Xk) ≥

Do đó

Bây giờ viết bất đẳng thức thứ nhất trong (4.7) cho X\* bất đẳng thức thứ hai cho Xk và cộng hai bất đẳng thức lại. Điều này dẫn đến

Từ bồ đề 4.1 suy ra rằng

(4.8)

Ta dễ dàng nhận thấy vì y\* = [y\* +δk(X)]+ theo bồ đề 4.3, ta có:

Vì phép chiếu lên tập lồi là một phép co, do đó

Ở đây chúng ta đã sử dụng L thay cho L() để viết ngắn gọn. Dưới các giả định về kích thước của δk, ta có 2δk − L2 ≥ β với mọi k ≥ 1và một số β > 0. Khi đó

(4.9)

Bài toán (3.1) với các ràng buộc tuyến tính có thể được giảm về (3.4) bằng cách chọn

Từ đó ta có hệ quả sau:

Hệ quả 4.5 Giả sử dãy kích thước bước thoả mãn điều kiện

0 < inf δk ≤ sup δk < 2/.

Khi đó dãy {Xk} thu được từ (3.3) hội tụ về nghiệm duy nhất của (3.1)

Gọi với được cho như trên, ta có |L()|2 = 2 và do đó định lý 4.4 đảm bảo hội tụ miễn là 0 < inf δk ≤ sup δk < 1/.

## **2.3 Kết quả mô phỏng**

### 2.3.1 Triển khai thuật toán

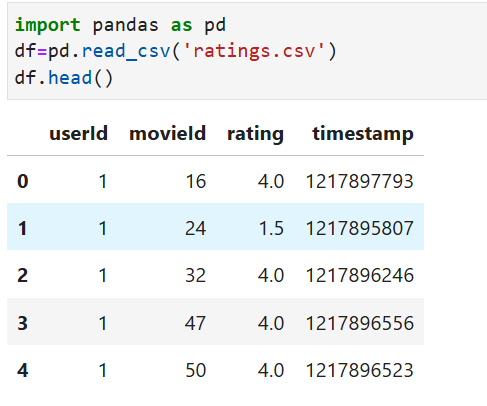
1 Dữ liệu đầu vào

1.1 Dữ liệu **MovieLens Rating** là một phần của bộ dữ liệu MovieLens — một tập dữ liệu phổ biến trong lĩnh vực hệ thống gợi ý (recommender systems). Dưới đây là mô tả chi tiết của bảng **ratings.csv** trong các phiên bản phổ biến của MovieLens

**- Thành phần chính của bảng** ratings.csv

| Cột | Ý nghĩa |
| --- | --- |
| userId | ID người dùng (người xem, người đánh giá phim) |
| movieId | ID phim (liên kết với bảng movies.csv) |
| rating | Điểm đánh giá phim, thường từ 0.5 đến 5.0 (theo bước 0.5) |
| timestamp | Dấu thời gian Unix (thời điểm người dùng đánh giá phim) |

- **Ví dụ dữ liệu**



- Giải thích timestamp

Là số giây tính từ **1970-01-01 00:00:00 UTC** (thời gian Unix).

Có thể chuyển thành định dạng ngày giờ để phân tích hành vi theo thời gian.

- Một số thông tin thống kê cơ bản

Số người dùng: hàng ngàn đến hàng trăm ngàn (tùy bộ 100k, 1M, 10M, 20M,…)

Số phim: tương ứng với hàng ngàn phim

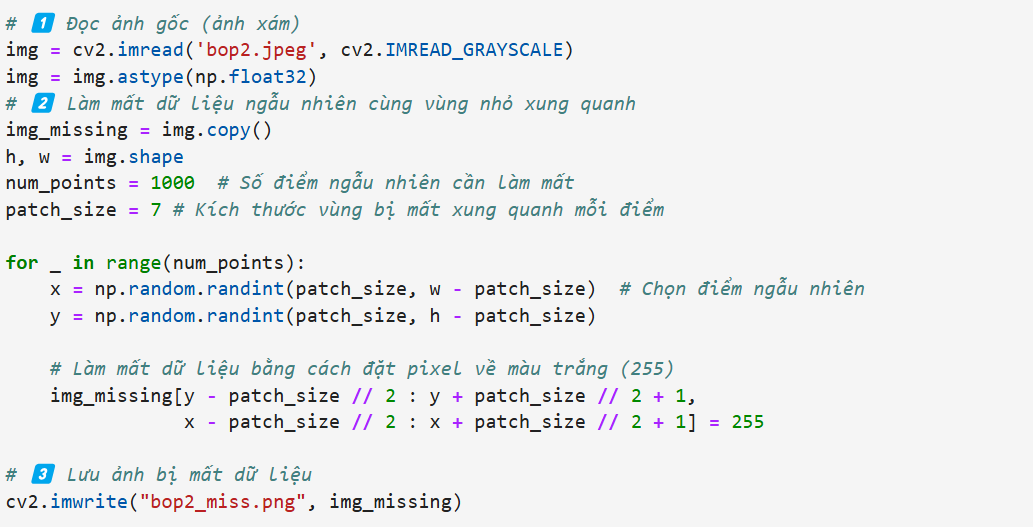
Số dòng đánh giá: bằng số lần người dùng đánh giá (userId + movieId là duy nhất)

- Các phiên bản phổ biến

| Tên bộ dữ liệu | Số ratings | Dung lượng |
| --- | --- | --- |
| MovieLens 100K | ~100,000 | ~5MB |
| MovieLens 1M | ~1,000,000 | ~6MB |
| MovieLens 10M | ~10,000,000 | ~70MB |
| MovieLens 20M | ~20,000,000 | ~200MB |

Ở đây chúng ta sử dụng dữ liệu MovieLens 100K.

1.2 Tạo ra 1 số ảnh bị mất pixel

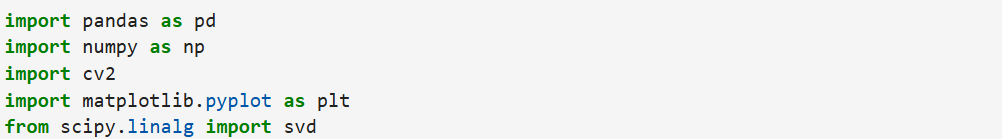


1 số hình ảnh bị làm mất pixel

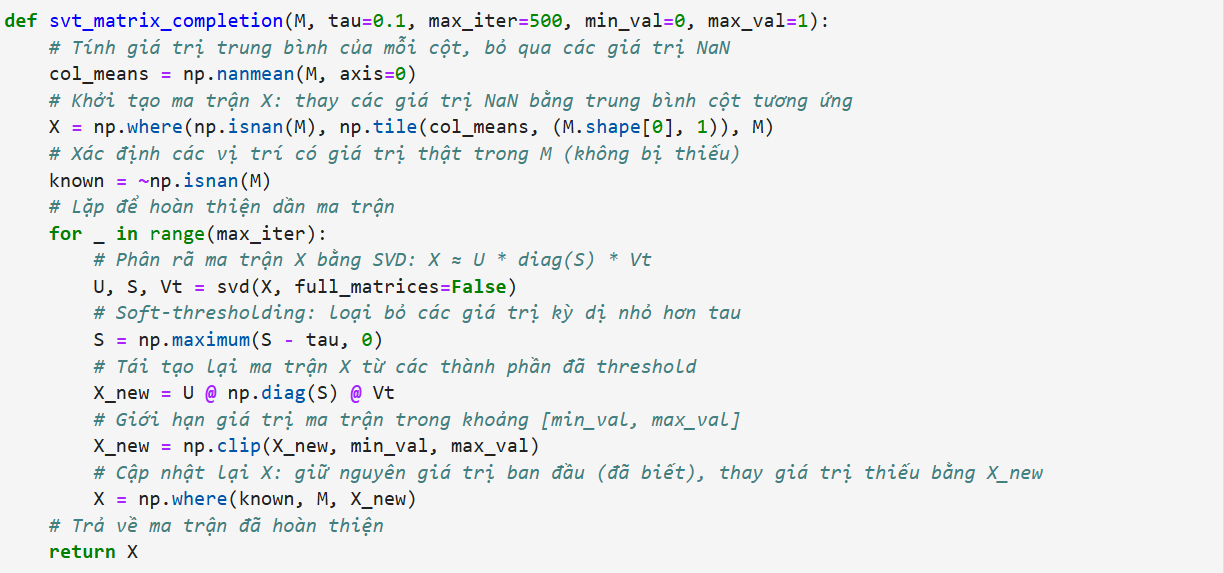
|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

2 Thuật toán SVT

- Khai báo 1 số thư viện cần sử dụng



- Thuật toán SVT



Trong đó :

M: ma trận đầu vào có thể chứa giá trị NaN (giá trị bị thiếu).

tau: ngưỡng dùng để lọc bớt các giá trị kỳ dị nhỏ (singular values) trong phân tích SVD.

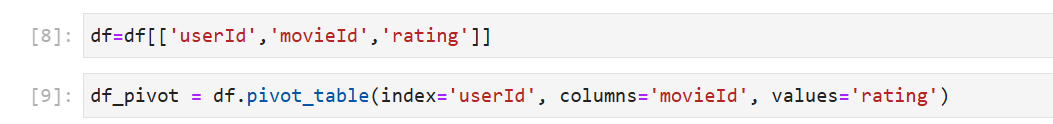
max\_iter: số vòng lặp tối đa.

min\_val, max\_val: dùng để ép giới hạn giá trị trong ma trận kết quả.

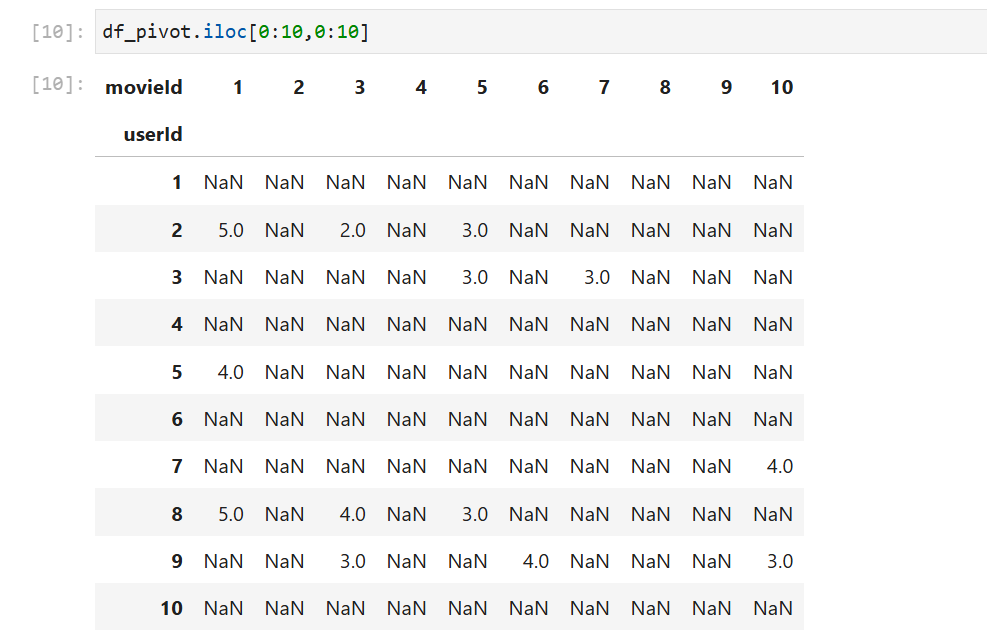
2.3.2 Áp dụng vào thuật toán

1 Dữ liệu Movie Lens

- Đưa dữ liệu về dạng ma trận

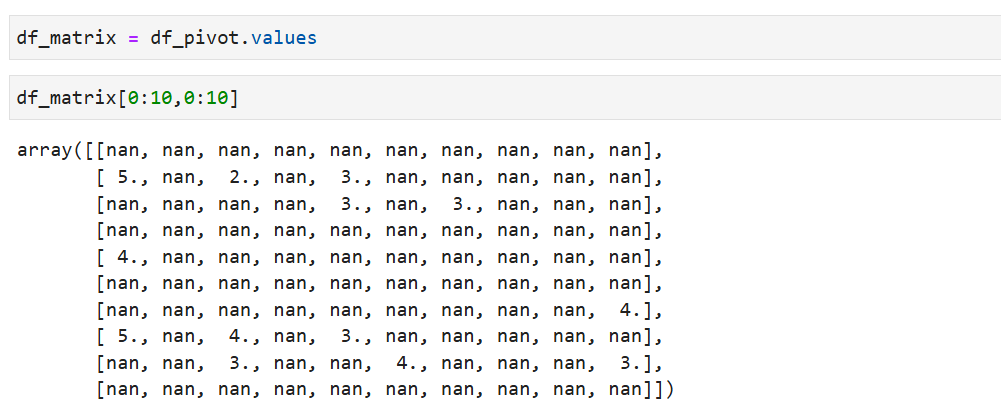


- Dữ liệu Movie Lens sau khi được đưa về dạng ma trận :



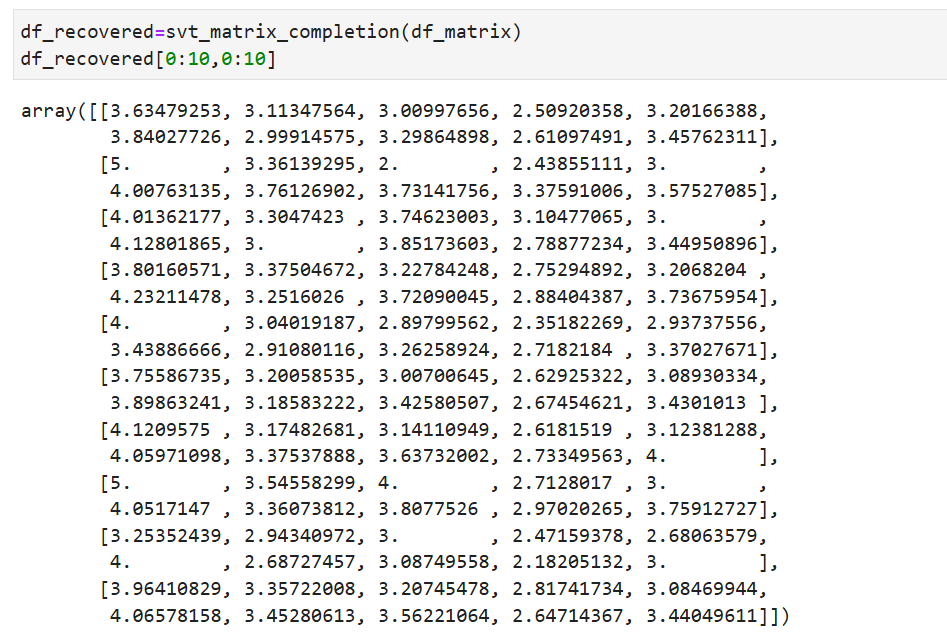
(1 phần con của dữ liệu)

- Đưa dữ liệu về dạng mảng 2 chiều đồng thời là dữ liệu đầu vào



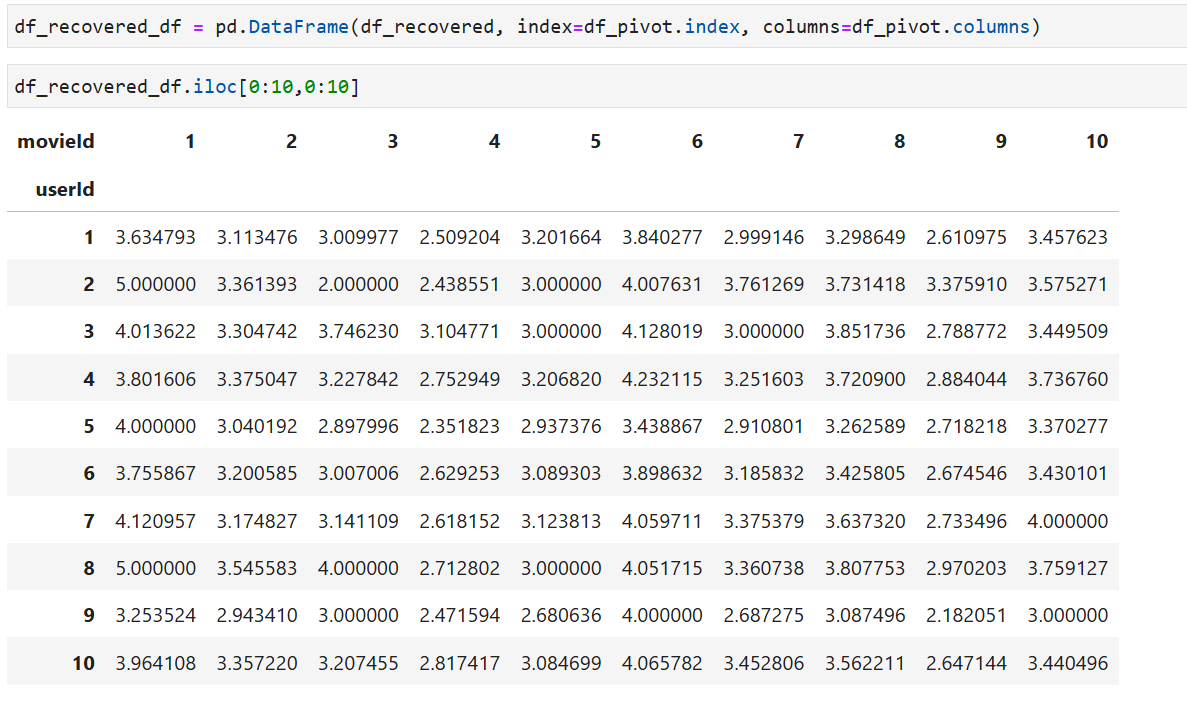
(1 phần con của dữ liệu)

- Kết quả đầu ra



(1 phần của kết quả)

- Đưa kết quả về dạng Data Frame



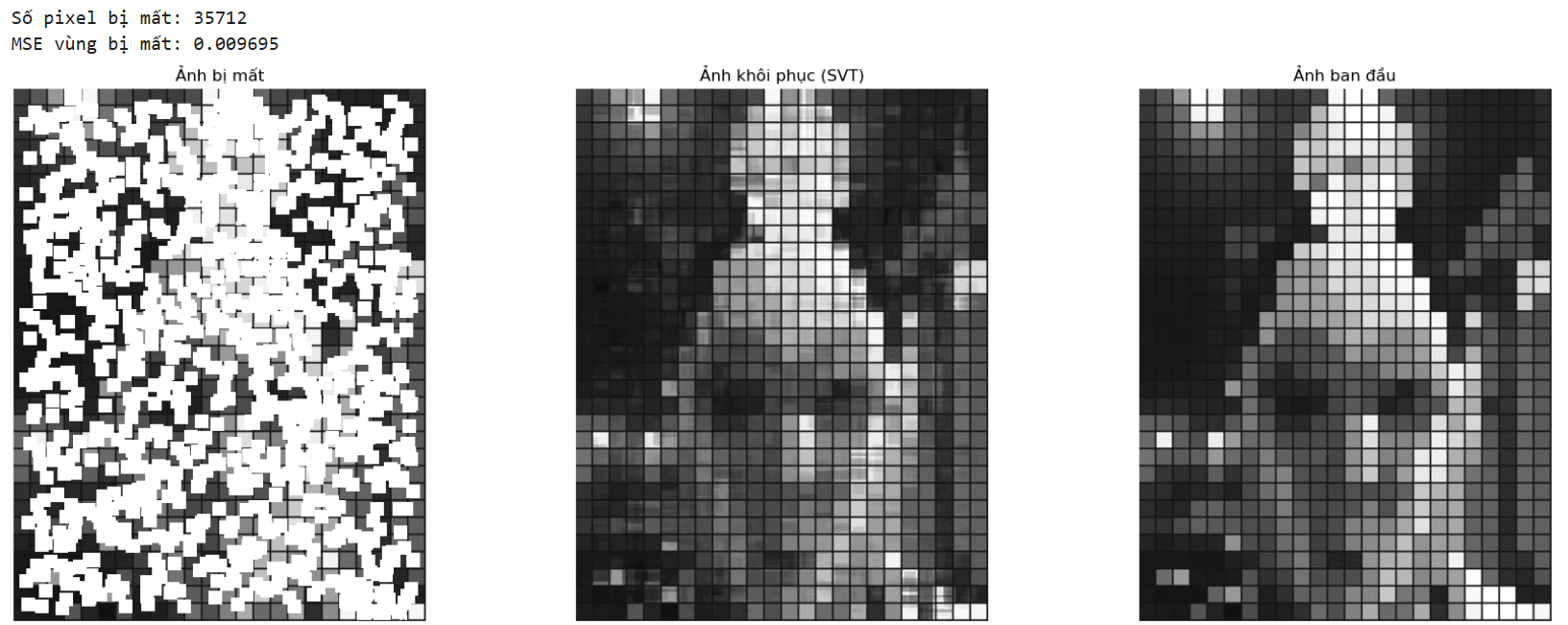
(1 phần của kết quả)

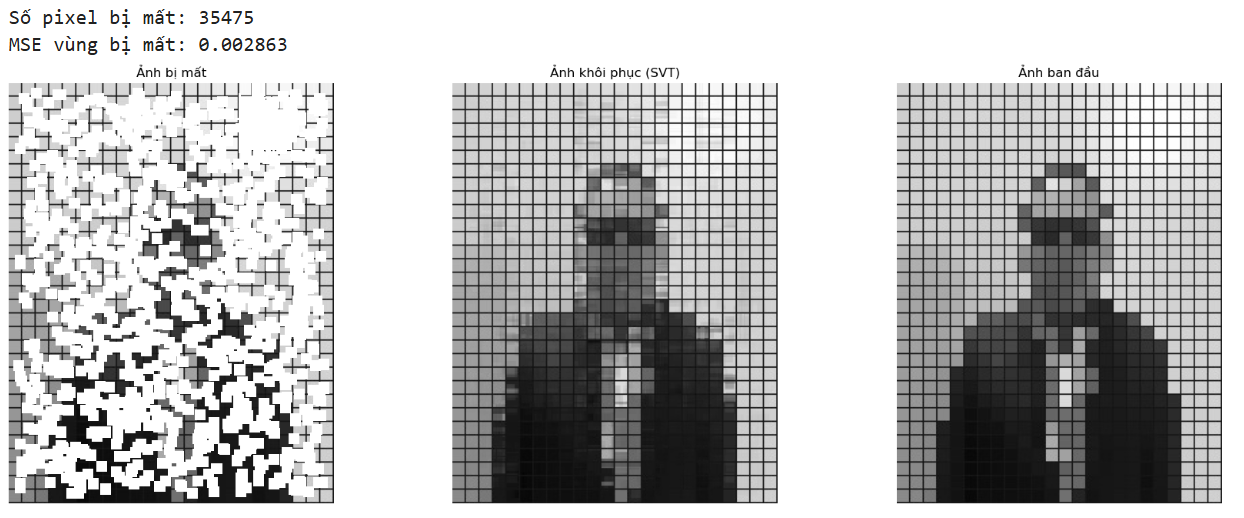
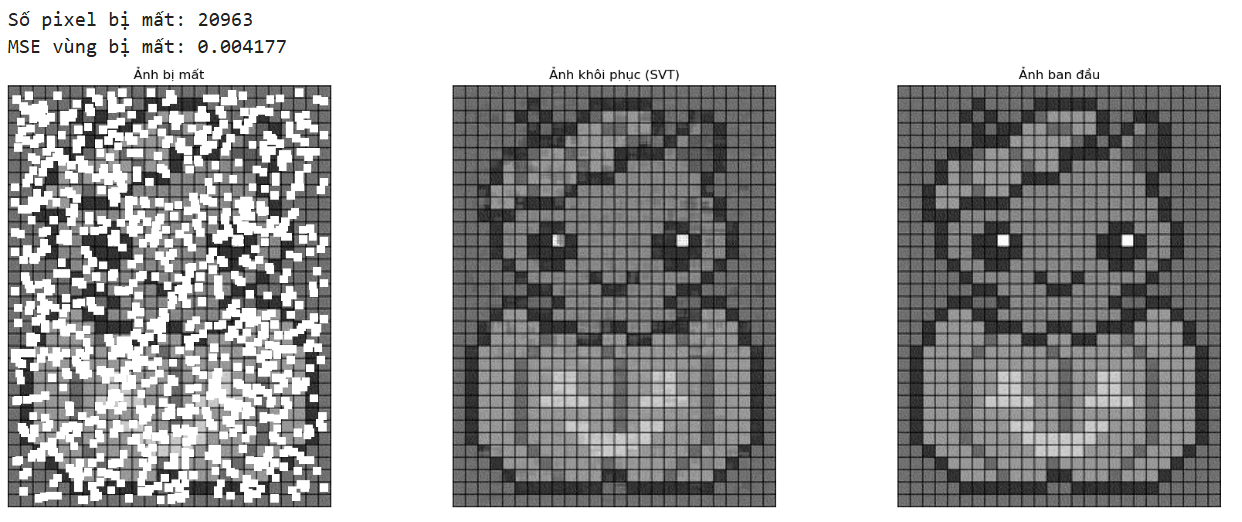
2 Dữ liệu ảnh

- Dữ liệu đầu vào

|  |  |  |
| --- | --- | --- |
|  |  |  |

- Kết quả đầu ra





Trong đó :

MSE là sai số bình phương trung bình

mse\_missing = np.mean((img[mask] - img\_recovered[mask])\*\*2)

ở đây mse\_missing tính **Sai số bình phương trung bình (MSE)** chỉ trên **các vùng ảnh bị mất dữ liệu** – tức là **nơi chứa NaN trong ảnh đầu vào**.

# **KẾT LUẬN**

Đề tài **“Tìm hiểu về bài toán khôi phục dữ liệu và ứng dụng”** đã thực hiện được một số nội dung chính sau:

* **Áp dụng thuật toán SVT (Singular Value Thresholding)** để khôi phục ảnh có vùng bị mất dữ liệu. Kết quả cho thấy ảnh đã được tái tạo lại khá tốt, với giá trị sai số MSE tại các vùng bị mất ở mức thấp, chứng minh tính hiệu quả của phương pháp.
* **Ứng dụng thuật toán SVT cho bài toán hoàn thiện ma trận đánh giá trong tập dữ liệu MovieLens**. Các giá trị xếp hạng (rating) bị thiếu đã được dự đoán lại dựa trên cấu trúc hạng thấp của ma trận gốc, giúp cải thiện khả năng gợi ý phim cho người dùng.
* **Thực nghiệm trên cả dữ liệu hình ảnh và dữ liệu bảng** (movie ratings) cho thấy thuật toán SVT có thể ứng dụng linh hoạt trong nhiều lĩnh vực khác nhau như thị giác máy tính, hệ thống gợi ý và khai phá dữ liệu.

# **Kiến nghị**

Từ những kết quả đạt được trong quá trình nghiên cứu và thực nghiệm đề tài “Tìm hiểu về bài toán khôi phục dữ liệu và ứng dụng”, nhóm em xin đề xuất một số kiến nghị như sau:

* Mở rộng nghiên cứu với các biến thể nâng cao của thuật toán SVT, cũng như so sánh với các phương pháp phục hồi dữ liệu khác (như Soft-Impute, OptSpace, Deep Matrix Factorization...) nhằm tìm ra hướng tiếp cận tối ưu hơn cho từng loại dữ liệu cụ thể.
* Ứng dụng vào thực tế bằng cách tích hợp thuật toán vào các hệ thống phần mềm có chức năng phục hồi dữ liệu ảnh, âm thanh, hoặc hệ thống gợi ý, đặc biệt trong các lĩnh vực như y tế, giáo dục và giải trí.
* Nâng cao hiệu quả xử lý thông qua việc kết hợp SVT với các kỹ thuật học sâu (deep learning) hoặc học máy (machine learning) để cải thiện khả năng tổng quát và hiệu suất trên dữ liệu lớn.

# **Tài liệu tham khảo**

1. Cai, J.-F., Candès, E. J., & Shen, Z. (2008). *A Singular Value Thresholding Algorithm for Matrix Completion*. arXiv:0810.3286v1 [math.OC].

2. Zhang, Y., Zhao, L., & Zhang, M. (2020). *A Survey on Matrix Completion*. arXiv:1901.10885v3 [cs.LG].

3. Candès, E. J., & Recht, B. (2009). *Exact Matrix Completion via Convex Optimization*. Foundations of Computational Mathematics, 9(6), 717–772.

4. Keshavan, R. H., Montanari, A., & Oh, S. (2010). *Matrix Completion from a Few Entries*. IEEE Transactions on Information Theory, 56(6), 2980–2998.