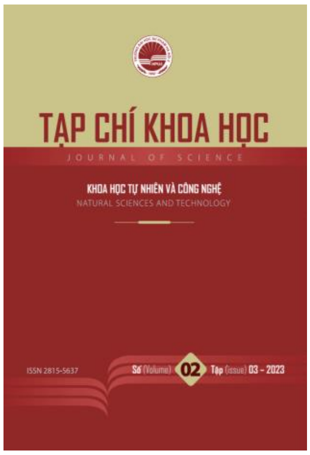
HPU2. Nat. Sci. Tech. Vol 02, issue 03 (2023), 34-41

HPU2 Journal of Sciences:
Natural Sciences and Technology

journal homepage: https://sj.hpu2.edu.vn



Article type: Research article

Singular value decomposition and applications in data processing and artificial intelligence

Thanh-Xuan Cao Thi*

University of Economics - Technology for Industries (UNETI), Ha Noi, Viet Nam

Abstract

Computing matrices is a crucial and prevalent topic in the field of data processing and artificial intelligence. One of the significant and effective methods for matrix manipulation is Singular Value Decomposition (SVD). Based on matrix computations using SVD, we can perform various complex operations such as dimensionality reduction, hidden information detection, optimization, and many other applications. Singular Value Decomposition is a valuable method in data science, allowing us to decompose a matrix into its fundamental components. Similar to Principal Component Analysis (PCA), SVD helps reduce the dimensionality of data while preserving the most important information. However, SVD can be applied to non-square, non-invertible matrices, and its ability to separate fundamental components enables us to analyze more complex data. SVD has a wide range of applications in practical scenarios.

Keywords: SVD, matrix, data

1. Introduction

In recent years, Artificial Intelligence (AI), and more specifically Machine Learning, has emerged as evidence of the fourth industrial revolution (1 - steam engine, 2 - electricity, 3 - information technology). Artificial Intelligence is infiltrating every aspect of our lives, and we may not even realize it. Google and Tesla's self-driving cars, Facebook's facial recognition system, Apple's virtual assistant Siri, Amazon's product recommendations, Netflix's movie suggestions, Google DeepMind's

* Corresponding author, E-mail: Cttxuan@uneti.edu.vn

https://doi.org/10.56764/hpu2.jos.2023.2.3.34-41

Received date: 13-10-2023 ; Revised date: 30-11-2023 ; Accepted date: 30-11-2023

AlphaGo in the game of Go, are just a few examples among countless applications of AI/Machine Learning [1].

Machine Learning is a subset of AI. According to the Wikipedia definition, Machine Learning is the subfield of computer science that "gives computers the ability to learn without being explicitly programmed." In simple terms, Machine Learning is a smaller field within Computer Science that enables computers to learn and adapt based on the input data without requiring explicit, hand-coded instructions.

Because of this, the research and development of Machine Learning algorithms as well as the creation of applications are high demands for computer scientists. The mathematical foundation of Machine Learning requires a fundamental understanding of linear algebra, optimization, and probability and statistics.

Matrix factorization, or matrix decomposition, is a fundamental technique with numerous important applications, particularly in machine learning. It involves decomposing a matrix into the product of several specialized matrices. This process has a wide range of benefits, including dimensionality reduction, data compression, exploring data properties, solving linear equations, clustering, and many other applications. One prominent application of matrix factorization is in recommendation systems.

Singular Value Decomposition (SVD) is a form of matrix factorization that finds extensive applications in problems related to inversion and data analysis. Nowadays, SVD is prevalent in practical applications such as digital signal processing, approximating values in engineering, information technology, machine learning, and is employed in search engines on various websites. However, theoretical research concerning SVD is a relatively new field and can be challenging for readers interested in delving into this fascinating topic. Therefore, this article aims to provide readers with the most basic knowledge about SVD factorization and offer an overview of its factorization form, as well as some essential properties and relevant consequences associated with this decomposition.

2. Theoretical basis

2.1. Machine learning

The concept of learning is broad, much like the notion of intelligence, and it does not have a precise definition. According to the dictionary, learning is the process of acquiring knowledge or skills either from others or by reading, studying, and memorizing. In a broader sense, learning also encompasses the process of extracting knowledge from observations and practical experiences.

Machine learning has two common meanings:

- Using computers to discover knowledge from data.

- The act of learning within a machine or agent.

- From a technological standpoint, machine learning is a subfield of artificial intelligence in which researchers study techniques for constructing and developing computer programs capable of adapting and "learning" from sample data or experience.

To date, there have been many definitions of this concept, but it is difficult to establish a universally accepted definition. The following definition is derived from T. Mitchell's definition and provides a mathematical perspective on how a learning program is researched and designed:

Definition 2.1: A computer program is considered to learn from data or experience E with respect to a class of tasks T and performance measure P if its performance on tasks T, as measured by P, improves with data or experience E.

According to this definition, one needs to optimize the performance measure P based on the analysis of data and experience E to find the best way to perform tasks T.

Example 2.1.

Retail Data Analysis for Supermarkets: Every day, supermarkets sell a large variety of items and record transaction data (shopping cart copies). From the retail data collected, we can analyze shopping baskets to predict how likely it is that a customer who buys item A will also buy item B. If this probability is high, we should place these items close together, making it convenient for customers and increasing sales compared to having customers search all over. Furthermore, with a good analytical model, we can also predict the quantity of goods needed in the near future, customer preferences, and based on that, make adaptive policy decisions. In this example, T represents forecasting, E represents the stored retail data, and P represents the accuracy of the forecasting results.

Problem 2.1.

The problem of fingerprint matching originates from two tasks: fingerprint identification and fingerprint verification. In the identification task, one needs to match a fingerprint image obtained during an investigation with fingerprint images stored in a database to determine if any of the stored fingerprints belong to the same rolled-out finger as the investigated image. In the verification task, one needs to verify whether a logged-in fingerprint image (also referred to as the probe) is indeed from the same finger as the registered image. Both of these tasks are reduced to the problem of matching pairs of fingerprint images: comparing a probe image with a stored image to answer whether they are from the same or different fingers. To build a fingerprint matching program, one requires a dataset containing pairs of images that either come from the same finger or from different fingers. Based on this dataset, an algorithm is applied to construct the program.

Problem 2.2.

Finding the Shortest Path for a Robot: In a grid network comprising n automated stations with varying distances between them, a robot needs to visit these stations exactly once. Assuming that the robot remembers the stations it has passed through and the distances between them, it also knows which stations it needs to inspect next. In this scenario, as it moves from one station to another, it will seek the shortest path to the next station. With a good learning strategy, over time, the robot will find shorter and potentially optimal paths. In this case, P and E correspond to the length of the paths and the routes the robot has discovered, and T represents the inspection route.

The strong penetration of advanced information technology, the development of the knowledge-based technology society, and the emergence of widespread application needs have led to various research scopes and typical applications:

Building pattern recognition systems for audiovisual devices in robots and in the field of automation, including tasks like handwriting recognition, speech-to-text conversion, and automatic image analysis.

Creating computer programs that can adapt to changing environments or perform tasks initially undefined, such as autonomous driving systems (for aircraft, cars, ships, etc.), games, or versatile robot control.

Extracting knowledge from data, especially from large databases, to aid in decision-making. For instance, market analysis, diagnosing patient illnesses, and determining treatment options through the analysis of stored medical records.

These applications demonstrate the significance of artificial intelligence and machine learning in addressing complex real-world problems across various domains.

Over the past few decades, scientific research and applications of machine learning have rapidly developed, incorporating advancements from various other fields. Here are the key areas that have significantly contributed to the research of machine learning:

Probability and Statistics Theory: This is the precursor to the field of machine learning, allowing for inference from specific observations to general conclusions through the achievements of random analysis.

Neurobiology Models: Studying the mechanisms, nonlinear processing, and structure of biological neural systems in general enables the creation of bio-inspired models and algorithms, especially neural networks.

Complexity Theory: Allows for the estimation of the computational complexity of learning tasks through training examples, error bounds, and computational procedures.

Adaptive Control Theory: Learning procedures for controlling processes to optimize predetermined objectives or learning to predict the next state of a controlled process.

Psychology: Allows for modeling real-world human responses, building efficient processing models, such as reinforcement learning.

Evolutionary Models: Researching evolutionary models enables the development of natural-inspired learning algorithms, such as Genetic Algorithms (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Artificial Immune Systems (AIS).

These interdisciplinary contributions have shaped and advanced the field of machine learning, enabling it to address a wide range of real-world problems and applications effectively.

2.2. Singular Value Decomposition

Singular Value Decomposition (SVD) is an important method in linear algebra and data processing. It allows the decomposition of a not necessarily square matrix into the product of three special matrices: a unitary matrix U, a diagonal matrix Σ, and another unitary matrix, the transpose of U (U*). Below is the definition, properties, and a specific example of the SVD method.

Definition of SVD

For an m x n matrix A, SVD decomposes A into the product of three special matrices:

The unitary matrix U with dimensions m x m: U*U = I (identity matrix).

The diagonal matrix Σ with dimensions m x n, where the values on the diagonal are non-negative and sorted in decreasing order.

The unitary matrix V* with dimensions n x n: V*V = I (identity matrix).

In other words, A = UΣV*.

Properties of SVD

a. U and V are unitary matrices, meaning that UU* = VV* = I.

b. The values on the diagonal of Σ are called singular values, typically arranged in decreasing order.

c. The singular values of A are precisely the square roots of the non-negative eigenvalues of the matrices AA* or A*A.

d. SVD is an important method for data compression, dimensionality reduction, and optimization analysis.

3. Singular Value Decomposition and applications

SVD has numerous practical applications in real-world problems, such as:

Dimensionality Reduction: Reducing the dimensionality of data helps us minimize computational complexity and avoid overfitting. This is particularly useful when working with high-dimensional data, such as images, audio, or text.

Discovering Hidden Information: SVD can decompose data into its underlying factors, helping us gain insights into data structures and uncover hidden relationships between components.

Image and Video Compression: By applying SVD, we can compress image and video data while preserving essential information.

Handling Missing Data: In scenarios where data is missing or incomplete, SVD enables us to reconstruct missing values without resorting to traditional imputation methods.

Recommendation Systems: In the field of recommendation systems, SVD can be applied to build recommendation systems based on user preferences.

To compute the Singular Value Decomposition (SVD) of a matrix in Python, we can use the NumPy library. We give a Python code of how to calculate the SVD for a matrix as follows:

```
# Create an example m x n matrix
A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
# Calculate the SVD of matrix A
U, S, VT = np.linalg.svd(A)
# U is an m x m unitary matrix
# S is an array containing the singular values sorted in descending order
# VT is the n x n unitary matrix, the transpose of V
print("Matrix U:")
print(U)
print("\nSingular values (Σ):")
print(np.diag(S))
print("\nMatrix V transpose:")
print(VT)
```

SVD (Singular Value Decomposition) is a versatile mathematical technique with numerous applications, and one of the most impressive applications is its use in digital image compression. This enables efficient transmission of digital images via satellite and the internet.

The fundamental idea behind image compression is to reduce the amount of transmitted information without losing essential details. In a digital image, each pixel is represented by three color values: Blue, Green, and Red, typically ranging from 0 to 255 (for 8-bit images). Therefore, for an image with dimensions of 340 X 280 pixels, we would need to store three matrices representing color information, each with the same size of 340 X 280, resulting in a total of 285,600 values to be stored. However, in practice, when transmitting or storing image data, we often don't need the images to have such high levels of detail. By using SVD, we can effectively eliminate a significant amount of unnecessary information.

To illustrate the specific use of SVD in digital image compression, let's consider two problems of image compression by reducing the number of color information while preserving the essential representation.

Problem 3.1. Image Compression using SVD

Let's assume we have a digital color image with dimensions 340x280 pixels. Each pixel in the image is represented by three color values: Red (R), Green (G), and Blue (B), each value ranging from 0 to 255.

To perform image compression using Singular Value Decomposition (SVD), we follow these steps:

Step 1: Convert the image into a 3D matrix with dimensions (340, 280, 3), where 3 represents the number of color channels.

Step 2: Apply SVD separately to each color channel (R, G, B). For example, for the Red channel (R), we calculate SVD as follows:

Python code: U_R, S_R, VT_R = np.linalg.svd(R_channel)

Step 3: Reduce the dimension of the diagonal matrix Σ by retaining only a certain number of important singular values. This reduces the size of the Σ matrix as well as the unitary matrices U and VT.

Step 4: Reconstruct the image using the reduced matrices obtained from Step 3: Python code: R_reconstructed = U_R[:, :k] @ np.diag(S_R[:k]) @ VT_R[:k, :]

Here, k is the number of important singular values you want to keep. This allows you to control the level of data compression. If you keep fewer singular values, the image will be compressed more, and details will be lost.

Step 5: Repeat Steps 2 to 4 for all color channels (G and B) of the image.

In the end, you will have three reconstructed matrices (one for each color channel). Combine them to create the compressed image. The resulting image may retain the most important information from the original image while discarding unnecessary information. The size of the Σ matrix and the number of important singular values (k) determine the degree of image compression.

This process illustrates how SVD can be applied to compress images effectively while preserving essential information.

Problem 3.2. Reducing the dimensionality of data

Reducing the dimensionality of data is a common challenge in data analysis and machine learning, and Singular Value Decomposition (SVD) is a powerful technique often employed to address this problem. The goal of dimensionality reduction is to represent a high-dimensional dataset in a lower-dimensional space while preserving its essential features and structure.

SVD achieves dimensionality reduction by decomposing a data matrix into three other matrices: U (the left singular vectors), Σ (a diagonal matrix containing the singular values), and V^T (the transpose of the right singular vectors). These singular values in Σ represent the importance or contribution of each component to the original data.To reduce the dimensionality of the data using SVD, we typically perform the following steps:

Step 1: Compute the SVD of the data matrix.

Step 2: Sort the singular values in Σ in descending order. These singular values represent the importance of the corresponding components in the data

Step 3: Select a subset of the top-k singular values and their corresponding singular vectors (columns in U and rows in V^T) to retain the most important components.

Step 4: Form a new data matrix using these selected components.

By retaining only the top-k components, where k is a user-defined parameter, we can significantly reduce the dimensionality of the data while retaining as much variance or information as possible. This reduced-dimensional representation can be used for various purposes, such as visualization, noise reduction, or as input for machine learning algorithms when dealing with high-dimensional datasets. Overall, SVD-based dimensionality reduction is a valuable tool for simplifying complex datasets and extracting meaningful information, making it a crucial technique in data preprocessing and analysis.

We give a Python code that demonstrates how to use Singular Value Decomposition (SVD) to perform Principal Component Analysis (PCA), a common data analysis technique for dimensionality reduction: import numpy as np from sklearn.decomposition import TruncatedSVD.

```
# Create a sample high-dimensional data matrix
data = np.random.rand(100, 50)  # 100 samples, 50 features
# Specify the number of components (dimensions) to reduce to
n_components = 10
# Apply SVD for dimensionality reduction
svd = TruncatedSVD(n_components=n_components)
reduced_data = svd.fit_transform(data)
# Check the explained variance ratio to understand information retained
explained_variance_ratio = svd.explained_variance_ratio_
total_variance_retained = sum(explained_variance_ratio)
print(f"Explained Variance Ratio for {n_components} components:")
print(explained_variance_ratio)
print(f"Total Variance Retained: {total_variance_retained:.2f}")
# The 'reduced_data' now contains the data in reduced dimensions
```

4. Conclusions

In practice, matrix computations play a crucial role in data science across a wide range of applications. The SVD method provides us with a powerful tool for processing and exploring complex data, addressing the challenges of handling large and diverse datasets.

References

[1] F. Anowar, S. Sadaoui and B. Selim, "Conceptual and empirical comparison of dimensionality reduction n algorithms," *Computer Science Review*, vol. 40, pp.100378, 2021, doi: 10.1016/j.cosrev.2021.100378

[2] F. Anowar and S. Sadaoui, "Incremental neural-network learning for big fraud data," *IEEE International Conference on Systems*, Man, and Cybernetics (SMC), IEEE, pp.3551-3557, 2020, doi: 10.1109/SMC42975.2020.9283136

[3] F. Anowar and S. Sadaoui, "Incremental learning framework for real-world fraud detection environment, " *Comput. Intell*, pp.1-22, 2021, http://dx.doi.org/10.1111/coin.12434, doi: 10.1111/coin.12434

[4] Z. Bai, E. Kaiser, J. L. Proctor, J. N. Kutz and S. L. Brunton, "Dynamic mode decomposition for compressive system identification," *AIAA Journal*, vol. 58, no. 2, pp.561-574, 2020, doi: 10.2514/1.J057870

[5] N. H. Cuong and Đ. C. Tung, "Nghiên cứu một số phương pháp học sâu và ứng dụng phát hiện ảnh giả mạo," *Giao thông Vân tải*, vol. 4, 2022.

[6] N. H. Cuong, "Phát hiện chỉnh sửa ảnh trên ảnh số dựa vào các phép phân tích ma trận, " *Tạp chí Khoa học GTVT ĐB Hội nghị KHCN lần thứ XXI*, 2018.

[7] N. Donthu, S. Kumar, D. Mukherjee, N. Pandey and W.M. Lim, "How to conduct a bibliometric analysis: An overview and guidelines," *Journal of Business Research*, vol. 133, pp.285-296, 2021, doi: 10.1016/j.jbusres.2021.04.070

[8] A. E. Ezugwu, A. K. Shukla, M. B. Agbaje, O. N. Oyelade, A. José-García and J. O. Agushaka, "Automatic clustering algorithms: a systematic review and bibliometric analysis of relevant literature," *Neural Computing and Applications*, vol. 33, no. 11, pp.6247-6306, 2021, doi: 10.1007/s00521-020-05395-4

[9] G. H. Golub and V. Loan, "Matrix Computations," *Baltimore, Johns Hopkins University Press*, 2012.

[10] Y. Jaradat, M. Masoud, I. Jannoud, A. Manasrah and M. Alia, "A Tutorial on Singular Value Decomposition with Applications on Image Compression and Dimensionality Reduction," *International Conference on Information Technology (ICIT)*, pp.769-772, 2021, doi: 10.1109/ICIT52682.2021.9491732

[11] Y. Jaradat, M. Alia, M. Masoud, A. Manasrah, I. Jebreil, A. Garaibeh, et al., "A bibliometric analysis of the International Journal of Advances in Soft Computing and its Applications: Research influence and Contributions," *International Journal of Advances in Soft Computing & Its Applications*, vol. 14, no. 2, 2022, doi: 10.15849/IJASCA.220720.12

[12] P. Jindal and D. Kumar, "A review on dimensionality reduction techniques, " *Int.J. Comput. Appl*, vol. 173, no. 2, pp.42-46, 2017, doi: 10.5120/ijca2017915260

[13] Y.Jaradat, M. Masoud, I. Jannoud, A. Manasrah and M. Alia, "A Tutorial on Singular Value Decomposition with Applications on Image Compression and Dimensionality Reduction," *2021 International Conference on Information Technology (ICIT)*, 2021, doi: 10.1109/ICIT52682.2021.9491732

[14] M. Masoud, Y. Jaradat, E. Rababa and A. Manasrah, "Turnover prediction using machine learning: Empirical study," Int. J. Advance Soft Compu. Appl, vol. 13, no. 1, 2021.

[15] B.N. Parlett and D.S. Scott, "Math Comput," vol. 33, pp.217-238, 1979, doi: 10.1090/S0025-5718-1979-0514820-3

[16] Đ. T. Phương, T. H. Hai, N. M. Tuong and B. T. T. Xuân, "Giáo trình Toán cao cấp 1," *NXB Đại học Thái Nguyên*, 2016.

[17] V. D. Tuan, "Python rất là cơ bản," 2016.

[18] A. I. Terekhov, "Bibliometric trends in quantum information processing, " *Scientific and Technical Information Processing*, vol. 47, no. 2, pp. 94-103, 2020, doi: 10.3103/S0147688220020021

[19] V. Spruyt, "The curse of dimensionality in classification," *Comput. Vision Dummies 21*, vol. **3**, pp.35-40, 2014.

[20] L.Van Der Maaten, E. Postma and J. Van den Herik, "Dimensionality reduction: A comparative review"*J. Mach. Learn. Res*, vol. 10, pp.66-71, 2009.

[21] I. Yamazaki, Z. Bai and H. Simon *et al*, "ACM Trans Math Softw," vol. 37, pp.1-18, 2010.

[22] C. Xu, U. Xu and K. Jing,"Fast algorithms for singular value decomposition and the inverse of nearly low-rank matrices," *National Science Review*, vol.10**,** 2023.