Proceedings of the 4th International Conference on Applied Physics and Mathematical Modeling
DOI: 10.54254/2753-8818/2026.37403

A Review of Matrix Factorization Algorithms: Theoretical Developments and Modern Applications

Fangyue Sun

School of Mathematics and Statistics, University of New South Wales, Sydney, Australia
2273879419@qq.com

Abstract. The effective processing and analysis of high dimensional data is crucial challenge in the big data era of computational science and engineering. As one of the most basic and powerful mathematical tools, matrix factorization can extract the low dimensional structures from high dimensional data and play an irreplaceable role in scientific computing and machine learning to achieve dimensionality reduction, feature extraction and information compression. This paper presents a systematic review of the theoretical basis and common usage of the matrix factorization algorithms and discusses the mathematical principles and the characteristics of the classical and spectral decomposition algorithms. In addition, cutting-edge trends and future directions are discussed including randomized algorithms, distributed computing, and decompositions based on deep learning. This research is carried out by systematic literature analysis, and the result of this analysis is the synthesis of existing knowledge in several fields of computation. The results show that classical matrix factorizations are well suited to deliver stable exact solutions to linear systems and modern probabilistic and tensor-based methods are necessary to obtain scalability in modern applications. Despite the ever-changing nature of the computational paradigms the matrix factorization remains a fundamental tool, which continues to be a critical link between pure linear algebra and applied data science.

Keywords: Matrix Factorization, Singular Value Decomposition, Dimensionality Reduction, Numerical Linear Algebra, Machine Learning

1. Introduction

With the advent of the age of "big data," the efficient processing and analysis of datasets with many dimensions have become fundamental problems in the areas of computational science and engineering. Matrix factorization is a powerful mathematical tool that can capture low dimensional structures in large high dimensional arrays, and it has been used extensively in dimensionality reduction, feature extraction and information compression [1]. These decomposition methods have found extensive use in various fields such as scientific computing, machine learning and data mining and have proven to be extremely efficient for the acceleration of the computation process [2]. Today while there are many scholarly reviews which have investigated related approaches to matrix factorization from particular and isolated points of view, most investigations concentrate on one algorithm, or one domain of narrow application [3]. Such a comprehensive review, which

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2753-8818/2026.37403

systematically incorporates the modern advancements in theory of linear algebra and the picture of industrial applications, is quite rare in the current literature [4].

Hence, in this paper three research objectives are addressed. Firstly, the development and theoretical background of matrix factorization algorithms are systematically analyzed, highlighting how their structure has evolved from classical exact algorithms to modern low-rank and probabilistic ones. Second, it carefully and explicitly analyzes the modern versions, operational trade-offs, and empirical results of these algorithms in complex domain problems such as recommender systems, image processing, and natural language understanding. Third, it gives a direction on the evolution of decomposition algorithms in high dimensional environments as well as a projection on the scalability bottlenecks and future evolutions. This research uses a systematic and rigorous literature analysis method to synthesize the advancements of literature and insights of the different domains from classical exact methods to modern approximate methods. The importance of the current study is to offer a comprehensive and coherent framework that unifies abstract mathematical and computational requirements, as well as the constraints of modern hardware platforms for the execution of computing tasks. The following sections are expected to provide extensive theoretical insights and concrete recommendations on algorithm optimization, specifically regarding numerical stabilization, parallelization and acceleration of computation for ultra large-scale sparse problems in modern artificial intelligence systems.

2. Classical factorization algorithms

The classical matrix factorizations are the foundations of numerical linear algebra. The key use of such theoretical constructs is to solve systems of linear equations, calculate the inverse of a matrix and/or calculate the determinant of a matrix without doing any direct, numerically unstable operations [5].

2.1. LU decomposition

Unless the matrix A is singular, the LU decomposition factorises a square matrix $\mathrm{A} \in \mathbb{R}^{\mathrm{n}\times\mathrm{n}}$ into the product of a lower triangular matrix L and an upper triangular matrix U, $\mathrm{A} = \mathrm{LU}$. If all the leading principal minors of the matrix A are nonzero, then a strict LU decomposition exists (and doesn't require row permutations).

The problem of finding the matrices L and U is essentially the same as the Gaussian elimination problem. It takes $\mathcal{O}\left(\mathrm{n}^3\right)$ to run an $\mathrm{n}\times\mathrm{n}$ matrix. In a practical computing setting however, the numerical stability of the factorization requires almost always that partial pivoting is used, resulting in the modified factorization $\mathrm{PA} = \mathrm{LU}$, where P is a permutation matrix [6]. With this permutation the possibility of division by very small pivots is avoided and the growth factor of the matrix elements is strictly bounded, and the risk of catastrophic cancellation errors is reduced. Two triangular systems, $\mathrm{Ly} = \mathrm{Pb}$ and $\mathrm{Ux} = \mathrm{y}$, can then be solved using forward and backward substitution, respectively, to reduce the linear system $\mathrm{Ax} = \mathrm{b}$ to their solution.

2.2. Cholesky decomposition

When a matrix A exhibits symmetric and positive-definite properties, meaning $\mathrm{A} = \mathrm{A}^{\mathrm{T}}$ and $\mathrm{x}^{\mathrm{T}}\mathrm{Ax} > 0$ for all non-zero vectors $\mathrm{x} \in \mathbb{R}^{\mathrm{n}}$, it admits a highly specialized and computationally efficient factorization known as the Cholesky decomposition. This is expressed as $\mathrm{A} = \mathrm{LL}^{\mathrm{T}}$, where L is a lower triangular matrix characterized by strictly positive diagonal entries [7].

DOI: 10.54254/2753-8818/2026.37403

The Cholesky algorithm requires exactly half the number of floating-point operations compared to standard LU decomposition, executing in approximately $1/3 \bullet n^3$ operations. It exhibits unconditional numerical stability, meaning pivoting is entirely unnecessary. This decomposition is extensively utilized in convex optimization algorithms, Monte Carlo simulations for generating correlated random variables, and efficiently solving the normal equations in ordinary least squares regression.

2.3. QR decomposition

The QR decomposition is a way to decompose a matrix $A \in \mathbb{R}^{m \times n}$ into an orthogonal matrix $Q \in \mathbb{R}^{m \times m}$, for which $Q^T Q = I$, and an upper triangular matrix $R \in \mathbb{R}^{m \times n}$, so that $A = QR$. This decomposition is closely related to the Gram-Schmidt orthogonalisation Theorem [8].

In numerical practice direct Gram-Schmidt procedures are often abandoned because of the intrinsic loss of orthogonality that occurs because of floating point rounding. Rather, numerical stability is the basis for systematically reducing the sub-diagonal elements by using Householder reflections or Givens rotations. QR algorithm is an essential mean to solve over-determined linear least-squares problems. To solve the easier problem, the objective function for the least squares problem is rewritten by substituting $A = QR$, so that the problem can be rewritten as $Rx = Q^T b$, thereby avoiding the numerical difficulties of forming the normal equations directly. In addition, QR factorization is the iterative "power engine" used for computing eigenvalues in the standard QR algorithm.

To draw inferences from classic paradigm synthesis, this paper compares the results from numerical complexity and stability analysis, which exhibits unique operation trade-offs. LU decomposition is a general-purpose solver and requires $\mathcal{O}(2/3 \bullet n^3)$ floating-point operations, whereas Cholesky decomposition takes advantage of the symmetry of the system to reduce the number of floating-point operations to $\mathcal{O}(1/3 \bullet n^3)$; it is a choice of solver used for positive definite systems. In contrast, the number of operations needed for QR decomposition of square matrices using Householder reflections is about $\mathcal{O}(4/3 \bullet n^3)$. QR, however, is very numerically stable for ill-conditioned matrices, because orthogonal transformations maintain the Euclidean norm and avoid the unwarranted amplification of round-off errors. Hence, there is a trade-off between the efficiency of the classical algorithms and sensitivity to condition numbers of matrices.

3. Advanced spectral and low-rank factorizations

Classical techniques are precise and optimal for well-conditioned systems, but today's data science often deals with naturally noisy, redundant data. Hence, the research emphasis has turned towards advanced spectral methods that can be used to obtain approximate, low-rank representations.

3.1. Singular Value Decomposition (SVD)

The Singular Value Decomposition is essentially known as the most important structural theorem in applied linear algebra. For any real rectangular matrix $A \in \mathbb{R}^{m \times n}$, a definitive factorization can be achieved as $A = U\Sigma V^T$ where U is an orthogonal matrix of all the left singular vectors, and V is an orthogonal matrix of all right singular vectors [9]. The diagonal matrix $\Sigma \in \mathbb{R}^{m \times n}$ is a diagonal matrix with the singular values arranged in decreasing order: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$, and r is the exact rank of matrix A .

DOI: 10.54254/2753-8818/2026.37403

The great power of SVD is its ability to present the best possible low rank approximation of any matrix in an unparalleled manner. The Eckart-Young-Mirsky theorem states that the truncation of the SVD to the first k singular values (where $k < r$), yielding $A_k = U_k \Sigma_k V_k^T$, generates a matrix A_k that serves as the closest rank-k matrix to the original A [10]. This optimality is true with respect to both Frobenius norm and spectral norm. The resulting truncation error is mathematically given by the sum of the squared singular values, where the singular values not kept in the approximation are squared. This is a strong property, which forms the basis for principal component analysis (PCA) and many data compression schemes.

Geometrically, the singular values σ_i are the lengths of the principal axes of the hyperellipsoid that is formed by transforming a unit hypersphere by A. In the actual data matrices, the singular values decay very quickly, which reflects the fact that these matrices are very linearly redundant, and it is possible to reduce the dimension considerably without losing much information. In real applications, practitioners often set an energy preservation threshold, and choose k to be the smallest value for which $\sum_{i=1}^{k} \sigma_i^2 / \sum_{i=1}^{r} \sigma_i^2 \geq \gamma$, with $0 < \gamma \leq 1$, being the desired fraction of energy to be preserved. On the other hand, the elbow point in which the signal turns to noise can be located with the help of scree plot heuristics and randomized profile likelihood methods. This spectral property makes SVD the ultimate measure of low-rank matrix approximation.

3.2. Non-negative Matrix Factorization (NMF)

The singular vectors in U and V are often negative in SVD which gives a mathematically optimal approximation. In empirical data sets, negative numbers may not have a physical interpretation or may be difficult to interpret, e.g., negative pixel intensities in image processing or negative word frequencies in text analysis. To address this discrepancy, Non-negative Matrix Factorization requires very strong non-negativity constraints.

With a non-negative data matrix $V \in \mathbb{R}_{\geq 0}^{m \times n}$ as input, NMF seeks to find two lower dimensional matrices $W \in \mathbb{R}_{\geq 0}^{m \times k}$ and $H \in \mathbb{R}_{\geq 0}^{k \times n}$ such that $V \approx WH$. This goal is usually cast as a non-convex optimization problem that is to minimize the Frobenius distance or Kullback-Leibler divergence between V and WH subject to $W \geq 0$ and $H \geq 0$ [11]. These multiplicative update rules for this optimization naturally, result in a parts-based representation. Additive combinations are strictly obeyed, and NMF will learn a set of basic, non-overlapping components of the data, which is very useful in some applications such as bioinformatics and pattern recognition.

4. Modern applications

The matrix factorization techniques have been able to make the leap from purely theoretical mathematical constructs to the algorithmic backbone of the global technology sector. The ability of such immense matrices to be broken down into functional components opens many modern technological applications.

4.1. Recommender systems and collaborative filtering

During the algorithmic evolution of recommender systems, the use of matrix factorization came to prominence, especially worldwide. In this setting, a user-item interaction matrix R is formed, which is typically very sparse; an average user interacts with only a small fraction of the total number of items. The model can be used to make accurate predictions of missing entries in the matrix by

DOI: 10.54254/2753-8818/2026.37403

factoring $R \approx PQ^T$, where P contains the user latent factors and Q contains the item latent factors [12].

Since the matrix R is sparsely distributed, traditional SVD cannot be calculated directly, without an excessive amount of imputation. Therefore, it is reformulated as an optimization problem on the set of known ratings. L2 regularisation is rigorously built in to avoid catastrophic overfitting on few data. The above objective function is most used in conjunction with Stochastic Gradient Descent (SGD) or Alternating Least Squares (ALS). ALS is especially attractive for distributed computing, splitting the problem into a sequence of fixing P and optimizing Q, which makes the objective function convex at each of the discrete steps and hence very large parallelism can be used.

4.2. Image compression and denoising

About digital image processing, a gray level photograph is mathematically represented as a very dense 2-D image matrix of pixel intensity values. The use of the truncated Singular Value Decomposition can yield very high compression of images. The computation and retention of only the top k singular vectors and singular values result in a sharp reduction in the amount of physical storage required from $m \times n$ entries to just $k(m + n + 1)$ parameters.

At the same time this breakdown serves as an integral high efficiency noise filter. Empirical models show that the energy of a random Gaussian noise is uniformly spread on all the singular values. On the other hand, the largest singular values of an image are very concentrated with the structural signal. So the high frequency noise contained in the rest of the spectrum will be thrown away by truncating the spectrum, not only to reduce the dimension of the file, but also to "denoise" the image mathematically.

4.3. Natural language processing (latent semantic analysis)

A corpus of text is traditionally represented in NLP as a sparse term-document matrix A, where each element A_{ij} is a measure of the term frequency-inverse document frequency (TF-IDF) of a particular word i in a particular document j. The application of truncated SVD to this linguistically based matrix is the basis of Latent Semantic Analysis (LSA).

LSA is a good way of mapping individual words and even whole texts into a common, continuous, low dimensional latent semantic space. A mathematical transformation enables the system to understand complex linguistic phenomena like synonymy and polysemy. Algorithms that apply the principle of cosine similarity of vectors in the decomposed orthogonal space, move beyond simple keyword matching and unlock true contextual meaning, which is the foundation of today's dense retrieval architectures.

5. Recent developments and future directions

With the vast size and rapid growth of data dimensions in this age, linear algebra is evolving into very advanced paradigms of factorization.

First, the field has been changed by the advent of Randomized Matrix Factorization. In the case of extremely large matrices, the classical deterministic algorithms whose time complexity is of cubic order are simply impracticable. Recently, new theories have been developed which use randomized algorithms with Gaussian random projection matrices. The projections are aimed at finding a lower dimensional subspace which for the most part captures the action of the original matrix. This

DOI: 10.54254/2753-8818/2026.37403

technique is then followed by deterministic decompositions performed only on the smaller sketch matrix, thus avoiding the computational bottleneck with provable bounds on the loss of accuracy.

Randomized SVD algorithms can be described as a two-stage paradigm. In the random step, a random test matrix, $\Omega \in \mathbb{R}^{n\times(k+p)}$, where p is the oversampling parameter, is multiplied with A to obtain a sample matrix, $Y = A\Omega$. By orthogonalizing Y with QR decomposition, the orthogonal basis Q will have the columns of the basis spanning the range of A and the basis Q will be orthonormal. In the deterministic stage, this matrix is projected onto this lower dimensional subspace by $B = Q^{T}A$ which gives a small matrix $B \in \mathbb{R}^{(k+p)\times n}$. Then, regular deterministic SVD is performed on $B = \tilde{U}\Sigma V^{T}$, and the left singular vectors of the original matrix B are reconstructed as $U = Q\tilde{U}$. This is a very decoupled framework which brings matrix operations to big data regime without sacrificing matrix geometry.

Secondly, Tensor Decompositions are at the front lines of the multilinear algebra. Increasingly, modern data structures have multidimensional properties, like functional magnetic resonance imaging (fMRI) maps that contain spatial, temporal and color properties. Algorithms like CANDECOMP/PARAFAC (CP) and Tucker Decomposition are higher order multidimensional versions of SVD and PCA. The direction of factorization development is obviously towards optimization of tensor networks and towards use of such multilinear decompositions directly in hidden layers of a deep neural network where parameter efficiency promises to be greatly boosted.

6. Conclusion

This paper has systematically presented the subject of matrix factorization algorithms from the exact classical algorithms to the sophisticated spectral approximation algorithms and the recent industrial implementations. The results show that the classical factorizations such as LU, Cholesky and QR are still of fundamental importance in the problem of secure numerical stability for deterministic systems. Moreover, higher order methods like SVD and NMF have been found essential in performing optimal low rank approximation and uncovering interpretable low order structure of complex high dimensional data sets. A review of the recent applications shows that these algebraic methods always provide the basic structure of the recommender systems, image compression protocols, latent semantic analysis, etc.

Even though this is an extensive synthesis, there are some limitations in the study. Because of the reliance in literature analysis, it is not possible to introduce novel empirical benchmarking in different hardware environments. Moreover, relationship of stochastic optimization methods and non-convex factorization landscapes is discussed in a general manner and not in a comprehensive numerical proof. More research is needed to fully investigate the combination of matrix factorization and quantum computing paradigm and deep neural network architectures. There is a need for new massively distributed tensor decompositions as well as non-linear, deep matrix factorization frameworks to address the scalability issue inherent in existing deterministic models.

References

[1] Golub, G. H., & Van Loan, C. F. (2013). *Matrix computations* (4th ed.). Johns Hopkins University Press.
[2] Strang, G. (2016). *Introduction to linear algebra* (5th ed.). Wellesley-Cambridge Press.
[3] Halko, N., Martinsson, P. G., & Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review, 53*(2), 217–288.
[4] Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature, 401*(6755), 788–791.

DOI: 10.54254/2753-8818/2026.37403

[5] Trefethen, L. N., & Bau, D. (1997). *Numerical linear algebra*. SIAM.
[6] Demmel, J. (1997). *Applied numerical linear algebra*. SIAM.
[7] Higham, N. J. (2002). *Accuracy and stability of numerical algorithms* (2nd ed.). SIAM.
[8] Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika, 1*(3), 211–218.
[9] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer, 42*(8), 30–37.
[10] Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review, 51*(3), 455–500.
[11] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*(6), 391–407.
[12] Paatero, P., & Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics, 5*(2), 111–126.