UNIVERSIDAD PANAMERICANA

Course: Simulation and data visualization

Final Project: Machine learning exploration

Maestra: Gabriel Castillo Cortés

Team 2:

Miguel Herrera Padilla

Eduardo Daniel Ramírez Prado

Carlos Esteban López Sánchez

Gabriel Guerra Rosales

INTRODUCTION

With the passage of time and the advances of technology, the amount of data available to study is huge; it's necessary to have a dynamic and fast way to analyze these data sets. That's where the necessity of machine learning, a branch of artificial intelligence, grows. Instead of using simple and fixed rules, machine learning identifies patterns to "learn" and be trained with this data. In a nutshell, it focuses on a system capable of learning from data in order to make predictions and decisions with this information. It's the way we can use large amounts of raw data to train the system in order to get real insights or compare relationships between multiple datasets. As we will see, thanks to this tool , we can now see the impact in a wide range of areas. With this report we want to focus primarily on some mathematical techniques (PCA, ICA, SVD) used in machine learning algorithms specifically in the area of unsupervised learning in order to better understand all the processes of training data and how these algorithms simplify and achieve a machine learning model's work.

DEVELOPMENT

Principal Component Analysis

Overview

Principal Component Analysis is an unsupervised statistical technique that allows you to reduce the dimensions of a dataset while keeping most of the information from the original dataset. It is attributed to Karl Pearson in 1901; however, its popularity rose when computers became widely available. In general terms, the way PCA works is by transforming a subset of potentially related variables into a smaller set that represents said correlation without the loss of important information (IBM, 2023). Nonetheless, it is important to take into consideration that while the "least important" variables are omitted, the new variables created will have different meanings than the original dataset, resulting in loss of interpretability (CS 357 Course Staff, 2025).

Key Features

1. **Linear Orthogonal Transformation:** The algorithm utilizes an orthogonal linear transformation to convert the data to a new coordinate system. This is done such that "the greatest variance by some scalar projection of the data comes to lie on the first coordinate", this is then known as the

first principal component, and then it goes on, the second greatest variance on the second coordinate, and so on and so forth (CS 357 Course Staff, 2025).

2. **Variance Maximization:** The first principal component is the direction that maximizes the variance of the projected data. Each subsequent component maximizes the remaining variance, taking into account the orthogonality constraint. In other words, PCA takes into account the most "informative" directions in terms of data spread (CS 357 Course Staff, 2025).
3. **Eigenvector Calculation:** Principal components can be thought of as eigenvectors of the data's covariance matrix. PCA is essentially an eigen-decomposition problem; it finds the eigenvalues and eigenvectors of the covariance matrix, which are the variance and component directions, respectively (CS 357 Course Staff, 2025).
4. **Dimension Reduction:** PCA allows the reduction of the original features to a number of principal components, following the restriction that the number of original features must be greater than the number of principal components. The number of components can be selected by plotting to retain a percentage of the total variance with fewer dimensions (CS 357 Course Staff, 2025).
5. **Variance Ratio Explainability:** Given that each principal component has an associated eigenvalue, we can calculate the variance percentage of a component by dividing the eigenvalue by the sum of all eigenvalues. This is useful to determine how many components are required to adequately represent the data (Liberti Hub, 2024).
6. **Uncorrelated Features:** PCA guarantees that transformed features are uncorrelated to each other as PCA axes are orthogonal. This is useful in regression analysis and modeling where highly correlated predictors are problematic (IBM, 2023).
7. **Mean Centering and Scaling:** PCA requires that the data is mean centered; it can be achieved by subtracting the mean of each feature so that the first principal component passes through the data's mean. Features are often also scaled to univ variance so that the variance is not dominated by units of measurement (Raschka, 2025).
8. **Computational Efficiency (SVD):** PCA can utilize Singular Value Decomposition of the data matrix. This algorithm provides the principal components and their singular value (through the use of the square roots of eigenvalues), making PCA feasible for high-dimensional data (Yu et al., 2017).
9. **Rank Reduction and Noise Filtering:** PCA is useful in problems such as noise filtering or compressing as it can be used with the assumption that smaller variance components represent

noise. This allows for denoising or data compression with minimal errors and using fewer dimensions (Antonelli et al., 2004).

10. **Curse of Dimensionality Mitigation:** PCA aims to reduce the dimensions of the dataset, which helps to mitigate the curse of dimensionality by reducing the number of features, enhancing the model's performance. It also helps to prevent overfitting by removing extraneous features and will reduce computational and storage costs (IBM, 2023).
11. **Loss of Interpretability:** As stated before, the new variables of the dataset are combinations of the previous original features, which lose the clear meaning of the problem they were describing, increasing the complexity of interpreting the new data (CS 357 Course Staff, 2025).
12. **Linear Structure:** PCA works by capturing linear correlations in data, yet it may fail to analyze complex nonlinear relations as it looks for global linear variance. There are other variations, such as the Kernel PCA, to account for this problem, yet PCA will not be as effective as other nonlinear methods, such as t-SNE, to capture patterns like multimodal clusters (GeeksforGeeks, 2025).

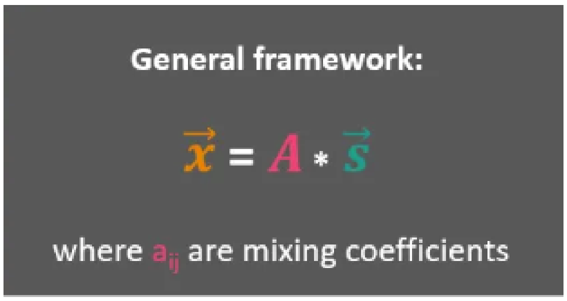
Industrial and Real World Applications

PCA is a very versatile tool and can be applied across various domains. For example, it can be used for image compression to preserve the important structures while reducing the image size. In genomics, the dimensionality of the dataset can be distilled to fewer components while identifying disease-related gene clusters. In finance PCA can help in risk management by analyzing related financial indicators, such as market index or interest rate movements. Moreover, it can help in marketing to do customer segmentation, where the key customer attributes can be reduced while highlighting the key dimensions. Finally, another common use is to be a preprocessing step in the machine learning pipelines to reduce feature dimension and mitigate overfitting (Liberti Hub, 2024). This helps a lot in algorithms such as regression or SVM, and its ability to eliminate multicollinearity makes it valuable in regression analysis. Overall PCA is an excellent tool for both exploratory analysis and a dimensional reduction technique (IBM, 2023).

Independent Component Analysis

Overview

Independent component analysis is a technique in unsupervised learning that unmixes and separates the blind sources. What it does is that it decomposes a multivariate signal into independent subcomponents in this way we can separate mixed signals into the original signals. The objective of this model is to find the linear transformation of data that can maximize the independence of the components.



To describe the mathematical representation, we can see it as vector notation, where vector **X** are the signals from vector **S,** multiplied with mixing coefficients, represented as matrix **A**

Image by Jonas Dieckmann on Medium

When having multiple variables, these signals can be seen as curves; nevertheless, when measuring the signals, we will receive a mixed linear combination. With ICA (Independent Component Analysis), we want to recover the original signals.

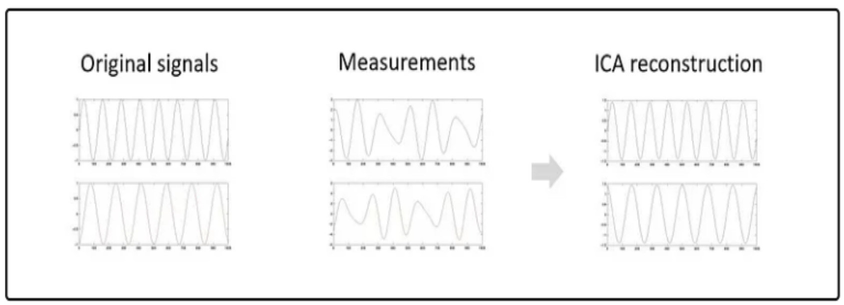


Image by Jonas Dieckmann on Medium

The cocktail party problem

In order to make this model more clear, there's a famous explanation for this. let's imagine we are at a party. We as humans don't have any problem distinguishing the voices of our friends, the music, and the sound of rain. In fact, it's a mix of multiple sounds with no relation to each other, and we can detect each signal separately, but with a model, this is not something easy, and we can't just know the different signals coming. This is what the model will do.

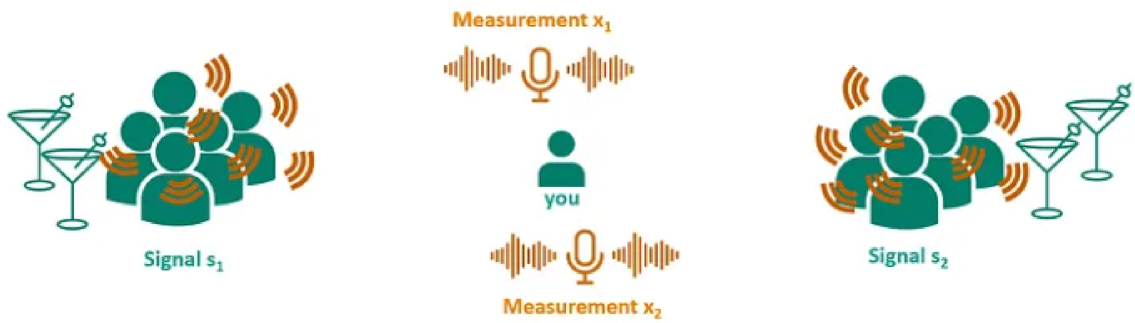


Image by Jonas Dieckmann on Medium

Features

1. **Statistical Independence:** The statistical independence between components means that the occurrence of one component does not provide information for the other ones.
2. **Non-Gaussianity:** The base of this feature is in the statistical explanation, that a sum of independent random variables tends to a gaussian distribution; if all sources are gaussian, the model would not be able to separate them
3. **Blind Source Separation:** This allows the separation of mixed signals without having any knowledge about the mixing process.
4. **No Orthogonality Constraint:** The model does not considerate the angles of the independent components; we only care about the statistical independence, so we can have any arbitrary angle and any scaling
5. **Higher-Order Statistics Optimization:** ICA uses higher-order statistics to achieve independence; this is a key difference between other models like PCA. Thanks to this, PCA is able to separate signals the other ones do not.
6. **Ambiguities in Scale and Order:** The model only determines independence by permutation and scaling; the model does not rank the signals by importance.
7. **Requires Whitening:** This means, before entering the processing we remove any correlation between the signals and scale our data to get the identity matrix and simplify the ICA algorithm
8. **ICA Algorithms:** We can use numerical methods to maximize the non-Gaussianity, like FastICA or infomaxICA.

9. **Applicability to Square or Overcomplete Cases:** Means that the number of observed mixes is equal to the number of independent sources. If we have more sources than mixtures makes the process more challenging.
10. **Independence vs Uncorrelated:** ICA uses independent components but not necessarily orthogonal or ordered as PCA
11. **Linearity and Limitations:** We assume a mix of sources. some of the possible limitations are the sound outside the signals and also large datasets
12. **Non-Parametric and Unsupervised:** ICA is a unsupervised training model; we don't need to categorize or expected output to supervise. it will discover the factors in data that are hidden, by pure statistics

Industrial and Real World Applications

It appears that ICA applies only for signals like audio, but it has wide-world applications, starting obviously with audio signal processing, as we saw, especially to separate voices or sounds that we need to get independent. For example, for telecommunications, also, in the area of biomedics, it's essential to separate the neural sources and clean brain signal data. In addition, for image processing, sometimes we can have overlapped images, so we can use ICA to separate the independent images or remove specific noise from an image. Also, in finance, we can find factors that change the driving price movement and get the specific source. As a last example, in data compression, the vectors that are independent can be obtained for the representation of the image compressed.

Singular Value Decomposition

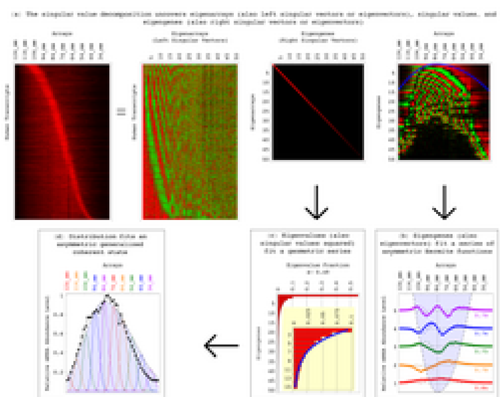
Singular Value Decomposition (SVD) is a core linear algebra process that reduces any input matrix, regardless of dimensions or symmetry properties, into three more manageable pieces. The theoretical decomposition explains the major structural properties contained within the original matrix and has wide application in theoretical models and practical applications.

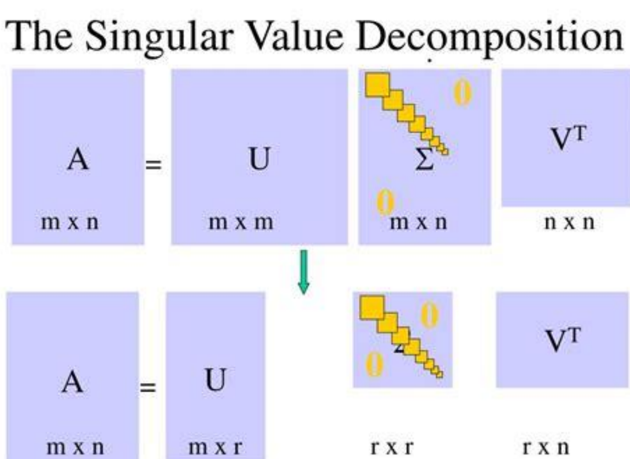
In essence, SVD separates a matrix into three parts:

- A segment describes the dominant trends or patterns that are seen in the data.
- The significance or strength of each of those patterns is determined
- The concluding part explains how these patterns fit into the overall framework.

The principal values in this process, which are known as singular values, represent the importance of every pattern. The values are always non-negative and are ranked in a decreasing manner, making it easier to see which components contribute most to the structure of the matrix.

SVD is significantly superior because of its ability to work with any type of matrix and extend the concept of eigenvalue decomposition that applies to certain types of square matrices. Geometrically, the procedure can be viewed as a combination of rotations and rescalings that transform one space into another.





Uses of SVD in the industry

SVD is widely used behind the scenes in many applications due to its power in simplifying linear problems. For recommender systems, it enables collaborative filtering, as used in the Netflix Prize solution. In natural language processing, SVD powers Latent Semantic Indexing to match queries and documents based on latent meaning. In computer vision, it enables image compression, image denoising, face detection, and 3D reconstruction. Signal processing applications utilize SVD for filtering signals and noise separation, particularly in sonar and radar. It is used by engineers for solving least squares problems, system calibration, and data compression. In finance, neuroscience, and climate science, SVD helps by dimensionality reduction and uncovering significant patterns.

Core Features of SVD

1. **Orthonormal Decomposition:** SVD decomposes any matrix into orthonormal sets of vectors that span the column and row spaces.
2. **Singular Values as Scaling Factors:** The singular values quantify how a matrix stretches or contracts data along each corresponding singular vector. The values are always nonnegative and ordered in a decreasing list.
3. **Rank Revelation:** The number of non-zero singular values equals the rank of the matrix. This makes SVD a precise tool for detecting linear dependencies and determining matrix rank.
4. **Optimal Low-Rank Approximation:** SVD provides the best approximation of a matrix with reduced rank by retaining the top singular values. This is optimal in terms of minimizing reconstruction error.
5. **Stable Pseudo-Inverse Computation:** Singular value decomposition allows the Moore-Penrose pseudoinverse to be computed in a way that is numerically stable, even if the matrix is ill-conditioned or not invertible.
6. **Numerical Robustness:** The decomposition is stable under small perturbations, and orthonormal components ensure that SVD handles near-singular or noisy matrices gracefully.
7. **Dimensionality Reduction and Latent Structure Discovery:** SVD allows projecting high-dimensional data onto a lower-dimensional space while preserving the most significant structure.

8. **Compression and Efficient Representation:** By keeping only the largest singular values and discarding the rest, SVD enables storage-efficient approximations of large matrices. This underpins practical compression techniques in images, signals, and text, achieving significant space savings with minimal quality loss.

9. **Noise Filtering and Signal Enhancement:** Small singular values often correspond to noise or irrelevant variation. By removing them, SVD can clean data while retaining essential patterns. This is useful in image denoising, sensor signal enhancement, and smoothing of experimental measurements.

10. **Insight into Matrix Geometry and Transformations:** SVD exposes the intrinsic geometry of a linear transformation by identifying the principal directions along which data is stretched or compressed. This helps in visualizing the action of a matrix and understanding its condition number, stability, and effect on vector norms.
11. **Support for Ill-Conditioned or Incomplete Problems:** SVD handles rank-deficient and non-square matrices robustly. It provides meaningful solutions to systems of equations that are underdetermined, overdetermined, or nearly singular—making it vital in inverse problems and regularization techniques.
12. **Adaptability via Variants:** Although classical SVD uses orthogonal components, it inspires variants like Non-negative Matrix Factorization or sparse approximations when interpretability or specific constraints are needed. These adaptations extend SVD's usefulness across specialized domains.

CONCLUSION

In this work, we explore how certain mathematical techniques help us handle large volumes of information more efficiently through the use of unsupervised machine learning. We observe that PCA, ICA, and SVD each offer their own advantages. PCA, for instance, allows us to reduce the dimensionality of data while preserving as much variance as possible, which is very useful for image compression, data visualization in 2D or 3D, and noise reduction in complex databases. ICA specializes in separating mixed signals and is commonly used in the analysis of biomedical signals as well as in noise removal from audio recordings. SVD, on the other hand, is widely used in recommendation engines, image compression, and information retrieval, as it enables the decomposition of a matrix into fundamental components and the construction of more manageable approximations that retain the essential information.

Thanks to these techniques, we can preserve the most relevant aspects of the data and eliminate what introduces "noise," thereby improving both computational efficiency and the interpretability of models in applications we use in our daily lives, from recommendations on social media and streaming services to the processing of images and audio signals.

It is truly impressive to see how mathematics and programming work together to make technology faster and smarter. Without a doubt, these tools will continue to be essential in the future, as we are constantly generating data around the world. Understanding and applying these techniques will provide future generations with the ability to transform raw data into useful insights and predictive models—making their study and understanding well worth the effort.

REFERENCES

CS 357 Course Staff. (2025). *Principal Component Analysis (PCA)*. University of Illinois at Urbana-Champaign. https://cs357.cs.illinois.edu/textbook/notes/pca.html

IBM. (2023). *What is principal component analysis (PCA)?* https://www.ibm.com/think/topics/principal-component-analysis

Liberti Hub. (2024, November 8). *Principal Component Analysis (PCA): An intuitive guide for everyone*. Medium. https://medium.com/@libertihub/principal-component-analysis-pca-an-intuitive-guide-for-everyone-5ea5431d7c17 Medium

Antonelli, P., Revercomb, H., Tobin, D., Sromovsky, L., Howell, H., Garcia, R., Huang, A., Best, F. A., Dutcher, S., & Smith, W. L. (2004). *A principal component noise filter for high spectral resolution infrared measurements*. Journal of Geophysical Research: Atmospheres, 109(D23). https://doi.org/10.1029/2004JD004862

Raschka, S. (2025, May 12). Does mean centering or feature scaling affect a principal component analysis? https://sebastianraschka.com/faq/docs/pca-scaling.html

Yu, W., Gu, Y., Li, J., Liu, S., & Li, Y. (2017). Single-pass PCA of large high-dimensional data. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI-17)* (pp. 3350–3356). International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2017/468

Vaidya, P. G., Anand, S. P. S., & Nagaraj, N. (2007). *A Non-linear Generalization of Singular Value Decomposition and its Application to Cryptanalysis*. arXiv preprint arXiv:0711.4910.

Roy, S., Ghosh, A., & Basu, A. (2023). *Analysis of the rSVDdpd Algorithm: A Robust Singular Value Decomposition Method using Density Power Divergence*. arXiv preprint arXiv:2307.10591.

Santos, A. R., Santos, M. A., Baumbach, J., McCulloch, J. A., Oliveira, G. C., Silva, A., Miyoshi, A., & Azevedo, V. (2011). *A singular value decomposition approach for improved taxonomic classification of biological sequences*. BMC Genomics, 12(Suppl 4), S11. https://doi.org/10.1186/1471-2164-12-S4-S11

GeeksforGeeks. (2025, May). *Difference between PCA vs t-SNE*. https://www.geeksforgeeks.org/difference-between-pca-vs-t-sne/