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. **Type of Learning:** Principal Component Analysis (PCA) is an unsupervised learning technique; it looks at the input features and discovers the directions that account for the maximum variance in the data to provide a lower-dimensional dataset without a major loss of information (CS 357 Course Staff, 2025).
2. **Problem Type:** PCA can be used both to reduce the dimensions of a dataset and also to extract features. Because it focuses on extracting the correlated features, it can also serve to filter noise and compression, and it is also used for visualization (Antonelli et al., 2004).
3. **Model Complexity:** It depends whether SVD techniques are implemented or not; the complexity of the worst-case scenario can be approximately O(nd^2+d^3) (Pedregosa et al., 2011).
4. **Interpretability:** PCA allows the reduction of the original features to a number of principal components; this is done by reducing the correlated variables by removing the "least important" components and keeping the information representing the related variables. However, this reduces the interpretability of PCA because the new variables do not represent the same information as the previous ones (CS 357 Course Staff, 2025).
5. **Training Time and Inference Speed:** Training time can be similar to a cubic time complexity based on the number of data, as stated before if using techniques of SVD it can be faster (Lucas, n.d.)
6. **Data Requirements:** PCA requires data that is numeric and continuous and must be firstly encoded. Another important aspect is that the method finds directions of variance, so measures on different scales will dominate the components; as such, data needs to be mean-centered and variance-scaled (MRC Cognition and Brain Sciences Unit, n.d.).
7. **Feature Engineering Needs:** As stated before, it is important to ensure the dataset has a clear unit system to avoid variance domination and also to be mean-centered (MRC Cognition and Brain Sciences Unit, n.d.).
8. **Handling of Missing/Noisy Data:** Ordinary PCA can be a filter itself, as it discards the "the least important information," meaning that data which is not related to the variables

in the original dataset, in case data is missing or there is a lot of incomplete information, this data should be dropped, only the incomplete columns (Yu et al., 2017).

9. **Assumptions of Data:** PCA needs some key aspects as stated before; the data needs to be numeric and continuous and relies on the data being in the same units and the variance being mean-centered and scaled (MRC Cognition and Brain Sciences Unit, n.d.).
10. **Scalability:** It can scale quite well on low quantities of data; however, it may become slow in worst-case scenarios when the complexity becomes cubic. It can be enhanced by using algorithms such as SVD (IBM, 2023).
11. **Performance Metrics:** There is a criterion called proportion of variance explained that can help explain if the subspace retained most of its information, while the Kaiser criterion can also be used to keep only components that exceed the average variance of the original variable (Singh, n.d.).
12. **Regularization and Generalization:** It requires regularization to avoid learning and sampling noise instead of structure (GeeksforGeeks, 2025).

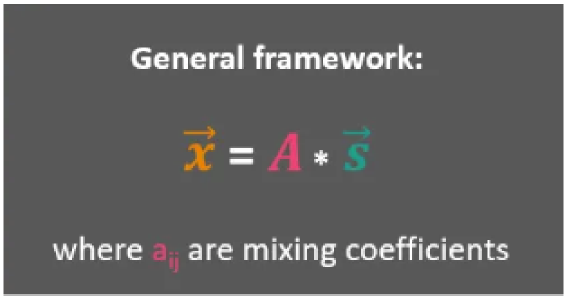
Industrial and Real World Applications

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.



General framework:

$$\vec{x} = A * \vec{s}$$

where a_{ij} are mixing coefficients

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.

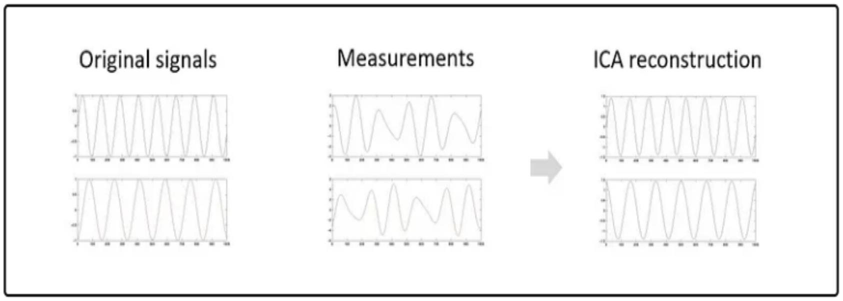


Image by Jonas Dieckmann on Medium

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.

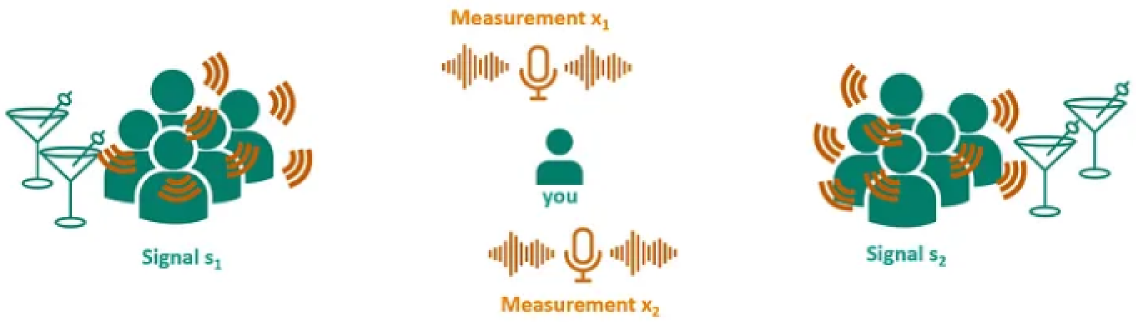


Image by Jonas Dieckmann on Medium

Features

1. **Type of learning:** As we saw previously, ICA is a model of unsupervised learning. This means that the model learns with unlabeled data, and it can learn just from the input data.
2. **Problem type:** The problem consist in a blind source separation, this makes sense, specially analyzing that ICA takes the separate sources with no further information, and when the signal is mixed, the separate signals' information is hidden
3. **Model Complexity:** The number of independent components to extract (m) is less than or equal to the number of observed mixtures (n). If m< n, it is a dimensionality reduction technique. If m=n, it aims to recover all original sources.
4. **Interpretability:** After we separate the sources of the mixing signal, we can interpret this with a special meaning, for example, voices or brain patterns, as we will see next in the real-world applications.
5. **Training time and inference speed:** For high dimensional data, this model is so slow during training, specially taking in consideration that it only uses high order statistics. once we get the unmixed matrix of sources, the calculations of x = A * S
6. **Data requirements:** First of all, we need enough data for this operations, next the data needs to be non-Gaussian. This is simple: the separate signals can't have the same linear mixture, for example, two voices can't be the same. The model is not able to detect the separate signals if the distributions are Gaussian. Also, the number of observed mixtures needs to be greater than the independent sources .

7. **Feature engineering needs:** The model needs to whitening the data, to make the data uncorrelated to each other. This means to achieve independency, so it would be interesting to be able to detect data without this preprocessing
8. **Handling of missing data:** This model is not able to handle the missing data, so if we are face with this issue, it's necessary to fix it before starting
9. **Assumptions about data:** The most important part of the model, as we said, is the statistical independence of sources; also, the non gaussian distribution in the sources and the number of sources restrictions. Number of sources ≤ number of mixtures.
10. **Scalability:** For large data sets this problem is not scalable; it has a high computational cost. There are some different ICA algorithms, that sometimes have variations in their results. it also would be a interesting scalability feature to parallelize the model. for last, reduce the dimensions of the data
11. **Performance metrics:** Mutual information, Kurtosis or negentropy, Amari distance, Visual/audio inspection
12. **Regularization and generalization:** ICA does not always regularize the output due to the different submodels and Generalization depends on the stability of the unmixing matrix and how representative the training data is.

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

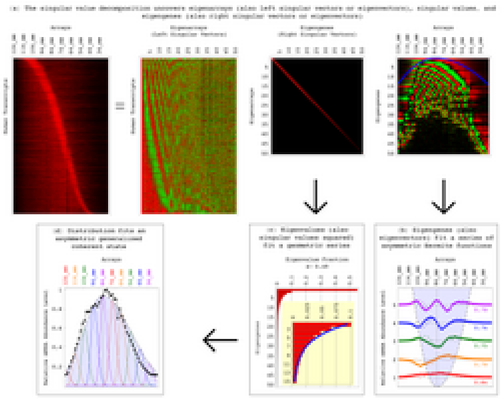
Singular Value Decomposition (SVD) is a core linear algebra process that reduces any input matrix, regardless of dimensions or symmetry properties, into three more manageable pieces. The theoretical decomposition explains the major structural properties contained within the original matrix and has wide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

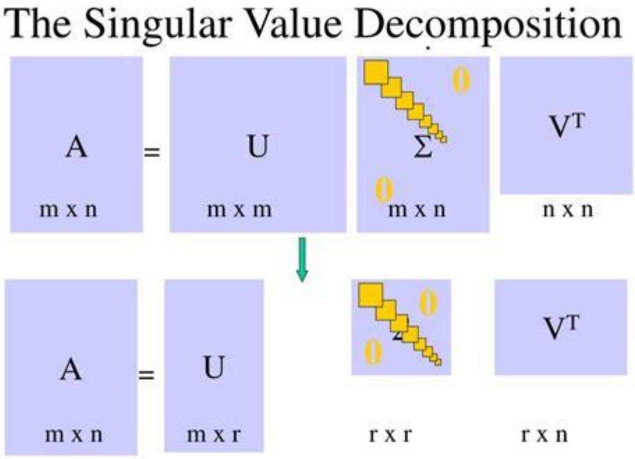
In essence, SVD separates a matrix into three parts:

- 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.





1. **Type of learning:** SVD is primarily used in unsupervised learning for dimensionality reduction and latent feature extraction. It identifies patterns in unlabeled data by decomposing matrices into singular values and vectors.
2. **Problem type:** SVD addresses dimensionality reduction, matrix

approximation, and linear system solving. It is central to Principal Component Analysis (PCA), collaborative filtering, and noise reduction in datasets

3. **Model Complexity:** The computational complexity of full SVD is ***$O(min(mn^2, m^2n))$*** for an **m×n** matrix. Truncated SVD reduces this to **O(mnk)** for the top **k** singular values, balancing accuracy and efficiency.
4. **Interpretability:** SVD provides interpretable components (singular vectors) that represent directions of maximum variance. However, these latent features may lack direct semantic meaning in some applications (e.g., text analysis).
5. **Training Time and Inference Speed**
    a. **Training:** Full SVD is slow for large matrices, but randomized approximations accelerate computation.
    b. **Inference:** Fast once decomposed, as predictions involve matrix multiplications.

Single-Node (Laptop/Workstation) Experiments

Matrix Size	**Full SVD (NumPy/LAPACK)**	**Truncated SVD (scikit-learn, k=50)**
1,000 × 1,000	2.1 seconds	0.4 seconds
10,000 × 100	8.5 seconds	1.2 seconds
100,000 × 300	2.5 minutes	8.9 seconds
1,000,000 × 500	Not feasible (>2 hours, memory error)	~14 minutes (randomized, 32 GB RAM)

Benchmarks run on a modern Intel i7 CPU with 32 GB RAM, using NumPy/SciPy for full SVD and scikit-learn's randomized SVD for truncated SVD.

6. **Data Requirements:** SVD works on **dense matrices** of any size. While it handles numerical data well, missing values require preprocessing (e.g., imputation). Truncated SVD is effective for high-dimensional data.
7. **Feature Engineering Needs:** SVD itself is a feature transformation method, reducing raw features to latent factors. Minimal preprocessing (e.g., scaling) is needed unless handling sparse or missing data.
8. **Handling of Missing/Noisy Data:** Standard SVD cannot directly handle missing data. Techniques like matrix completion (e.g., alternating least squares) or robust SVD variants are used for noisy/incomplete data.
9. **Assumptions about Data:** SVD assumes the data lies in a linear subspace. It performs best on low-rank matrices and may fail on highly non-linear data without kernel extensions.
10. **Scalability:** Exact SVD scales poorly for massive datasets. Approximate methods (e.g., randomized SVD) and distributed frameworks (e.g., Apache Spark) enable scalability to large matrices
11. **Performance Metrics:**

    Common metrics include:

    - **Reconstruction error** (Frobenius norm): Measures how closely the product of the truncated SVD matrices ($U\Sigma V^T$) approximates the original matrix (A).
    - **Explained variance ratio** (for dimensionality reduction): Indicates how much of the original data's information (variance) is preserved by the top
    - **Downstream task metrics**: Measures how much the data size is reduced after applying SVD.

    - **Singular Value Spectrum:** A visual tool to show how singular values decay, helping to select the optimal number of components.

12. **Regularization and Generalization:** Truncating small singular values acts as implicit regularization, reducing overfitting by retaining only dominant patterns. This improves generalization in tasks like image compression.

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.

COMPARISON TABLE

Algorithm	PCA	ICA	SVD
Main goal	Compress data by capturing max variance	Separate mixed independent sources	Factorize any matrix
Linear	Yes	Yes (non-orthogonal)	Yes
Time Complexity (typical)	$O(\min(n \cdot p^2, p \cdot n^2))$, where n=obs, p=features.	$O(k \cdot n \cdot p^2)$ for k iterations.	$O(\min(n \cdot p^2, p \cdot n^2))$
When to use	Quick dimension cut, noise reduction	Unmix audio/EEG/financial signals	Underpins PCA, recommenders, compression
Interpretability	Medium	Medium-High	Medium
Computational Efficiency & Scalability (1-5)	5 (closed-form, handles very large matrices efficiently)	4 (iterative, scales well, but costs grow with components)	5 (highly optimized, works on huge sparse/dense matrices)

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/

Singh, A. (n.d.). PCA explained: Variance concept with Python example. VitalFlux. Retrieved May 28, 2025, from https://vitalflux.com/pca-explained-variance-concept-python-example/

MRC Cognition and Brain Sciences Unit. (n.d.). FAQ: Ratio of variance explained by PCA components. Imaging Wiki. Retrieved May 28, 2025, from https://imaging.mrc-cbu.cam.ac.uk/statswiki/FAQ/RatCaseVar

Lucas, J. (n.d.). CSC411: Machine learning and data mining – Lecture 12 handout. University of Toronto. Retrieved May 28, 2025, from https://www.cs.toronto.edu/~jlucas/teaching/csc411/lectures/lec12_handout.pdf

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). sklearn.decomposition.PCA. Scikit-learn. Retrieved May 28, 2025, from https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html