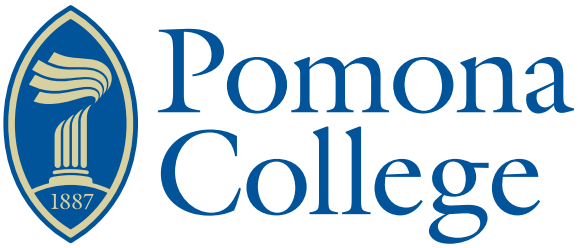


Senior Thesis in Mathematics

Exploration of Dimension Reduction Methods: Theory and Applications

Author:
Necdet Canim

Advisor:
Prof. Johanna S. Hardin

Submitted to Pomona College in Partial Fulfillment
of the Degree of Bachelor of Arts

May 5, 2025

Abstract

Having a greater number of dimensions in a dataset does not always lead to more efficient analysis results. In this study, the author explores different methods for dimension reduction. The specific methods focused are Principal Component Analysis (PCA), Contrastive Principal Component Analysis (cPCA), Independent Component Analysis (ICA), and Nonnegative Matrix Factorization (NMF). These techniques are applied to different datasets to provide a comparative analysis of their strengths and weaknesses. The mathematical foundations of each method are examined in detail, followed by systematic application to diverse datasets from autism screening, online streaming analytics, speech processing, and synthetic high-dimensional simulations. Each method's performance is evaluated based on Frobenius and L1 norm reconstruction error and interpretability across these datasets. The results demonstrate that no single technique outperforms others across all contexts. Instead, method suitability depends on the data structure and the analysis goal. The thesis highlights the importance of aligning method selection with research objectives and data characteristics, while also offering insights into future research directions.

Contents

Chapter 1

Introduction

In modern data science, the increasing complexity and dimensionality of datasets present significant challenges for analysis, visualization, and interpretation. High-dimensional data often contain redundant or irrelevant information, which complicates data processing and can obscure meaningful patterns. For instance, in fields such as genomics, finance, and image processing, datasets with hundreds or even thousands of variables are common, yet only a small fraction of these variables may contribute to the underlying structure or trends. Extracting only the necessary information—including the contributing variables—is of utmost importance.

In recent years, dimension reduction methods have emerged as tools to address this dimensionality challenge. These techniques aim to transform high-dimensional data into a lower-dimensional space while retaining as much relevant information as possible. By reducing dimensions, these methods simplify data representation, enhance computational efficiency, and improve model performance. Additionally, they enable the visualization of complex datasets in lower dimensions.

1.1 Problem: Curse of Dimensionality

Curse of dimensionality is the challenge that arises when analyzing high-dimensional datasets. As the number of dimensions increases, the volume of the search space expands exponentially, making the data points increasingly sparse. This phenomenon impacts a variety of domains, including machine learning, image recognition, and bioinformatics, where high-dimensional data is common.

One of the key consequences of the curse of dimensionality is the growing difficulty in effectively sampling the data space. As dimensionality increases, the amount of data required to achieve reliable results grows at a higher rate. This issue, often called

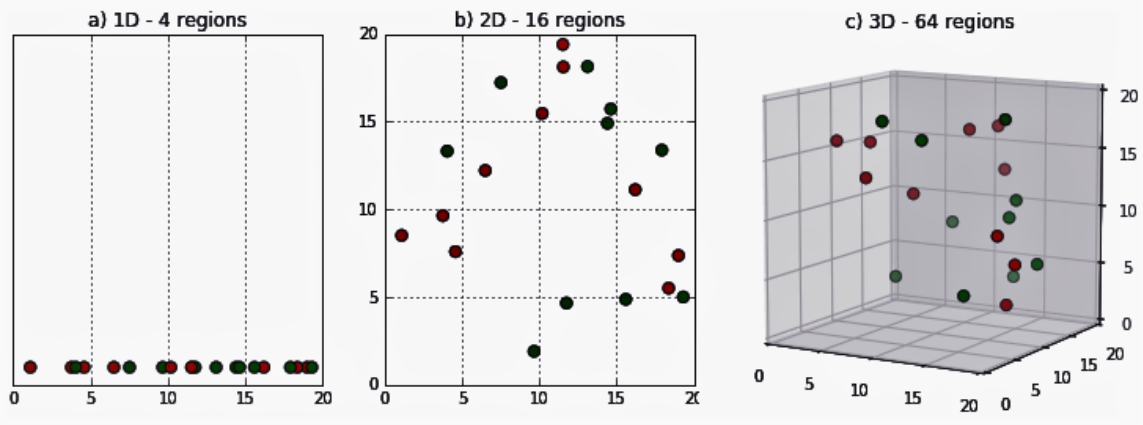


Figure 1.1: Illustration of how increasing dimensionality affects the relative volume occupied by data points. In higher dimensions, the data points become sparse, leading to difficulties in classification and clustering tasks [Raj, 2021].

the Hughes phenomenon, was extensively studied by [Shahshahani and Landgrebe, 1994], who demonstrated how sparsity in high-dimensional spaces can degrade the performance of classification algorithms.

Figure 1.1 shows data points in 1D, 2D, and 3D search spaces. As the search space grows in dimensions, the ratio of the search space to the area that the desired data points occupy increases. This ratio becomes even greater when the comparison is extended to dimensions greater than 3.

1.2 A Brief Survey of Dimension Reduction Methods

Dimension reduction techniques are crucial in modern data analysis to solve the *curse of dimensionality*. There are AI-supported, computational, theoretical, and practical techniques used to reduce the dimensions of a dataset. The methods that are explored in this thesis include the Standard Principal Component Analysis (PCA), Contrastive Principal Component Analysis (cPCA), Independent Component Analysis (ICA), and Non-Negative Matrix Factorization (NMF). Gaining a better understanding of the mathematical foundations behind these methods is important to get a sense of which method thrives under which conditions.

Principal Component Analysis (PCA) is one of the most widely used dimension reduction techniques. It identifies the directions that maximize the variance of the

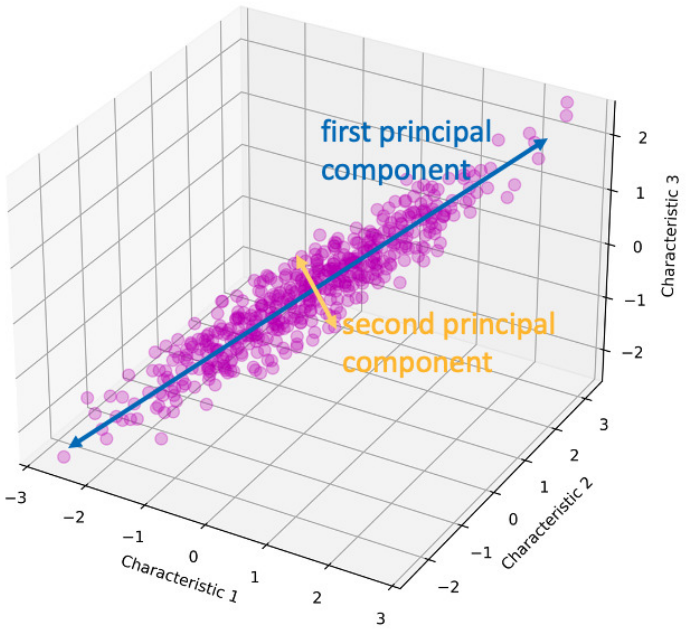


Figure 1.2: Graphical representation of Principal Component Analysis (PCA), showing how data points are projected onto principal components that maximize variance while reducing dimensionality [Follette, 2023].

data projections, capturing the most significant patterns. Figure 1.2 shows what that visually means for an example dataset.

Contrastive PCA (cPCA) is an extension of Principal Component Analysis (PCA) designed to identify the principal components that differentiate one dataset (the target dataset) from another (the background dataset). Compared to traditional PCA, which finds the directions of maximum variance in a single dataset, cPCA focuses on identifying components that capture variance unique to the target dataset.

Independent Component Analysis (ICA) is a technique used for separating a signal into independent components. Unlike PCA, ICA aims to find components that are independent of each other. Figure 1.3 is a flowchart summarizing the ICA process. (See Chapter 2: Theory for a detailed exploration of this process.)

Non-Negative Matrix Factorization (NMF) is a method for decomposing a non-negative matrix into two lower-dimensional non-negative matrices. Unlike PCA and ICA, which allow negative values in the factors, NMF has a non-negativity constraint.

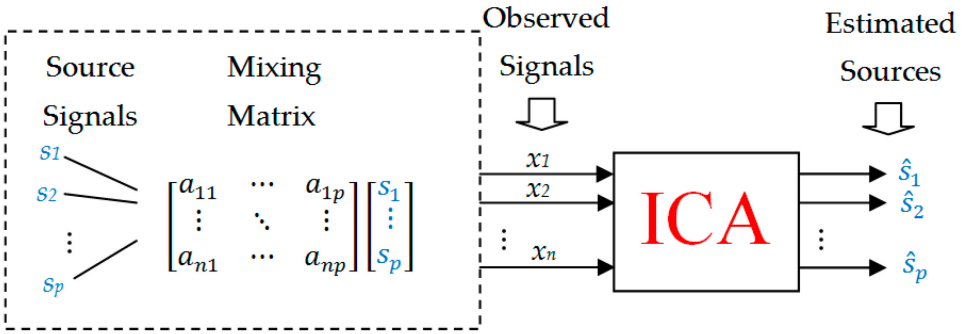


Figure 1.3: Diagram of the Independent Component Analysis (ICA) process, illustrating the separation of mixed signals into statistically independent components using matrix factorization techniques [Mika et al., 2020].

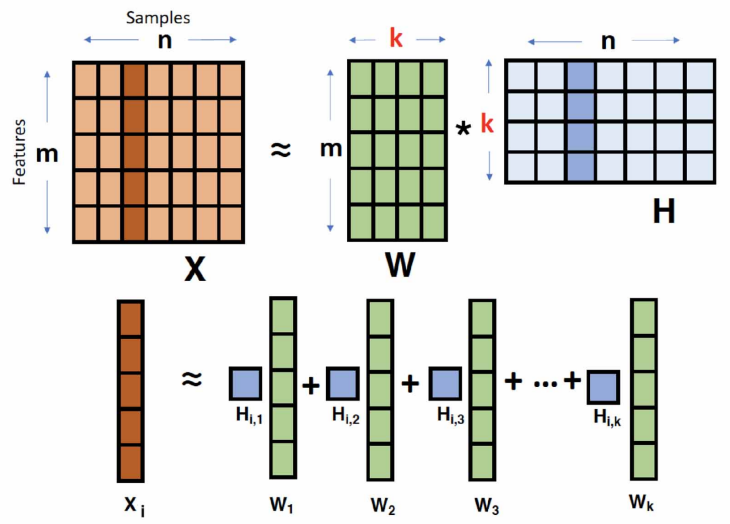


Figure 1.4: Conceptual depiction of Non-Negative Matrix Factorization (NMF), showing how an input matrix is decomposed into two lower-dimensional non-negative matrices, revealing latent structures in the data [Nebgen et al., 2021].

Figure 3.1 visualizes the matrices involved in the NMF process.

1.3 Prior Work and Literature Review

Because they are useful while analyzing high dimension datasets, dimension reduction methods have been extensively studied in data science and mathematics.

Principal Component Analysis (PCA) remains one of the most widely used dimension reduction methods. [Gewers et al., 2021] highlights PCA's role in simplifying high-dimensional datasets. [Kokiopoulou et al., 2011] explore trace optimization and eigenproblems, which are central to methods like PCA. Another operation that is crucial for PCA is accurate covariance matrix estimation. [Ledoit and Wolf, 2003] introduce new techniques to improve covariance matrix estimates.

While PCA is powerful, it has limitations in identifying unique patterns in the specific setting of the presence of background noise. To address this, [Abid et al., 2018] introduced Contrastive Principal Component Analysis (cPCA), which enhances PCA by comparing a target dataset against a background dataset to isolate target-specific variance.

The statistical principles of ICA are detailed in [Nordhausen and Oja, 2018], providing a comparative perspective on how ICA differs from cPCA in applications where signals overlap.

[Baron, 2019] and [Ivezić et al., 2020] illustrate the practical applications of dimension reduction methods, particularly in astronomy. Baron's review demonstrates how PCA is applied to analyze astronomical datasets. Similarly, Ivezić provides a detailed discussion of the relevance of PCA and ICA to high-dimensional astronomical data.

1.4 Thesis Objectives

The overarching goals of this thesis are to understand the mathematics behind the dimension reduction methods better and to compare them by applying them on different datasets from various fields.

This thesis explores a range of dimension reduction techniques, including linear methods like Principal Component Analysis (PCA), its contrastive variant (cPCA), Independent Component Analysis (ICA) and Nonnegative Matrix Factorization (NMF). These methods have proven to be critical to address *curse of dimensionality* in various data science and mathematics projects.

Sub-objectives include exploring the mathematical background behind various dimension reduction techniques, comparing their assumptions, and evaluating their effectiveness on specific datasets.

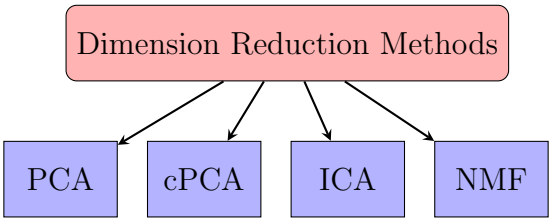


Figure 1.5: Flowchart of Dimension Reduction Methods that the author aims to explore. These methods are PCA, cPCA, ICA, and NMF.

By the end of this thesis, reader will gain a deeper understanding of Principal Component Analysis (PCA), Contrastive Principal Component Analysis (cPCA), Independent Component Analysis (ICA), and Nonnegative Matrix Factorization (NMF).

Chapter 2

Theory

This section explores the mathematical foundations of the dimension reduction methods in detail, providing a comprehensive understanding of their principles and assumptions. I begin with classical method Principal Component Analysis (PCA) and then extend to more modern approaches Contrastive PCA (cPCA), Independent Component Analysis (ICA), and Non-negative Matrix Factorization (NMF).

2.1 Standard Principal Component Analysis (PCA)

PCA aims to capture the most variation in a dataset with the fewest number of dimensions to retain as much information as possible while reducing the dimension. Mathematically, this is achieved by finding a unit projection vector $r \in \mathbb{R}^m$ that maximizes the variance of the projected data. Let $X \in \mathbb{R}^{n \times m}$ be a mean-centered data matrix with n data points and m features (dimensions). The projected data $Xr \in \mathbb{R}^n$ represents a 1D projection of the original data onto the direction of r. When I project each data point in X onto r, I obtain a single scalar value per sample. X's symmetric square covariance matrix $\text{Cov}(X) \in \mathbb{R}^{m \times m}$ is defined as

$$\text{Cov}(X) = \frac{1}{n} X^T X \tag{2.1}$$

where $X^T \in \mathbb{R}^{m \times n}$ is the transpose of X.

Then, the variance of the projected data Xr (a scalar) is

$$\text{Var}(Xr) = r^T \text{Cov}(X) r \tag{2.2}$$

Our goal is to maximize this variance subject to the constraint that r is a unit vector, meaning

$$r^T r = 1 \tag{2.3}$$

To maximize $r^T \text{Cov}(X) r$ under the constraint $r^T r = 1$, I introduce a Lagrange multiplier λ and define the Lagrangian:

$$\mathcal{L}(r, \lambda) = r^T \text{Cov}(X) r - \lambda(r^T r - 1) \tag{2.4}$$

To find the extrema, I take the derivative of $\mathcal{L}$ with respect to r and set it equal to zero vector. Given a scalar function $f : \mathbb{R}^d \to \mathbb{R}$ that depends on a vector $r \in \mathbb{R}^d$, the gradient of $f(r)$ with respect to r is a column vector. In our case, for the Lagrangian, this means

$$\frac{\partial \mathcal{L}}{\partial r} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial r_1} \\ \frac{\partial \mathcal{L}}{\partial r_2} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial r_d} \end{bmatrix} \in \mathbb{R}^d$$

This gradient measures how small changes in each element of r affect $\mathcal{L}$. Now, I can show the derivative as

$$\frac{\partial}{\partial r} \mathcal{L}(r, \lambda) = \frac{\partial}{\partial r} \left(r^T \text{Cov}(X) r - \lambda(r^T r - 1) \right) = \mathbf{0} \tag{2.5}$$

Using the matrix calculus identity $\frac{\partial}{\partial r}(r^T A r) = 2Ar$ (for a square symmetric matrix A), I get

$$2\text{Cov}(X) r - 2\lambda r = \mathbf{0} \tag{2.6}$$

Simplifying the above expression gives

$$\text{Cov}(X) r = \lambda r \tag{2.7}$$

I showed that the covariance matrix $\text{Cov}(X)$ times the projection vector r gives the eigenvalue equation. This shows that r must be an eigenvector of the covariance matrix $\text{Cov}(X)$, and λ is the corresponding eigenvalue. This is the first principal component.

After finding the first principal component, I want to compute additional principal components. The k-th principal component is the direction that maximizes the variance of the projected data, while being orthogonal to all previously computed components.

The k-th principal component (which I denote by r_k) is found by maximizing the variance $\text{Var}(X r_k) = r_k^T \text{Cov}(X) r_k$, subject to the constraints that:

- r_k is a unit vector: $r_k^T r_k = 1$,
- r_k is orthogonal to the previously computed components: $r_k^T r_j = 0$ for all $j < k$.

These conditions ensure that each new principal component captures variance in a direction not explained by the earlier components.

To incorporate the new constraints, I use Lagrange multipliers again. For the unit length constraint, I have a multiplier λ_k, and for each orthogonality constraint, I have the Lagrange multiplier μ_j for $j = 1, 2, \ldots, k-1$. So, the Lagrangian for the k-th principal component becomes

$$\mathcal{L}(r_k, \lambda_k, \mu_1, \ldots, \mu_{k-1}) = r_k^T \mathrm{Cov}(X) r_k - \lambda_k (r_k^T r_k - 1) - \sum_{j=1}^{k-1} \mu_j (r_k^T r_j) \tag{2.8}$$

To find the extrema, I take the derivative of the Lagrangian with respect to the vector r_k

$$\frac{\partial \mathcal{L}}{\partial r_k} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial (r_k)_1} \\ \frac{\partial \mathcal{L}}{\partial (r_k)_2} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial (r_k)_d} \end{bmatrix} \in \mathbb{R}^d$$

and set it equal to zero vector:

$$\frac{\partial \mathcal{L}}{\partial r_k} = \frac{\partial}{\partial r_k} \left(r_k^T \mathrm{Cov}(X) r_k - \lambda_k (r_k^T r_k - 1) - \sum_{j=1}^{k-1} \mu_j (r_k^T r_j) \right) = \mathbf{0} \tag{2.9}$$

The derivative of the $r_k^T \mathrm{Cov}(X) r_k$ term is

$$\frac{\partial}{\partial r_k} (r_k^T \mathrm{Cov}(X) r_k) = 2\mathrm{Cov}(X) r_k \tag{2.10}$$

Next, the derivative of $-\lambda_k (r_k^T r_k - 1)$ is

$$\frac{\partial}{\partial r_k} \left(-\lambda_k (r_k^T r_k - 1) \right) = -2\lambda_k r_k \tag{2.11}$$

The derivative of the last term $-\sum_{j=1}^{k-1} \mu_j (r_k^T r_j)$ is

$$\frac{\partial}{\partial r_k}\left(-\sum_{j=1}^{k-1}\mu_j(r_k^T r_j)\right) = -\sum_{j=1}^{k-1}\mu_j r_j \tag{2.12}$$

Thus, the gradient of the Lagrangian is

$$2\text{Cov}(X)r_k - 2\lambda_k r_k - \sum_{j=1}^{k-1}\mu_j r_j = \mathbf{0} \tag{2.13}$$

Dividing by 2 and modifying the equation yield

$$\text{Cov}(X)r_k = \lambda_k r_k + \frac{1}{2}\sum_{j=1}^{k-1}\mu_j r_j \tag{2.14}$$

This is the equation for the k-th principal component.

Let's start by considering the case k=2: I know that r_1 is the first eigenvector of $\text{Cov}(X)$. Now, I want to find the second principal component r_2.

$$\text{Cov}(X)r_2 = \lambda_2 r_2 + \frac{1}{2}\mu_1 r_1 \tag{2.15}$$

Multiplying both sides with r_1^T yields

$$r_1^T\text{Cov}(X)r_2 = \lambda_2 r_1^T r_2 + \frac{\mu_1}{2}r_1^T r_1 = \lambda_2 r_1^T r_2 + \frac{\mu_1}{2} \tag{2.16}$$

because $r_1^T r_1 = 1$.

Our constraint states that r_1 and r_2 are orthogonal, $r_1^T r_2 = 0$, which implies that

$$r_1^T\text{Cov}(X)r_2 = 0 + \frac{\mu_1}{2} \Rightarrow \frac{\mu_1}{2} = 0 \Rightarrow \mu_1 = 0 \tag{2.17}$$

Thus, I have

$$\text{Cov}(X)r_2 = \lambda_2 r_2 + \frac{1}{2}\mu_1 r_1 \Rightarrow \text{Cov}(X)r_2 = \lambda_2 r_2 \tag{2.18}$$

which confirms that r_2 is the eigenvector associated with λ_2.

Now, for the case $k = 3$, to find the third principal component r_3, I start with

$$\text{Cov}(X)r_3 = \lambda_3 r_3 + \frac{1}{2}(\mu_1 r_1 + \mu_2 r_2) \tag{2.19}$$

By multiplying both sides of the equation with r_1^T, I get

$$r_1^T \text{Cov}(X) r_3 = \lambda_3 r_1^T r_3 + \frac{1}{2}(\mu_1 r_1^T r_1 + \mu_2 r_1^T r_2) \tag{2.20}$$

Since $r_1^T r_1 = 1$, $r_1^T r_2 = 0$, and $r_1^T r_3 = 0$,

$$r_1^T \text{Cov}(X) r_3 = \frac{1}{2}\mu_1 \tag{2.21}$$

Similarly, multiplying both sides of the equation (2.19) with r_2^T instead of r_1^T yields

$$r_2^T \text{Cov}(X) r_3 = \lambda_3 r_2^T r_3 + \frac{1}{2}(\mu_1 r_2^T r_1 + \mu_2 r_2^T r_2) \tag{2.22}$$

Since $r_2^T r_2 = 1$, $r_2^T r_1 = 0$, and $r_2^T r_3 = 0$,

$$r_2^T \text{Cov}(X) r_3 = \frac{1}{2}\mu_2 \tag{2.23}$$

This means I have

$$r_1^T \text{Cov}(X) r_3 = \frac{1}{2}\mu_1 \quad \text{and} \quad r_2^T \text{Cov}(X) r_3 = \frac{1}{2}\mu_2 \tag{2.24}$$

Since r_3 is orthogonal to r_1 and r_2 by our constraints, these terms simplify to 0, leading to

$$\mu_1 = 0 \quad \text{and} \quad \mu_2 = 0 \tag{2.25}$$

Therefore, equation (2.19) becomes

$$\text{Cov}(X) r_3 = \lambda_3 r_3 \tag{2.26}$$

which confirms that r_3 is the eigenvector associated with λ_3.

This completes the derivation for the k-th principal component up to $k = 3$, and the greater k cases follow analogous proofs.

In this section, I provided a full derivation of the PCA optimization problem using Lagrange multipliers, showing how the principal components arise as eigenvectors of the data covariance matrix. I extended the derivation to include the general case of the k-th principal component under orthogonality constraints, proving the eigenstructure iteratively up to $k = 3$. This foundation serves as a benchmark for later extensions and comparisons.

2.2 Contrastive Principal Component Analysis (cPCA)

cPCA's objective is to maximize the contrastive variance to highlight the difference between target and background datasets with the fewest number of components.

Given a target dataset $X_t \in \mathbb{R}^{n_t \times d}$ and a background dataset $X_b \in \mathbb{R}^{n_b \times d}$, let $\Sigma_t = \text{Cov}(X_t) \in \mathbb{R}^{d \times d}$ and $\Sigma_b = \text{Cov}(X_b) \in \mathbb{R}^{d \times d}$ be the symmetric square covariance matrices of the target and background datasets, where n_t and n_b are number of data points in the target and background datasets, respectively.

The contrastive variance is defined as

$$r^T(\Sigma_t - \alpha\Sigma_b)r \tag{2.27}$$

where α is a user defined scalar parameter that controls the balance between target-specific variance and background variance. Because Σ_t and Σ_b are symmetric, $\Sigma_t - \alpha\Sigma_b$ is symmetric, too.

Analogous to the standard PCA, to find the first principal component, I solve the eigenvalue problem:

$$(\Sigma_t - \alpha\Sigma_b)r = \lambda r \tag{2.28}$$

To obtain the k-th principal component in cPCA, I maximize the contrastive variance in a direction orthogonal to the previously computed components. Analogous to the standard PCA, the Lagrangian for the k-th contrastive principal component is:

$$\mathcal{L}(r_k, \lambda_k, \mu_1, \ldots, \mu_{k-1}) = r_k^T(\Sigma_t - \alpha\Sigma_b)r_k - \lambda_k(r_k^T r_k - 1) - \sum_{j=1}^{k-1} \mu_j(r_k^T r_j) \tag{2.29}$$

I can now take the derivative of the Lagrangian with respect to r_k and set it equal to zero:

$$\frac{\partial \mathcal{L}}{\partial r_k} = 2(\Sigma_t - \alpha\Sigma_b)r_k - 2\lambda_k r_k - \sum_{j=1}^{k-1} \mu_j r_j = 0 \tag{2.30}$$

Dividing by 2, I get the following equation:

$$(\Sigma_t - \alpha\Sigma_b)r_k = \lambda_k r_k + \frac{1}{2}\sum_{j=1}^{k-1} \mu_j r_j \tag{2.31}$$

I know that r_1 is the first contrastive principal component, obtained by solving the equation

$$(\Sigma_t - \alpha\Sigma_b)r_1 = \lambda_1 r_1 \tag{2.32}$$

where λ_1 is the largest eigenvalue of $\Sigma_t - \alpha\Sigma_b$ and r_1 is the corresponding eigenvector.

Now, I want to find the second contrastive principal component r_2.

$$(\Sigma_t - \alpha\Sigma_b)r_2 = \lambda_2 r_2 + \frac{1}{2}\mu_1 r_1 \tag{2.33}$$

Similar to PCA, by our constraints, r_2 is a unit vector orthogonal to r_1.

Multiplying both sides with r_1^T (transpose of the first contrastive principal component), I get

$$r_1^T(\Sigma_t - \alpha\Sigma_b)r_2 = \lambda_2 r_1^T r_2 + \frac{\mu_1}{2} \tag{2.34}$$

because $r_1^T r_1 = 1$.

Since r_1 and r_2 are orthogonal, $r_1^T r_2 = 0$, which implies

$$r_1^T(\Sigma_t - \alpha\Sigma_b)r_2 = 0 + \frac{\mu_1}{2} \Rightarrow \frac{\mu_1}{2} = 0 \Rightarrow \mu_1 = 0 \tag{2.35}$$

Thus, equation (2.33) becomes

$$(\Sigma_t - \alpha\Sigma_b)r_2 = \lambda_2 r_2 \tag{2.36}$$

which confirms that r_2 is the eigenvector associated with λ_2.

For the case $k = 3$, to find the third contrastive principal component r_3, I start with

$$(\Sigma_t - \alpha\Sigma_b)r_3 = \lambda_3 r_3 + \frac{1}{2}(\mu_1 r_1 + \mu_2 r_2) \tag{2.37}$$

By our constraints, r_3 is orthogonal to both r_1 and r_2. By multiplying both sides of the equation with r_1^T and r_2^T separately, I get

$$r_1^T(\Sigma_t - \alpha\Sigma_b)r_3 = \frac{1}{2}\mu_1 \quad \text{and} \quad r_2^T(\Sigma_t - \alpha\Sigma_b)r_3 = \frac{1}{2}\mu_2 \tag{2.38}$$

Since r_3 is orthogonal to both r_1 and r_2, these equations equate to 0, leading to

$$\mu_1 = 0 \quad \text{and} \quad \mu_2 = 0 \tag{2.39}$$

Therefore, after substituting these into equation (2.37), I am left with

$$(\Sigma_t - \alpha\Sigma_b)r_3 = \lambda_3 r_3 \tag{2.40}$$

which confirms that r_3 is the eigenvector associated with λ_3.

This completes the derivation for the k-th contrastive principal component up to $k = 3$, and the greater k cases follow analogous proofs, similar to PCA.

In this section, building on the standard PCA formulation, I derived the contrastive PCA (cPCA) optimization problem using a contrastive covariance matrix and demonstrated that the solution also reduces to an eigenvalue problem. I showed how orthogonality constraints apply analogously to the standard case and provided full derivations for the first three contrastive components. These results clarify how cPCA isolates directions enriched in target-specific variance.

2.3 Independent Component Analysis (ICA)

The primary goal of ICA is to recover the independent source signals from observed mixtures. The key characteristic of ICA is *Statistical Independence*: In ICA, I assume that each source signal is statistically independent of the others. Independence is a stronger requirement than uncorrelation (as in PCA) because it implies that the joint probability distribution of the sources can be expressed as a product of their individual distributions. This means that ICA is effective in cases where multiple overlapping signals need to be separated, such as in audio or EEG data.

To achieve signal separation, ICA relies on the assumption that the source signals are *non-Gaussian*. The reason for this is that Gaussian variables are symmetric and have only second-order statistics (like mean and variance) that vary. This makes it challenging to distinguish between Gaussian sources.

Non-Gaussianity is crucial because, by the Central Limit Theorem, the sum of independent random variables tends toward a Gaussian distribution regardless of their individual distributions. Thus, if the observed signals are mixtures of multiple sources, they will likely be closer to Gaussian than the original, individual sources.

Let $X \in \mathbb{R}^{n\times m}$ represent the observed, whitened ($\mathrm{Cov}(X) = I$) and mean-centered data matrix, where each row corresponds to an observed channel (individual data collection/measurement, e.g., each microphone recording sound), and each column represents data samples collected at a time. So, there are n sources and data samples collected over m times. The ICA model assumes that

$$X = AS,$$

where

- $A \in \mathbb{R}^{n\times n}$ is the mixing matrix,

- $S \in \mathbb{R}^{n \times m}$ contains n independent, distinct underlying source signals that generate observed/measured mixture.

ICA makes the assumption that the mixing process that generates X is linear.

The goal of ICA is to estimate the independent components S by finding a matrix $W \in \mathbb{R}^{n \times n}$ such that

$$\hat{S} = W^T X \tag{2.41}$$

where $\hat{S}$ represents the *estimated* independent components, not the exact components.

I aim to maximize the measure of non-Gaussianity of $\hat{S} = W^T X$. Two popular measures of non-Gaussianity used in ICA are *negentropy* (negative entropy) and *kurtosis* (the fourth-order moment of the distribution, which indicates the peakedness). In this thesis, *negentropy* is used, which is derived from the concept of entropy. The entropy $H(y)$ of a variable y is the expected amount of information needed to describe the state of a variable, in other words, a measure of uncertainty.

The differential entropy (in nats) of a continuous variable y with probability density function $p(y)$ is given by:

$$H(y) = -\int_{-\infty}^{\infty} p(y) \ln p(y)\, dy$$

[Papoulis and Pillai, 2002].

For a Gaussian-distributed random variable $Y_{Gaussian} \sim \mathcal{N}(\mu, \sigma^2)$ with mean μ and variance σ^2, the probability density function is

$$p(y_{Gaussian}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}$$

Substituting this into the entropy formula and solving the integral, I obtain:

$$H(y_{Gaussian}) = \frac{1}{2} \ln(2\pi e \sigma^2)$$

[Cover and Thomas, 1991].

A Gaussian variable has the highest entropy among all distributions with the same variance by the principle of maximum entropy and the definition of maximum entropy probability distribution [Cover and Thomas, 1991].

In ICA, y is defined by

$$y = w^T X \tag{2.42}$$

where w^T is a row vector, representing a single row of W^T, which is a single column of W.

The row vector w^T can be thought of as a set of weights applied to observed signals in X. Since w^T is part of the separation matrix W^T, each row of W^T represents a specific weight configuration aimed at extracting one independent component. The expression $w^T X$ represents a linear combination of the observed signals. This combination produces a new signal, y, which is an estimate of one of the underlying independent components I am trying to uncover from the mixed signals in X.

I iteratively adjust w^T until y (the estimated independent component) exhibits maximum non-Gaussianity. By doing this for each column of W, I aim to decompose X into components that are as statistically independent as possible.

Negentropy $J(y)$ is defined as

$$J(y) = H(y_{\text{Gaussian}}) - H(y) \tag{2.43}$$

where $H(y_{\text{Gaussian}})$ is the entropy of a Gaussian random variable with the same variance as y [Meyer-Baese and Schmid, 2014]. Maximizing negentropy helps us identify non-Gaussian signals, which correspond to the independent components.

Since exact negentropy is difficult to compute because the distribution of Y is unknown [Hyvärinen et al., 2001], it is often approximated:

$$J(y) \approx (\mathbb{E}[G(y)] - \mathbb{E}[G(y_{\text{Gaussian}})])^2 \tag{2.44}$$

where $G(y)$ is a function chosen to capture the deviation from Gaussianity, such as y^4 and $\log(\cosh(y))$ [Oja and Hyvarinen, 2000]. One can calculate $\mathbb{E}[G(y)]$ by utilizing Monte Carlo estimation or analytical computation.

The G(y) = y^4 function is highly sensitive to kurtosis because it grows rapidly for large values. However, it is symmetric and does not capture skewness. The G(y) = $\log(\cosh(y))$ function is more moderate in its sensitivity to extreme values, so it is less sensitive to kurtosis. However, because of its non-quadratic, less symmetric nature, it can capture skewness more effectively than y^4.

The separation matrix W is initialized either randomly or as the identity matrix [Montoya-Martínez et al., 2017, Tharwat, 2021]. This matrix is then adjusted iteratively to maximize the independence of the estimated components.

The iterative process in ICA is essential for refining the separation matrix W to maximize the independence (non-Gaussianity) of the estimated source signals.

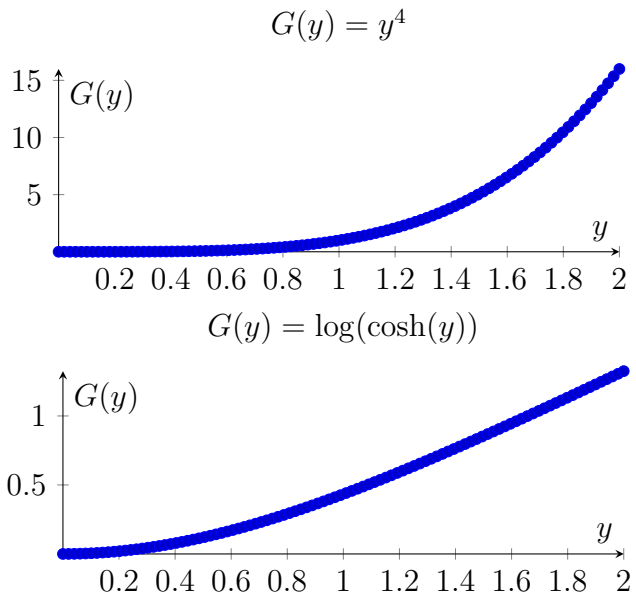


Figure 2.1: Behaviors of $G(y) = y^4$ and $G(y) = \log(\cosh(y))$. These functions are plotted in the first quadrant in the domain 0<y<2. The first emphasizes higher-order moments, while the latter is more robust to extreme values.

2.3.1 The Iterative Process in ICA and The Update Rule

Because Negentropy is defined as the difference between $\mathbb{E}[G(y)]$ and $\mathbb{E}[G(y_{\text{Gaussian}})]$ in equation (2.44) and also $\mathbb{E}[G(y_{\text{Gaussian}})]$ is the maximum possible value for any $\mathbb{E}[G(y)]$, maximizing the Negentropy is equivalent to minimizing the function $\mathbb{E}\{G(y)\}$, where $y = w^T X$. $\mathbb{E}\{G(y)\}$ can be expressed as

$$\mathbb{E}\{G(y)\} = \mathbb{E}\{G(w^T X)\} \tag{2.45}$$

The objective function with a constraint that $w^T w = 1$ is defined using the Lagrange multiplier method to minimize $\mathbb{E}\{G(y)\}$:

$$J(w) = \mathbb{E}\{G(w^T X)\} - \frac{\beta}{2}(w^T w - 1), \tag{2.46}$$

where the second term ensures that w is constrained to have unit norm.

Following the derivation given by [Wang, 2021] and [Hyvärinen et al., 2001], I use a Newtonian method for iterative optimization:

$$w \leftarrow w - J_F^{-1}(w) F(w), \tag{2.47}$$

where $F(w)$ is the the gradient of the objective function $J(w)$ with respect to the vector w. $J_F(w)$ is the Jacobian matrix of $F(w)$, which consists of the partial derivatives of the gradient function.

Taking the derivative of $J(w)$ with respect to the vector w yields

$$F(w) = \frac{\partial J(w)}{\partial w} = \begin{bmatrix} \frac{\partial J(w)}{\partial w_1} \\ \frac{\partial J(w)}{\partial w_2} \\ \vdots \\ \frac{\partial J(w)}{\partial w_d} \end{bmatrix} = \frac{\partial(\mathbb{E}\{G(w^T X)\})}{\partial w} - \frac{\partial(\frac{\beta}{2}(w^T w) - 1)}{\partial w} \tag{2.48}$$

$$F(w) = \mathbb{E}\{\frac{\partial(G(w^T X))}{\partial w}\} - \frac{\beta}{2}(2w) = \mathbb{E}\{\frac{\partial(G(w^T X))}{\partial w}\} - \beta w \tag{2.49}$$

where differentiation is moved inside because expectation is a linear operator.

According to [Hyvärinen et al., 2001], the $F(w)$ equation becomes

$$F(w) = \mathbb{E}\{X g(w^T X)\} - \beta w, \tag{2.50}$$

where $g(w^T X)$ is the derivative of $G(w^T X)$. Now, the Jacobian matrix $J_F(w)$ is

$$J_F(w) = \frac{\partial F(w)}{\partial w} = \begin{bmatrix} \frac{\partial F_1}{\partial w_1} & \frac{\partial F_1}{\partial w_2} & \cdots & \frac{\partial F_1}{\partial w_d} \\ \frac{\partial F_2}{\partial w_1} & \frac{\partial F_2}{\partial w_2} & \cdots & \frac{\partial F_2}{\partial w_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_d}{\partial w_1} & \frac{\partial F_d}{\partial w_2} & \cdots & \frac{\partial F_d}{\partial w_d} \end{bmatrix} = \frac{\partial \mathbb{E}\{X g(w^T X)\}}{\partial w} - \beta \frac{\partial w}{\partial w} \tag{2.51}$$

Then, similar to the derivation of $F(w)$,

$$J_F(w) = \mathbb{E}\{\frac{\partial(X g(w^T X))}{\partial w}\} - \beta I \tag{2.52}$$

Now, following the derivation by [Hyvärinen et al., 2001], I get

$$J_F(w) = \mathbb{E}\{X X^T (g'\left(w^\top X\right)\} - \beta I, \tag{2.53}$$

where $g'(z)$ is the derivative of $g(z)$.

Assuming that XX^T and $g'\left(w^\top X\right)$ are approximately uncorrelated,

$$\mathbb{E}\{X X^T (g'\left(w^\top X\right)\} \approx \mathbb{E}[X X^T]\mathbb{E}[g'\left(w^\top X\right)]$$

Since X is whitened initially, I can assume directly that $\mathbb{E}[X X^T] = I$.

Thus, the Jacobian becomes

$$J_F(w) = \mathbb{E}\{g'(w^\top X)\}I - \beta I \tag{2.54}$$

Factorizing:

$$J_F(w) = \left(\mathbb{E}\{g'(w^\top X)\} - \beta\right) I \tag{2.55}$$

which means that the gradient becomes diagonal. Since $J_F(w)$ is diagonal, its inverse is

$$J_F(w)^{-1} = \frac{1}{\mathbb{E}\{g'(w^\top X)\} - \beta} I \tag{2.56}$$

Substituting $J_F(w)^{-1}$ to the Newtonian update, I get

$$w \leftarrow w - \frac{\mathbb{E}\{Xg(w^\top X)\} - \beta w}{\mathbb{E}\{g'(w^\top X)\} - \beta} \tag{2.57}$$

Multiplying both sides of the equation with $-\mathbb{E}\{g'(w^\top X)\} + \beta$ yields

$$(-\mathbb{E}\{g'(w^\top X)\} + \beta)w \leftarrow (-\mathbb{E}\{g'(w^\top X)\} + \beta)w + \mathbb{E}\{Xg(w^\top X)\} - \beta w \tag{2.58}$$

$$(-\mathbb{E}\{g'(w^\top X)\} + \beta)w \leftarrow -\mathbb{E}\{g'(w^\top X)\}w + \mathbb{E}\{Xg(w^\top X)\} \tag{2.59}$$

Because $\beta - \mathbb{E}\{g'(w^\top X)\}$ is a scalar, it just scales the update step, affecting only the step size. So, I can simplify the update rule as

$$w \leftarrow \mathbb{E}\{Xg(w^T X)\} - \mathbb{E}\{g'(w^T X)\}w \tag{2.60}$$

Then, the expected values are calculated either via Monte Carlo estimation or using analytical solutions.

As described in [Hyvärinen et al., 2001], different ICA algorithms employ various optimization methods. One of the most commonly used approaches is FastICA, which uses the following steps:

1. **Center and Whiten the Data:** Center the data X (subtract the mean) and whiten it (transform to unit variance) by using PCA. Whitening ensures that the components are uncorrelated and reduces the dimension.

2. **Nonlinear Transformation:** For each component w, apply a nonlinearity $G(y)$, such as $G(y) = \tanh(y)$ or $G(y) = y^4$.

3. **Update Rule:** Update w by maximizing the non-Gaussianity of $w^T X$ using:

$$w \leftarrow \mathbb{E}\{Xg(w^T X)\} - \mathbb{E}\{g'(w^T X)\}w \tag{2.61}$$

4. **Decorrelate the Components:** To prevent different vectors from converging to the same maxima, FastICA decorrelates vectors sequentially. This decorrelation is based on Gram-Schmidt orthogonalization. Before normalization, the projection of the current vector w_{k+1} onto all previously computed vectors is subtracted:
$$w_{k+1}^{+} \leftarrow w_{k+1} - \sum_{j=1}^{k} (w_{k+1}^{T} w_j) w_j. \tag{2.62}$$

5. **Normalization:** After decorrelation, normalize w_{k+1}^{+} to ensure unit length:
$$w_{k+1} \leftarrow \frac{w_{k+1}^{+}}{\|w_{k+1}^{+}\|}. \tag{2.63}$$

6. **Convergence Check:** The update process continues until the old and new weight vectors w and w^{+} are in the same direction, i.e., their dot product converges to 1:
$$w \cdot w^{+} \approx 1. \tag{2.64}$$
If this convergence criterion is not met within a predetermined number of iterations, the weight calculation is stopped and restarted with another random initialization.

7. **Construct the Demixing Matrix W:** After calculating all n weight vectors, construct the demixing matrix W by stacking $w_1, \ldots, w_n$.

8. **Compute the Independent Components:** Once W is computed, the estimated independent components $\hat{S}$ are obtained by applying W^T to the whitened data X.

In this thesis, I use scikit-learn's FastICA algorithm, which begins by centering and whitening the input data using PCA by default. PCA reduces the dimensions by projecting the data onto its leading principal components that retain the most variance. After this preprocessing, FastICA applies its core procedure to extract statistically independent components. Unlike PCA, which seeks uncorrelated directions of maximal variance, Independent Component Analysis (ICA) aims to identify a linear transformation such that the resulting components are as statistically independent as possible. FastICA uses a fixed-point iteration algorithm to compute the weight vectors that maximize non-Gaussianity, iteratively updating each direction

while maintaining orthogonality to previously found components. The final independent components are linear combinations of the input features and are intended to approximate the independent factors that underlie the observed data.

In this section, I derived the ICA objective from first principles by introducing negentropy as a non-Gaussianity measure and formulating the constrained optimization problem. I followed the derivation of the FastICA update rule using Newton's method, including the Jacobian and final simplified expression. I also outlined how whitening and fixed-point iteration contribute to component estimation, solidifying the theoretical basis for ICA's mechanism.

2.4 Non-Negative Matrix Factorization (NMF)

Given a non-negative data matrix $X \in \mathbb{R}^{n \times m}$, NMF seeks to approximate it by the product of two non-negative matrices $W \in \mathbb{R}^{n \times l}$ and $H \in \mathbb{R}^{l \times m}$, where l is a number chosen to be smaller than m and n:

$$X \approx WH \tag{2.65}$$

In this thesis, I focus on the Multiplicative Update Rule version of NMF, which is one of the most widely used NMF algorithms.

The factor matrices W and H are found by minimizing the difference between X and WH, which is equivalent to finding

$$\arg\min_{W,H} \frac{1}{2}\|X - WH\|_F^2 \tag{2.66}$$

where $\|\cdot\|_F$ is the Frobenius norm, defined as

$$\|X - WH\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} \left(X_{ij} - (WH)_{ij}\right)^2 . \tag{2.67}$$

To ensure non-negativity of the factor matrices W and H, a multiplicative update process is used instead of additive updates. This involves designing step sizes in such a way that each update guarantees non-negativity. The goal is to minimize $\frac{1}{2}\|X - WH\|_F^2$, which can be expressed as

$$\begin{aligned} \|X - WH\|_F^2 &= \mathrm{tr}\{(X - WH)^\top (X - WH)\} \\ &= \mathrm{tr}\{X^\top X - X^\top WH - (WH)^\top X + (WH)^\top WH\} \end{aligned} \tag{2.68}$$

where tr() denotes the trace of a matrix.

Because $\text{tr}(X) = \text{tr}(X^\top)$, $\text{tr}(X^\top WH) = \text{tr}((X^\top WH)^\top) = \text{tr}((WH)^\top X)$. Then,

$$\begin{aligned} \|X - WH\|_F^2 &= \text{tr}(X^\top X - 2(WH)^\top X + H^\top W^\top WH) \\ &= \|X\|_F^2 - 2\,\text{tr}((WH)^\top X) + \text{tr}(H^\top W^\top WH) \end{aligned} \tag{2.69}$$

since $\|X\|_F^2 = \text{tr}(X^\top X)$. So,

$$f(W,H) = \frac{1}{2}\|X - WH\|_F^2 = \frac{1}{2}\|X\|_F^2 - \text{tr}((WH)^\top X) + \frac{1}{2}\text{tr}(H^\top W^\top WH) \tag{2.70}$$

The gradient with respect to W is

$$\nabla_W f(W,H) = \nabla_W(\frac{1}{2}\|X\|_F^2 - \text{tr}((WH)^\top X) + \frac{1}{2}\text{tr}(H^\top W^\top WH)) \tag{2.71}$$

$$\nabla_W f(W,H) = \nabla_W \frac{1}{2}\|X\|_F^2 - \nabla_W \text{tr}((WH)^\top X) + \nabla_W \frac{1}{2}\text{tr}(H^\top W^\top WH) \tag{2.72}$$

Because

$$\nabla_W \frac{1}{2}\|X\|_F^2 = 0 \tag{2.73}$$

and

$$\text{tr}((WH)^\top X) = \text{tr}(((WH)^\top X)^\top) = \text{tr}(X^\top WH), \tag{2.74}$$

the gradient equation becomes

$$\nabla_W f(W,H) = -\nabla_W \text{tr}(X^\top WH) + \nabla_W \frac{1}{2}\text{tr}(H^\top W^\top WH) \tag{2.75}$$

According to [Ang, 2019], some useful matrix calculus equations are

$$\nabla_X \text{tr}(AXB) = A^\top B^\top \tag{2.76}$$

$$\nabla_X \text{tr}(X^\top AX) = (A + A^\top)X \tag{2.77}$$

$$\nabla_X \text{tr}(B^\top X^\top XB) = 2XBB^\top \tag{2.78}$$

Since $\nabla_W \text{tr}(AWB) = A^\top B^\top$, $\nabla_W \text{tr}(X^\top WH) = (X^\top)^\top H^\top = XH^\top$. Also, because $\nabla_W \text{tr}(B^\top W^\top WB) = 2WBB^\top$, $\nabla_W \frac{1}{2}\text{tr}(H^\top W^\top WH) = WHH^\top$. So, the gradient equation can be rewritten as

$$\nabla_W f(W,H) = -XH^\top + WHH^\top \tag{2.79}$$

Similarly, the gradient with respect to H is

$$\nabla_H f(W,H) = \nabla_H \frac{1}{2}\|X\|_F^2 - \nabla_H \operatorname{tr}((WH)^\top X) + \nabla_H \frac{1}{2}\operatorname{tr}(H^\top W^\top WH) \tag{2.80}$$

$$\nabla_H f(W,H) = -\nabla_H \operatorname{tr}((WH)^\top X) + \nabla_H \frac{1}{2}\operatorname{tr}(H^\top W^\top WH) \tag{2.81}$$

$$\nabla_H f(W,H) = -\nabla_H \operatorname{tr}(X^\top WH) + \nabla_H \frac{1}{2}\operatorname{tr}(H^\top W^\top WH) \tag{2.82}$$

I know that $\nabla_H \operatorname{tr}(X^\top WH) = (X^\top W)^\top$. Since $\nabla_H \operatorname{tr}(H^\top AH) = (A + A^\top)H$, $\nabla_H \frac{1}{2}\operatorname{tr}(H^\top W^\top WH) = \nabla_H \frac{1}{2}\operatorname{tr}(H^\top (W^\top W)H) = \frac{1}{2}((W^\top W) + (W^\top W)^\top)H$, which is equal to $W^\top WH$ because $(W^\top W) = (W^\top W)^\top$. So,

$$\nabla_H f(W,H) = -(X^\top W)^\top + W^\top WH \tag{2.83}$$

which means

$$\nabla_H f(W,H) = -W^\top X + W^\top WH \tag{2.84}$$

According to [Ang, 2019], the Alternating Gradient Descent method states that the estimates of matrices at the step after the kth step are

$$W_{k+1} = W_k - t_k^W \nabla_W f(W_k, H_k), \tag{2.85}$$

$$H_{k+1} = H_k - t_k^H \nabla_H f(W_{k+1}, H_k). \tag{2.86}$$

where t_k^W and t_k^H are step sizes.

Then, the NMF gradient descent update rules are

$$W_{k+1} = W_k - t_k^W (W_k H_k H_k^\top - X H_k^\top) \tag{2.87}$$

$$H_{k+1} = H_k - t_k^H (W_{k+1}^\top W_{k+1} H_k - W_{k+1}^\top X) \tag{2.88}$$

Instead of a common step size for the whole matrix, I will let each element have a different step size. Also, the updating process itself will be element-wise. This choice guarantees non-negativity. Then, the update rules become

$$[W_{k+1}]_{ij} = [W_k]_{ij} - [t_k^W]_{ij} \left([W_k H_k H_k^\top]_{ij} - [X H_k^\top]_{ij}\right) \tag{2.89}$$

$$[H_{k+1}]_{ij} = [H_k]_{ij} - [t_k^H]_{ij} \left([W_{k+1}^\top W_{k+1} H_k]_{ij} - [W_{k+1}^\top X]_{ij}\right) \tag{2.90}$$

Then, for W, I can choose

$$[t_k^W]_{ij} = \frac{[W_k]_{ij}}{[W_k H_k H_k^T]_{ij}} \tag{2.91}$$

For W, the element-wise update rules become

$$[W_{k+1}]_{ij} = [W_k]_{ij} - \frac{[W_k]_{ij}}{[W_k H_k H_k^T]_{ij}} \left([W_k H_k H_k^\top]_{ij} - [X H_k^\top]_{ij}\right) \tag{2.92}$$

$$[W_{k+1}]_{ij} = [W_k]_{ij} - [W_k]_{ij} + [W_k]_{ij} \frac{[X H_k^\top]_{ij}}{[W_k H_k H_k^T]_{ij}} \tag{2.93}$$

$$[W_{k+1}]_{ij} = [W_k]_{ij} \frac{[X H_k^\top]_{ij}}{[W_k H_k H_k^T]_{ij}} \tag{2.94}$$

Similarly, for H, the step size can be chosen as

$$[t_k^H]_{ij} = \frac{[H_k]_{ij}}{[W_{k+1}^T W_{k+1} H_k]_{ij}} \tag{2.95}$$

This leads to the element-wise update rules:

$$[H_{k+1}]_{ij} = [H_k]_{ij} - \frac{[H_k]_{ij}}{[W_{k+1}^T W_{k+1} H_k]_{ij}} \left([W_{k+1}^T W_{k+1} H_k]_{ij} - [W_{k+1}^T X]_{ij}\right) \tag{2.96}$$

$$[H_{k+1}]_{ij} = [H_k]_{ij} - [H_k]_{ij} + \frac{[H_k]_{ij}[W_{k+1}^T X]_{ij}}{[W_{k+1}^T W_{k+1} H_k]_{ij}} \tag{2.97}$$

$$[H_{k+1}]_{ij} = [H_k]_{ij} \cdot \frac{[W_{k+1}^T X]_{ij}}{[W_{k+1}^T W_{k+1} H_k]_{ij}} \tag{2.98}$$

The final multiplicative element-wise update rules are

$$W_{k+1} = W_k \circ \frac{X H_k^T}{W_k H_k H_k^T} \tag{2.99}$$

$$H_{k+1} = H_k \circ \frac{W_{k+1}^T X}{W_{k+1}^T W_{k+1} H_k} \tag{2.100}$$

where $\circ$ denotes element-wise multiplication.

The updates retain non-negativity of W and H because all terms in X, W, H, and the gradients are non-negative.

These updates are repeated until convergence. The multiplicative update rules minimize the same objective function as the gradient descent update rule but in a way that respects the non-negativity constraint.

These multiplicative update rules form the foundation of the Multiplicative Update Algorithm for NMF, ensuring that the factorized matrices W and H remain

non-negative throughout the iterative process. By iteratively refining these matrices, the algorithm effectively extracts meaningful, lower-dimensional representations from high-dimensional non-negative data.

In this section, for Non-negative Matrix Factorization (NMF), I derived the Frobenius-norm objective and used matrix calculus to compute gradients with respect to both factor matrices W and H. I showed how alternating gradient descent steps lead to multiplicative update rules, and rigorously proved their correctness through element-wise formulations. These derivations justify the non-negativity-preserving structure of the NMF optimization.

Chapter 3

Application and Results

3.1 Datasets and Pre-processing

3.1.1 Datasets

I use datasets from diverse domains to evaluate the performance of dimension reduction techniques:

- **Combined Autism Dataset:** Three datasets were sourced from autism screening studies, targeting different age groups. All three datasets have the same variables, including demographic information, responses to screening questions, and the final screening score:
  - *Adolescent Dataset:* This dataset contains 104 instances with 21 attributes [Thabtah, 2017a],
  - *Child Dataset:* Comprising 292 instances and 21 attributes [Thabtah, 2017c], and
  - *Adult Dataset:* With 704 instances and 21 attributes [Thabtah, 2017b].

  The three datasets are combined to create the Combined Autism dataset.
- **Twitch Dataset:** The Twitch dataset includes user engagement metrics, game preferences, and channel statistics from the Twitch streaming platform and statistics from other social media platforms. This dataset is sourced from Kaggle [Richard_V, 2022].
- **Accent Dataset:** The Accent dataset contains Mel-Frequency Cepstral Coefficients (MFCC) features extracted from speech samples of people with 6

different accents (American, British, Spanish, Italian, German, and French). These features are commonly used in speech recognition and classification tasks. This dataset is sourced from the UC Irvine Machine Learning Repository [Speaker, 2020].

- **High Dimensional Synthetic Datasets:** Two synthetic datasets are sourced from OSF (Open Science Framework). [Rodrigues and Moreira, 2021]:
  - *dim_512 Dataset:* This dataset contains 512-dimensional 1024 instances grouped in 16 clusters.
  - *dim_1024 Dataset:* Similar to dim_512, this dataset has 1024 instances, but with 1024 dimensions also grouped in 16 clusters.

Statistical information about each dataset is given on Table 3.1.

Table 3.1: Summary of Unprocessed Datasets

Dataset	**Instances**	**Features**	**Missing**	**Min**	**Max**
Autism Adolescent	104	21	0	1.0	16.0
Autism Child	292	21	4	0.0	11.0
Autism Adult	704	21	2	0.0	383.0
Twitch	30	116	0	-19,755	6,172,546,000
Accent	329	13	0	-15.66	21.45
Dim_512	1024	512	0	1.0	226.0
Dim_1024	1024	1024	0	1.0	226.0

3.1.2 Pre-processing

To ensure consistency in the application of dimension reduction methods, the following pre-processing steps were applied to each dataset:

1. **Handling Missing Values:** Datasets were examined for missing values. Records with missing entries were either imputed using the median value (for numerical features) or the mode (for categorical features).

2. **Encoding Categorical Variables:** Categorical attributes were transformed using one-hot encoding, creating binary indicators for each category.

3. **Standardization of Features:** Numerical features were standardized to have zero mean and unit variance to ensure that features with larger magnitudes did not dominate the results.

4. **Normalization:** All datasets were normalized to the range [0, 1] for consistency across different scales.

Table 3.2: Summary of Pre-Processed Datasets

Dataset	**Instances**	**Features**
Combined Autism	1101	125
Twitch	30	148
Accent	329	18
Dim_512	1024	512
Dim_1024	1024	1024

Table 3.2 reports information about the pre-processed versions of the datasets.

The synthetic dim_512 and dim_1024 datasets have 16 clusters, meaning that the data points are divided into 16 groups. The intragroup distances are almost negligible compared to the intergroup distances in the original dataset, which made it almost impossible to apply methods. To fix this, I increased the intragroup distances while keeping the centroids the same. The modified version of the dim_512 dataset is plotted in Figure 3.1 as an example.

3.2 Application of the Methods

This section describes how principal component analysis (PCA), independent component analysis (ICA), nonnegative matrix factorization (NMF), and contrast principal component analysis (cPCA) are applied to the datasets introduced earlier. Each method is implemented with automated component selection and visualization. For each dataset, an analysis pipeline is executed as follows:

1. **PCA**: The number of components is chosen automatically to retain at least 95% of the variance. The Frobenius and L1 norms of the reconstruction error are computed by projecting back from the component output.

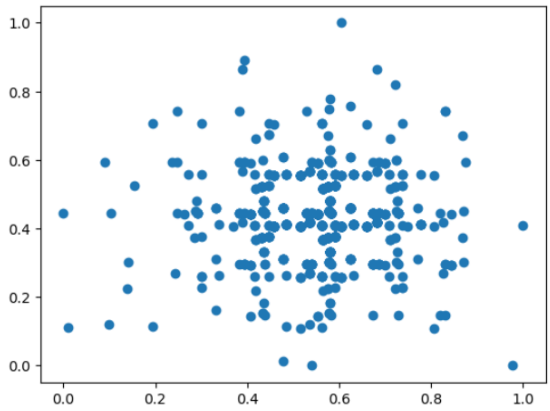


Figure 3.1: The final version of the pre-processed synthetic dim_512 dataset after modifying the intragroup distances. Randomly chosen two variables are plotted against each other. The axes are unitless.

2. **ICA**: ICA is tested with a range of component counts (from 1 to half of the input dimension). For each count, the reconstruction error using the estimated mixing matrix is computed, and the number of components that minimized this error was selected.

3. **NMF**: Input features are first rescaled to [0,1] using MinMax scaling, as required by the nonnegativity constraint of NMF. Component counts ranging from 1 to half of the feature count are tested. The configuration minimizing the reconstruction error is retained.

4. **cPCA**: Each dataset is split into a target and background subset. For Accent and Combined Autism, specific groups (U.S. English speakers and adults, respectively) were chosen as the target. The number of components is computed for a fixed contrastive parameter $\alpha = 2.6$, selecting the dimensionality that minimized the reconstruction error on the filtered target.

The optimal number of components was determined by minimizing the Frobenius or L1 reconstruction error.

Reconstruction error computation involved two steps. First, the dataset was projected into a lower-dimensional space using the projection vectors, and then it was reconstructed back to its original space using the pseudoinverse of those vectors. The difference between the original and reconstructed data gave the reconstruction error.

The projections that result in the lowest reconstruction error were retained for each method. The final comparison among methods was done by calculating the relative errors using these optimized projections:

For the Frobenius version, the total squared difference between all original and reconstructed values was calculated and then divided by the total Frobenius norm of the dataset. This gave a relative measure of how large the error is compared to the total signal strength.

For the L1 version, I used the average of the absolute differences instead of squared differences. The relative error was obtained by dividing this absolute error by the sum of the absolute values in the dataset. This provided a scale-invariant comparison of how much information is lost relative to the overall magnitude of the dataset.

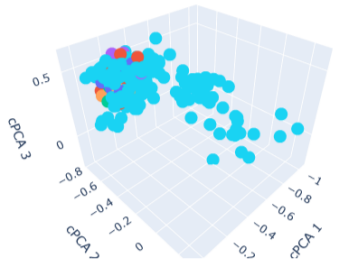
For each dataset, the following outputs are stored:

- Table of number of components and reconstruction errors for PCA, ICA, and NMF (including relative errors),
- Table of optimal component count, target error, full error, and relative error for cPCA,
- 3D scatter plots generated by using the top 3 components,
- 2D pairplots of the top 5 components across all methods, and
- Heatmaps of loading matrices (Most extreme holdings for each component are also stored).

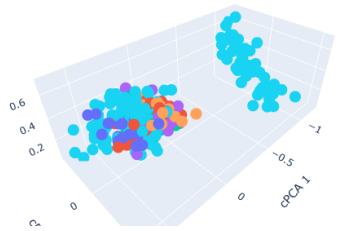
The full Python code implementing this pipeline, including plotting, is provided in the Appendix.

3.3 Results

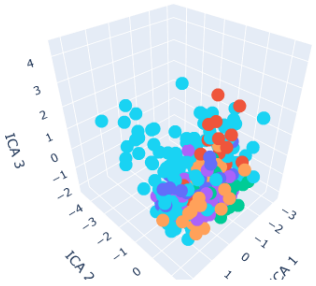
3.3.1 Accent Dataset



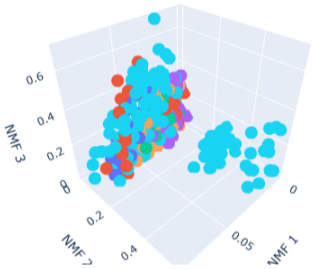
(a) cPCA - Frobenius



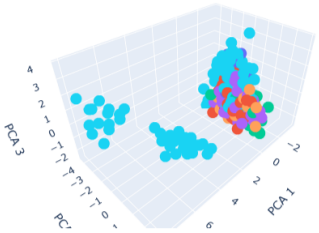
(b) cPCA - L1



(c) ICA



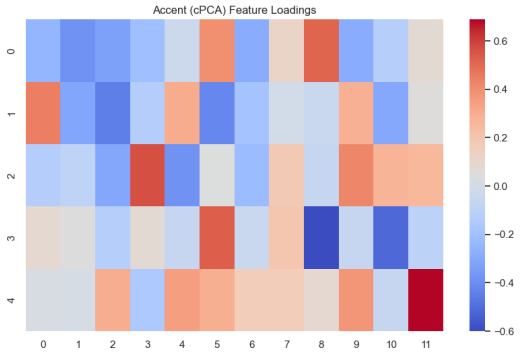
(d) NMF



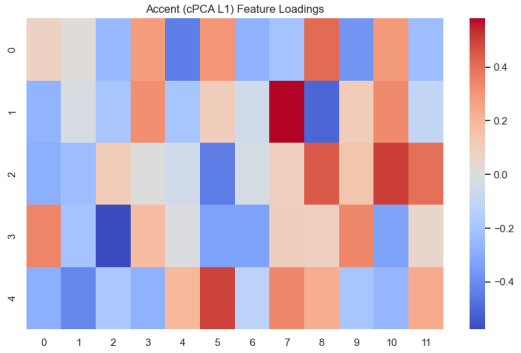
(e) PCA

• French • Spanish • German • British • Italian • American

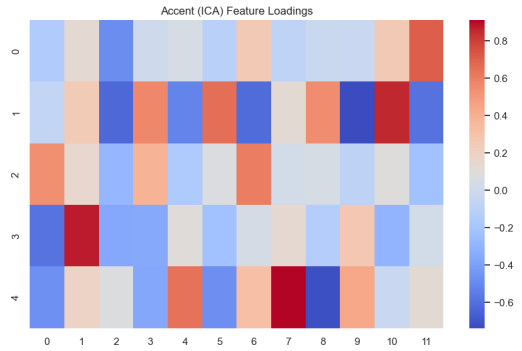
Figure 3.2: Color-coded 3D plots of methods' top 3 components are represented by unitless axes. Frobenius and L1 norm results differ for only cPCA.



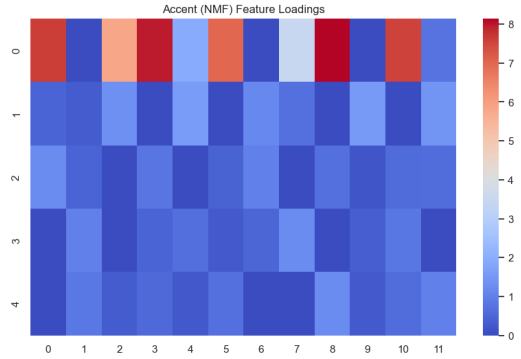
(a) cPCA - Frobenius Heatmap



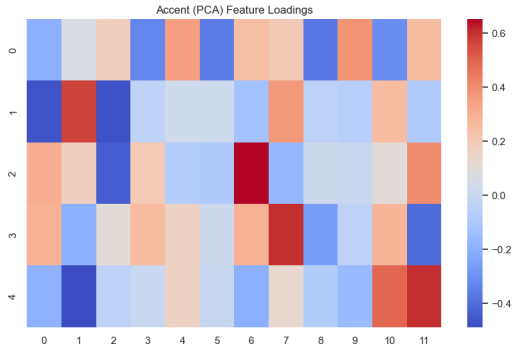
(b) cPCA - L1 Heatmap



(c) ICA Heatmap

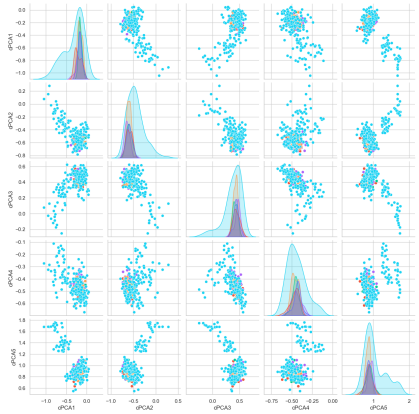


(d) NMF Heatmap

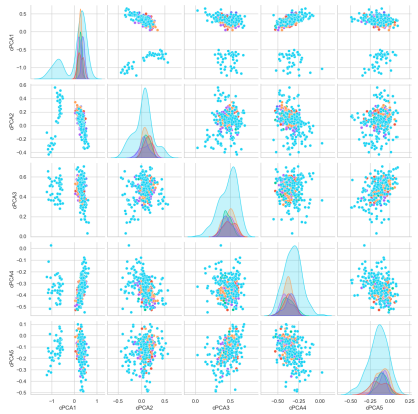


(e) PCA Heatmap

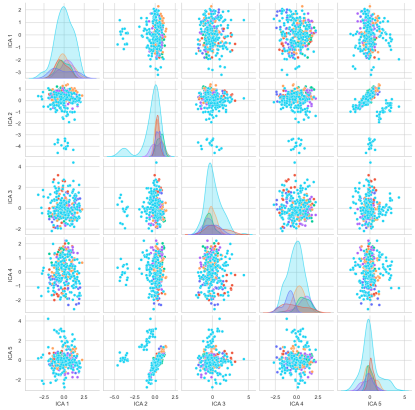
Figure 3.3: Heatmaps of top components show which variables dominate each component. The colors are based on the loadings of Variables, which are shown on the x-axes, and the numbers on y-axes correspond to the specific top components. Both axes start at 0; however, the datasets' indices start at 1. In other words, 0th component on a heatmap plot corresponds to the 1st, and so on.



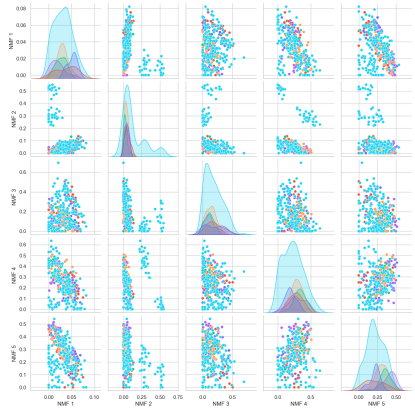
(a) cPCA - Frobenius Pairplot



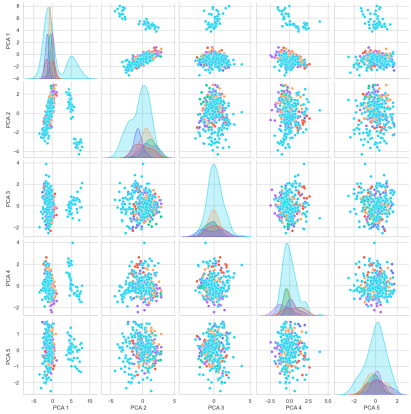
(b) cPCA - L1 Pairplot



(c) ICA Pairplot



(d) NMF Pairplot



(e) PCA Pairplot

French Spanish German British Italian American

Figure 3.4: Color-coded pairplots visualize the relationships between pairs of top 5 components.

Table 3.3: Top 7 Extreme Loadings for Accent (PCA - Frobenius / L1)

Component	Top 7 Features and Loadings
1	X10 (0.3815), X9 (-0.3752), X6 (-0.3629), X5 (0.3523), X4 (-0.3361), X11 (-0.3106), X12 (0.2551)
2	X2 (0.5692), X3 (-0.4726), X1 (-0.4666), X8 (0.3687), X11 (0.2599), X7 (-0.1275), X12 (-0.0831)
3	X7 (0.6520), X3 (-0.4426), X12 (0.4018), X1 (0.3085), X4 (0.2018), X2 (0.1741), X8 (-0.1689)
4	X8 (0.6063), X12 (-0.4055), X1 (0.2953), X7 (0.2939), X11 (0.2803), X9 (-0.2629), X4 (0.2572)
5	X12 (0.6033), X11 (0.4941), X2 (-0.4879), X7 (-0.2041), X1 (-0.1934), X5 (0.1619), X10 (-0.1602)

Table 3.4: Top 7 Extreme Loadings for Accent (ICA - Frobenius / L1)

Component	Top 7 Features and Loadings
1	X12 (0.7112), X3 (-0.4852), X11 (0.2669), X7 (0.2634), X1 (-0.1526), X2 (0.1351), X6 (-0.1185)
2	X11 (0.8563), X10 (-0.7372), X6 (0.6584), X3 (-0.6283), X7 (-0.6150), X12 (-0.5900), X4 (0.5652)
3	X7 (0.6061), X1 (0.5400), X4 (0.3853), X3 (-0.2805), X12 (-0.2225), X5 (-0.1628), X2 (0.1548)
4	X2 (0.8743), X1 (-0.5831), X3 (-0.3627), X4 (-0.3511), X11 (-0.2992), X10 (0.2781), X6 (-0.2246)
5	X8 (0.9125), X9 (-0.7237), X5 (0.6459), X6 (-0.4788), X1 (-0.4677), X10 (0.4417), X4 (-0.3653)

Table 3.5: Top 7 Extreme Loadings for Accent (NMF - Frobenius / L1)

Component	Top 7 Features and Loadings
1	X9 (8.1438), X4 (7.9661), X1 (7.6314), X11 (7.5676), X6 (7.0240), X3 (5.8382), X8 (3.4543)
2	X5 (1.5766), X10 (1.5305), X12 (1.4296), X3 (1.3044), X7 (1.1769), X8 (0.6848), X1 (0.4673)
3	X1 (1.2393), X7 (1.0001), X4 (0.7733), X9 (0.6989), X12 (0.6151), X11 (0.5759), X2 (0.4659)
4	X8 (1.2247), X2 (1.0118), X11 (0.8106), X5 (0.6436), X7 (0.5079), X4 (0.4521), X10 (0.3403)
5	X9 (1.2121), X12 (0.9939), X2 (0.8325), X6 (0.6722), X11 (0.6205), X4 (0.5536), X3 (0.2990)

Table 3.6: Top 7 Extreme Loadings for Accent (cPCA - Frobenius)

Component	Top 7 Features and Loadings
1	X9 (0.5155), X6 (0.4004), X2 (-0.3842), X3 (-0.3380), X7 (-0.2914), X10 (-0.2904), X1 (-0.2504)
2	X3 (-0.4459), X1 (0.4417), X6 (-0.4209), X2 (-0.3167), X11 (-0.3080), X5 (0.3030), X10 (0.2878)
3	X4 (0.5667), X10 (0.4247), X5 (-0.3824), X3 (-0.3116), X11 (0.2739), X12 (0.2583), X7 (-0.2258)
4	X9 (-0.6003), X6 (0.5320), X11 (-0.5161), X8 (0.1998), X3 (-0.1270), X12 (-0.1018), X1 (0.0872)
5	X12 (0.6884), X10 (0.3765), X5 (0.3463), X3 (0.2962), X6 (0.2925), X7 (0.1645), X4 (-0.1620)

Table 3.7: Top 7 Extreme Loadings for Accent (cPCA - L1)

Component	Top 7 Features and Loadings
1	X5 (-0.4392), X9 (0.4100), X10 (-0.3727), X6 (0.3012), X11 (0.2888), X4 (0.2847), X7 (-0.2793)
2	X8 (0.5817), X9 (-0.5080), X11 (0.3329), X4 (0.3226), X1 (-0.2706), X5 (-0.1993), X3 (-0.1947)
3	X11 (0.5081), X6 (-0.4488), X9 (0.4463), X12 (0.3981), X1 (-0.2848), X2 (-0.2363), X10 (0.1454)
4	X3 (-0.5768), X1 (0.3495), X10 (0.3390), X6 (-0.3349), X7 (-0.3327), X11 (-0.3294), X2 (-0.2102)
5	X6 (0.4965), X2 (-0.4142), X8 (0.3408), X1 (-0.2893), X4 (-0.2786), X11 (-0.2580), X12 (0.2402)

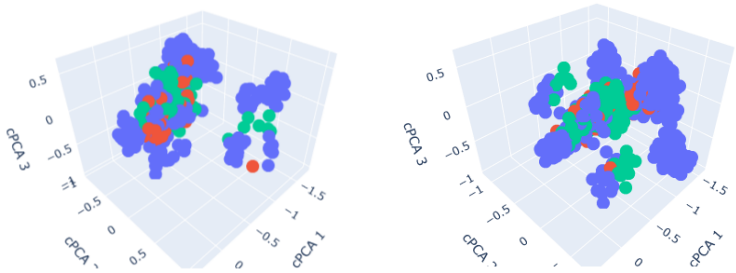
Figure 3.2 shows 3D scatter plots of the Accent dataset projected using cPCA, ICA, NMF, and PCA. While all methods are successful in distinguishing a group of American datapoints from the rest even by using the top 3 components, cPCA - L1 and PCA reveal the most distinct separation. On the 3D plot of cPCA - Frobenius, it can be seen clearly that the method's separating power comes mostly from the second component. Meanwhile, ICA provides moderate separability.

Figure 3.3 presents component heatmaps. On that figure, cPCA - Frobenius, cPCA - L1, and ICA show broader loadings that are closer to 0, whereas NMF highlights a smaller set of features with high intensity and assigns some variables low loadings. Interestingly, none of the variables of components higher than the first one have a loading greater than 4 on the NMF heatmap.

In Figure 3.4, pairplots show that cPCA achieves better cluster compactness than other methods. Also, the NMF pairplot indicates that it produces more sparse clusters compared to other methods. The PCA pairplot shows how powerful the first component is at separating.

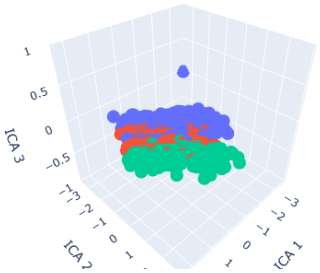
Tables 3.3 to 3.7 list the top 7 extreme loadings per method. It is interesting to see that the most extreme loading is positive and the second most extreme leading is negative for almost all 5 components in ICA and PCA. cPCA results do not exhibit this, but there is a different trend there: The common variables included in the first components of the L1 and Frobenius versions of cPCA have the same sign (X9 and X6 have positive loadings while X7 and X10 have negative loadings on both versions' tables). Also, NMF yields large magnitudes on only some MFCC (Mel-Frequency Cepstral Coefficients).

3.3.2 Combined Autism Dataset

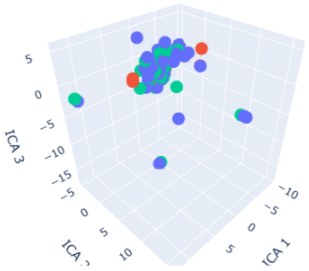


(a) cPCA - Frobenius

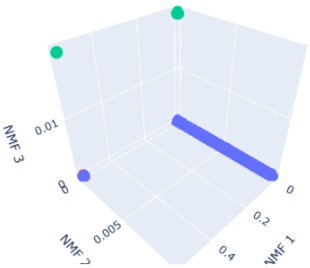
(b) cPCA - L1



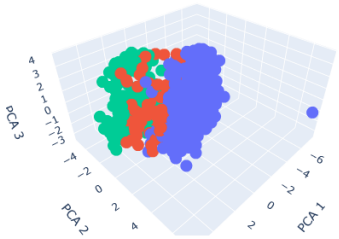
(c) ICA - L1



(d) ICA - Frobenius



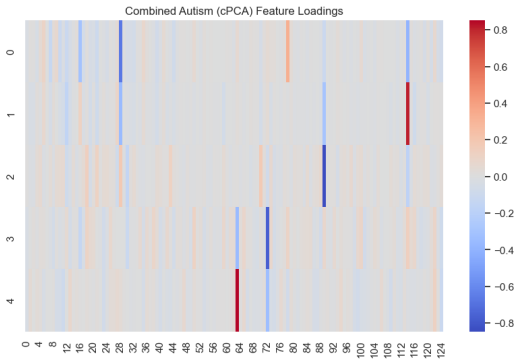
(e) NMF



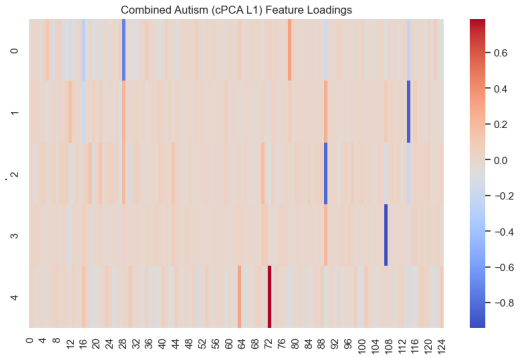
(f) PCA

• Adolescent • Adult • Child

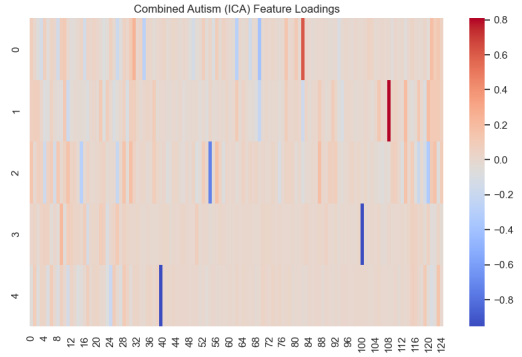
Figure 3.5: Color-coded 3D plots outputted by the methods applied on the Combined Autism dataset are shown with unitless axes. Both Frobenius and L1 results of cPCA and ICA are included because they are substantially different.



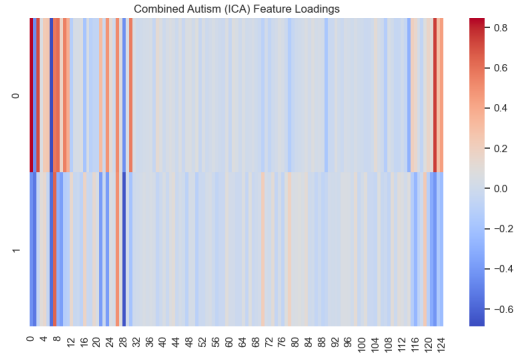
(a) cPCA - Frobenius Heatmap



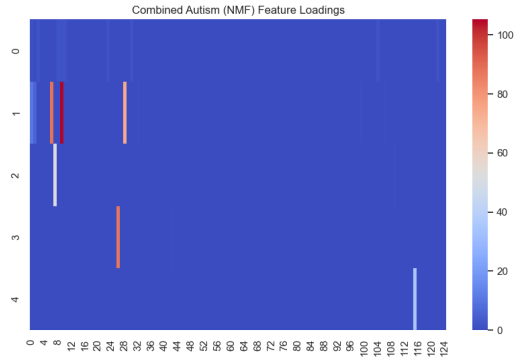
(b) cPCA - L1 Heatmap



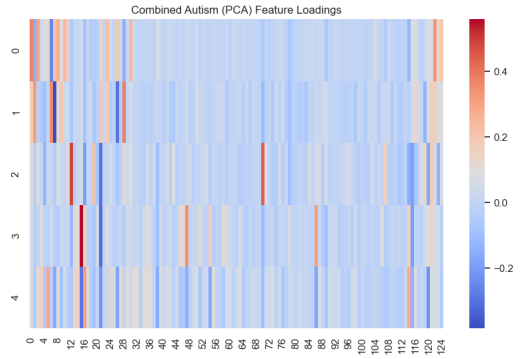
(c) ICA - Frobenius Heatmap



(d) ICA - L1 Heatmap

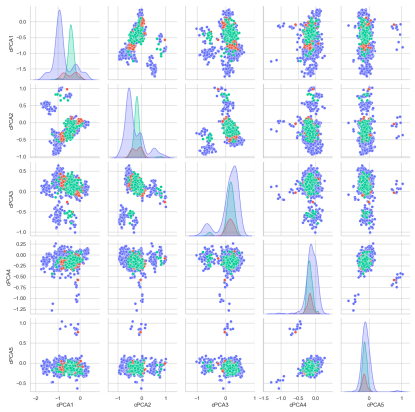


(e) NMF Heatmap

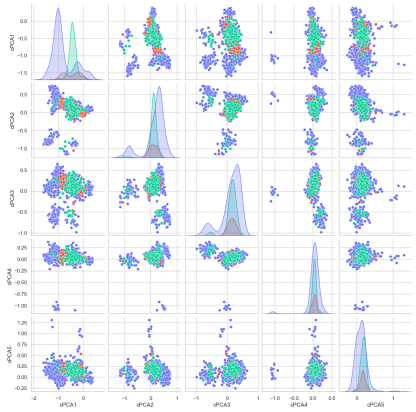


(f) PCA Heatmap

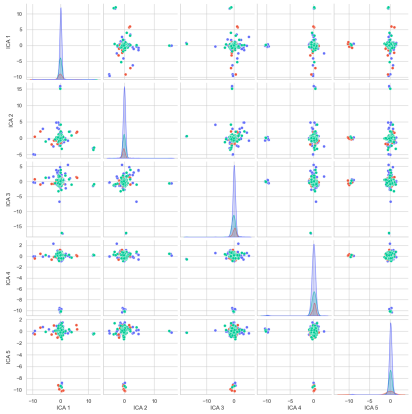
Figure 3.6: Heatmaps show the loadings outputted by the methods applied on the Combined Autism dataset. Heatmaps of both Frobenius and L1 versions of cPCA and ICA are included.



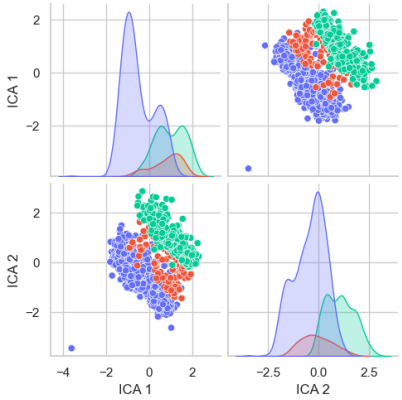
(a) cPCA - Frobenius Pairplot



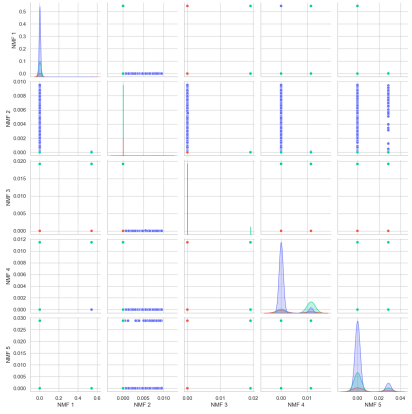
(b) cPCA - L1 Pairplot



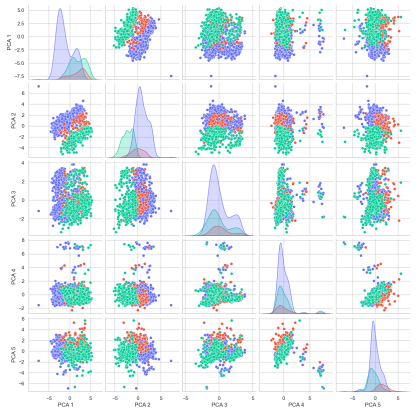
(c) ICA - Frobenius Pairplot



(d) ICA - L1 Pairplot



(e) NMF Pairplot



(f) PCA Pairplot

● Adolescent ● Adult ● Child

Figure 3.7: Color-coded pairplots hint at which components are better at distinguishing groups of datapoints.

Table 3.8: Top 7 Extreme Loadings for Combined Autism (PCA - Frobenius / L1)

Component	Top 7 Features and Loadings
1	result (0.3747), Class/ASD_b'YES' (0.3354), A6_Score_b'1' (0.2976), A9_Score_b'1' (0.2726), age_desc_b'18 and more' (-0.2526), A5_Score_b'1' (0.2452), A3_Score_b'1' (0.2366)
2	relation_b'Self' (0.3852), age_desc_b'4-11 years' (-0.3844), age_desc_b'18 and more' (0.3562), age (0.3152), relation_b'Parent' (-0.3022), A1_Score_b'1' (0.1908), ethnicity_b'White-European' (0.1880)
3	ethnicity_b'Asian' (0.4864), contry_of_res_b'India' (0.4433), ethnicity_b'White-European' (-0.2962), ethnicity_b'South Asian' (0.2213), ethnicity_b'Middle Eastern ' (-0.1947), contry_of_res_b'United Kingdom' (-0.1919), A8_Score_b'1' (0.1837)
4	ethnicity_b'Latino' (0.5585), contry_of_res_b'Brazil' (0.3396), ethnicity_b'White-European' (-0.3156), contry_of_res_b'Mexico' (0.3142), contry_of_res_b'United Kingdom' (-0.2021), ethnicity_b'Middle Eastern ' (0.1769), contry_of_res_b'Costa Rica' (0.1667)
5	ethnicity_b'Middle Eastern ' (0.3105), ethnicity_b'Latino' (-0.2924), age_desc_b'12-16 years' (0.2909), contry_of_res_b'United Arab Emirates' (0.2732), austim_b'yes' (-0.2285), ethnicity_b'White-European' (-0.2155), contry_of_res_b'Mexico' (-0.1801)

Table 3.9: Top 7 Extreme Loadings for Combined Autism (ICA - Frobenius)

Component	Top 7 Features and Loadings
1	contry_of_res_b'Lebanon' (0.6177), contry_of_res_b'Iceland' (-0.4029), contry_of_res_b'Finland' (-0.2889), contry_of_res_b'Anguilla' (-0.2782), contry_of_res_b'Albania' (0.2486), A8_Score_b'1' (0.1851), relation_b'Parent' (-0.1774)
2	contry_of_res_b'Sweden' (0.8089), contry_of_res_b'Iceland' (-0.2135), contry_of_res_b'Ukraine' (0.2029), austim_b'yes' (0.1950), A7_Score_b'1' (-0.1716), contry_of_res_b'Lebanon' (-0.1660), contry_of_res_b'South Korea' (0.1241)
3	contry_of_res_b'Costa Rica' (-0.7378), austim_b'yes' (-0.3404), ethnicity_b'Latino' (-0.2875), relation_b'Parent' (-0.2042), contry_of_res_b'Uruguay' (-0.2039), A5_Score_b'1' (0.1890), contry_of_res_b'Nepal' (0.1831)
4	contry_of_res_b'Russia' (-0.9539), A1_Score_b'1' (0.2074), ethnicity_b'Others' (-0.1214), relation_b'Parent' (0.1123), A4_Score_b'1' (-0.0895), A7_Score_b'1' (0.0887), ethnicity_b'White-European' (0.0876)
5	contry_of_res_b'Austria' (-0.9509), relation_b'Health care professional' (-0.1977), A2_Score_b'1' (0.1478), A7_Score_b'1' (-0.1337), relation_b'Self' (0.1091), gender_b'm' (0.1037), age_desc_b'18 and more' (0.0977)

Table 3.10: Top 7 Extreme Loadings for Combined Autism (ICA - L1)

Component	Top 7 Features and Loadings
1	result (0.8466), Class/ASD_b'YES' (0.7541), A6_Score_b'1' (0.7121), age_desc_b'18 and more' (-0.6873), A9_Score_b'1' (0.6189), age_desc_b'4-11 years' (0.5777), A5_Score_b'1' (0.5656)
2	relation_b'Self' (-0.6580), age_desc_b'4-11 years' (0.6387), age_desc_b'18 and more' (-0.5714), age (-0.5274), relation_b'Parent' (0.4932), result (-0.4391), Class/ASD_b'YES' (-0.4213)

Table 3.11: Top 7 Extreme Loadings for Combined Autism (NMF - Frobenius / L1)

Component	Top 7 Features and Loadings
1	Class/ASD_b'YES' (1.8380), A6_Score_b'1' (1.8380), A5_Score_b'1' (1.8379), contry_of_res_b'South Africa' (1.8379), A9_Score_b'1' (1.8370), A3_Score_b'1' (1.8354), A4_Score_b'1' (1.8352)
2	A1_Score_b'1' (105.1923), age_desc_b'18 and more' (88.1413), relation_b'Self' (73.9951), result (10.5082), age (5.8761), contry_of_res_b'Romania' (0.7624), contry_of_res_b'AmericanSamoa' (0.6113)
3	age_desc_b'4-11 years' (52.0342), contry_of_res_b'Syria' (0.4422), contry_of_res_b'Saudi Arabia' (0.3413), contry_of_res_b'Pakistan' (0.3201), contry_of_res_b'Iraq' (0.3011), contry_of_res_b'Kuwait' (0.1427), contry_of_res_b'Georgia' (0.1370)
4	relation_b'Parent' (87.1380), contry_of_res_b'Bahrain' (0.4163), contry_of_res_b'Costa Rica' (0.2684), contry_of_res_b'Georgia' (0.1506), contry_of_res_b'Burundi' (0.1437), contry_of_res_b'Anguilla' (0.1283), contry_of_res_b'South Korea' (0.1133)
5	contry_of_res_b'United Kingdom' (34.7669), age (0.3076), ethnicity_b'White-European' (0.0568), age_desc_b'12-16 years' (0.0172), relation_b'Self' (0.0117), jundice_b'yes' (0.0091), A1_Score_b'1' (0.0076)

Table 3.12: Top 7 Extreme Loadings for Combined Autism (cPCA - Frobenius)

Component	Top 7 Features and Loadings
1	relation_b'Self' (-0.6738), contry_of_res_b'United Arab Emirates' (-0.4007), contry_of_res_b'Jordan' (0.3334), ethnicity_b'Middle Eastern ' (-0.3130), age_desc_b'4-11 years' (-0.1271), ethnicity_b'Asian' (-0.1132), contry_of_res_b'New Zealand' (-0.1120)
2	contry_of_res_b'United Arab Emirates' (0.8087), relation_b'Self' (-0.3618), contry_of_res_b'New Zealand' (-0.3011), ethnicity_b'Asian' (-0.1615), ethnicity_b'Middle Eastern ' (0.1167), relation_b'Relative' (0.1009), ethnicity_b'White-European' (-0.0877)
3	contry_of_res_b'New Zealand' (-0.8465), relation_b'Self' (0.1863), contry_of_res_b'India' (0.1805), contry_of_res_b'United Arab Emirates' (-0.1800), A5_Score_b'1' (-0.1464), ethnicity_b'Pasifika' (0.1405), ethnicity_b'White-European' (0.1228)
4	contry_of_res_b'Iran' (-0.7687), contry_of_res_b'France' (-0.3580), contry_of_res_b'United Arab Emirates' (0.1368), ethnicity_b'Middle Eastern ' (-0.1365), contry_of_res_b'Canada' (0.1224), A5_Score_b'1' (-0.1139), ethnicity_b'Pasifika' (0.1100)
5	contry_of_res_b'France' (0.8530), contry_of_res_b'Iran' (-0.3382), Class/ASD_b'YES' (0.1251), A10_Score_b'1' (-0.1146), A7_Score_b'1' (-0.0942), ethnicity_b'Black' (0.0808), contry_of_res_b'Oman' (0.0762)

Table 3.13: Top 7 Extreme Loadings for Combined Autism (cPCA - L1)

Component	Top 7 Features and Loadings
1	relation_b'Self' (-0.7276), contry_of_res_b'Jordan' (0.3240), ethnicity_b'Middle Eastern ' (-0.2797), contry_of_res_b'United Arab Emirates' (-0.2474), contry_of_res_b'New Zealand' (-0.1627), ethnicity_b'Asian' (-0.1537), age_desc_b'4-11 years' (-0.1340)
2	contry_of_res_b'United Arab Emirates' (-0.8650), contry_of_res_b'New Zealand' (0.2678), relation_b'Self' (0.2273), ethnicity_b'Middle Eastern ' (-0.1760), ethnicity_b'Asian' (0.1581), contry_of_res_b'Sri Lanka' (0.0935), relation_b'Relative' (-0.0922)
3	contry_of_res_b'New Zealand' (-0.8204), relation_b'Self' (0.1944), contry_of_res_b'India' (0.1925), contry_of_res_b'United Arab Emirates' (-0.1914), ethnicity_b'Pasifika' (0.1531), A5_Score_b'1' (-0.1484), contry_of_res_b'Bangladesh' (0.1292)
4	contry_of_res_b'Sri Lanka' (-0.9400), contry_of_res_b'New Zealand' (0.2030), contry_of_res_b'India' (0.0964), ethnicity_b'Asian' (-0.0906), contry_of_res_b'Bangladesh' (0.0816), contry_of_res_b'Indonesia' (0.0671), contry_of_res_b'United Arab Emirates' (-0.0571)
5	contry_of_res_b'Iran' (0.7875), contry_of_res_b'France' (0.3132), contry_of_res_b'United Arab Emirates' (-0.1335), ethnicity_b'Middle Eastern ' (0.1312), contry_of_res_b'Canada' (-0.1254), A5_Score_b'1' (0.1120), Class/ASD_b'YES' (-0.1092)

In Figure 3.5, L1 and Frobenius norms yield different embeddings for cPCA and ICA, with L1 generally offering sharper age-group separations. ICA - L1 and PCA achieve the best separation, while the top 3 NMF components do not produce structure on the 3D plot. It is important to remember that the preprocessed Combined

Autism dataset has more than 100 variables. So, showing a 3D embedding might not be helpful for methods that output high dimensional reconstructed datasets.

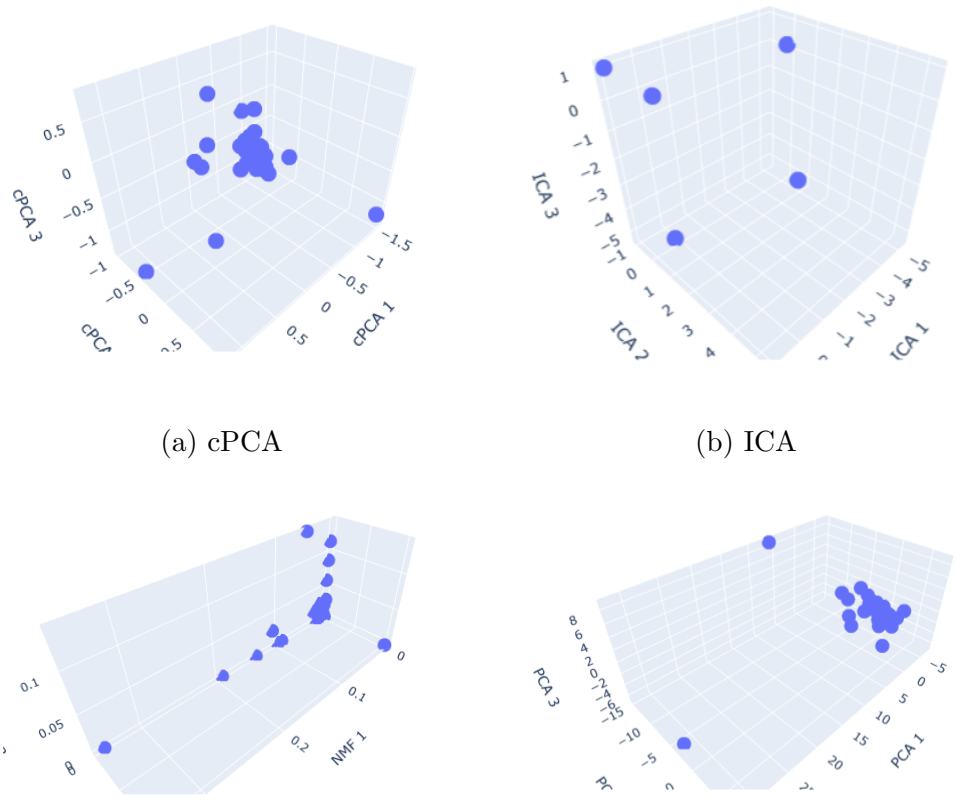
In Figure 3.6, heatmap comparison across methods shows interesting differences and similarities: Most of the variables with the darkest colors in cPCA - Frobenius and cPCA - L1 are the same variables; ICA - Frobenius and ICA - L1 heatmaps are almost completely different. In fact, it is surprising to see that ICA - L1 reduced the dimension to only 2. This is much smaller than what I expected.

A notable difference between the NMF heatmaps of the Accent and Combined Autism is that almost all variables of the top component have small loadings in the latter. This was the opposite in the Accent dataset.

Figure 3.7 demonstrates that PCA and ICA (L1) yield high visual separability among Adolescent, Child, and Adult groups. NMF picks out sparse contrasts again.

Tables 3.8 to 3.13 provide loadings that support the interpretation: cPCA is sensitive to geographic and relational contrasts, ICA - Frobenius to underrepresented regions, and NMF to specific test questions. Moreover, each of the NMF's components that are higher than the first component has one variable that dominates everything else. PCA focuses on a mix of screening scores and demographic indicators. A notable observation is that cPCA - Frobenius and cPCA - L1's first 3 components have almost the same most extreme variables, with different loadings.

3.3.3 Twitch Dataset



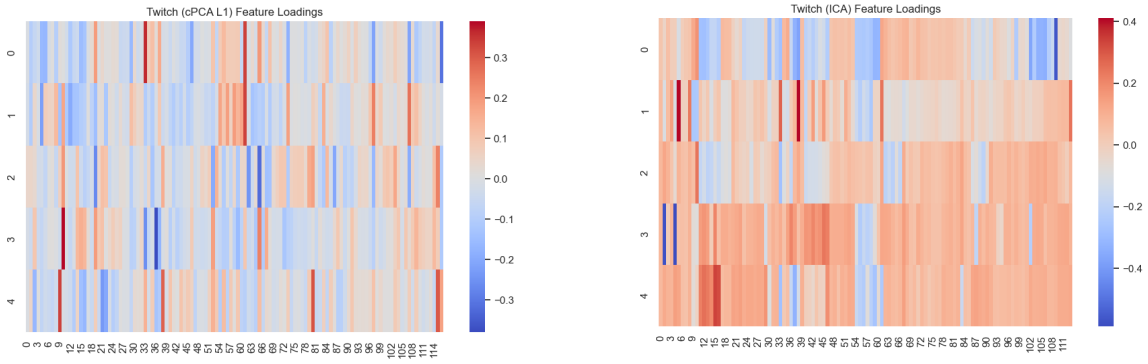
(a) cPCA

(b) ICA

(c) NMF

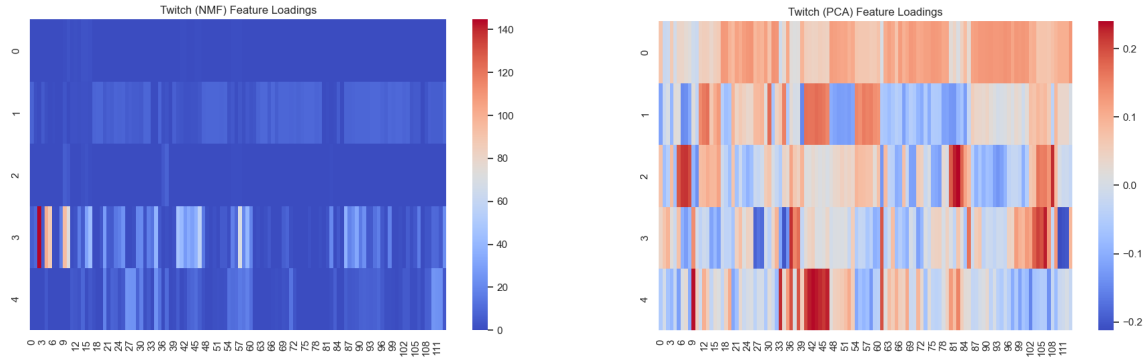
(d) PCA

Figure 3.8: 3D plots show the first 3 components outputted by methods applied on the Twitch dataset. For this dataset, Frobenius and L1 results are the same.



(a) cPCA Heatmap

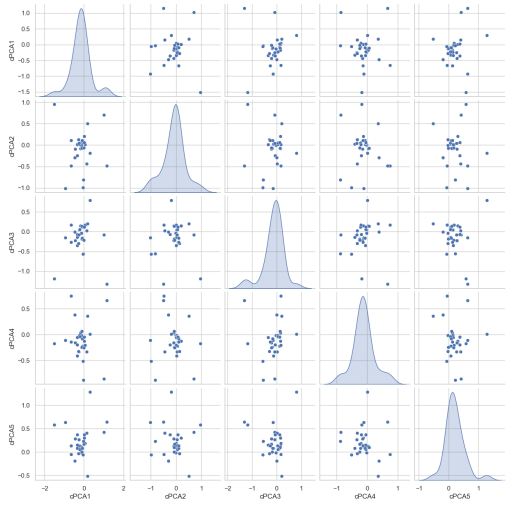
(b) ICA Heatmap



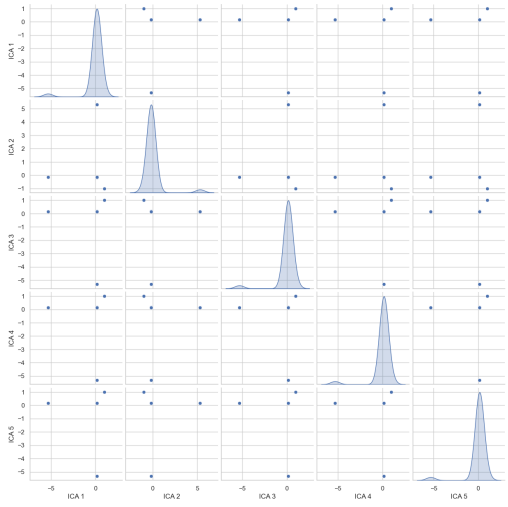
(c) NMF Heatmap

(d) PCA Heatmap

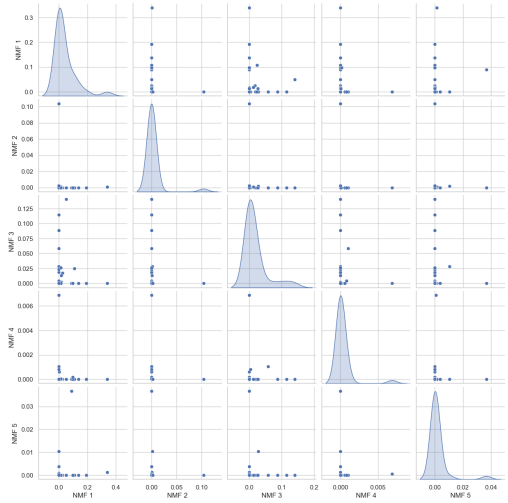
Figure 3.9: Heatmaps show the loadings of the variables in the top 5 components that the methods outputted. For this dataset, Frobenius and L1 results are the same.



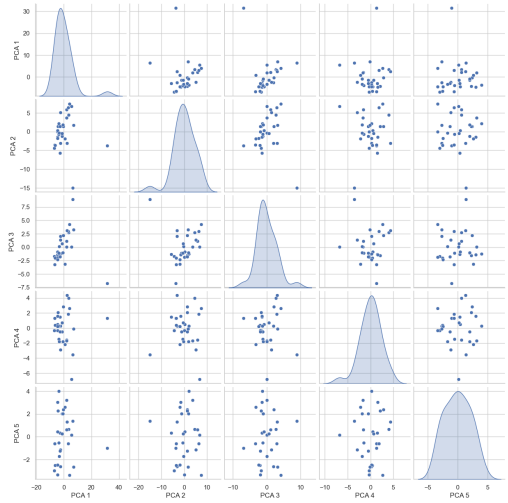
(a) cPCA Pairplot



(b) ICA Pairplot



(c) NMF Pairplot



(d) PCA Pairplot

Figure 3.10: Pairplots of the top 5 components across methods applied on the Twitch dataset are compared.

Table 3.14: Top 7 Extreme Loadings for Twitch (PCA - Frobenius / L1)

Component	Top 7 Features and Loadings
1	_03/21_views (0.1340), total_followers_tw (0.1339), t_views_gain (0.1325), _07/21_views (0.1315), _02/21_views (0.1315), total_views_tw (0.1311), _08/21_views (0.1303)
2	total_h_stream_tw (0.1744), avg_th_h_tw (0.1711), t_h_stream_tw (0.1681), avg_mo_h_tw (0.1676), avg_fr_h_tw (0.1663), avg_we_h_tw (0.1649), avg_tu_h_tw (0.1647)
3	_04/20_foll_balance (0.2398), _05/20_foll_balance (0.2275), _02/20_views (0.2237), tot_views_yt (0.2171), subs_yt (0.2165), n_videos_yt (0.2008), n_posts_twi (0.1891)
4	_04/20_views (0.2308), mst_cat_avg_w_channels (0.2190), avg_w_cunc_tw (-0.2123), _06/20_views (0.2085), avg_m_cunc_tw (-0.2044), mst_cat_h_tw (-0.2034), avg_t_cunc_tw (-0.2012)
5	avg_we_h_tw (0.2378), year_creation_yt (0.2358), avg_th_h_tw (0.2320), avg_tu_h_tw (0.2293), avg_su_h_tw (0.2287), avg_fr_h_tw (0.2254), A18flag_tw (0.2232)

Table 3.15: Top 7 Extreme Loadings for Twitch (ICA - Frobenius / L1)

Component	Top 7 Features and Loadings
1	_01/20_views (-0.5668), avg_n_cat_per_stream_tw (-0.3754), _08/20_views (-0.3685), _04/20_views (-0.3402), _06/20_views (-0.3327), mst_cat_avg_w_viewers (-0.3296), _05/20_views (-0.3294)
2	avg_n_cat_per_stream_tw (0.4108), n_posts_twi (0.4097), A18flag_tw (0.2784), tot_active_on_fr (-0.2513), year_creation_yt (-0.2509), female_flag (0.2461), avg_views_per_stream_tw (0.2441)
3	A18flag_tw (-0.2784), year_creation_tw (0.2440), _10/21_foll_balance (-0.2081), _09/21_views (-0.2018), _12/21_foll_balance (-0.2002), t_h_stream_tw (-0.1956), _08/21_foll_balance (-0.1837)
4	followee_twi (-0.5886), followee_insta (-0.5744), avg_sa_h_tw (0.2408), tot_active_on_tu (-0.2315), _08/20_views (-0.2214), avg_su_h_tw (0.2200), mst_cat_avg_w_channels (0.2109)
5	m_d_active_tw (0.3384), mst_cat_avg_w_viewers (-0.3296), t_d_active_tw (0.3220), A18flag_tw (-0.2784), tot_active_on_sa (-0.2679), m_h_stream_tw (0.2533), female_flag (-0.2461)

Table 3.16: Top 7 Extreme Loadings for Twitch (NMF - Frobenius / L1)

Component	Top 7 Features and Loadings
1	year_creation_tw (2.6245), m_d_active_tw (2.5117), w_d_active_tw (2.5099), t_d_active_tw (1.9552), m_h_stream_tw (1.1518), w_h_stream_tw (1.0614), avg_sa_h_tw (0.9232)
2	_10/20_views (9.4299), _05/21_views (9.3601), _08/21_foll_balance (9.3526), _09/20_foll_balance (9.3365), _11/20_foll_balance (9.3111), _10/21_foll_balance (9.2817), _09/20_views (9.2739)
3	mst_cat_avg_w_viewers (7.5242), year_creation_yt (6.0669), year_creation_tw (4.5053), mst_cat_avg_w_channels (4.1858), m_d_active_tw (2.6042), w_d_active_tw (1.6748), tot_active_on_tu (1.0315)
4	n_posts_insta (144.7039), followee_twi (101.1386), year_creation_yt (94.2542), n_posts_twi (87.1092), year_creation_tw (72.7770), tot_active_on_th (69.0822), avg_su_h_tw (56.5505)
5	avg_w_cunc_tw (25.1963), w_h_watched (25.1497), m_h_watched (24.5396), t_h_watched (24.1834), mst_cat_h_tw (23.9481), avg_m_cunc_tw (23.9406), avg_t_cunc_tw (23.2694)

Table 3.17: Top 7 Extreme Loadings for Twitch (cPCA - Frobenius / L1)

Component	Top 7 Features and Loadings
1	A18flag_tw (0.3619), female_flag (0.3373), language_tw_spanish (-0.3088), _03/20_views (-0.2160), _01/21_views (-0.2156), avg_su_h_tw (-0.2021), _09/21_foll_balance (0.2009)
2	female_flag (0.3326), _03/20_views (0.2644), _01/21_views (0.2434), followee_twi (-0.2242), m_h_stream_tw (-0.1982), tot_active_on_fr (0.1921), _01/21_foll_balance (0.1882)
3	_09/21_foll_balance (-0.3236), t_followers_gain (-0.2654), _12/21_foll_balance (-0.2464), language_tw_german (0.2405), _12/21_views (-0.2072), year_creation_tw (0.2033), avg_sa_foll_gain (-0.1962)
4	year_creation_tw (0.3895), mst_cat_avg_w_channels (-0.3792), A18flag_tw (-0.2562), _09/21_foll_balance (0.2406), t_followers_gain (0.1921), avg_sa_foll_gain (0.1866), mst_cat_avg_w_viewers (-0.1725)
5	year_creation_yt (0.3371), _06/20_foll_balance (0.3137), language_tw_german (0.3048), avg_n_cat_per_stream_tw (0.2919), m_peak_viewers (-0.2374), t_peak_viewers (-0.2031), language_tw_spanish (0.1812)

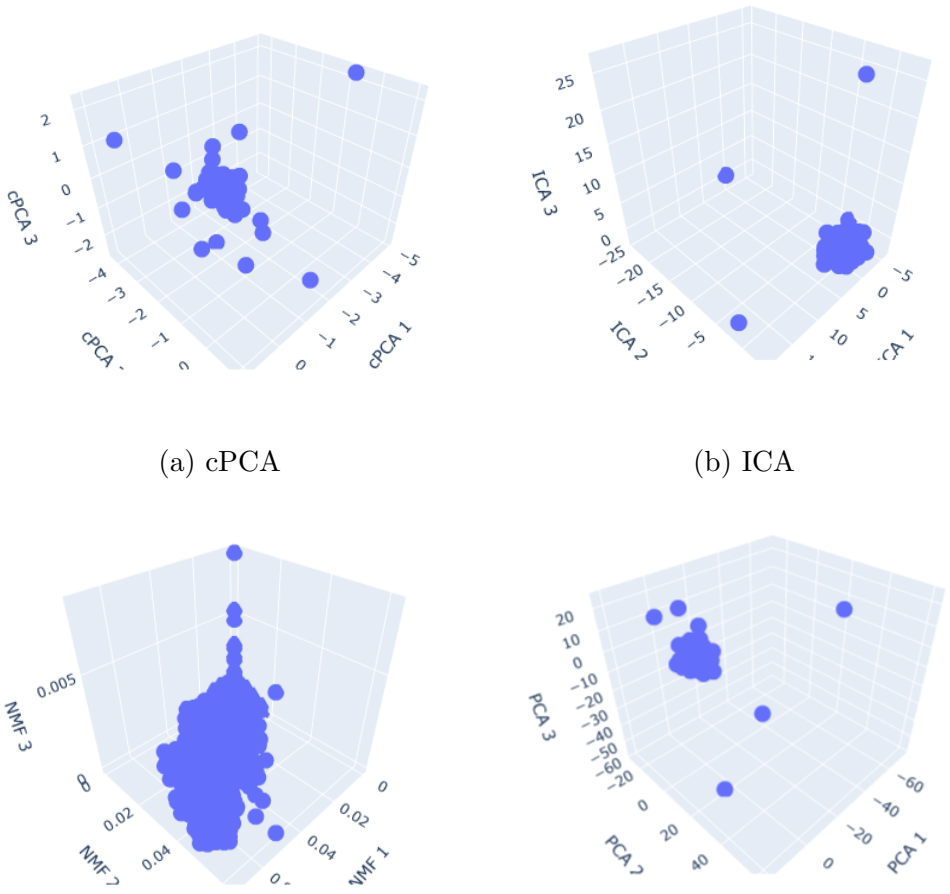
As shown in Figure 3.8, ICA forms diagonally structured clusters and cPCA pushing certain users to extreme corners of the plot, especially along the first component. NMF creates vertically stretched patterns, indicating uneven loadings across dimensions, while PCA results in a concentrated cloud.

Figure 3.9 confirms that PCA shows almost uniform contributions across variables to the first component. NMF, as expected, yields sparse and interpretable heatmaps.

In Figure 3.10, on the cPCA plot, most of the distributions are centered with nominal skew. This shows how much this method depends on the dataset having contrasting datapoints. When the datapoints do not have contrastive properties or values, it outputs concentrated groups located at the center.

Tables 3.14 to 3.17 show that cPCA and ICA emphasize complementary subsets of metadata. PCA covers broader indicators, and NMF identifies extreme influencers and content trends. ICA and cPCA separate user types more clearly than PCA or NMF. cPCA achieves clean contrastive projections using metadata such as age restrictions and gender, while PCA captures cumulative trends (e.g., total followers or views).

3.3.4 Dim512 Dataset



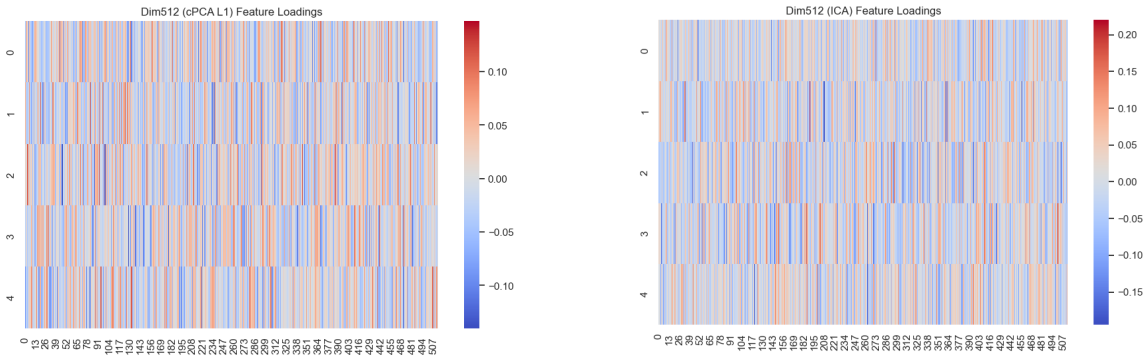
(a) cPCA

(b) ICA

(c) NMF

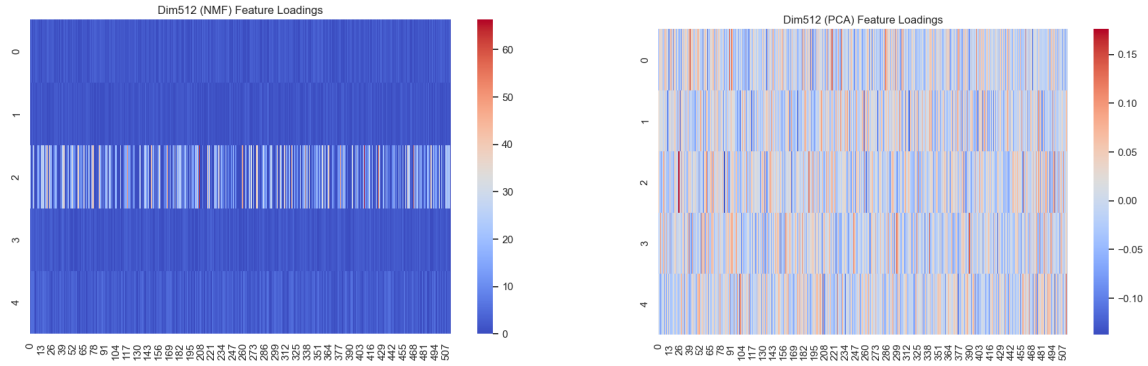
(d) PCA

Figure 3.11: 3D plot comparison across methods for the synthetic dataset Dim512.



(a) cPCA Heatmap

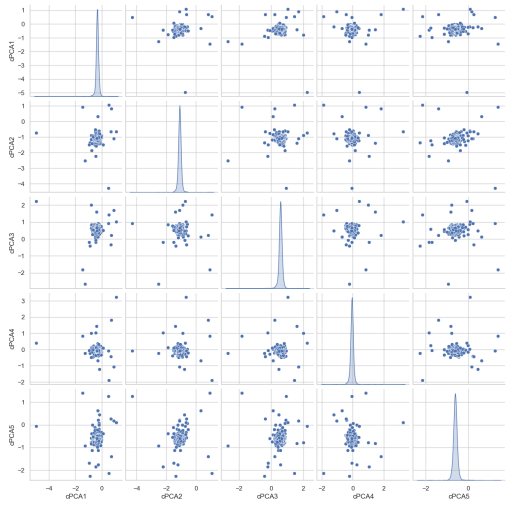
(b) ICA Heatmap



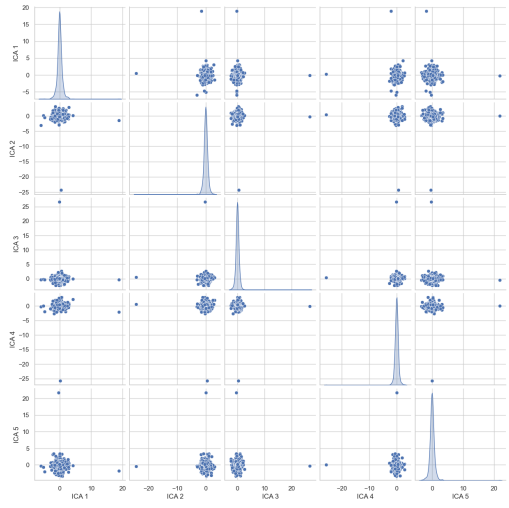
(c) NMF Heatmap

(d) PCA Heatmap

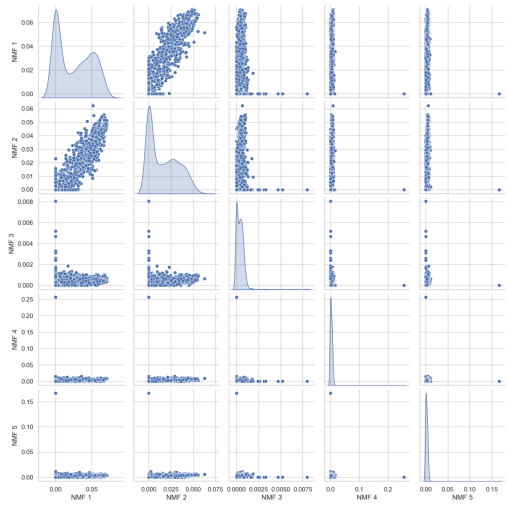
Figure 3.12: Heatmaps of the top 5 components show which synthetic variables dominate the components.



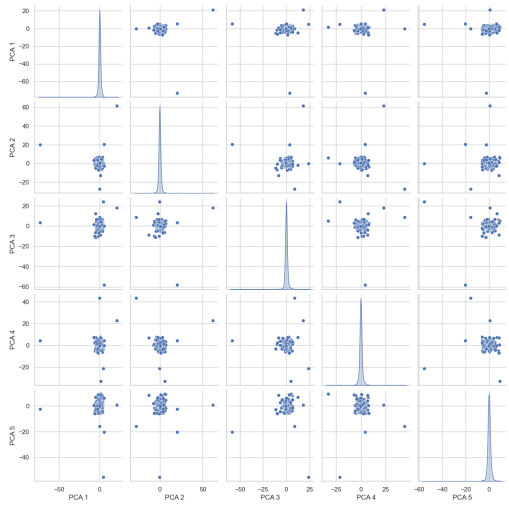
(a) cPCA Pairplot



(b) ICA Pairplot



(c) NMF Pairplot



(d) PCA Pairplot

Figure 3.13: Pairplots of the top 5 components are compared across methods.

Table 3.18: Top 7 Extreme Loadings for Dim512 (PCA - Frobenius / L1)

Component	Top 7 Features and Loadings
1	col_88 (0.1432), col_229 (0.1378), col_217 (0.1372), col_39 (0.1322), col_91 (0.1315), col_61 (-0.1146), col_398 (0.1144)
2	col_257 (0.1330), col_314 (-0.1218), col_302 (0.1135), col_455 (0.1069), col_405 (-0.1049), col_174 (-0.1036), col_494 (0.1028)
3	col_25 (0.1761), col_82 (-0.1296), col_188 (0.1176), col_496 (-0.1167), col_166 (-0.1162), col_133 (0.1157), col_457 (0.1126)
4	col_389 (0.1353), col_339 (0.1328), col_323 (0.1293), col_491 (0.1281), col_153 (-0.1248), col_38 (0.1234), col_298 (0.1197)
5	col_510 (0.1591), col_496 (-0.1375), col_234 (-0.1133), col_464 (0.1081), col_102 (0.1046), col_469 (-0.1018), col_101 (-0.1001)

Table 3.19: Top 7 Extreme Loadings for Dim512 (ICA - Frobenius / L1)

Component	Top 7 Features and Loadings
1	col_24 (-0.1443), col_141 (-0.1422), col_49 (-0.1229), col_418 (0.1222), col_345 (-0.1216), col_272 (0.1212), col_444 (0.1130)
2	col_189 (0.2201), col_455 (-0.1947), col_207 (-0.1868), col_246 (-0.1571), col_336 (0.1502), col_84 (0.1441), col_33 (0.1387)
3	col_350 (-0.1836), col_294 (-0.1722), col_331 (-0.1521), col_42 (-0.1517), col_292 (0.1501), col_376 (0.1477), col_160 (-0.1420)
4	col_185 (0.1828), col_261 (-0.1695), col_500 (0.1620), col_85 (-0.1591), col_3 (-0.1588), col_55 (0.1495), col_202 (0.1490)
5	col_120 (-0.1822), col_261 (-0.1762), col_199 (-0.1560), col_43 (0.1530), col_146 (0.1445), col_348 (0.1353), col_403 (0.1351)

Table 3.20: Top 7 Extreme Loadings for Dim512 (NMF - Frobenius / L1)

Component	Top 7 Features and Loadings
1	col_427 (4.7885), col_21 (4.5763), col_256 (4.4263), col_210 (4.3034), col_158 (4.2665), col_198 (4.2435), col_354 (4.0724)
2	col_416 (4.0220), col_281 (4.0206), col_125 (3.8242), col_104 (3.6336), col_465 (3.5896), col_381 (3.5401), col_393 (3.4753)
3	col_407 (66.3408), col_206 (64.0948), col_424 (57.1425), col_259 (56.4948), col_467 (52.6120), col_118 (50.3967), col_167 (48.7842)
4	col_243 (3.9933), col_486 (3.9399), col_321 (3.8953), col_101 (3.8569), col_469 (3.8423), col_141 (3.8196), col_7 (3.8123)
5	col_171 (6.0089), col_324 (5.9351), col_382 (5.9345), col_289 (5.9193), col_3 (5.9164), col_492 (5.9033), col_192 (5.8792)

Table 3.21: Top 7 Extreme Loadings for Dim512 (cPCA - Frobenius / L1)

Component	Top 7 Features and Loadings
1	col_470 (-0.1190), col_215 (-0.1109), col_408 (0.1108), col_451 (0.1097), col_318 (0.1094), col_357 (-0.1055), col_165 (-0.1047)
2	col_235 (-0.1202), col_124 (0.1168), col_81 (0.1155), col_427 (-0.1097), col_393 (-0.1096), col_293 (-0.1091), col_463 (-0.1070)
3	col_385 (0.1475), col_46 (-0.1322), col_86 (-0.1278), col_344 (0.1249), col_99 (-0.1199), col_92 (0.1179), col_201 (-0.1136)
4	col_128 (-0.1400), col_178 (-0.1304), col_481 (-0.1245), col_440 (0.1169), col_469 (-0.1109), col_268 (0.1101), col_220 (-0.1098)
5	col_80 (-0.1375), col_470 (-0.1345), col_219 (-0.1285), col_279 (0.1231), col_132 (0.1144), col_94 (0.1081), col_76 (-0.1065)

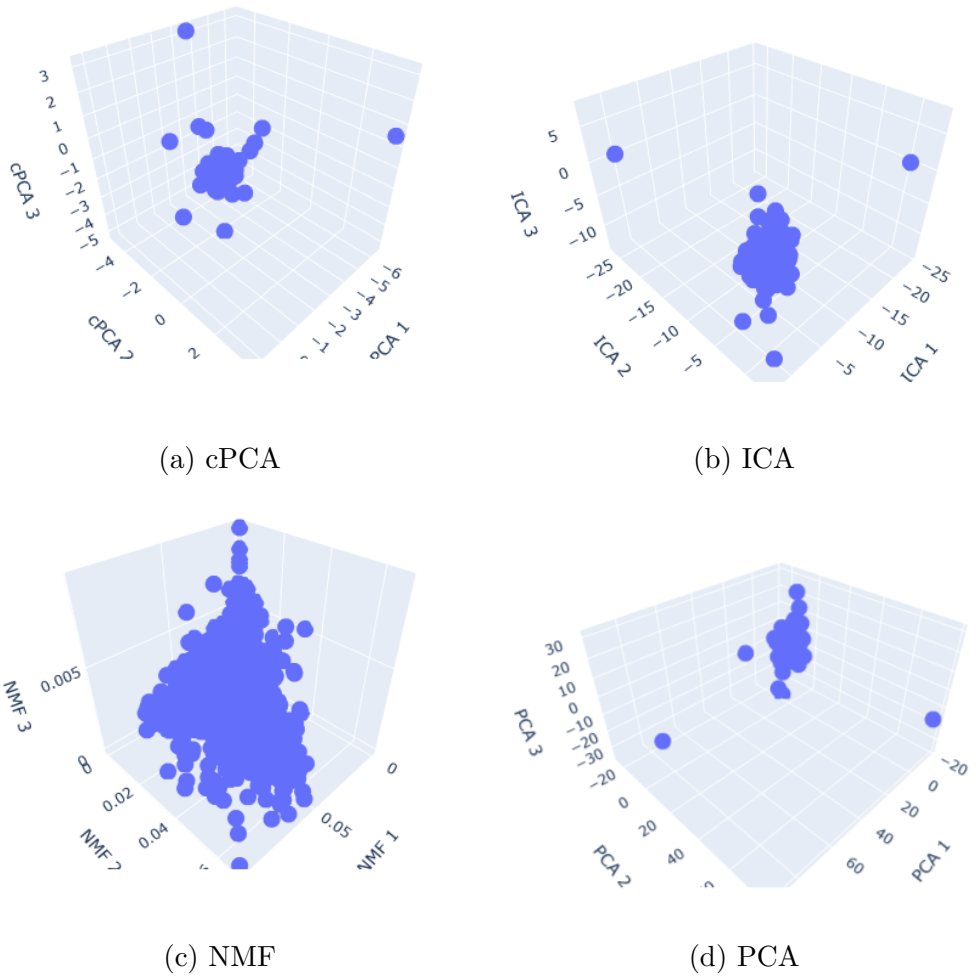
Figure 3.11 shows that all four methods uncover clusters in the Dim512 dataset, but their embeddings differ in density and orientation. PCA and ICA reveal horizontal spreads with distinct clusters, while cPCA forms more vertically aligned groups. NMF, on the other hand, produces narrow clusters, reflecting its emphasis on sparsity.

In Figure 3.12, PCA distributes loadings relatively evenly across variables, with moderate values. ICA selects more localized features, generating components built from a few moderate-strength variables. NMF shows sparsity, with only a handful of features exhibiting very large loadings, in the 3rd component. However, other NMF components do not exhibit this behavior. cPCA displays a wide spread of loadings, with no extremely dominant variable—likely a consequence of the lack of background contrast in synthetic data.

As Figure 3.13 illustrates, the pairplots show that ICA and PCA produce more tightly packed clusters with sharp boundaries, suggesting more localized encoding. cPCA shows diffuse grouping—indicating reduced effectiveness in the absence of meaningful background-target contrast.

Loadings in Tables 3.18 to 3.21 indicate that PCA and cPCA distribute loadings widely. ICA emphasizes moderate independence. NMF yields sharp, interpretable loading values, with some exceeding 60 in magnitude.

3.3.5 Dim1024 Dataset



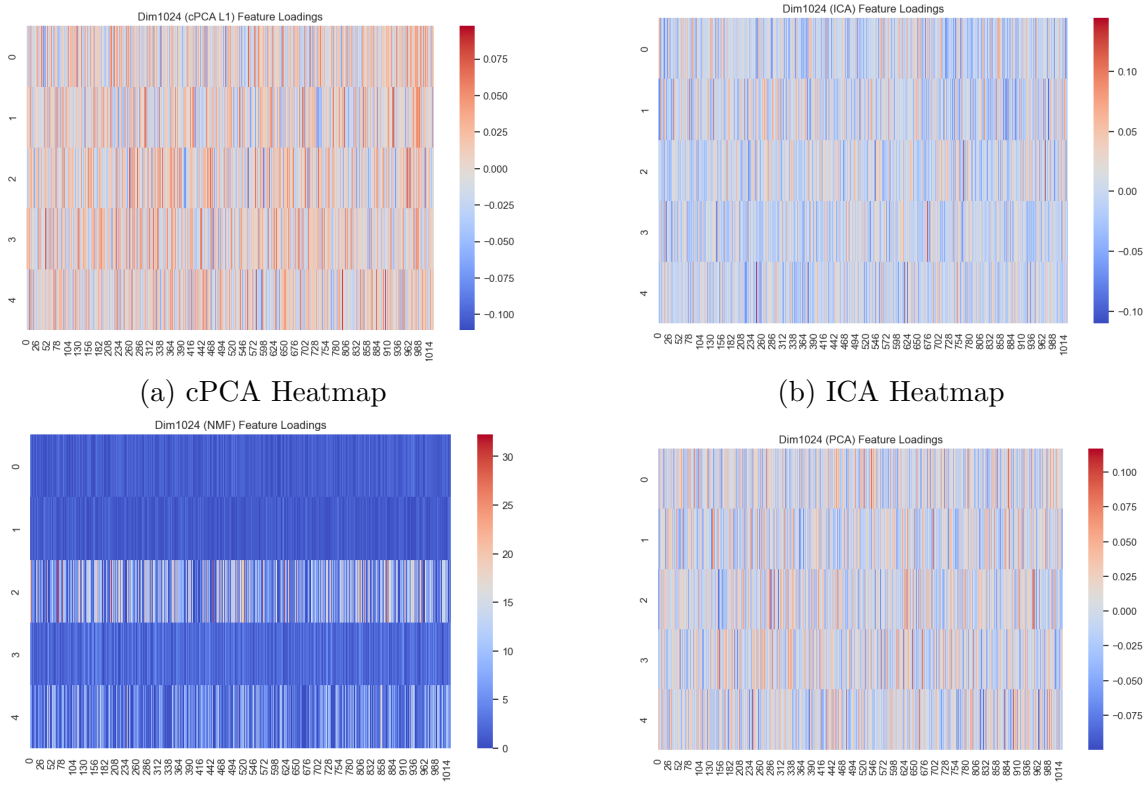
(a) cPCA

(b) ICA

(c) NMF

(d) PCA

Figure 3.14: 3D plots of the components are compared across methods for the synthetic dataset Dim1024.



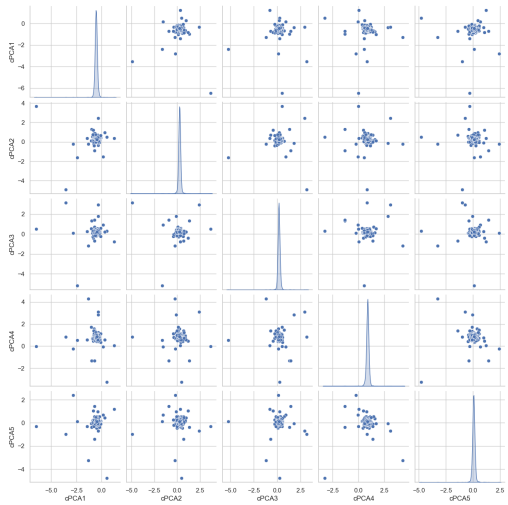
(a) cPCA Heatmap

(b) ICA Heatmap

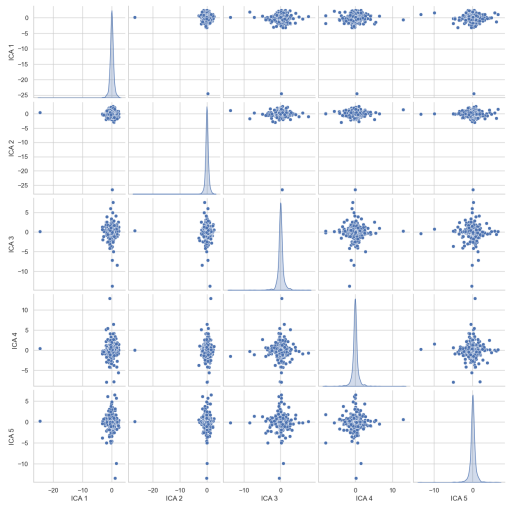
(c) NMF Heatmap

(d) PCA Heatmap

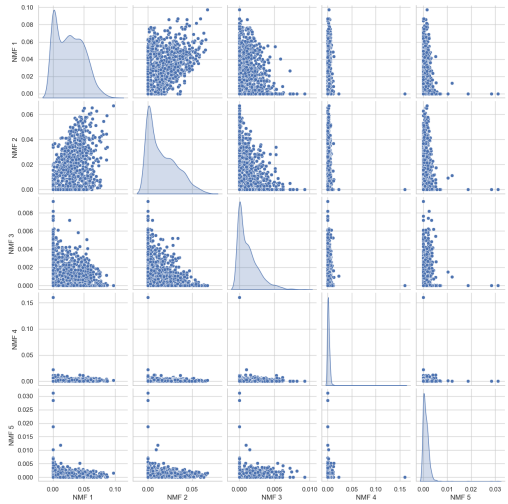
Figure 3.15: Heatmaps make it easier to compare methods by looking at the loadings of the variables of the top components.



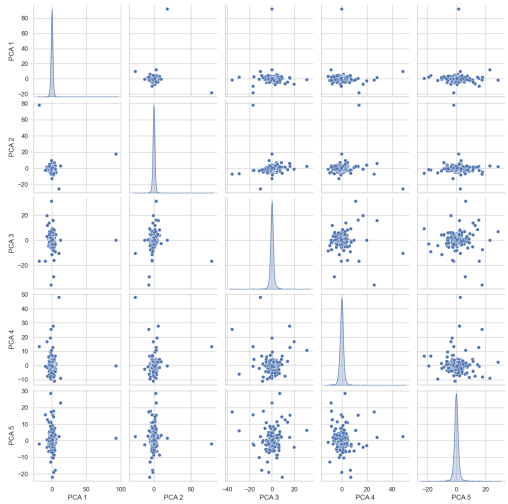
(a) cPCA Pairplot



(b) ICA Pairplot



(c) NMF Pairplot



(d) PCA Pairplot

Figure 3.16: Pairplots of the top 5 components of the methods are graphed.

Table 3.22: Top 7 Extreme Loadings for Dim1024 (PCA - Frobenius / L1)

Component	Top 7 Features and Loadings
1	col_886 (0.1003), col_295 (0.0975), col_539 (0.0925), col_103 (0.0908), col_667 (0.0904), col_935 (-0.0866), col_274 (0.0850)
2	col_696 (0.1028), col_553 (-0.0992), col_940 (0.0931), col_137 (0.0913), col_857 (0.0828), col_915 (-0.0827), col_134 (-0.0826)
3	col_283 (0.1171), col_949 (0.1122), col_304 (-0.0974), col_294 (0.0908), col_455 (0.0905), col_630 (0.0891), col_152 (-0.0876)
4	col_668 (0.0960), col_643 (0.0927), col_811 (-0.0920), col_558 (-0.0880), col_773 (0.0868), col_793 (0.0853), col_289 (0.0808)
5	col_238 (0.1019), col_382 (-0.0999), col_899 (0.0997), col_823 (-0.0995), col_135 (-0.0975), col_759 (-0.0967), col_964 (-0.0938)

Table 3.23: Top 7 Extreme Loadings for Dim1024 (ICA - Frobenius / L1)

Component	Top 7 Features and Loadings
1	col_642 (0.0985), col_850 (0.0975), col_834 (-0.0900), col_971 (-0.0894), col_604 (-0.0886), col_248 (-0.0882), col_668 (-0.0830)
2	col_78 (0.1120), col_975 (-0.1036), col_363 (0.0966), col_815 (0.0933), col_232 (-0.0863), col_525 (-0.0856), col_83 (-0.0853)
3	col_764 (-0.1098), col_791 (-0.1043), col_847 (0.0990), col_969 (-0.0977), col_542 (-0.0870), col_37 (-0.0835), col_685 (0.0802)
4	col_673 (0.1328), col_145 (-0.1044), col_739 (0.0962), col_15 (0.0911), col_325 (0.0859), col_470 (0.0848), col_396 (-0.0844)
5	col_245 (0.1445), col_416 (0.1043), col_623 (-0.0992), col_747 (0.0971), col_559 (0.0946), col_748 (0.0932), col_65 (-0.0928)

Table 3.24: Top 7 Extreme Loadings for Dim1024 (NMF - Frobenius / L1)

Component	Top 7 Features and Loadings
1	col_827 (4.3236), col_954 (3.5926), col_491 (3.5752), col_569 (3.5290), col_935 (3.5229), col_173 (3.4772), col_151 (3.4459)
2	col_938 (4.0269), col_286 (4.0233), col_992 (3.7889), col_617 (3.5798), col_735 (3.5574), col_1006 (3.5200), col_344 (3.5141)
3	col_68 (32.2813), col_197 (31.4645), col_520 (30.0847), col_443 (29.8076), col_819 (28.2869), col_620 (27.3040), col_173 (26.2480)
4	col_45 (5.9975), col_788 (5.9935), col_141 (5.9778), col_226 (5.9376), col_882 (5.8852), col_647 (5.8104), col_436 (5.7905)
5	col_855 (19.9745), col_236 (19.2876), col_116 (18.6209), col_125 (18.3171), col_932 (17.2302), col_674 (17.1260), col_524 (16.3982)

Table 3.25: Top 7 Extreme Loadings for Dim1024 (cPCA - Frobenius / L1)

Component	Top 7 Features and Loadings
1	col_449 (-0.0890), col_286 (-0.0860), col_558 (0.0845), col_574 (0.0843), col_42 (-0.0806), col_642 (-0.0805), col_243 (-0.0765)
2	col_48 (0.0964), col_698 (-0.0886), col_921 (0.0883), col_300 (0.0870), col_799 (0.0864), col_695 (0.0846), col_335 (-0.0818)
3	col_558 (-0.0995), col_818 (0.0934), col_997 (-0.0919), col_831 (-0.0850), col_278 (0.0821), col_708 (0.0805), col_399 (-0.0804)
4	col_284 (-0.1108), col_607 (-0.0909), col_651 (0.0905), col_319 (-0.0875), col_685 (0.0859), col_131 (0.0857), col_415 (0.0856)
5	col_578 (-0.1101), col_587 (-0.1043), col_794 (0.0978), col_903 (0.0939), col_591 (-0.0928), col_246 (0.0897), col_365 (0.0895)

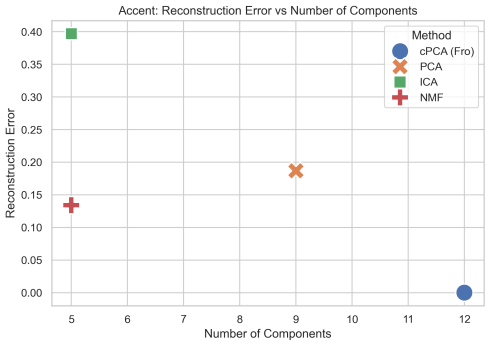
As seen in Figure 3.14, all methods identify separable clusters. cPCA provides more centered concentrated embeddings, whereas ICA and PCA outputs embeddings that are closer to the edges of the 3D space. NMF compresses the data into spread-out and more linear shape.

In Figure 3.15, PCA and ICA assign moderate loadings across a range of variables. NMF demonstrates pronounced sparsity in the 3rd and 5th components, with some loadings reaching values exceeding 30. cPCA shows heatmaps with varying but mild loadings, consistent with the expectation of results after a contrastive method is applied to data lacking explicit background.

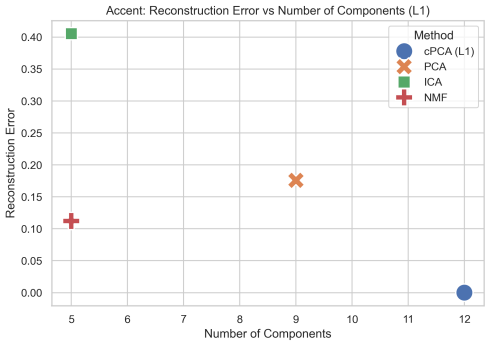
The pairplots in Figure 3.16 show that PCA and cPCA isolate dense cores, but offer little inter-cluster separation. ICA, meanwhile, preserves the dataset's spread better but produces looser clusters. NMF's first 3 components provide minimally concentrated datapoints, but the 4th and 5th components align them along one axis.

Tables 3.22 to 3.25 show consistency with Dim512 in method behavior. Notably, NMF achieves sparse and large loading values in the 3rd component (e.g., 32.2813 for the variable *col_68*) in small groups of variables.

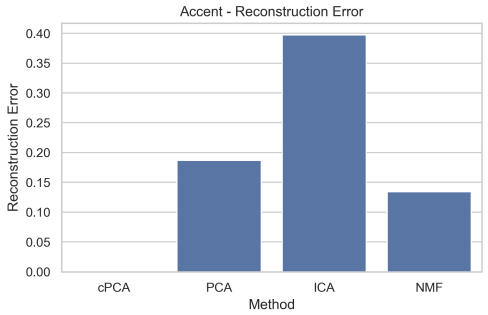
3.3.6 Final Results



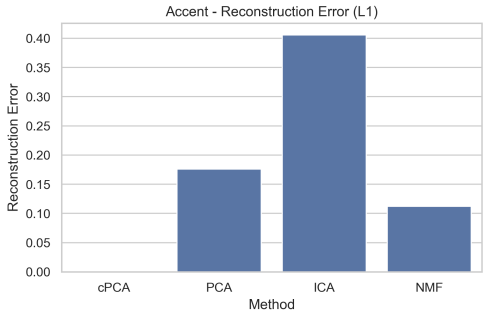
(a) Frobenius Norm Error vs Number of Components



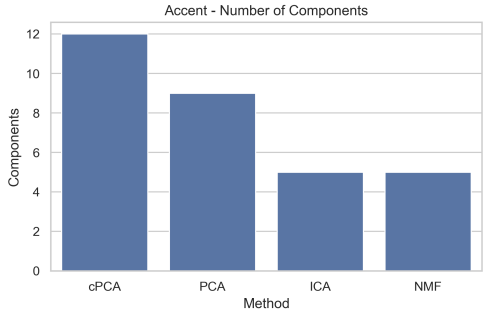
(b) L1 Norm Error vs Number of Components



(c) Frobenius Norm Errors

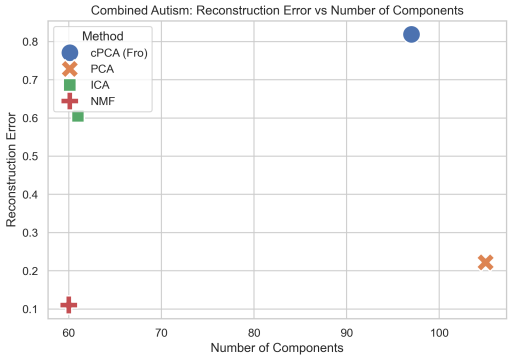


(d) L1 Norm Errors

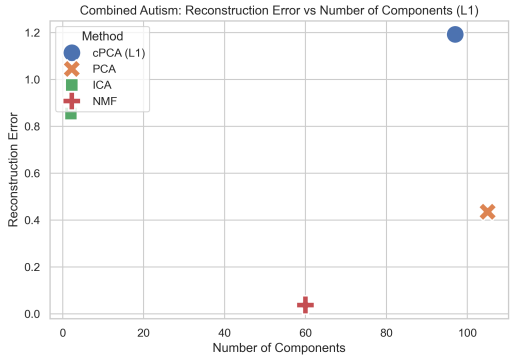


(e) Number of Components

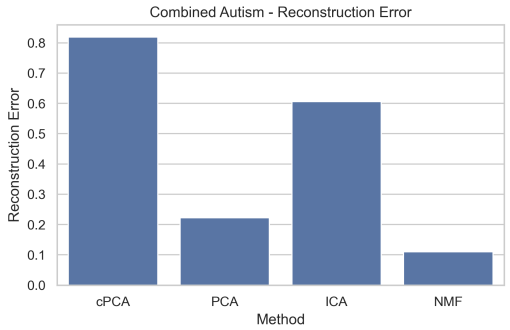
Figure 3.17: Final results for the Accent dataset include barplots and error vs. number of component scatter plots. Frobenius and L1 norms yield equal number of components.



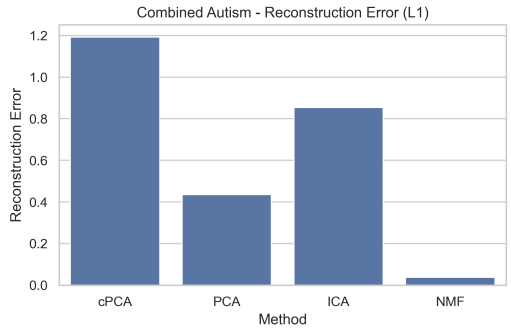
(a) Frobenius Error vs Number of Components



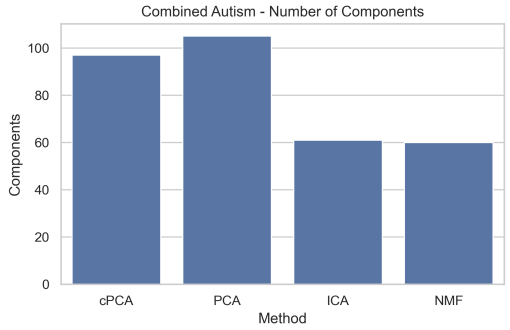
(b) L1 Norm Error vs Number of Components



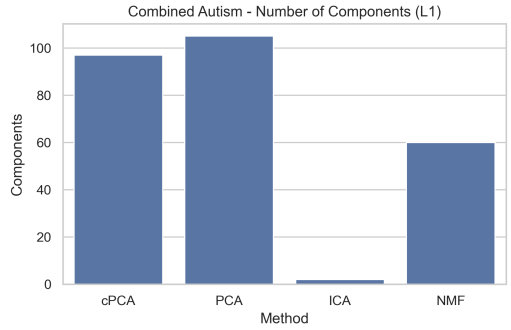
(c) Frobenius Norm Errors



(d) L1 Norm Errors

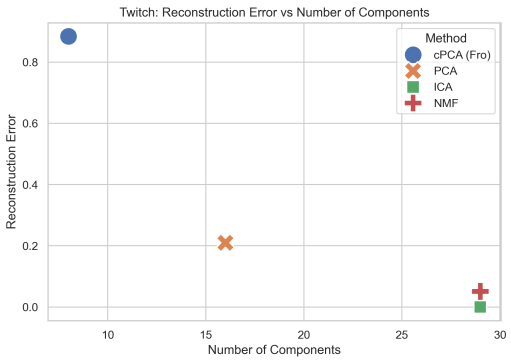


(e) Number of Components (Frobenius)

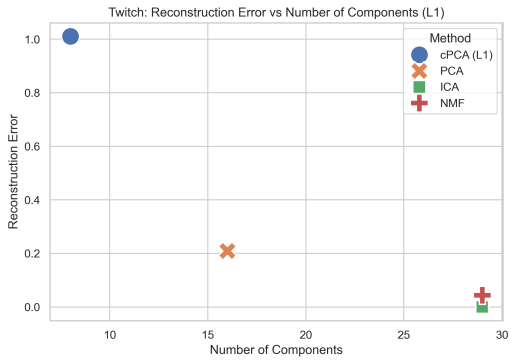


(f) Number of Components (L1)

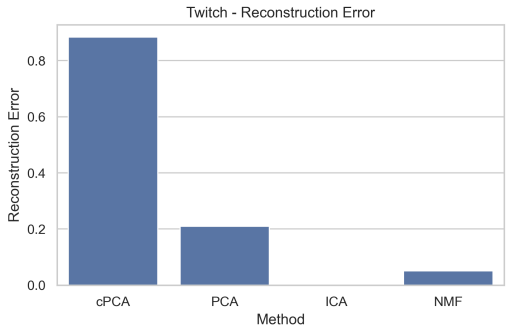
Figure 3.18: Final results for the Combined Autism dataset include barplots and error vs. number of component scatter plots. Frobenius and L1 norms yield different results for this dataset. That is why both panels (e) and (f) are shown.



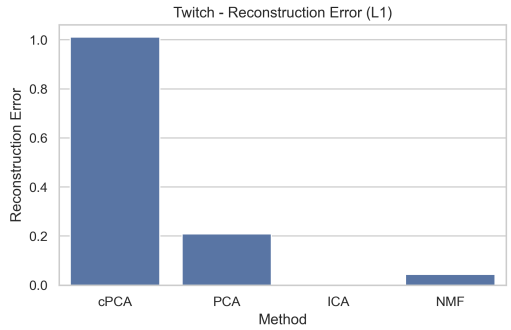
(a) Frobenius Error vs Number of Components



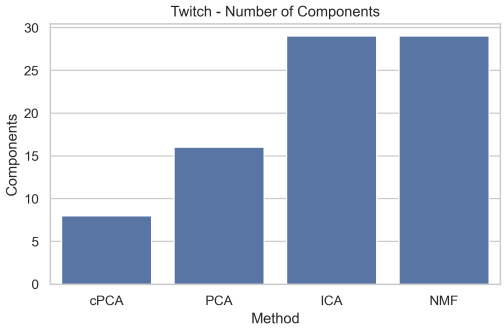
(b) L1 Norm Error vs Number of Components



(c) Frobenius Norm Errors

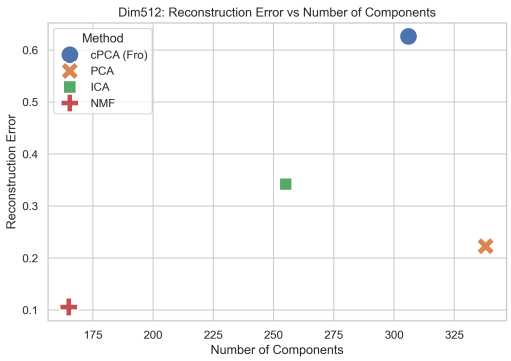


(d) L1 Norm Errors

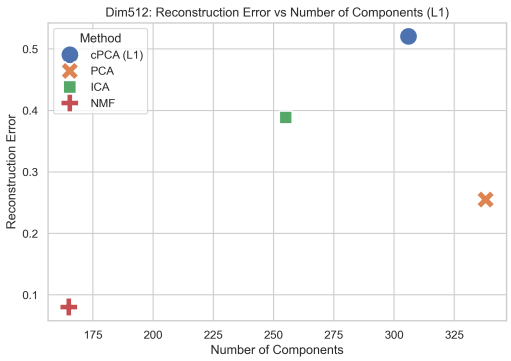


(e) Number of Components

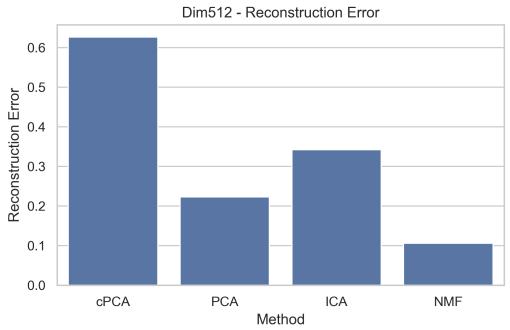
Figure 3.19: Final results for the Twitch dataset include barplots and error vs. number of component scatter plots. Frobenius and L1 norms yield equal number of components.



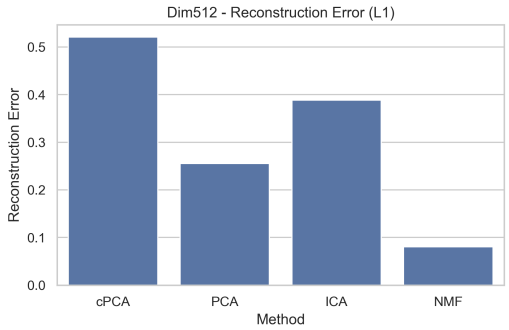
(a) Frobenius Error vs Number of Components



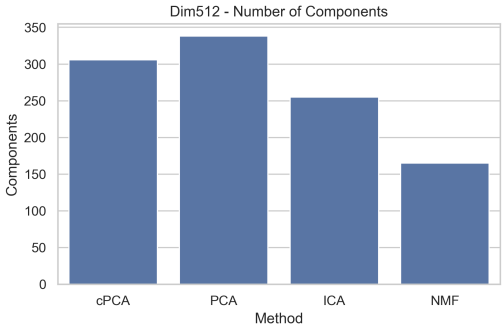
(b) L1 Norm Error vs Number of Components



(c) Frobenius Norm Errors

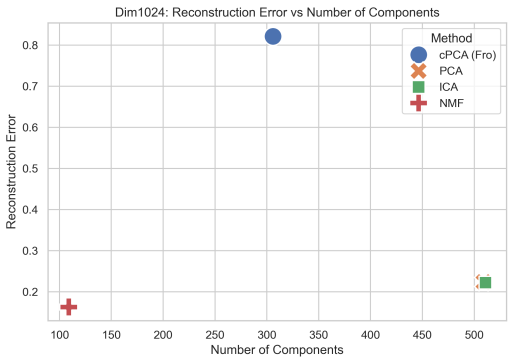


(d) L1 Norm Errors

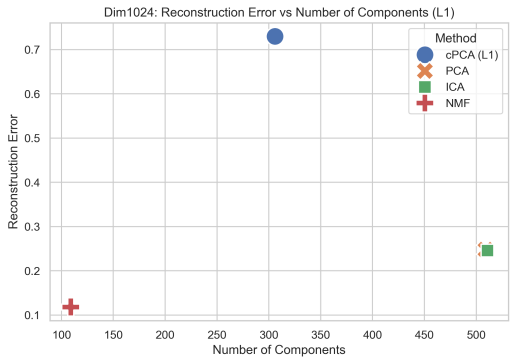


(e) Number of Components

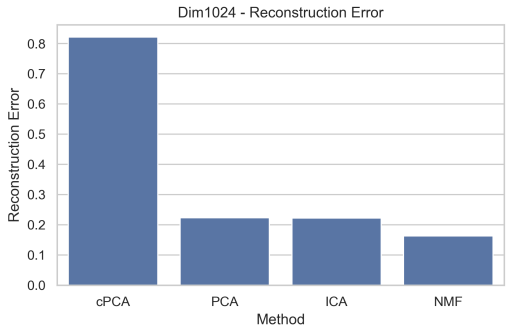
Figure 3.20: Final results for the Dim512 dataset include barplots and error vs. number of component scatter plots. Frobenius and L1 norms yield equal number of components.



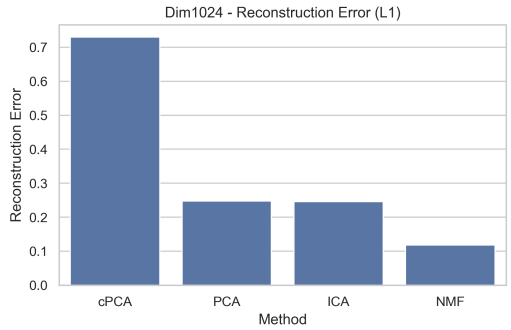
(a) Frobenius Error vs Number of Components



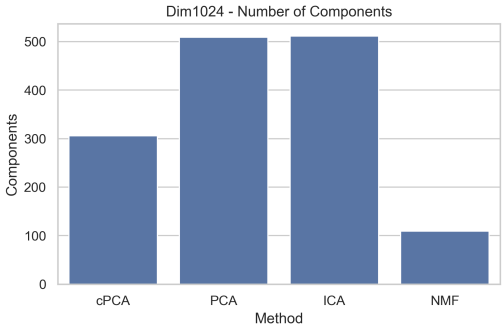
(b) L1 Norm Error vs Number of Components



(c) Frobenius Norm Errors



(d) L1 Norm Errors



(e) Number of Components

Figure 3.21: Final results for the Dim1024 dataset include barplots and error vs. number of component scatter plots. Frobenius and L1 norms yield equal number of components.

Tables 3.26 and 3.27 report the original number of dimensions, number of remaining dimensions after applying each method, and the relative reconstruction errors (Frobenius in Table 3.24 and L1 in Table 3.25).

Table 3.26: Final Frobenius Results Table

Dataset	Original Dimension	cPCA Components	cPCA Error	PCA Components	PCA Error	ICA Components	ICA Error	NMF Components	NMF Error
Combined Autism	125	97	0.8187	105	0.2220	61	0.6056	60	0.1106
Twitch	148	8	0.8838	16	0.2096	29	0.0000	29	0.0508
Accent	18	12	0.0000	9	0.1868	5	0.3971	5	0.1341
Dim512	512	306	0.6262	338	0.2227	255	0.3419	165	0.1058
Dim1024	1024	306	0.8211	509	0.2235	511	0.2219	109	0.1628

Table 3.27: Final L1 Results Table

Dataset	Original Dimension	cPCA Components	cPCA Error	PCA Components	PCA Error	ICA Components	ICA Error	NMF Components	NMF Error
Combined Autism	125	97	1.1920	105	0.4352	2	0.8542	60	0.0375
Twitch	148	8	1.0104	16	0.2090	29	0.0000	29	0.0434
Accent	18	12	0.0000	9	0.1758	5	0.4055	5	0.1123
Dim512	512	306	0.5205	338	0.2551	255	0.3887	165	0.0804
Dim1024	1024	306	0.7297	509	0.2477	511	0.2459	109	0.1183

Figures 3.17 to 3.21 summarize the final results across all datasets by displaying relative construction error trends and the number of selected components for each method under both Frobenius and L1 norms. These visualizations help reveal patterns and tradeoffs.

In Figure 3.17 (Accent), all methods except ICA show comparable error trends between the Frobenius and L1 norms. It might look like cPCA achieves perfect reconstruction on both metrics, demonstrating its exceptional contrastive power in accent classification; however, it selects almost all original dimensions as components. PCA and NMF follow closely with low errors, while ICA underperforms with noticeably higher reconstruction loss.

Figure 3.18 (Combined Autism) highlights a divergence between the Frobenius and L1 norms. Under Frobenius, NMF yields the lowest error with the fewest components. In contrast, under L1, ICA uses the fewest number of components, only two, an unexpectedly aggressive compression that likely indicates a problem. cPCA, while still selecting many components, shows relatively high L1 error—demonstrating that it does not benefit from the same sparsity as ICA or NMF on this dataset.

In Figure 3.19 (Twitch), cPCA achieves the worst reconstruction error under both norms, with only 8 components. This might be caused by the fact that this dataset does not have groups with high contrast. Meanwhile, ICA performs with 29 components and 0 error under L1, which indicates that there is a problem, similar to the Combined Autism dataset's results. NMF provides the lowest error with the greatest number of components among all non-ICA methods.

Figures 3.20 and 3.21 (Dim512 and Dim1024) show the challenge of applying dimensionality reduction to high-dimensional synthetic datasets. PCA consistently provides a strong balance between error minimization and compression. cPCA, despite using over 300 components, has the highest relative reconstruction error under both metrics, showing how ineffective it is when applied to datasets with no clear contrast. NMF performs best, using fewer than a third of the original dimensions. ICA again shows low L1 and Frobenius errors.

Tables 3.26 and 3.27 report the original number of features, selected component counts, and relative reconstruction errors across all methods and datasets. They confirm that PCA is an all-rounder with solid performance across both Frobenius and L1 norms, cPCA excels in datasets with clear contrast and struggles in synthetic settings, ICA can minimize reconstruction error dramatically——but sometimes at the cost of instability, and NMF yields interpretable representations with strong Frobenius performance.

These final results highlight that no method dominates; effectiveness depends heavily on dataset structure and evaluation norm. Method selection should thus be guided by whether interpretability, compression, or contrastive insight is prioritized.

Chapter 4

Conclusion

4.1 Comparison of Results

4.1.1 Observations and Findings

PCA demonstrated stable and balanced performance across all datasets, particularly in high-dimensional settings like Dim512 and Dim1024. It consistently preserved global structure, yielded low Frobenius and L1 reconstruction errors, and selected moderate component counts. PCA's components often reflected the overall variance pattern, with the first component resembling an average over all variables.

cPCA excelled in datasets where a clear contrast between target and background existed—namely the Accent and Combined Autism datasets. However, its performance declined significantly on synthetic datasets, where contrastive structure was absent, emphasizing its dependence on meaningful background-target divergence.

ICA selected components based on statistical independence and frequently highlighted underrepresented or behaviorally distinct features. On the Twitch dataset, it uniquely achieved zero L1 error using only two components—a result that, while efficient, raises concerns about L1 optimization. In high-dimensional synthetic datasets, ICA's Frobenius errors were consistently higher than those of PCA and NMF, limiting its utility for reconstruction-driven tasks.

NMF produced sparse and interpretable representations. It performed especially well on the Combined Autism dataset under the Frobenius norm, yielding the lowest error across all methods. While its L1 performance deteriorated on synthetic datasets, NMF remained highly competitive under the Frobenius norm.

4.1.2 Summary of the Analysis of Extreme Loadings

Accent Dataset: PCA and ICA consistently emphasized key MFCC (Mel-Frequency Cepstral Coefficients) variables like X10, X9, and X6. ICA's loadings showed opposing signs, enhancing contrast, while NMF's extreme sparsity concentrated magnitude on just a few coefficients. cPCA captured accent group differences using clear directional contrasts in loadings, making it particularly useful for speech classification.

Combined Autism Dataset: PCA and ICA highlighted demographic and screening features (e.g., age descriptors, Class/ASD flag, relation). ICA's emphasis on rare countries and ethnicities revealed subgroup sensitivity. NMF produced components with extreme magnitudes (e.g., A1 score > 100), underscoring interpretable links to clinical traits. cPCA emphasized relational and regional contrasts, often surfacing meaningful sociocultural markers.

Twitch Dataset: ICA uncovered behavioral signals like average category count per stream, largely ignored by other methods. PCA emphasized scale-related variables such as total views and followers. NMF isolated influencers via extreme metrics like number of posts and follower count. cPCA extracted demographic contrast, notably gender and age restrictions, offering distinct group separation.

Synthetic Datasets: In Dim512 and Dim1024, PCA retained variance structure with broadly distributed loadings. ICA and NMF extracted localized or sparse features, with NMF producing loadings exceeding 60 (e.g., col 407, col 68), supporting variable selection or compression tasks. cPCA showed limited value due to a lack of structured background contrast.

4.1.3 Overall Performance Comparison

Table 4.1: Method Strengths Summary

Method	Strengths	Best For	Notes
PCA	Variance preservation	Synthetic Datasets	Strong error-performance tradeoff
cPCA	Target-background contrast	Accent, Twitch	Requires contrastive structure
ICA	Statistical independence	Twitch, Autism	Sensitive; can over-optimize L1
NMF	Sparsity, interpretability	Autism, Twitch	Strong Frobenius performance

4.1.4 Key Trends Across Datasets

- **Dimensionality Sensitivity:** As dimensionality increased, ICA and NMF showed elevated L1 errors and unstable behavior. PCA remained reliable and

scalable, while cPCA's performance plateaued in synthetic scenarios.

- **Contrast Dependency:** cPCA performed best when contrastive structure was meaningful and well-defined. In its absence (e.g., synthetic datasets), its performance significantly declined.
- **Interpretability vs. Accuracy Tradeoff:** PCA and cPCA generally yielded lower reconstruction errors, whereas ICA and NMF offered more interpretable components—often at the cost of reconstruction accuracy. Hybrid strategies may be needed based on the application goal.

4.1.5 Frobenius vs. L1 Norm Error Metrics

This thesis evaluated reconstruction errors using both Frobenius and L1 norms to better capture different methodological strengths.

- **Frobenius norm** emphasizes squared deviations, making it variance-sensitive and better suited for evaluating global structural retention.
- **L1 norm** penalizes deviations linearly and is more robust to outliers. It tends to favor methods that generate sparse approximations.

These differences led to divergent outcomes across methods. In the Combined Autism dataset, PCA and NMF outperformed under Frobenius, while ICA achieved nearly perfect L1 reconstruction using just two components—likely an overfit. Similar behavior was observed in the Twitch dataset.

In synthetic datasets, PCA showed consistency under both metrics, while ICA and NMF had more volatility under L1. For cPCA, performance was consistent across norms in real-world data, but Frobenius provided more meaningful comparisons when contrast was weak.

In summary, Frobenius norm provided stable, structure-aware error assessments, while L1 revealed sparsity and interpretability advantages. The choice of metric should align with the research objective.

4.2 Implications for Further Research and Future Work

This study reinforces that no single dimension reduction method excels in every context. Selection should be tailored to the dataset's structure and the goal—whether error minimization, interpretability, or contrastive insight.

For instance, a two-stage pipeline—PCA for initial compression followed by NMF for interpretable structure—may provide a practical balance. Similarly, ICA's independence may complement PCA's variance-based structure in exploratory data analysis.

cPCA's success on real-world datasets with structured contrast, such as Accent, suggests promise for domains like medical imaging, fraud detection, or social behavior analysis—fields where separating meaningful signal from structured noise is essential.

Building on this thesis, several future research directions emerge:

- **Automatic Parameter Selection:** Methods like cPCA depend heavily on parameters such as α. Auto-tuning via optimization could be helpful.
- **Visual Clarity Metrics:** Developing objective scores for visual separation in 2D and 3D plots would improve comparative evaluations.
- **Incorporating Manifold Learning:** Nonlinear methods (e.g., t-SNE, Isomap, LLE) could uncover structure in cases where non-linear methods flatten curvature [Saul and Roweis, 2000]. A comparative extension to the methods explored in this thesis would improve the work.

Appendix A

Code

A.1 My Code

The code for the application of methods on datasets is available at `https://github.com/cannecdet222/MathThesis`.

A.2 Imported Libraries

I used the Python machine learning library scikit-learn, specifically the sklearn.decomposition module. This module includes PCA, FastICA, and NMF functions.

For cPCA, I used the *contrastive* Python library which can be found at the following github page: `https://github.com/abidlabs/contrastive` .

Acknowledgements

Special thanks to Prof. Hardin, Prof. Garcia, and the Pomona College Mathematics and Statistics Department.

Bibliography

[Abid et al., 2018] Abid, A., Zhang, M. J., Bagaria, V. K., and Zou, J. (2018). Exploring patterns enriched in a dataset with contrastive principal component analysis. *Nature Communications*, 9(1):2134.

[Ang, 2019] Ang, A. (2019). Non-negative Matrix Factorization and Multiplicative Update.

[Baron, 2019] Baron, D. (2019). Machine Learning in Astronomy: a practical overview.

[Cover and Thomas, 1991] Cover, T. and Thomas, J. (1991). *Elements of Information Theory*. Wiley Series in Telecommunications and Signal Processing. Wiley.

[Follette, 2023] Follette, K. B. (2023). An introduction to high contrast differential imaging of exoplanets and disks. *Publications of the Astronomical Society of the Pacific*, 135(1051):093001.

[Gewers et al., 2021] Gewers, F. L., Ferreira, G. R., Arruda, H. F. D., Silva, F. N., Comin, C. H., Amancio, D. R., and Costa, L. D. F. (2021). Principal Component Analysis: A Natural Approach to Data Exploration. *ACM Computing Surveys*, 54(4):1–34.

[Hyvärinen et al., 2001] Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. Wiley-So Inc.

[Ivezić et al., 2020] Ivezić, , Connolly, A. J., VanderPlas, J. T., and Gray, A. (2020). Dimensionality and Its Reduction. In *Statistics, Data Mining, and Machine Learning in Astronomy*, A Practical Python Guide for the Analysis of Survey Data, Updated Edition, pages 281–310. Princeton University Press, rev - revised edition.

[Kokiopoulou et al., 2011] Kokiopoulou, E., Chen, J., and Saad, Y. (2011). Trace optimization and eigenproblems in dimension reduction methods. *Numerical Linear Algebra with Applications*, 18(3):565–602.

[Ledoit and Wolf, 2003] Ledoit, O. and Wolf, M. (2003). Honey, i shrunk the sample covariance matrix. Working Paper 691, UPF Economics and Business.

[Meyer-Baese and Schmid, 2014] Meyer-Baese, A. and Schmid, V. (2014). Chapter 8 - transformation and signal-separation neural networks. In Meyer-Baese, A. and Schmid, V., editors, *Pattern Recognition and Signal Analysis in Medical Imaging (Second Edition)*, pages 245–289. Academic Press, Oxford, second edition edition.

[Mika et al., 2020] Mika, D., Budzik, G., and Józwik, J. (2020). Single channel source separation with ica-based time-frequency decomposition. *Sensors*, 20(7).

[Montoya-Martínez et al., 2017] Montoya-Martínez, J., Cardoso, J.-F., and Gramfort, A. (2017). Caveats with stochastic gradient and maximum likelihood based ica for eeg. In *Latent Variable Analysis and Signal Separation: 13th International Conference, LVA/ICA 2017, Grenoble, France, February 2017*. Springer. `https://hal.science/hal-01451432`.

[Nebgen et al., 2021] Nebgen, B. T., Vangara, R., Hombrados-Herrera, M. A., Kuksova, S., and Alexandrov, B. S. (2021). A neural network for determination of latent dimensionality in non-negative matrix factorization. *Machine Learning: Science and Technology*, 2(2):025012.

[Nordhausen and Oja, 2018] Nordhausen, K. and Oja, H. (2018). Independent component analysis: A statistical perspective. *WIREs Computational Statistics*, 10(5):e1440.

[Oja and Hyvarinen, 2000] Oja, E. and Hyvarinen, A. (2000). Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430.

[Papoulis and Pillai, 2002] Papoulis, A. and Pillai, S. U. (2002). *Probability*. McGraw-hill.

[Raj, 2021] Raj, N. (2021). The Curse of Dimensionality and its Cure.

[Richard_V, 2022] Richard_V (2022). Twitchers statistics - high dimensional.

[Rodrigues and Moreira, 2021] Rodrigues, C. M. F. and Moreira, I. M. S. (2021). Dimensionality Reduction Data Sets. Publisher: OSF.

[Saul and Roweis, 2000] Saul, L. K. and Roweis, S. T. (2000). An introduction to locally linear embedding. *unpublished. Available at: http://www. cs. toronto. edu/˜ roweis/lle/publications. html*.

[Shahshahani and Landgrebe, 1994] Shahshahani, B. and Landgrebe, D. (1994). The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Transactions on Geoscience and Remote Sensing*, 32(5):1087–1095.

[Speaker, 2020] Speaker (2020). Speaker Accent Recognition.

[Thabtah, 2017a] Thabtah, F. (2017a). Autistic Spectrum Disorder Screening Data for Adolescent.

[Thabtah, 2017b] Thabtah, F. (2017b). Autistic Spectrum Disorder Screening Data for Adult.

[Thabtah, 2017c] Thabtah, F. (2017c). Autistic Spectrum Disorder Screening Data for Children.

[Tharwat, 2021] Tharwat, A. (2021). Independent component analysis: An introduction. *Applied Computing and Informatics*, 17(2):222–249.

[Wang, 2021] Wang, R. (2021). FastICA Algorithm.