Introduction to Principal Component Analysis and Indepedent Component Analysis

Tristan A. Hearn

Bioscience and Technology Branch, NASA Glenn Research Center

May 29, 2010

Table of Contents

Introduction
    Blind Seperation Problem

PCA

ICA

Cocktail Party Problem

Application to Images

References

Introduction

Princial Component Analysis (PCA) and Independent Component Analysis (ICA) are both types of transformations that may be performed on a given matrix: $A \in \mathbb{R}^{M \times N}$

- The basis vectors are computed to satisfy statistical properties associated with given data
- Orthogonal or biorthogonal

Introduction

Princial Component Analysis (PCA) and Independent Component Analysis (ICA) are both types of transformations that may be performed on a given matrix: $A \in \mathbb{R}^{M \times N}$

- The basis vectors are computed to satisfy statistical properties associated with given data
- Orthogonal or biorthogonal

Introduction

Princial Component Analysis (PCA) and Independent Component Analysis (ICA) are both types of transformations that may be performed on a given matrix: $A \in \mathbb{R}^{M \times N}$

- The basis vectors are computed to satisfy statistical properties associated with given data
- Orthogonal or biorthogonal

Introduction

PCA and ICA both seek a linear transformation on A such that the column (or row) vectors of A, represented in the new basis, maximize some measure related to **statistical independence**.

Definition

Two Random variables are said to be **independent** if and only if their joint density is the product of the marginal densities:

$$f_{X_1,X_2}(x,y) = f_{X_1}(x)\, f_{X_2}(y)$$

The Blind Seperation Problem (BSS)

Problem

Consider solving:

$$\begin{aligned} x_1(t) &= \overbrace{a_{11}s_1(t) + \ldots + a_{1N}s_N(t)}^{?} \\ &\vdots \\ x_N(t) &= \underbrace{a_{N1}s_1(t) + \ldots + a_{NN}s_N(t)}_{?} \end{aligned}$$

$$\Rightarrow \begin{bmatrix} x_1(t) \\ \vdots \\ x_N(t) \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix}}_{?} \begin{bmatrix} s_1(t) \\ \vdots \\ s_N(t) \end{bmatrix} \Rightarrow \underbrace{AS}_{?} = X$$

where $x_i, s_i \in \mathbb{R}^{M \times 1}$

The Blind Seperation Problem (BSS)

Problem

$$\begin{aligned} x_1(t) &= \overbrace{a_{11}s_1(t) + \ldots + a_{1N}s_N(t)}^{?} \\ &\vdots \\ x_N(t) &= \underbrace{a_{N1}s_1(t) + \ldots + a_{NN}s_N(t)}_{?} \end{aligned}$$

Consider solving:

$$\Rightarrow \begin{bmatrix} x_1(t) \\ \vdots \\ x_N(t) \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix}}_{?} \begin{bmatrix} s_1(t) \\ \vdots \\ s_N(t) \end{bmatrix} \Rightarrow \underbrace{AS}_{?} = X$$

where $x_i, s_i \in \mathbb{R}^{M \times 1}$

The Blind Seperation Problem (BSS)

Problem

$$\begin{aligned} x_1(t) &= \overbrace{a_{11}s_1(t) + \ldots + a_{1N}s_N(t)}^{?} \\ &\vdots \\ x_N(t) &= \underbrace{a_{N1}s_1(t) + \ldots + a_{NN}s_N(t)}_{?} \end{aligned}$$

Consider solving:

$$\Rightarrow \begin{bmatrix} x_1(t) \\ \vdots \\ x_N(t) \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} s_1(t) \\ \vdots \\ s_N(t) \end{bmatrix}}_{?} \Rightarrow \underbrace{AS}_{?} = X$$

where $x_i, s_i \in \mathbb{R}^{M \times 1}$

Principal Component Analysis

PCA aims to compute a 'more meaningful' basis in which to represent given data

- 'More meaningful': should reduce noise and redundancy in the data

Goal: to seperate sources, filter data, and reveal 'hidden' dynamics. PCA begins by assuming that the transformation to the new basis is linear:

$$\mathbf{PX} = \mathbf{Y} \Rightarrow \mathbf{y_i} = \begin{bmatrix} \mathbf{p_i x_i} \\ \vdots \\ \mathbf{p_i x_i} \end{bmatrix}$$

where $\mathbf{x_i}, \mathbf{y_i}$ represent columns of the source and transformed data matrices $\mathbf{X}, \mathbf{Y}$ and $\mathbf{p_i}$ represents a row of the transform matrix $\mathbf{P}$. So the rows of $\mathbf{P}$ form a new basis for the columns of $\mathbf{X}$; they are the **Principal Components** of the given data.

Principal Component Analysis

PCA aims to compute a 'more meaningful' basis in which to represent given data

- 'More meaningful': should reduce noise and redundancy in the data

Goal: to seperate sources, filter data, and reveal 'hidden' dynamics. PCA begins by assuming that the transformation to the new basis is linear:

$$\mathbf{PX} = \mathbf{Y} \Rightarrow \mathbf{y_i} = \begin{bmatrix} \mathbf{p_i x_i} \\ \vdots \\ \mathbf{p_i x_i} \end{bmatrix}$$

where $\mathbf{x_i}, \mathbf{y_i}$ represent columns of the source and transformed data matrices $\mathbf{X}, \mathbf{Y}$ and $\mathbf{p_i}$ represents a row of the transform matrix $\mathbf{P}$. So the rows of $\mathbf{P}$ form a new basis for the columns of $\mathbf{X}$; they are the **Principal Components** of the given data.

Principal Component Analysis

PCA aims to compute a 'more meaningful' basis in which to represent given data

- 'More meaningful': should reduce noise and redundancy in the data

Goal: to seperate sources, filter data, and reveal 'hidden' dynamics.

PCA begins by assuming that the transformation to the new basis is linear:

$$\mathbf{PX} = \mathbf{Y} \Rightarrow \mathbf{y_i} = \begin{bmatrix} \mathbf{p_i x_i} \\ \vdots \\ \mathbf{p_i x_i} \end{bmatrix}$$

where $\mathbf{x_i}, \mathbf{y_i}$ represent columns of the source and transformed data matrices $\mathbf{X}, \mathbf{Y}$ and $\mathbf{p_i}$ represents a row of the transform matrix $\mathbf{P}$. So the rows of $\mathbf{P}$ form a new basis for the columns of $\mathbf{X}$; they are the **Principal Components** of the given data.

Principal Component Analysis

PCA aims to compute a 'more meaningful' basis in which to represent given data

- 'More meaningful': should reduce noise and redundancy in the data

Goal: to seperate sources, filter data, and reveal 'hidden' dynamics. PCA begins by assuming that the transformation to the new basis is linear:

$$\mathbf{PX} = \mathbf{Y} \Rightarrow \mathbf{y_i} = \begin{bmatrix} \mathbf{p_i x_i} \\ \vdots \\ \mathbf{p_i x_i} \end{bmatrix}$$

where $\mathbf{x_i}, \mathbf{y_i}$ represent columns of the source and transformed data matrices $\mathbf{X}, \mathbf{Y}$ and $\mathbf{p_i}$ represents a row of the transform matrix $\mathbf{P}$. So the rows of $\mathbf{P}$ form a new basis for the columns of $\mathbf{X}$; they are the **Principal Components** of the given data.

Principal Component Analysis

PCA aims to compute a 'more meaningful' basis in which to represent given data

- 'More meaningful': should reduce noise and redundancy in the data

Goal: to seperate sources, filter data, and reveal 'hidden' dynamics. PCA begins by assuming that the transformation to the new basis is linear:

$$\mathbf{PX} = \mathbf{Y} \Rightarrow \mathbf{y_i} = \begin{bmatrix} \mathbf{p_i x_i} \\ \vdots \\ \mathbf{p_i x_i} \end{bmatrix}$$

where $\mathbf{x_i}, \mathbf{y_i}$ represent columns of the source and transformed data matrices $\mathbf{X}, \mathbf{Y}$ and $\mathbf{p_i}$ represents a row of the transform matrix $\mathbf{P}$. So the rows of $\mathbf{P}$ form a new basis for the columns of $\mathbf{X}$; they are the **Principal Components** of the given data.

Principal Component Analysis

PCA aims to compute a 'more meaningful' basis in which to represent given data

- 'More meaningful': should reduce noise and redundancy in the data

Goal: to seperate sources, filter data, and reveal 'hidden' dynamics. PCA begins by assuming that the transformation to the new basis is linear:

$$\mathbf{PX} = \mathbf{Y} \Rightarrow \mathbf{y_i} = \begin{bmatrix} \mathbf{p_i x_i} \\ \vdots \\ \mathbf{p_i x_i} \end{bmatrix}$$

where $\mathbf{x_i}, \mathbf{y_i}$ represent columns of the source and transformed data matrices $\mathbf{X}, \mathbf{Y}$ and $\mathbf{p_i}$ represents a row of the transform matrix $\mathbf{P}$. So the rows of $\mathbf{P}$ form a new basis for the columns of $\mathbf{X}$; they are the **Principal Components** of the given data.

Principal Component Analysis

To minimize redundancy in the new basis, the sampled data should be **un-correlated** in the new basis.

Definition

Two random samples $\mathbf{x}, \mathbf{y}$ are **un-correlated** if their **sample covariance** is 0:

$$\sigma^2_{\mathbf{x},\mathbf{y}} = \frac{1}{n-1}(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{y} - \bar{\mathbf{y}})^T = 0$$

and $\sigma^2_{\mathbf{x},\mathbf{x}} > 0$ is simply the variance of $\mathbf{x}$.

Principal Component Analysis

To minimize redundancy in the new basis, the sampled data should be **un-correlated** in the new basis.

Definition

Two random samples $\mathbf{x}, \mathbf{y}$ are **un-correlated** if their **sample covariance** is 0:

$$\sigma^2_{\mathbf{x},\mathbf{y}} = \frac{1}{n-1}\left(\mathbf{x} - \bar{\mathbf{x}}\right)\left(\mathbf{y} - \bar{\mathbf{y}}\right)^T = 0$$

and $\sigma^2_{\mathbf{x},\mathbf{x}} > 0$ is simply the variance of $\mathbf{x}$.

Principal Component Analysis

To minimize redundancy in the new basis, the sampled data should be **un-correlated** in the new basis.

Definition

n random samples $y_1, y_2, \ldots$ are **un-correlated** if their **sample covariance matrix** is **diagonal**:

$$\mathbf{S_Y} = \frac{1}{n-1}\left(\mathbf{Y} - \bar{\mathbf{Y}}\mathbf{1}\right)\left(\mathbf{Y} - \bar{\mathbf{Y}}\mathbf{1}\right)^T = \begin{bmatrix} a_1 & & 0 \\ & \ddots & \\ 0 & & a_n \end{bmatrix}$$

- $\mathbf{S_Y}$ is always a square, symmetric matrix
- Diagonal elements are the individual **variances** of $\mathbf{y_1}, \mathbf{y_2}, \ldots$
- Off-diagonal elements are the **covariances** of $\mathbf{y_1}, \mathbf{y_2}, \ldots$
- $\mathbf{S_Y}$ quantifies the correlation between all possible pairings of $\{\mathbf{y_1}, \ldots, \mathbf{y_n}\}$

Principal Component Analysis

To minimize redundancy in the new basis, the sampled data should be **un-correlated** in the new basis.

Definition

n random samples $y_1, y_2, \ldots$ are **un-correlated** if their **sample covariance matrix** is **diagonal**:

$$\mathbf{S_Y} = \frac{1}{n-1}\left(\mathbf{Y} - \bar{\mathbf{Y}}\mathbf{1}\right)\left(\mathbf{Y} - \bar{\mathbf{Y}}\mathbf{1}\right)^T = \begin{bmatrix} a_1 & & 0 \\ & \ddots & \\ 0 & & a_n \end{bmatrix}$$

- $\mathbf{S_Y}$ is always a square, symmetric matrix
- Diagonal elements are the individual **variances** of $\mathbf{y_1}, \mathbf{y_2}, \ldots$
- Off-diagonal elements are the **covariances** of $\mathbf{y_1}, \mathbf{y_2}, \ldots$
- $\mathbf{S_Y}$ quantifies the correlation between all possible pairings of $\{\mathbf{y_1}, \ldots, \mathbf{y_n}\}$

Principal Component Analysis

To minimize redundancy in the new basis, the sampled data should be **un-correlated** in the new basis.

Definition

n random samples $y_1, y_2, \ldots$ are **un-correlated** if their **sample covariance matrix** is **diagonal**:

$$\mathbf{S_Y} = \frac{1}{n-1}\left(\mathbf{Y} - \bar{\mathbf{Y}}\mathbf{1}\right)\left(\mathbf{Y} - \bar{\mathbf{Y}}\mathbf{1}\right)^T = \begin{bmatrix} a_1 & & 0 \\ & \ddots & \\ 0 & & a_n \end{bmatrix}$$

- $\mathbf{S_Y}$ is always a square, symmetric matrix
- Diagonal elements are the individual **variances** of $\mathbf{y_1}, \mathbf{y_2}, \ldots$
- Off-diagonal elements are the **covariances** of $\mathbf{y_1}, \mathbf{y_2}, \ldots$
- $\mathbf{S_Y}$ quantifies the correlation between all possible pairings of $\{\mathbf{y_1}, \ldots, \mathbf{y_n}\}$

Principal Component Analysis

To minimize redundancy in the new basis, the sampled data should be **un-correlated** in the new basis.

Definition

n random samples $y_1, y_2, \ldots$ are **un-correlated** if their **sample covariance matrix** is **diagonal**:

$$\mathbf{S_Y} = \frac{1}{n-1}\left(\mathbf{Y} - \bar{\mathbf{Y}}\mathbf{1}\right)\left(\mathbf{Y} - \bar{\mathbf{Y}}\mathbf{1}\right)^T = \begin{bmatrix} a_1 & & 0 \\ & \ddots & \\ 0 & & a_n \end{bmatrix}$$

- $\mathbf{S_Y}$ is always a square, symmetric matrix
- Diagonal elements are the individual **variances** of $\mathbf{y_1}, \mathbf{y_2}, \ldots$
- Off-diagonal elements are the **covariances** of $\mathbf{y_1}, \mathbf{y_2}, \ldots$
- $\mathbf{S_Y}$ quantifies the correlation between all possible pairings of $\{\mathbf{y_1}, \ldots, \mathbf{y_n}\}$

Principal Component Analysis

To minimize redundancy in the new basis, the sampled data should be **un-correlated** in the new basis.

Definition

n random samples $y_1, y_2, \ldots$ are **un-correlated** if their **sample covariance matrix** is **diagonal**:

$$\mathbf{S_Y} = \frac{1}{n-1}\left(\mathbf{Y} - \bar{\mathbf{Y}}\mathbf{1}\right)\left(\mathbf{Y} - \bar{\mathbf{Y}}\mathbf{1}\right)^T = \begin{bmatrix} a_1 & & 0 \\ & \ddots & \\ 0 & & a_n \end{bmatrix}$$

- $\mathbf{S_Y}$ is always a square, symmetric matrix
- Diagonal elements are the individual **variances** of $\mathbf{y_1}, \mathbf{y_2}, \ldots$
- Off-diagonal elements are the **covariances** of $\mathbf{y_1}, \mathbf{y_2}, \ldots$
- $\mathbf{S_Y}$ quantifies the correlation between all possible pairings of $\{\mathbf{y_1}, \ldots, \mathbf{y_n}\}$

Principal Component Analysis

- So to remove redundancy, we must find new basis vectors (Principal Components) such that the covariance matrix of the transformed data is diagonal.
- PCA also assumes that the basis vectors are **orthogonal**, to simplify the computation of the new basis.

Definition

Two vectors $\mathbf{x}, \mathbf{y}$ are said to be **orthogonal** if their dot product is zero:

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n} x_i y_i = 0$$

Principal Component Analysis

- So to remove redundancy, we must find new basis vectors (Principal Components) such that the covariance matrix of the transformed data is diagonal.
- PCA also assumes that the basis vectors are **orthogonal**, to simplify the computation of the new basis.

Definition

Two vectors $\mathbf{x}, \mathbf{y}$ are said to be **orthogonal** if their dot product is zero:

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n} x_i y_i = 0$$

Principal Component Analysis

Summary of assumptions:

- Linearity of the transformation.
- The sample mean and sample variance are **sufficient statistics** for the underlying seperation problem.
- Large variances in $\mathbf{X}$ correspond to important dynamics in the underlying system.
- The principal components are orthogonal.

Definition

A function $T(\mathbf{x})$ is said to be a **sufficient statistic** for the random variable $\mathbf{x}$ if the conditional probability distribution of $\mathbf{x}$, given $T(\mathbf{x})$, is not a function of any unknown distribution parameters: $P(X = x|\, T(x), \theta \in \Omega) = P(X = x|\, T(x))$

Principal Component Analysis

Summary of assumptions:

- Linearity of the transformation.
- The sample mean and sample variance are **sufficient statistics** for the underlying seperation problem.
- Large variances in $\mathbf{X}$ correspond to important dynamics in the underlying system.
- The principal components are orthogonal.

Definition

A function $T(\mathbf{x})$ is said to be a **sufficient statistic** for the random variable $\mathbf{x}$ if the conditional probability distribution of $\mathbf{x}$, given $T(\mathbf{x})$, is not a function of any unknown distribution parameters: $P(X = x|\, T(x), \theta \in \Omega) = P(X = x|\, T(x))$

Principal Component Analysis

Summary of assumptions:

- Linearity of the transformation.
- The sample mean and sample variance are **sufficient statistics** for the underlying seperation problem.
- Large variances in $\mathbf{X}$ correspond to important dynamics in the underlying system.
- The principal components are orthogonal.

Definition

A function $T(\mathbf{x})$ is said to be a **sufficient statistic** for the random variable $\mathbf{x}$ if the conditional probability distribution of $\mathbf{x}$, given $T(\mathbf{x})$, is not a function of any unknown distribution parameters: $P(X = x|\, T(x), \theta \in \Omega) = P(X = x|\, T(x))$

Principal Component Analysis

Summary of assumptions:

- Linearity of the transformation.
- The sample mean and sample variance are **sufficient statistics** for the underlying seperation problem.
- Large variances in $\mathbf{X}$ correspond to important dynamics in the underlying system.
- The principal components are orthogonal.

Definition

A function $T(\mathbf{x})$ is said to be a **sufficient statistic** for the random variable $\mathbf{x}$ if the conditional probability distribution of $\mathbf{x}$, given $T(\mathbf{x})$, is not a function of any unknown distribution parameters: $P(X = x|\, T(x), \theta \in \Omega) = P(X = x|\, T(x))$

Principal Component Analysis

Summary of assumptions:

- Linearity of the transformation.
- The sample mean and sample variance are **sufficient statistics** for the underlying seperation problem.
- Large variances in $\mathbf{X}$ correspond to important dynamics in the underlying system.
- The principal components are orthogonal.

Definition

A function $T(\mathbf{x})$ is said to be a **sufficient statistic** for the random variable $\mathbf{x}$ if the conditional probability distribution of $\mathbf{x}$, given $T(\mathbf{x})$, is not a function of any unknown distribution parameters: $P(X = x|\, T(x), \theta \in \Omega) = P(X = x|\, T(x))$

Principal Component Analysis

Solving for the PCs:

- WLOG, assume $\bar{\mathbf{X}}$ is normalized with zero mean.
- Seek an orthonormal matrix $\mathbf{P}$ (where $\mathbf{Y} = \mathbf{PX}$) such that $\mathbf{S_Y} = \frac{1}{\mathbf{n-1}}\mathbf{YY^T}$ is diagonalized. The rows of $\mathbf{P}$ will be the **principal components** of $\mathbf{X}$.

So:

$$S_Y = \frac{1}{n-1} Y Y^T = P \underbrace{\left(\frac{1}{n-1} X X^T \right)}_{\text{symmetric!}} P^T$$

Principal Component Analysis

Solving for the PCs:

- WLOG, assume $\bar{\mathbf{X}}$ is normalized with zero mean.
- Seek an orthonormal matrix $\mathbf{P}$ (where $\mathbf{Y} = \mathbf{PX}$) such that $\mathbf{S_Y} = \frac{\mathbf{1}}{\mathbf{n-1}}\mathbf{YY^T}$ is diagonalized. The rows of $\mathbf{P}$ will be the **principal components** of $\mathbf{X}$.

So:

$$S_Y = \frac{1}{n-1} Y Y^T = P \underbrace{\left(\frac{1}{n-1} X X^T \right)}_{\text{symmetric!}} P^T$$

Principal Component Analysis

Solving for the PCs:

- WLOG, assume $\bar{\mathbf{X}}$ is normalized with zero mean.
- Seek an orthonormal matrix $\mathbf{P}$ (where $\mathbf{Y} = \mathbf{PX}$) such that $\mathbf{S_Y} = \frac{\mathbf{1}}{\mathbf{n-1}}\mathbf{YY^T}$ is diagonalized. The rows of $\mathbf{P}$ will be the **principal components** of $\mathbf{X}$.

So:

$$S_Y = \frac{1}{n-1} Y Y^T = P \underbrace{\left(\frac{1}{n-1} X X^T \right)}_{\text{symmetric!}} P^T$$

Principal Component Analysis

- Any real, symmetric matrix is diagonalized by an orthonormal matrix of its eigenvectors.
- Therefore, normalizing the data matrix $\mathbf{X}$ and computing the eigenvectors of $\frac{1}{n-1}\mathbf{X}\mathbf{X}^{\mathsf{T}} = \mathbf{S}_{\mathbf{X}}$ with give the principal components!
- Best approach: the **singular value decomposition**

Principal Component Analysis

- Any real, symmetric matrix is diagonalized by an orthonormal matrix of its eigenvectors.
- Therefore, normalizing the data matrix $\mathbf{X}$ and computing the eigenvectors of $\frac{1}{n-1}\mathbf{X}\mathbf{X}^{\mathbf{T}} = \mathbf{S}_{\mathbf{X}}$ with give the principal components!
- Best approach: the **singular value decomposition**

Principal Component Analysis

- Any real, symmetric matrix is diagonalized by an orthonormal matrix of its eigenvectors.
- Therefore, normalizing the data matrix $\mathbf{X}$ and computing the eigenvectors of $\frac{1}{n-1}\mathbf{X}\mathbf{X}^{\mathbf{T}} = \mathbf{S}_{\mathbf{X}}$ with give the principal components!
- Best approach: the **singular value decomposition**

Principal Component Analysis

Definition

The **singular value decomposition** of a real $m \times xn$ matrix $\mathbf{X}$ is given by:

$$\mathbf{X} = \mathbf{U\Sigma V^T}$$

where $\mathbf{U}$ is an $m \times m$ matrix containing the eigenvectors of XX^T, $\mathbf{V}$ is an $n \times n$ matrix containing the eigenvectors of X^TX, and $\mathbf{\Sigma}$ is an $m \times n$ matrix with the square roots of the eigenvalues of $\mathbf{XX^T}$ along its main diagonal.

- The singular values σ (elements of Σ) or ordered from greatest to least, and each correspond to a basis vector in U and V.
- Dimension reduction: choose a minimum acceptable value for the σs; consider as the principal components only the vectors corresponding to σs larger than the chosen threshold.

Principal Component Analysis

Definition

The **singular value decomposition** of a real $m \times xn$ matrix $\mathbf{X}$ is given by:

$$\mathbf{X} = \mathbf{U\Sigma V^T}$$

where $\mathbf{U}$ is an $m \times m$ matrix containing the eigenvectors of XX^T, $\mathbf{V}$ is an $n \times n$ matrix containing the eigenvectors of X^TX, and $\mathbf{\Sigma}$ is an $m \times n$ matrix with the square roots of the eigenvalues of $\mathbf{XX^T}$ along its main diagonal.

- The singular values σ (elements of Σ) or ordered from greatest to least, and each correspond to a basis vector in U and V.
- Dimension reduction: choose a minimum acceptable value for the σs; consider as the principal components only the vectors corresponding to σs larger than the chosen threshold.

Principal Component Analysis

Definition

The **singular value decomposition** of a real $m \times xn$ matrix $\mathbf{X}$ is given by:

$$\mathbf{X} = \mathbf{U\Sigma V^T}$$

where $\mathbf{U}$ is an $m \times m$ matrix containing the eigenvectors of XX^T, $\mathbf{V}$ is an $n \times n$ matrix containing the eigenvectors of X^TX, and $\mathbf{\Sigma}$ is an $m \times n$ matrix with the square roots of the eigenvalues of $\mathbf{XX^T}$ along its main diagonal.

- The singular values σ (elements of Σ) or ordered from greatest to least, and each correspond to a basis vector in U and V.
- Dimension reduction: choose a minimum acceptable value for the σs; consider as the principal components only the vectors corresponding to σs larger than the chosen threshold.

Principal Component Analysis

- The SVD is a very important matrix factorization with a wide variety of applications.

For PCA, note that:

$$\mathbf{Z} = \frac{1}{\sqrt{n-1}}\mathbf{X}^\mathsf{T} \Rightarrow \mathbf{Z}^\mathsf{T}\mathbf{Z} = \left(\frac{1}{\sqrt{n-1}}\mathbf{X}^\mathsf{T}\right)^\mathsf{T}\left(\frac{1}{\sqrt{n-1}}\mathbf{X}^\mathsf{T}\right)$$

$$= \frac{1}{n-1}\left(\mathbf{X}^\mathsf{T}\right)^T\mathbf{X}^\mathsf{T} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^\mathsf{T} = \mathbf{S_X}$$

- So the matrix $\mathbf{V}$ given by the SVD of $\mathbf{Z}$ will give the eigenvectors of S_X, which are the principal components! Therefore $\mathbf{P} = \mathbf{V}^\mathsf{T}$.
- Once $\mathbf{P}$ is found, the data can be transformed: $\mathbf{Y} = \mathbf{PX}$

Principal Component Analysis

- The SVD is a very important matrix factorization with a wide variety of applications.

For PCA, note that:

$$\mathbf{Z} = \frac{1}{\sqrt{n-1}}\mathbf{X}^{\mathsf{T}} \Rightarrow \mathbf{Z}^{\mathsf{T}}\mathbf{Z} = \left(\frac{1}{\sqrt{n-1}}\mathbf{X}^{\mathsf{T}}\right)^{\mathsf{T}}\left(\frac{1}{\sqrt{n-1}}\mathbf{X}^{\mathsf{T}}\right)$$

$$= \frac{1}{n-1}\left(\mathbf{X}^{\mathsf{T}}\right)^{T}\mathbf{X}^{\mathsf{T}} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^{\mathsf{T}} = \mathbf{S}_{\mathbf{X}}$$

- So the matrix $\mathbf{V}$ given by the SVD of $\mathbf{Z}$ will give the eigenvectors of S_X, which are the principal components! Therefore $\mathbf{P} = \mathbf{V}^{\mathsf{T}}$.
- Once $\mathbf{P}$ is found, the data can be transformed: $\mathbf{Y} = \mathbf{P}\mathbf{X}$

Principal Component Analysis

- The SVD is a very important matrix factorization with a wide variety of applications.

For PCA, note that:

$$\mathbf{Z} = \frac{1}{\sqrt{n-1}}\mathbf{X}^\mathbf{T} \Rightarrow \mathbf{Z}^\mathbf{T}\mathbf{Z} = \left(\frac{1}{\sqrt{n-1}}\mathbf{X}^\mathbf{T}\right)^\mathbf{T}\left(\frac{1}{\sqrt{n-1}}\mathbf{X}^\mathbf{T}\right)$$

$$= \frac{1}{n-1}\left(\mathbf{X}^\mathbf{T}\right)^T\mathbf{X}^\mathbf{T} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^\mathbf{T} = \mathbf{S_X}$$

- So the matrix $\mathbf{V}$ given by the SVD of $\mathbf{Z}$ will give the eigenvectors of S_X, which are the principal components! Therefore $\mathbf{P} = \mathbf{V}^\mathbf{T}$.
- Once $\mathbf{P}$ is found, the data can be transformed: $\mathbf{Y} = \mathbf{PX}$

Principal Component Analysis

- The SVD is a very important matrix factorization with a wide variety of applications.

For PCA, note that:

$$\mathbf{Z} = \frac{1}{\sqrt{n-1}}\mathbf{X}^{\mathbf{T}} \Rightarrow \mathbf{Z}^{\mathbf{T}}\mathbf{Z} = \left(\frac{1}{\sqrt{n-1}}\mathbf{X}^{\mathbf{T}}\right)^{\mathbf{T}}\left(\frac{1}{\sqrt{n-1}}\mathbf{X}^{\mathbf{T}}\right)$$

$$= \frac{1}{n-1}\left(\mathbf{X}^{\mathbf{T}}\right)^{T}\mathbf{X}^{\mathbf{T}} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^{\mathbf{T}} = \mathbf{S_X}$$

- So the matrix $\mathbf{V}$ given by the SVD of $\mathbf{Z}$ will give the eigenvectors of S_X, which are the principal components! Therefore $\mathbf{P} = \mathbf{V}^{\mathbf{T}}$.
- Once $\mathbf{P}$ is found, the data can be transformed: $\mathbf{Y} = \mathbf{PX}$

Principal Component Analysis

- The SVD is a very important matrix factorization with a wide variety of applications.

For PCA, note that:

$$\mathbf{Z} = \frac{1}{\sqrt{\mathbf{n}-\mathbf{1}}}\mathbf{X}^{\mathbf{T}} \Rightarrow \mathbf{Z}^{\mathbf{T}}\mathbf{Z} = \left(\frac{1}{\sqrt{\mathbf{n}-\mathbf{1}}}\mathbf{X}^{\mathbf{T}}\right)^{\mathbf{T}}\left(\frac{1}{\sqrt{\mathbf{n}-\mathbf{1}}}\mathbf{X}^{\mathbf{T}}\right)$$

$$= \frac{1}{n-1}\left(\mathbf{X}^{\mathbf{T}}\right)^{T}\mathbf{X}^{\mathbf{T}} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^{\mathbf{T}} = \mathbf{S_X}$$

- So the matrix $\mathbf{V}$ given by the SVD of $\mathbf{Z}$ will give the eigenvectors of S_X, which are the principal components! Therefore $\mathbf{P} = \mathbf{V}^{\mathbf{T}}$.
- Once $\mathbf{P}$ is found, the data can be transformed: $\mathbf{Y} = \mathbf{PX}$

Principal Component Analysis

2D Example

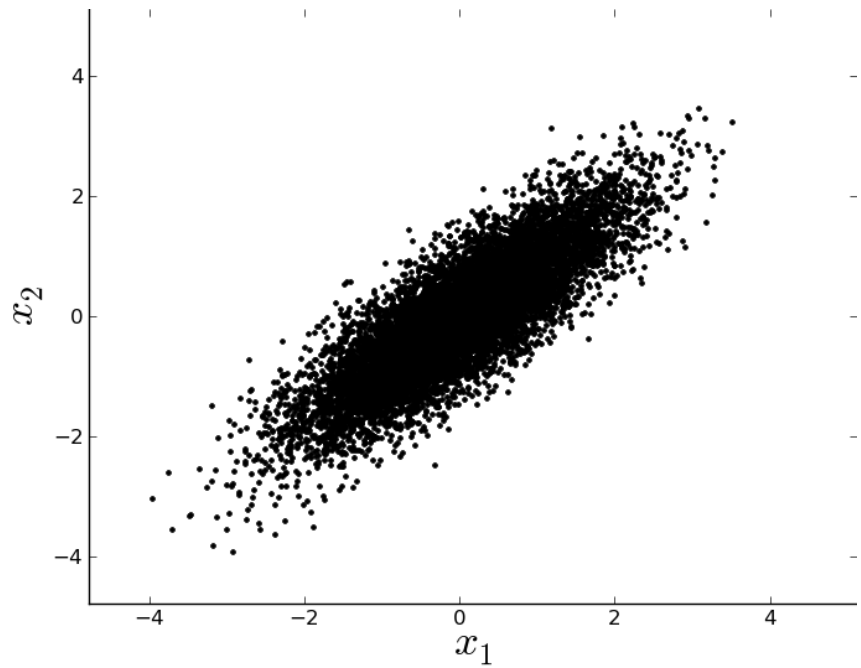
Let $x_1 = [x_{1,1}, \ldots, x_{1,1000}], x_2 = [x_{2,1}, \ldots, x_{2,1000}]$ be random variables such that $x_{1,i} \overset{i.i.d.}{\sim} P_1$ and $x_{2,j} \overset{i.i.d.}{\sim} P_2 \ \forall i, j$ with the two distributions P_1, P_2 unknown.

- So, x_1, x_2 are two different measurement types (sensors, etc) each containing 1000 measurements.

Principal Component Analysis

2D Example

- We can plot x_1 vs x_2 data to show that they are strongly correlated:



Principal Component Analysis

The SVD of $X = [x_1, x_2]^T$ is computed to be:

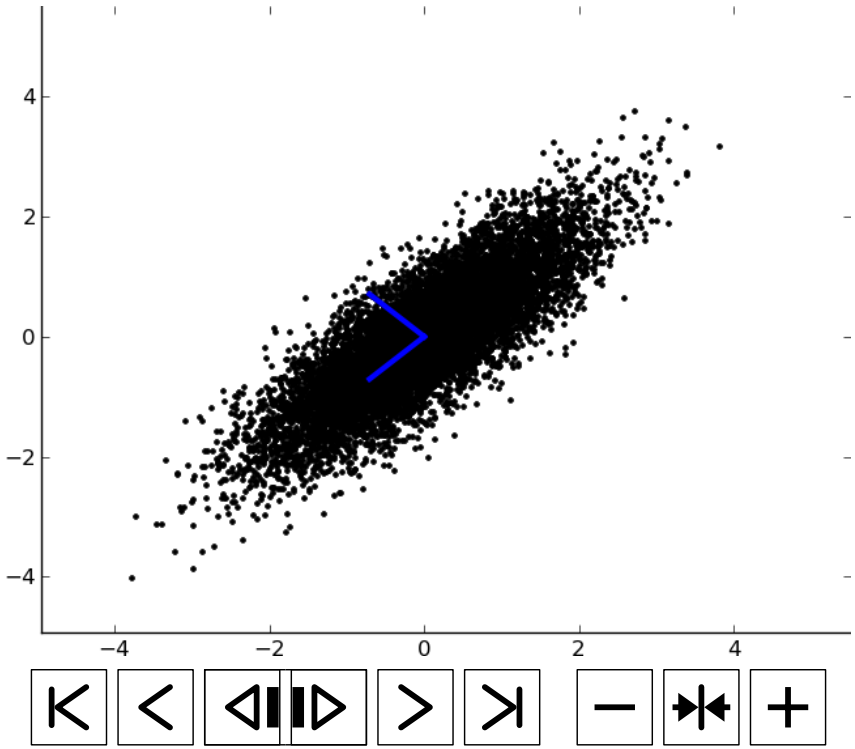
$$U = \begin{bmatrix} 3.77 \times 10^{-2} & \cdots & -3.61 \times 10^{-2} \\ \vdots & \ddots & \vdots \\ -4.57 \times 10^{-2} & \cdots & 0.97 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 142.85 & 0 \\ 0 & 43.61 \end{bmatrix}$$

$$V^T = \begin{bmatrix} 0.63 & 0.77 \\ -0.77 & 0.63 \end{bmatrix}$$

Principal Component Analysis

2D Example



PCA provides a transformation into a new basis in which the data becomes uncorrelated.

Principal Component Analysis

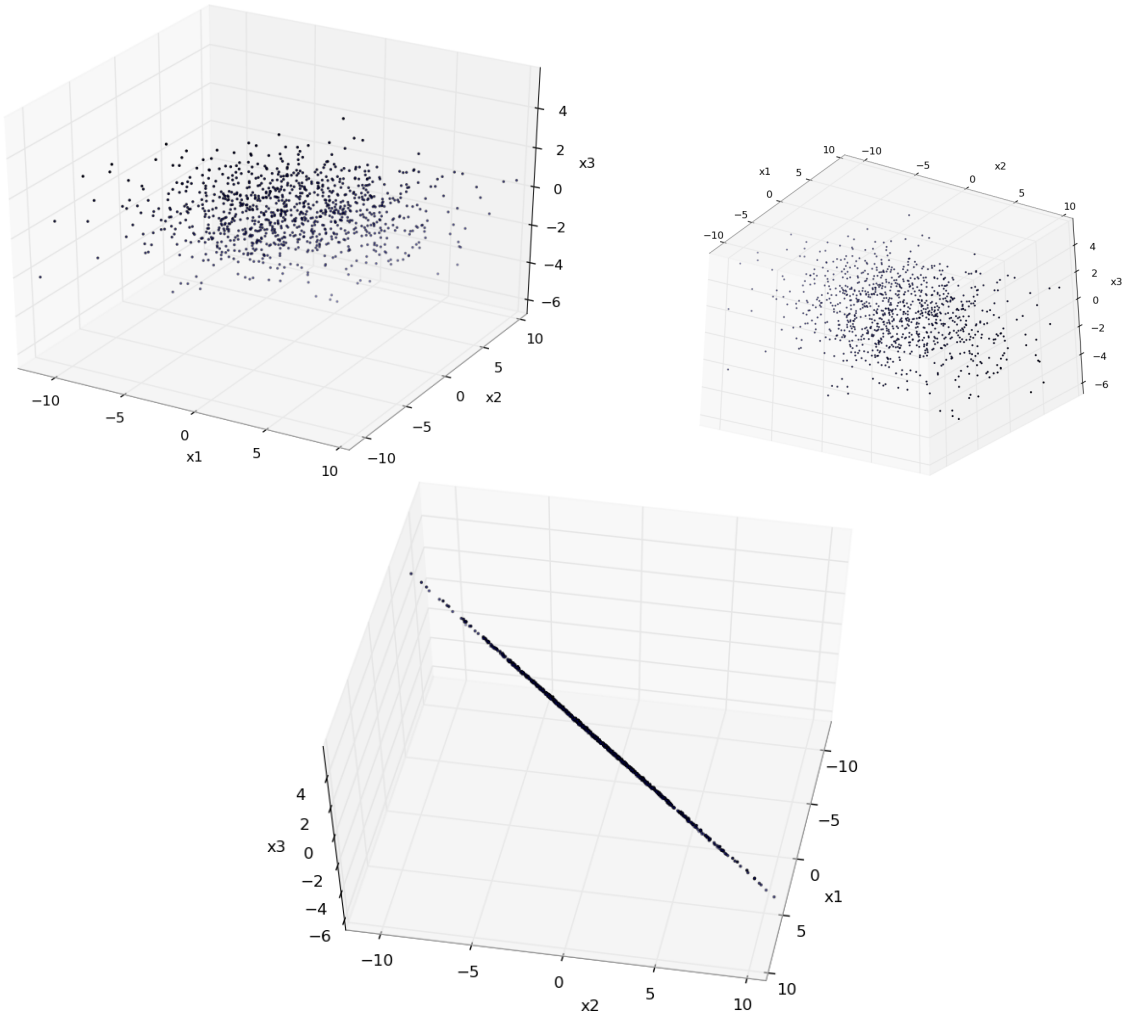
3D Example

Let us introduct a new component, so that the data is 3 dimensional: $x_3 = x_1 - x_2$

- $\Rightarrow x_3$ provides no new information about the underlying system!
- Thanks to the SVD, the PCA provides a mechanism for detecting this and removing the redundant dimension.

Principal Component Analysis

3D Example



Principal Component Analysis

The SVD of $X = [x_1, x_2]^T$ is computed to be:

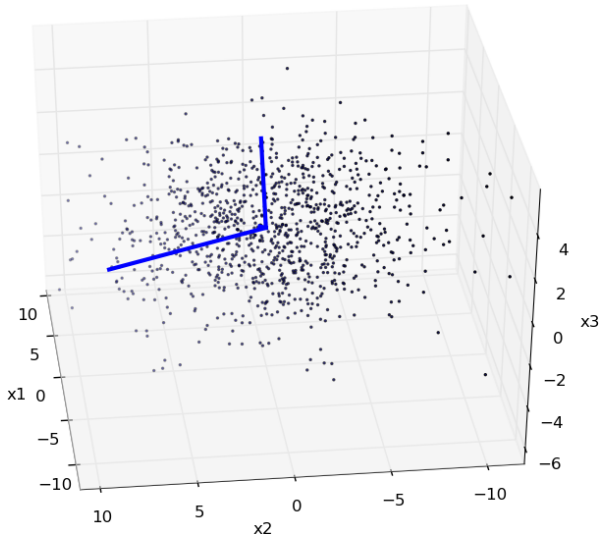
$$U = \begin{bmatrix} 3.77 \times 10^{-2} & \cdots & -3.61 \times 10^{-2} \\ \vdots & \ddots & \vdots \\ -4.57 \times 10^{-2} & \cdots & 0.97 \end{bmatrix}$$

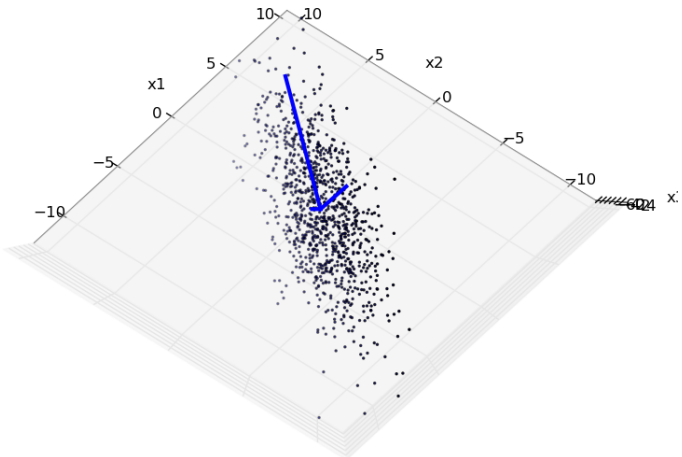
$$\Sigma = \begin{bmatrix} 142.97 & 0 & 0 \\ 0 & 73.35 & 0 \\ 0 & 0 & 4.29 \times 10^{-14} \end{bmatrix}$$

$$V^T = \begin{bmatrix} 0.61 & 0.77 & -0.16 \\ 0.54 & -0.25 & 0.80 \\ -0.577 & 0.577 & 0.577 \end{bmatrix}$$

Principal Component Analysis

3D Example





Since the singular value corresponding to third PC is small, the contribution of that axis in the new basis is minimal

$\Rightarrow$ Projection onto the first two PCs is sufficient to charectorize the data!

Principal Component Analysis

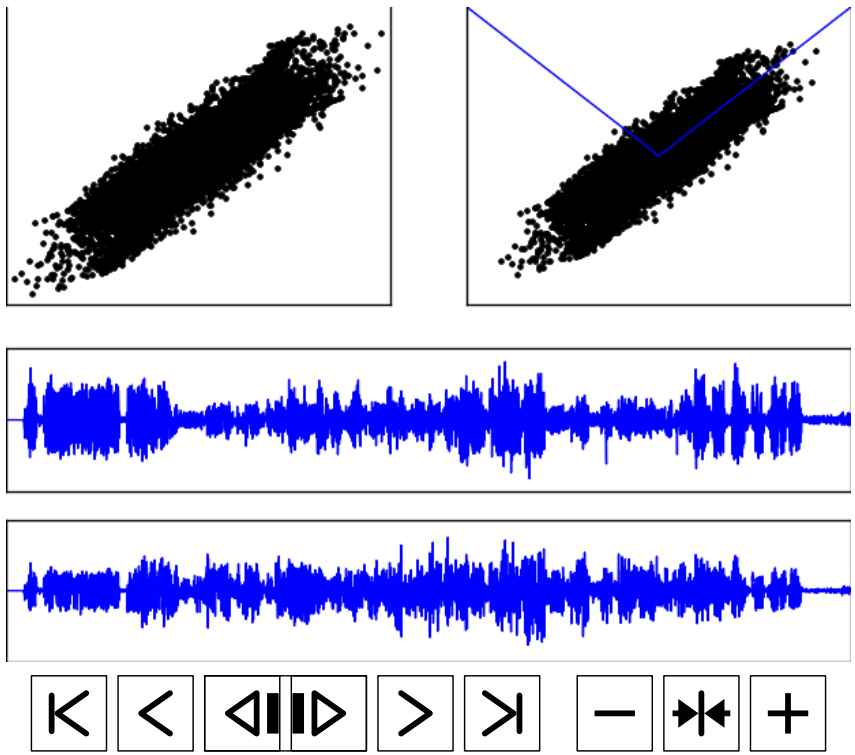
2-Source Audio Example

▷

▷

Principal Component Analysis

2-Source Audio Example



▷

▷

- In the previous two examples, PCA was not successfull in completely seperating the mixed signals.
- What is needed: A transformation driven by a stronger measure of independence.

Independent Component Analysis

ICA, like PCA, aims to compute a 'more meaningful' basis in which to represent given data.

- 'More meaningful': should reduce noise and redundancy in the data

Goal: to seperate sources, filter data, and reveal 'hidden' dynamics. ICA also begins by assuming that the transformation to the new basis is linear:

$$\mathbf{WX} = \mathbf{Y} \Rightarrow \mathbf{y_i} = \begin{bmatrix} \mathbf{w_i x_i} \\ \vdots \\ \mathbf{w_i x_i} \end{bmatrix}$$

where $\mathbf{x_i}, \mathbf{y_i}$ represent columns of the source and transformed data matrices $\mathbf{X}, \mathbf{Y}$ and $\mathbf{w_i}$ represents a row of the transform matrix $\mathbf{W}$. So the rows of $\mathbf{W}$ form a new basis for the columns of $\mathbf{X}$; they are the **Independent Components** of the given data.

Independent Component Analysis

ICA, like PCA, aims to compute a 'more meaningful' basis in which to represent given data.

- 'More meaningful': should reduce noise and redundancy in the data

Goal: to seperate sources, filter data, and reveal 'hidden' dynamics. ICA also begins by assuming that the transformation to the new basis is linear:

$$\mathbf{WX} = \mathbf{Y} \Rightarrow \mathbf{y_i} = \begin{bmatrix} \mathbf{w_i x_i} \\ \vdots \\ \mathbf{w_i x_i} \end{bmatrix}$$

where $\mathbf{x_i}, \mathbf{y_i}$ represent columns of the source and transformed data matrices $\mathbf{X}, \mathbf{Y}$ and $\mathbf{w_i}$ represents a row of the transform matrix $\mathbf{W}$. So the rows of $\mathbf{W}$ form a new basis for the columns of $\mathbf{X}$; they are the **Independent Components** of the given data.

Independent Component Analysis

ICA, like PCA, aims to compute a 'more meaningful' basis in which to represent given data.

- 'More meaningful': should reduce noise and redundancy in the data

Goal: to seperate sources, filter data, and reveal 'hidden' dynamics.

ICA also begins by assuming that the transformation to the new basis is linear:

$$\mathbf{WX} = \mathbf{Y} \Rightarrow \mathbf{y_i} = \begin{bmatrix} \mathbf{w_i x_i} \\ \vdots \\ \mathbf{w_i x_i} \end{bmatrix}$$

where $\mathbf{x_i}, \mathbf{y_i}$ represent columns of the source and transformed data matrices $\mathbf{X}, \mathbf{Y}$ and $\mathbf{w_i}$ represents a row of the transform matrix $\mathbf{W}$. So the rows of $\mathbf{W}$ form a new basis for the columns of $\mathbf{X}$; they are the **Independent Components** of the given data.

Independent Component Analysis

ICA, like PCA, aims to compute a 'more meaningful' basis in which to represent given data.

- 'More meaningful': should reduce noise and redundancy in the data

Goal: to seperate sources, filter data, and reveal 'hidden' dynamics. ICA also begins by assuming that the transformation to the new basis is linear:

$$\mathbf{WX} = \mathbf{Y} \Rightarrow \mathbf{y_i} = \begin{bmatrix} \mathbf{w_i x_i} \\ \vdots \\ \mathbf{w_i x_i} \end{bmatrix}$$

where $\mathbf{x_i}, \mathbf{y_i}$ represent columns of the source and transformed data matrices $\mathbf{X}, \mathbf{Y}$ and $\mathbf{w_i}$ represents a row of the transform matrix $\mathbf{W}$. So the rows of $\mathbf{W}$ form a new basis for the columns of $\mathbf{X}$; they are the **Independent Components** of the given data.

Independent Component Analysis

ICA, like PCA, aims to compute a 'more meaningful' basis in which to represent given data.

- 'More meaningful': should reduce noise and redundancy in the data

Goal: to seperate sources, filter data, and reveal 'hidden' dynamics. ICA also begins by assuming that the transformation to the new basis is linear:

$$\mathbf{WX} = \mathbf{Y} \Rightarrow \mathbf{y_i} = \begin{bmatrix} \mathbf{w_i x_i} \\ \vdots \\ \mathbf{w_i x_i} \end{bmatrix}$$

where $\mathbf{x_i}, \mathbf{y_i}$ represent columns of the source and transformed data matrices $\mathbf{X}, \mathbf{Y}$ and $\mathbf{w_i}$ represents a row of the transform matrix $\mathbf{W}$. So the rows of $\mathbf{W}$ form a new basis for the columns of $\mathbf{X}$; they are the **Independent Components** of the given data.

Independent Component Analysis

ICA, like PCA, aims to compute a 'more meaningful' basis in which to represent given data.

- 'More meaningful': should reduce noise and redundancy in the data

Goal: to seperate sources, filter data, and reveal 'hidden' dynamics. ICA also begins by assuming that the transformation to the new basis is linear:

$$\mathbf{WX} = \mathbf{Y} \Rightarrow \mathbf{y_i} = \begin{bmatrix} \mathbf{w_i x_i} \\ \vdots \\ \mathbf{w_i x_i} \end{bmatrix}$$

where $\mathbf{x_i}, \mathbf{y_i}$ represent columns of the source and transformed data matrices $\mathbf{X}, \mathbf{Y}$ and $\mathbf{w_i}$ represents a row of the transform matrix $\mathbf{W}$. So the rows of $\mathbf{W}$ form a new basis for the columns of $\mathbf{X}$; they are the **Independent Components** of the given data.

Independent Component Analysis

However, unlike PCA:

- The vectors of the new basis are not assumed to be orthogonal.
- Directions of highest variance are not assumed to be strongly charectoristic of the underlying dynamics of the system.
- Measures based on higher order statistics (> 2) are assumed to be necessary to seperate the sources in a problem.
- There is no standard measure of independence or computational algorithm to perform ICA.
  - Algorithms are iterative and tend to be much more computationally expensive than the SVD.
  - In general, well-posedness is not guaranteed.

Independent Component Analysis

However, unlike PCA:

- The vectors of the new basis are not assumed to be orthogonal.
- Directions of highest variance are not assumed to be strongly charectoristic of the underlying dynamics of the system.
- Measures based on higher order statistics (> 2) are assumed to be necessary to seperate the sources in a problem.
- There is no standard measure of independence or computational algorithm to perform ICA.
  - Algorithms are iterative and tend to be much more computationally expensive than the SVD.
  - In general, well-posedness is not guaranteed.

Independent Component Analysis

However, unlike PCA:

- The vectors of the new basis are not assumed to be orthogonal.
- Directions of highest variance are not assumed to be strongly charectoristic of the underlying dynamics of the system.
- Measures based on higher order statistics (> 2) are assumed to be necessary to seperate the sources in a problem.
- There is no standard measure of independence or computational algorithm to perform ICA.
    - Algorithms are iterative and tend to be much more computationally expensive than the SVD.
    - In general, well-posedness is not guaranteed.

Independent Component Analysis

However, unlike PCA:

- The vectors of the new basis are not assumed to be orthogonal.
- Directions of highest variance are not assumed to be strongly charectoristic of the underlying dynamics of the system.
- Measures based on higher order statistics (> 2) are assumed to be necessary to seperate the sources in a problem.
- There is no standard measure of independence or computational algorithm to perform ICA.
    - Algorithms are iterative and tend to be much more computationally expensive than the SVD.
    - In general, well-posedness is not guaranteed.

Independent Component Analysis

However, unlike PCA:

- The vectors of the new basis are not assumed to be orthogonal.
- Directions of highest variance are not assumed to be strongly charectoristic of the underlying dynamics of the system.
- Measures based on higher order statistics (> 2) are assumed to be necessary to seperate the sources in a problem.
- There is no standard measure of independence or computational algorithm to perform ICA.
  - Algorithms are iterative and tend to be much more computationally expensive than the SVD.
  - In general, well-posedness is not guaranteed.

Independent Component Analysis

However, unlike PCA:

- The vectors of the new basis are not assumed to be orthogonal.
- Directions of highest variance are not assumed to be strongly charectoristic of the underlying dynamics of the system.
- Measures based on higher order statistics (> 2) are assumed to be necessary to seperate the sources in a problem.
- There is no standard measure of independence or computational algorithm to perform ICA.
  - Algorithms are iterative and tend to be much more computationally expensive than the SVD.
  - In general, well-posedness is not guaranteed.

Independent Component Analysis

However, unlike PCA:

- The vectors of the new basis are not assumed to be orthogonal.
- Directions of highest variance are not assumed to be strongly charectoristic of the underlying dynamics of the system.
- Measures based on higher order statistics (> 2) are assumed to be necessary to seperate the sources in a problem.
- There is no standard measure of independence or computational algorithm to perform ICA.
  - Algorithms are iterative and tend to be much more computationally expensive than the SVD.
  - In general, well-posedness is not guaranteed.

Independent Component Analysis

Also:

- There is no framework for reducing the dimensionality of data within ICA (must perform PCA first!)
- Computationally efficient estimators used to approximate higher order statistics are typically biased.
- The variances of the original sources cannot be recovered.
- The signs of the original sources cannot be recovered.
- Any ordering of the sources which existed prior to mixing cannot be recovered.

Independent Component Analysis

Also:

- There is no framework for reducing the dimensionality of data within ICA (must perform PCA first!)
- Computationally efficient estimators used to approximate higher order statistics are typically biased.
- The variances of the original sources cannot be recovered.
- The signs of the original sources cannot be recovered.
- Any ordering of the sources which existed prior to mixing cannot be recovered.

Independent Component Analysis

Also:

- There is no framework for reducing the dimensionality of data within ICA (must perform PCA first!)
- Computationally efficient estimators used to approximate higher order statistics are typically biased.
- The variances of the original sources cannot be recovered.
- The signs of the original sources cannot be recovered.
- Any ordering of the sources which existed prior to mixing cannot be recovered.

Independent Component Analysis

Also:

- There is no framework for reducing the dimensionality of data within ICA (must perform PCA first!)
- Computationally efficient estimators used to approximate higher order statistics are typically biased.
- The variances of the original sources cannot be recovered.
- The signs of the original sources cannot be recovered.
- Any ordering of the sources which existed prior to mixing cannot be recovered.

Independent Component Analysis

Also:

- There is no framework for reducing the dimensionality of data within ICA (must perform PCA first!)
- Computationally efficient estimators used to approximate higher order statistics are typically biased.
- The variances of the original sources cannot be recovered.
- The signs of the original sources cannot be recovered.
- Any ordering of the sources which existed prior to mixing cannot be recovered.

Independent Component Analysis

- Seek $\mathbf{W}, \mathbf{Y}$ such that $\mathbf{Y} = \mathbf{W}^{-1}\mathbf{X}$ and each row of $\mathbf{Y}$ maximizes some high-order measure of independence.
- Typical perspectives:
  - Maximum liklihood
  - Direct high-order moments
  - Maximization of mutual information
  - Maximization of negative information entropy
- The optimization for any choice of the above measures is motivated by the **Central Limit Theorem**.

Independent Component Analysis

- Seek $\mathbf{W}, \mathbf{Y}$ such that $\mathbf{Y} = \mathbf{W}^{-1}\mathbf{X}$ and each row of $\mathbf{Y}$ maximizes some high-order measure of independence.
- Typical perspectives:
  - Maximum liklihood
  - Direct high-order moments
  - Maximization of mutual information
  - Maximization of negative information entropy
- The optimization for any choice of the above measures is motivated by the **Central Limit Theorem**.

Independent Component Analysis

- Seek $\mathbf{W}, \mathbf{Y}$ such that $\mathbf{Y} = \mathbf{W}^{-1}\mathbf{X}$ and each row of $\mathbf{Y}$ maximizes some high-order measure of independence.
- Typical perspectives:
  - Maximum liklihood
  - Direct high-order moments
  - Maximization of mutual information
  - Maximization of negative information entropy
- The optimization for any choice of the above measures is motivated by the **Central Limit Theorem**.

Independent Component Analysis

- Seek $\mathbf{W}, \mathbf{Y}$ such that $\mathbf{Y} = \mathbf{W}^{-1}\mathbf{X}$ and each row of $\mathbf{Y}$ maximizes some high-order measure of independence.
- Typical perspectives:
  - Maximum liklihood
  - Direct high-order moments
  - Maximization of mutual information
  - Maximization of negative information entropy
- The optimization for any choice of the above measures is motivated by the **Central Limit Theorem**.

Independent Component Analysis

- Seek $\mathbf{W}, \mathbf{Y}$ such that $\mathbf{Y} = \mathbf{W}^{-1}\mathbf{X}$ and each row of $\mathbf{Y}$ maximizes some high-order measure of independence.
- Typical perspectives:
  - Maximum liklihood
  - Direct high-order moments
  - Maximization of mutual information
  - Maximization of negative information entropy
- The optimization for any choice of the above measures is motivated by the **Central Limit Theorem**.

Independent Component Analysis

- Seek $\mathbf{W}, \mathbf{Y}$ such that $\mathbf{Y} = \mathbf{W}^{-1}\mathbf{X}$ and each row of $\mathbf{Y}$ maximizes some high-order measure of independence.
- Typical perspectives:
  - Maximum liklihood
  - Direct high-order moments
  - Maximization of mutual information
  - Maximization of negative information entropy
- The optimization for any choice of the above measures is motivated by the **Central Limit Theorem**.

Independent Component Analysis

- Seek $\mathbf{W}, \mathbf{Y}$ such that $\mathbf{Y} = \mathbf{W}^{-1}\mathbf{X}$ and each row of $\mathbf{Y}$ maximizes some high-order measure of independence.
- Typical perspectives:
  - Maximum liklihood
  - Direct high-order moments
  - Maximization of mutual information
  - Maximization of negative information entropy
- The optimization for any choice of the above measures is motivated by the **Central Limit Theorem**.

Independent Component Analysis

Central Limit Theorem (Lyapunov)

Let $X_n, n \in \mathbb{N}$ be any sequence of independent random variables; each with finite mean μ_n and variance σ_n^2. Define $S_N^2 = \sum_{i=1}^{n} \sigma_i^2$. If for some $\delta > 0$ the expectations $\mathbb{E}\left[[X_k]^{2+\delta}\right]$ are finite for every $k \in \mathbb{N}$ and the condition $\lim_{N\to\infty} \frac{1}{S_N^{2+\delta}} \sum_{i=1}^{N} \mathbb{E}\left[[X_n - \mu_n]^{2+\delta}\right] = 0$ is satisfied, then:

$$\frac{\sum_{i=1}^{N} (X_n - \mu_n)}{S_n} \xrightarrow{\text{distr.}} \text{Normal}\,(0,1) \text{ as } N \to \infty$$

Independent Component Analysis

Heuristic arguement:

- The sum of any group of independent random variables is 'more gaussian' they any of the individual random variables.
- Assume that none of the original sources has a gaussian distribution:
  - Then minimizing gaussinity w.r.t. higher order statistical measures should seperate the sources in $\mathbf{X}$!

Independent Component Analysis

Heuristic arguement:

- The sum of any group of independent random variables is 'more gaussian' they any of the individual random variables.
- Assume that none of the original sources has a gaussian distribution:
  - Then minimizing gaussinity w.r.t. higher order statistical measures should seperate the sources in $\mathbf{X}$!

Independent Component Analysis

Heuristic arguement:

- The sum of any group of independent random variables is 'more gaussian' they any of the individual random variables.
- Assume that none of the original sources has a gaussian distribution:
  - Then minimizing gaussinity w.r.t. higher order statistical measures should seperate the sources in $\mathbf{X}$!

Independent Component Analysis

Definition

The **Kurtosis** of a a random variable x is defined to be:

$$\kappa(x) = E\left[x^4\right] - 3\left(E\left[y^2\right]\right)^2$$

- Kurtosis is a measure of 'peakedness' and thickness of tails for a distribution.
- Note that if x is gaussian:
  $\kappa(x) = 3\left(E\left[y^2\right]\right)^2 - 3\left(E\left[y^2\right]\right)^2 = 0$
- So, simultaneously minimizing $\left|\kappa\left(\mathbf{Y_1}\right)\right|, \ldots, \left|\kappa\left(\mathbf{Y_m}\right)\right|$ or $\left(\kappa\left(\mathbf{Y_1}\right)\right)^2, \ldots, \left(\kappa\left(\mathbf{Y_m}\right)\right)^2$ can provide a basis where the recovered sources are (in one sense) maximally non-gaussian.

Independent Component Analysis

Definition

The **Kurtosis** of a a random variable x is defined to be:

$$\kappa(x) = E\left[x^4\right] - 3\left(E\left[y^2\right]\right)^2$$

- Kurtosis is a measure of 'peakedness' and thickness of tails for a distribution.
- Note that if x is gaussian: $\kappa(x) = 3\left(E\left[y^2\right]\right)^2 - 3\left(E\left[y^2\right]\right)^2 = 0$
- So, simultaneously minimizing $|\kappa(\mathbf{Y_1})|, \ldots, |\kappa(\mathbf{Y_m})|$ or $(\kappa(\mathbf{Y_1}))^2, \ldots, (\kappa(\mathbf{Y_m}))^2$ can provide a basis where the recovered sources are (in one sense) maximally non-gaussian.

Independent Component Analysis

Definition

The **Kurtosis** of a a random variable x is defined to be:

$$\kappa(x) = E\left[x^4\right] - 3\left(E\left[y^2\right]\right)^2$$

- Kurtosis is a measure of 'peakedness' and thickness of tails for a distribution.
- Note that if x is gaussian:
  $\kappa(x) = 3\left(E\left[y^2\right]\right)^2 - 3\left(E\left[y^2\right]\right)^2 = 0$
- So, simultaneously minimizing $|\kappa(\mathbf{Y_1})|, \ldots, |\kappa(\mathbf{Y_m})|$ or $(\kappa(\mathbf{Y_1}))^2, \ldots, (\kappa(\mathbf{Y_m}))^2$ can provide a basis where the recovered sources are (in one sense) maximally non-gaussian.

Independent Component Analysis

Definition

The **Kurtosis** of a a random variable x is defined to be:

$$\kappa(x) = E\left[x^4\right] - 3\left(E\left[y^2\right]\right)^2$$

- Kurtosis is a measure of 'peakedness' and thickness of tails for a distribution.
- Note that if x is gaussian:
  $\kappa(x) = 3\left(E\left[y^2\right]\right)^2 - 3\left(E\left[y^2\right]\right)^2 = 0$
- So, simultaneously minimizing $\left|\kappa\left(\mathbf{Y_1}\right)\right|, \ldots, \left|\kappa\left(\mathbf{Y_m}\right)\right|$ or $\left(\kappa\left(\mathbf{Y_1}\right)\right)^2, \ldots, \left(\kappa\left(\mathbf{Y_m}\right)\right)^2$ can provide a basis where the recovered sources are (in one sense) maximally non-gaussian.

Independent Component Analysis

Drawbacks of using kurtosis as an optimality critereon:

- Very sensitive to outliers.
- Note a robust measure of gaussinity.

A more suitable measure of gaussinity is required to produce stable ICA methods.

Independent Component Analysis

Definition

The **Differential Entropy** of a a continuous random variable X with density function $f_X(x)$is defined to be:

$$H(X) = -\int f_X(x) \log f_X(x)\, dx$$

- Can be interpreted as the degree of information carried by a random variable.
- Fundamental result in information theory: *A gaussian random variable has the greatest entropy among all random variables of equal variance.*

Independent Component Analysis

Definition

The **Differential Entropy** of a a continuous random variable X with density function $f_X(x)$is defined to be:

$$H(X) = -\int f_X(x) \log f_X(x)\, dx$$

- Can be interpreted as the degree of information carried by a random variable.
- Fundamental result in information theory: *A gaussian random variable has the greatest entropy among all random variables of equal variance.*

Independent Component Analysis

Definition

The **Differential Entropy** of a a continuous random variable X with density function $f_X(x)$is defined to be:

$$H(X) = -\int f_X(x) \log f_X(x)\, dx$$

- Can be interpreted as the degree of information carried by a random variable.
- Fundamental result in information theory: *A gaussian random variable has the greatest entropy among all random variables of equal variance*.

Independent Component Analysis

Consider the following:

Definition

The **Negative Entropy** (or **Negentropy**) of a a continuous random variable X with density function $f_X(x)$is defined to be:

$$J(X) = H(X_{gauss}) - H(X)$$

where X_{gauss} is a random variable with identical variance to X (or identical covariance matrix).

Advantages:

- Always non-negative; equal to 0 for a gaussian random variable.
- Not sensitive to sample outliers.

Independent Component Analysis

Consider the following:

Definition

The **Negative Entropy** (or **Negentropy**) of a a continuous random variable X with density function $f_X(x)$is defined to be:

$$J(X) = H(X_{gauss}) - H(X)$$

where X_{gauss} is a random variable with identical variance to X (or identical covariance matrix).

Advantages:

- Always non-negative; equal to 0 for a gaussian random variable.
- Not sensitive to sample outliers.

Independent Component Analysis

Consider the following:

Definition

The **Negative Entropy** (or **Negentropy**) of a a continuous random variable X with density function $f_X(x)$is defined to be:

$$J(X) = H(X_{gauss}) - H(X)$$

where X_{gauss} is a random variable with identical variance to X (or identical covariance matrix).

Advantages:

- Always non-negative; equal to 0 for a gaussian random variable.
- Not sensitive to sample outliers.

Independent Component Analysis

Difficulties:

- Negentropy optimization is computationally difficult to deal with directly.

Estimates:

- $J(X) \approx \frac{1}{12} E\left[y^3\right]^2 + \frac{1}{48} \kappa(y)^2$
  - Same problems as in the case of just using kurtosis!
- $J(X) \approx \sum_{i=1}^{n} k_i (E\left[G_i(y)\right] - E\left[G_i(v)\right])^2$, where $\{k_i\}$ are positive constants, v is a standard gaussian random variable and $\{G_i\}$ are some non-quadratic functions.

Independent Component Analysis

Difficulties:

- Negentropy optimization is computationally difficult to deal with directly.

Estimates:

- $J(X) \approx \frac{1}{12} E\left[y^3\right]^2 + \frac{1}{48} \kappa(y)^2$
  - Same problems as in the case of just using kurtosis!
- $J(X) \approx \sum_{i=1}^{n} k_i (E\left[G_i(y)\right] - E\left[G_i(v)\right])^2$, where $\{k_i\}$ are positive constants, v is a standard gaussian random variable and $\{G_i\}$ are some non-quadratic functions.

Independent Component Analysis

Difficulties:

- Negentropy optimization is computationally difficult to deal with directly.

Estimates:

- $J(X) \approx \frac{1}{12} E\left[y^3\right]^2 + \frac{1}{48} \kappa(y)^2$
  - Same problems as in the case of just using kurtosis!
- $J(X) \approx \sum_{i=1}^{n} k_i (E\left[G_i(y)\right] - E\left[G_i(v)\right])^2$, where $\{k_i\}$ are positive constants, v is a standard gaussian random variable and $\{G_i\}$ are some non-quadratic functions.

Independent Component Analysis

Typically:

- All of the G_i are the same function.
- Very good results have been demonstrated using:
  - $G(u) = \frac{1}{\alpha_1} \log[\cosh(\alpha_1 u)]$, for some constant $1 \leq \alpha_1 \leq 2$
  - $G(u) = -\exp\left(-u^2/2\right)$

Independent Component Analysis

Typically:

- All of the G_i are the same function.
- Very good results have been demonstrated using:
  - $G(u) = \frac{1}{\alpha_1} \log[\cosh(\alpha_1 u)]$, for some constant $1 \leq \alpha_1 \leq 2$
  - $G(u) = -\exp\left(-u^2/2\right)$

Independent Component Analysis

Typically:

- All of the G_i are the same function.
- Very good results have been demonstrated using:
  - $G(u) = \frac{1}{\alpha_1} \log[\cosh(\alpha_1 u)]$, for some constant $1 \leq \alpha_1 \leq 2$
  - $G(u) = -\exp\left(-u^2/2\right)$

Independent Component Analysis

Consider the computation of one the first independent component $\mathbf{w_1}$, based on maximizing negentropy: The maxima of the negentropy approximations of $\mathbf{w_1^T X}$ are obtained at certain optima of $\mathbb{E}\left[\mathbf{w_1^T X}\right]$. KKT conditions: optima of $\mathbb{E}\left[\mathbf{w_1^T X}\right]$ under the constraint $\mathbb{E}\left[\left(\mathbf{w_1^T X}\right)^2\right] = \|\mathbf{w_1}\| = 1$ are obtained at points where:

$$\mathbb{E}\left[\mathbf{w_1^T X}\right] - \beta \mathbf{w_1} = \mathbf{0}$$

So, the Jacobian has the form:

$$\mathbf{J}\left(\mathbf{w_1}\right) = \mathbb{E}\left[\mathbf{X X^T} g'\left(\mathbf{w_1^T X}\right)\right] - \beta \mathbf{I}$$

Independent Component Analysis

If we use the approximation $\mathbb{E}\left[\mathbf{X}\mathbf{X}^{\mathbf{T}}\mathbf{g}'\left(\mathbf{w}_1^{\mathbf{T}}\mathbf{X}\right)\right] \approx \mathbb{E}\left[\mathbf{X}\mathbf{X}^{\mathbf{T}}\right]\mathbb{E}\left[g'\left(\mathbf{w}_1^{\mathbf{T}}\mathbf{X}\right)\right] = \mathbb{E}\left[g'\left(\mathbf{w}_1^{\mathbf{T}}\mathbf{X}\right)\right]\mathbf{I}$, we get the Newton-Raphson iteration:

$$\mathbf{w_1} = \mathbf{w_1} - \left(\mathbf{E}\left[\mathbf{X}\mathbf{g}\left(\mathbf{w}_1^{\mathbf{T}}\mathbf{X}\right)\right] - \beta\mathbf{w_1}\right) / \left(\mathbf{E}\left[\mathbf{g}'\left(\mathbf{w}_1^{\mathbf{T}}\mathbf{X}\right)\right] - \beta\right)$$

Dividing both sides by $\beta - E\left[g'\left(\mathbf{w}_1^{\mathbf{T}}\mathbf{X}\right)\right]$ gives:

$$\mathbf{w}_1^{+} = \mathbf{E}\left[\mathbf{X}\mathbf{g}\left(\mathbf{w}_1^{\mathbf{T}}\mathbf{X}\right)\right] - \mathbf{E}\left[\mathbf{X}\mathbf{g}'\left(\mathbf{w}_1^{\mathbf{T}}\mathbf{X}\right)\right]\mathbf{w_1}$$

This is the basic iterate of the **FastICA** algorithm[4].

Independent Component Analysis

The computation of a single Independent Component (FastICA):

- Choose an initial random vector $\mathbf{w_1}$.
- Compute $\mathbf{w_1^+} = \mathbf{E}\left[\mathbf{X}\mathbf{g}\left(\mathbf{w_1^T}\mathbf{X}\right)\right] - \mathbf{E}\left[\mathbf{X}\mathbf{g}'\left(\mathbf{w_1^T}\mathbf{X}\right)\right]\mathbf{w_1}$
- Set $\mathbf{w_1} = \mathbf{w_1^+} / \left\|\mathbf{w_1^+}\right\|$.
- Repeat until convergence.

Independent Component Analysis

The computation of a single Independent Component (FastICA):

- Choose an initial random vector $\mathbf{w_1}$.
- Compute $\mathbf{w}_1^+ = \mathbf{E}\left[\mathbf{X}\mathbf{g}\left(\mathbf{w}_1^\mathsf{T}\mathbf{X}\right)\right] - \mathbf{E}\left[\mathbf{X}\mathbf{g}'\left(\mathbf{w}_1^\mathsf{T}\mathbf{X}\right)\right]\mathbf{w}_1$
- Set $\mathbf{w_1} = \mathbf{w}_1^+ / \left\|\mathbf{w}_1^+\right\|$.
- Repeat until convergence.

Independent Component Analysis

The computation of a single Independent Component (FastICA):

- Choose an initial random vector $\mathbf{w_1}$.
- Compute $\mathbf{w_1^+} = \mathbf{E}\left[\mathbf{X}\mathbf{g}\left(\mathbf{w_1^T X}\right)\right] - \mathbf{E}\left[\mathbf{X}\mathbf{g}'\left(\mathbf{w_1^T X}\right)\right]\mathbf{w_1}$
- Set $\mathbf{w_1} = \mathbf{w_1^+} / \left\|\mathbf{w_1^+}\right\|$.
- Repeat until convergence.

Independent Component Analysis

The computation of a single Independent Component (FastICA):

- Choose an initial random vector $\mathbf{w_1}$.
- Compute $\mathbf{w_1^+} = \mathbf{E}\left[\mathbf{X}\mathbf{g}\left(\mathbf{w_1^T X}\right)\right] - \mathbf{E}\left[\mathbf{X}\mathbf{g}'\left(\mathbf{w_1^T X}\right)\right]\mathbf{w_1}$
- Set $\mathbf{w_1} = \mathbf{w_1^+} / \left\|\mathbf{w_1^+}\right\|$.
- Repeat until convergence.

Independent Component Analysis

The computation of a single Independent Component (FastICA):

- Choose an initial random vector $\mathbf{w_1}$.
- Compute $\mathbf{w_1^+} = \mathbf{E}\left[\mathbf{X}\mathbf{g}\left(\mathbf{w_1^T X}\right)\right] - \mathbf{E}\left[\mathbf{X}\mathbf{g}'\left(\mathbf{w_1^T X}\right)\right]\mathbf{w_1}$
- Set $\mathbf{w_1} = \mathbf{w_1^+} / \left\|\mathbf{w_1^+}\right\|$.
- Repeat until convergence.

Independent Component Analysis

The computation of the full Independent Component Analysis (FastICA):

- Assume $\mathbf{w_1}, \ldots, \mathbf{w_p}$ independent components have been estimated.
- Run single-component method for a vector $\mathbf{w_{p+1}}$, and after every iteration subtract $\mathbf{w_{p+1}}$ from $\mathbf{w_{p+1}^T w_j w_j}$ for $j = 1, \ldots, p$:

$$\mathbf{w_{p+1}} = \mathbf{w_{p+1}} - \sum_{j=1}^{p} \mathbf{w_{p+1}^T w_j w_j}$$

- Renormalize $\mathbf{w_{p+1}}$:

$$\mathbf{w_{p+1}} = \mathbf{w_{p+1}} / \sqrt{\mathbf{w_{p+1}^T w_{p+1}}}$$

Repeat until $p = m$

Independent Component Analysis

The computation of the full Independent Component Analysis (FastICA):

- Assume $\mathbf{w_1}, \ldots, \mathbf{w_p}$ independent components have been estimated.
- Run single-component method for a vector $\mathbf{w_{p+1}}$, and after every iteration subtract $\mathbf{w_{p+1}}$ from $\mathbf{w_{p+1}^T w_j w_j}$ for $j = 1, \ldots, p$:

$$\mathbf{w_{p+1}} = \mathbf{w_{p+1}} - \sum_{j=1}^{p} \mathbf{w_{p+1}^T w_j w_j}$$

- Renormalize $\mathbf{w_{p+1}}$:

$$\mathbf{w_{p+1}} = \mathbf{w_{p+1}} / \sqrt{\mathbf{w_{p+1}^T w_{p+1}}}$$

Repeat until $p = m$

Independent Component Analysis

The computation of the full Independent Component Analysis (FastICA):

- Assume $\mathbf{w_1}, \ldots, \mathbf{w_p}$ independent components have been estimated.
- Run single-component method for a vector $\mathbf{w_{p+1}}$, and after every iteration subtract $\mathbf{w_{p+1}}$ from $\mathbf{w_{p+1}^T w_j w_j}$ for $j = 1, \ldots, p$:

$$\mathbf{w_{p+1}} = \mathbf{w_{p+1}} - \sum_{j=1}^{p} \mathbf{w_{p+1}^T w_j w_j}$$

- Renormalize $\mathbf{w_{p+1}}$:

$$\mathbf{w_{p+1}} = \mathbf{w_{p+1}} / \sqrt{\mathbf{w_{p+1}^T w_{p+1}}}$$

Repeat until $p = m$

Independent Component Analysis

The computation of the full Independent Component Analysis (FastICA):

- Assume $\mathbf{w_1}, \ldots, \mathbf{w_p}$ independent components have been estimated.
- Run single-component method for a vector $\mathbf{w_{p+1}}$, and after every iteration subtract $\mathbf{w_{p+1}}$ from $\mathbf{w_{p+1}^T w_j w_j}$ for $j = 1, \ldots, p$:

$$\mathbf{w_{p+1}} = \mathbf{w_{p+1}} - \sum_{j=1}^{p} \mathbf{w_{p+1}^T w_j w_j}$$

- Renormalize $\mathbf{w_{p+1}}$:

$$\mathbf{w_{p+1}} = \mathbf{w_{p+1}} / \sqrt{\mathbf{w_{p+1}^T w_{p+1}}}$$

Repeat until $p = m$

Independent Component Analysis

The computation of the full Independent Component Analysis (FastICA):

- Assume $\mathbf{w_1}, \ldots, \mathbf{w_p}$ independent components have been estimated.
- Run single-component method for a vector $\mathbf{w_{p+1}}$, and after every iteration subtract $\mathbf{w_{p+1}}$ from $\mathbf{w_{p+1}^T w_j w_j}$ for $j = 1, \ldots, p$:

$$\mathbf{w_{p+1}} = \mathbf{w_{p+1}} - \sum_{j=1}^{p} \mathbf{w_{p+1}^T w_j w_j}$$

- Renormalize $\mathbf{w_{p+1}}$:

$$\mathbf{w_{p+1}} = \mathbf{w_{p+1}} / \sqrt{\mathbf{w_{p+1}^T w_{p+1}}}$$

Repeat until $p = m$

Independent Component Analysis

Properties of FastICA[4][3][1]:

- Convergence is cubic (assuming the ICA data model).
- There is no step-size parameter to be chosen.
- Is a type of neural algorithm.
- Directly computes ICs using practically any non-linearity g.
  - The choice of g does affect performance.
- Parallel, distributed, computationally simple, requires little memory.
- May still prematurely converge to local optima.
- PCA must be conducted on $\mathbf{X}$ prior to use of FastICA.

Independent Component Analysis

Properties of FastICA[4][3][1]:

- Convergence is cubic (assuming the ICA data model).
- There is no step-size parameter to be chosen.
- Is a type of neural algorithm.
- Directly computes ICs using practically any non-linearity g.
  - The choice of g does affect performance.
- Parallel, distributed, computationally simple, requires little memory.
- May still prematurely converge to local optima.
- PCA must be conducted on $\mathbf{X}$ prior to use of FastICA.

Independent Component Analysis

Properties of FastICA[4][3][1]:

- Convergence is cubic (assuming the ICA data model).
- There is no step-size parameter to be chosen.
- Is a type of neural algorithm.
- Directly computes ICs using practically any non-linearity g.
  - The choice of g does affect performance.
- Parallel, distributed, computationally simple, requires little memory.
- May still prematurely converge to local optima.
- PCA must be conducted on $\mathbf{X}$ prior to use of FastICA.

Independent Component Analysis

Properties of FastICA[4][3][1]:

- Convergence is cubic (assuming the ICA data model).
- There is no step-size parameter to be chosen.
- Is a type of neural algorithm.
- Directly computes ICs using practically any non-linearity g.
  - The choice of g does affect performance.
- Parallel, distributed, computationally simple, requires little memory.
- May still prematurely converge to local optima.
- PCA must be conducted on $\mathbf{X}$ prior to use of FastICA.

Independent Component Analysis

Properties of FastICA[4][3][1]:

- Convergence is cubic (assuming the ICA data model).
- There is no step-size parameter to be chosen.
- Is a type of neural algorithm.
- Directly computes ICs using practically any non-linearity g.
  - The choice of g does affect performance.
- Parallel, distributed, computationally simple, requires little memory.
- May still prematurely converge to local optima.
- PCA must be conducted on $\mathbf{X}$ prior to use of FastICA.

Independent Component Analysis

Properties of FastICA[4][3][1]:

- Convergence is cubic (assuming the ICA data model).
- There is no step-size parameter to be chosen.
- Is a type of neural algorithm.
- Directly computes ICs using practically any non-linearity g.
  - The choice of g does affect performance.
- Parallel, distributed, computationally simple, requires little memory.
- May still prematurely converge to local optima.
- PCA must be conducted on $\mathbf{X}$ prior to use of FastICA.

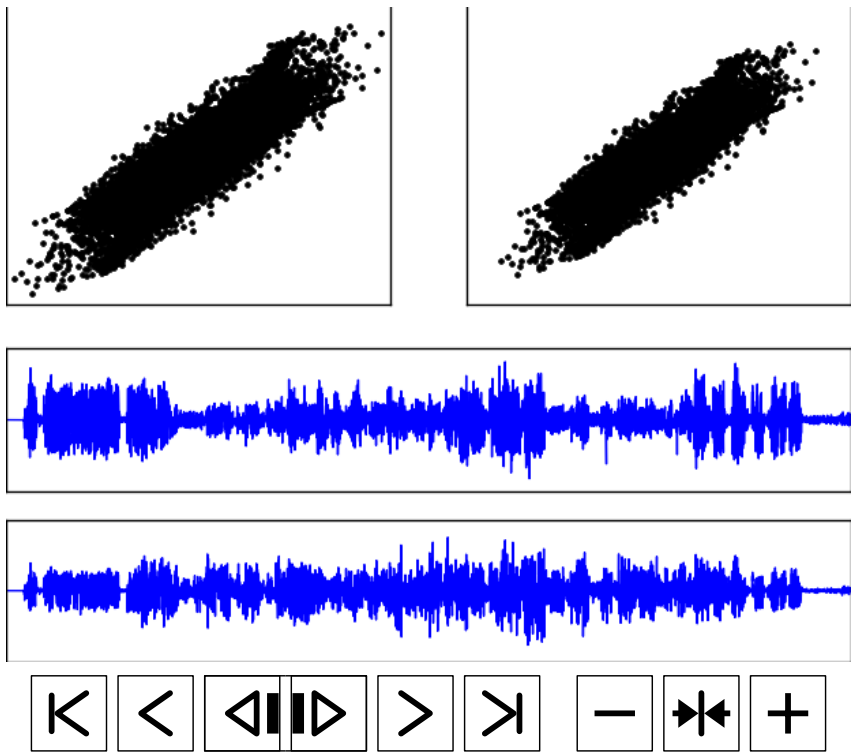
Independent Component Analysis

Properties of FastICA[4][3][1]:

- Convergence is cubic (assuming the ICA data model).
- There is no step-size parameter to be chosen.
- Is a type of neural algorithm.
- Directly computes ICs using practically any non-linearity g.
  - The choice of g does affect performance.
- Parallel, distributed, computationally simple, requires little memory.
- May still prematurely converge to local optima.
- PCA must be conducted on $\mathbf{X}$ prior to use of FastICA.

Independent Component Analysis

2-Source Audio Example

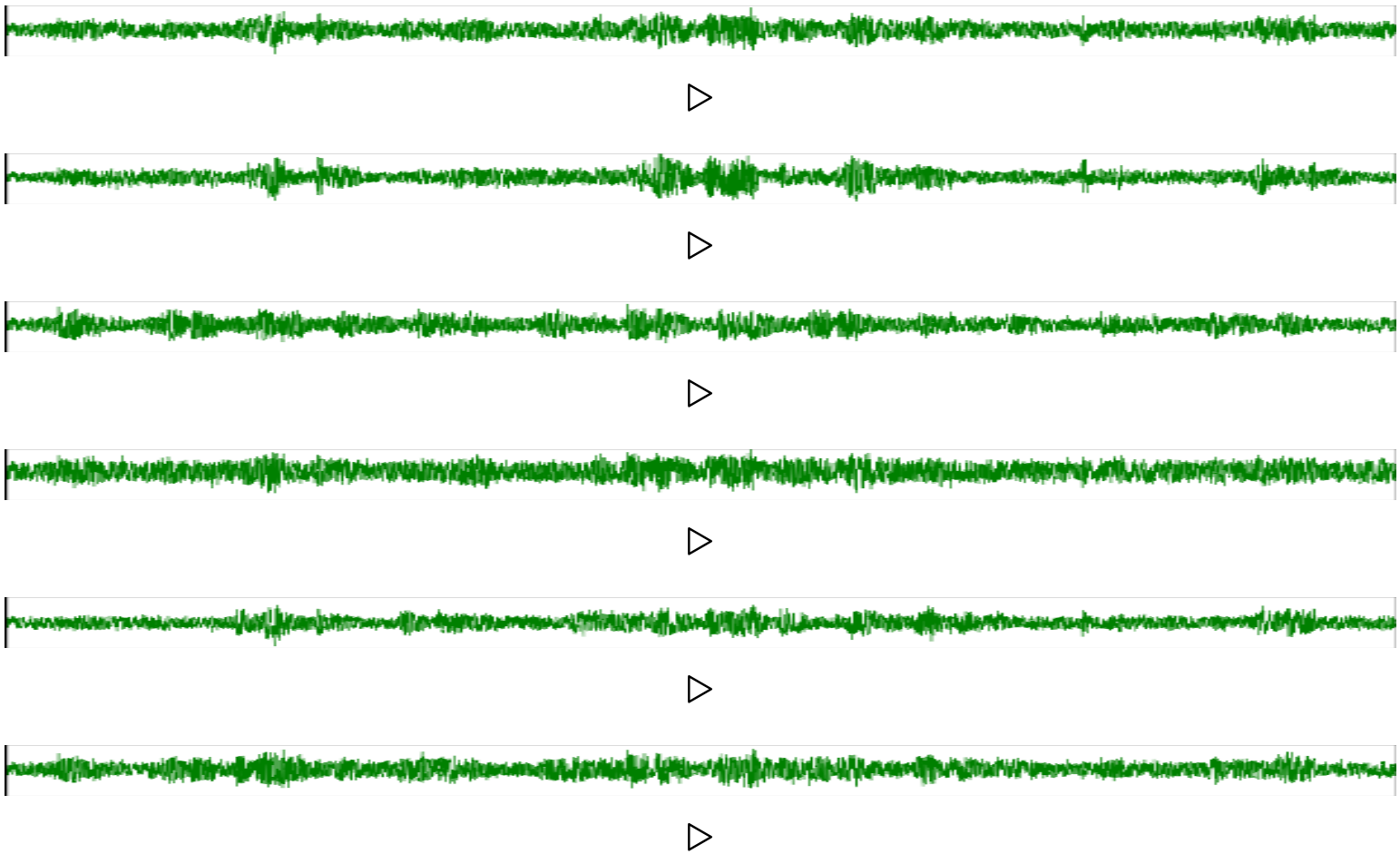


▷

▷

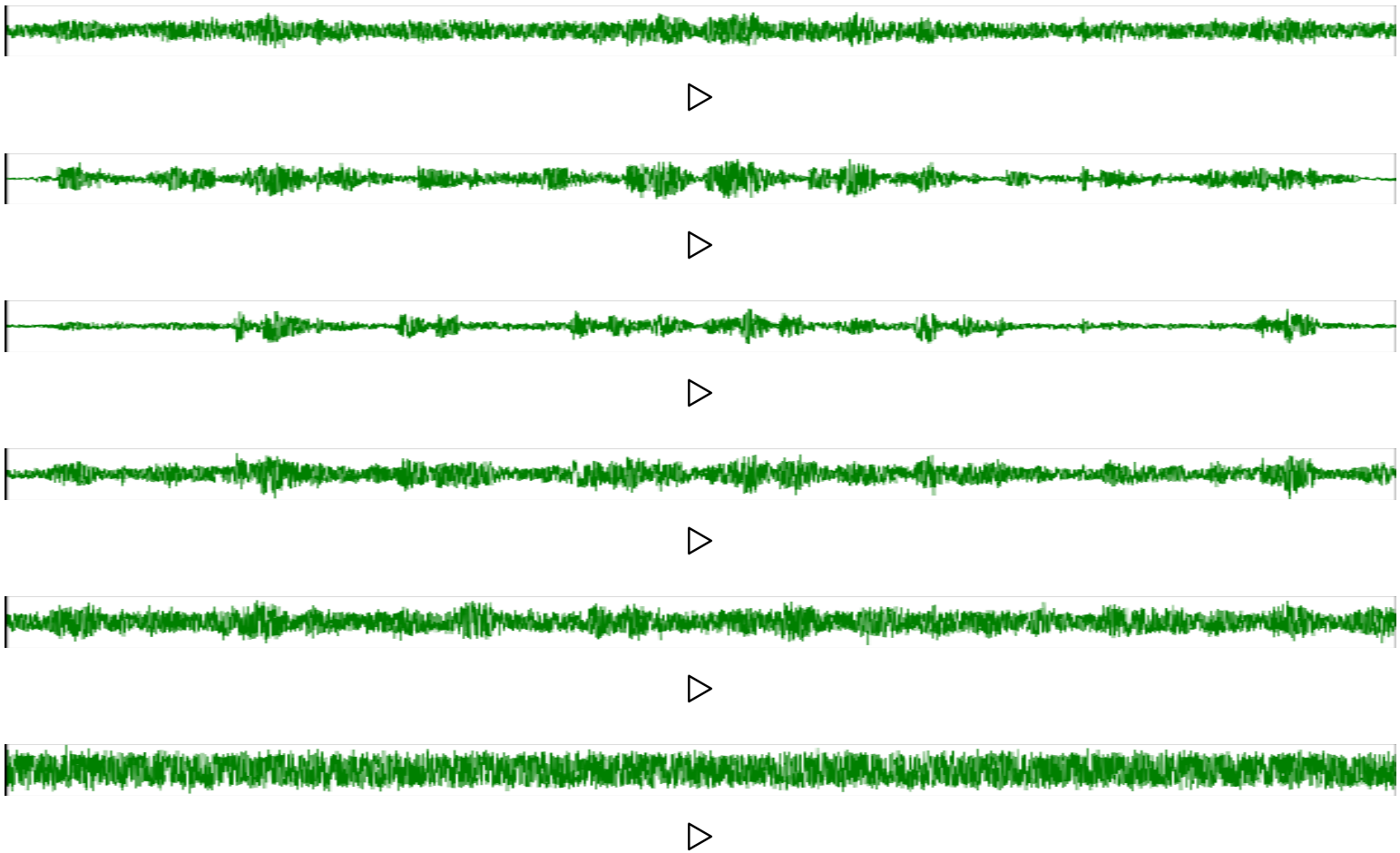
Example (Cocktail Party Problem)

These 6 audio recordings are assumed to be a linear mix of unknown sources (via multiplication with an unknown matrix):



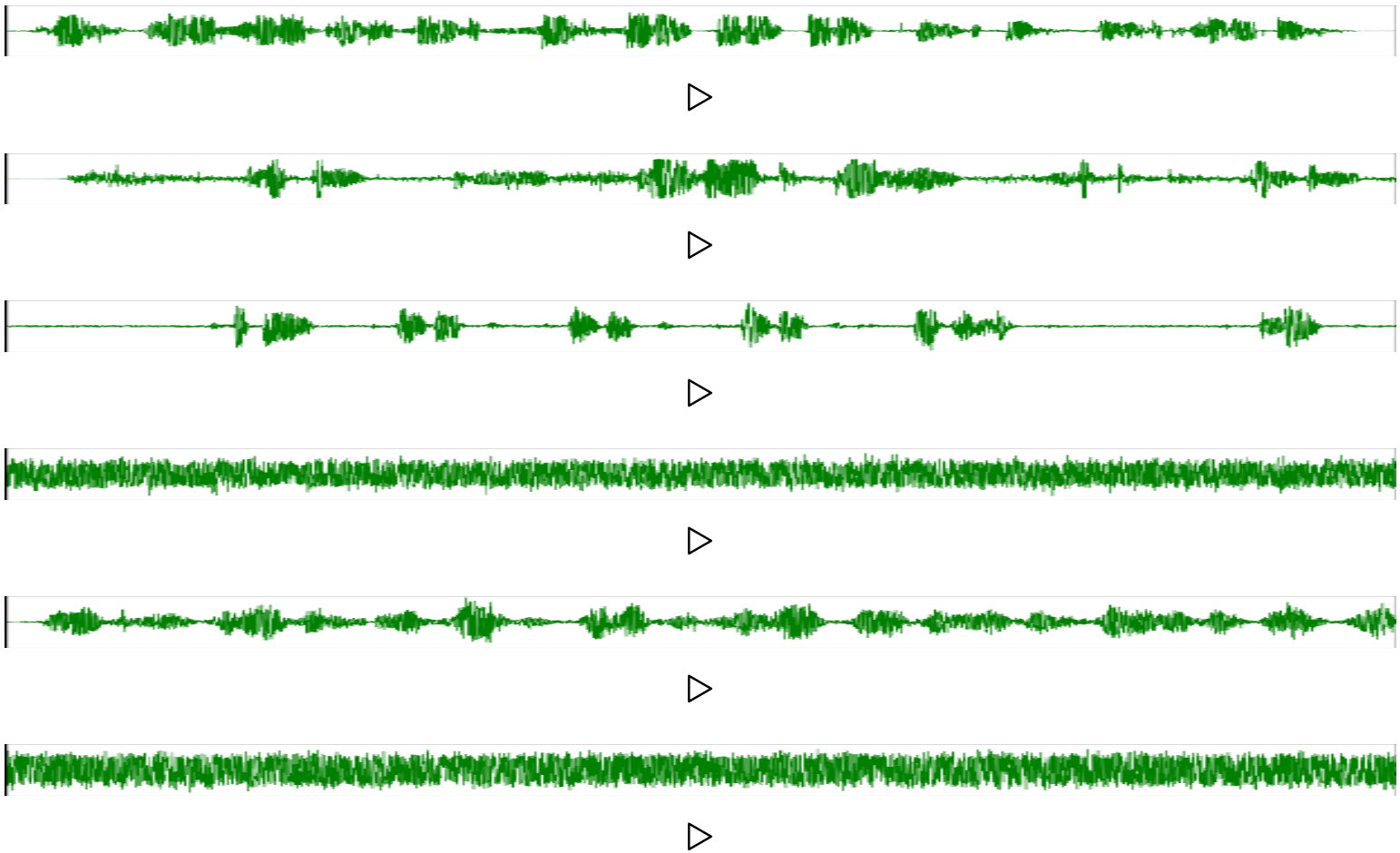
Seperation via PCA

The following 6 signals were retrieved from the mixed sources using PCA:



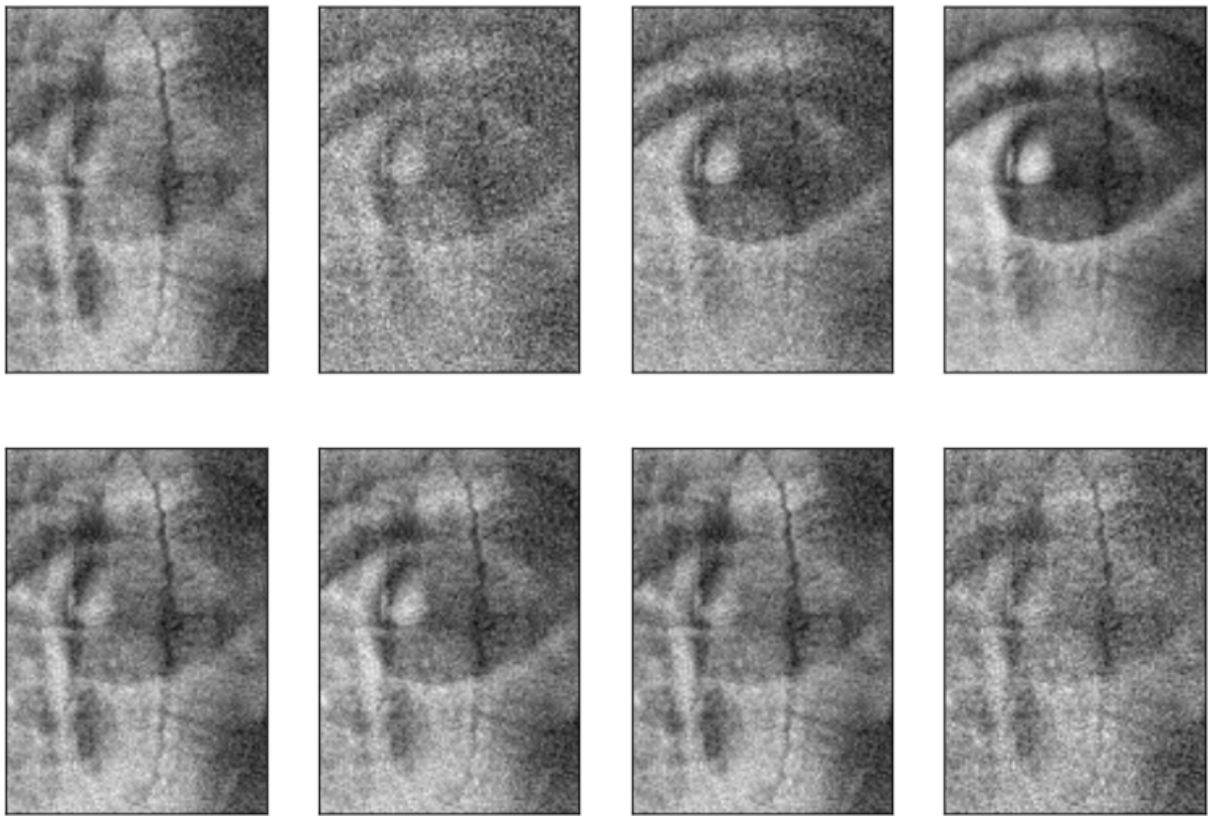
Seperation via ICA

The following 6 signals were retrieved from the mixed sources using ICA:

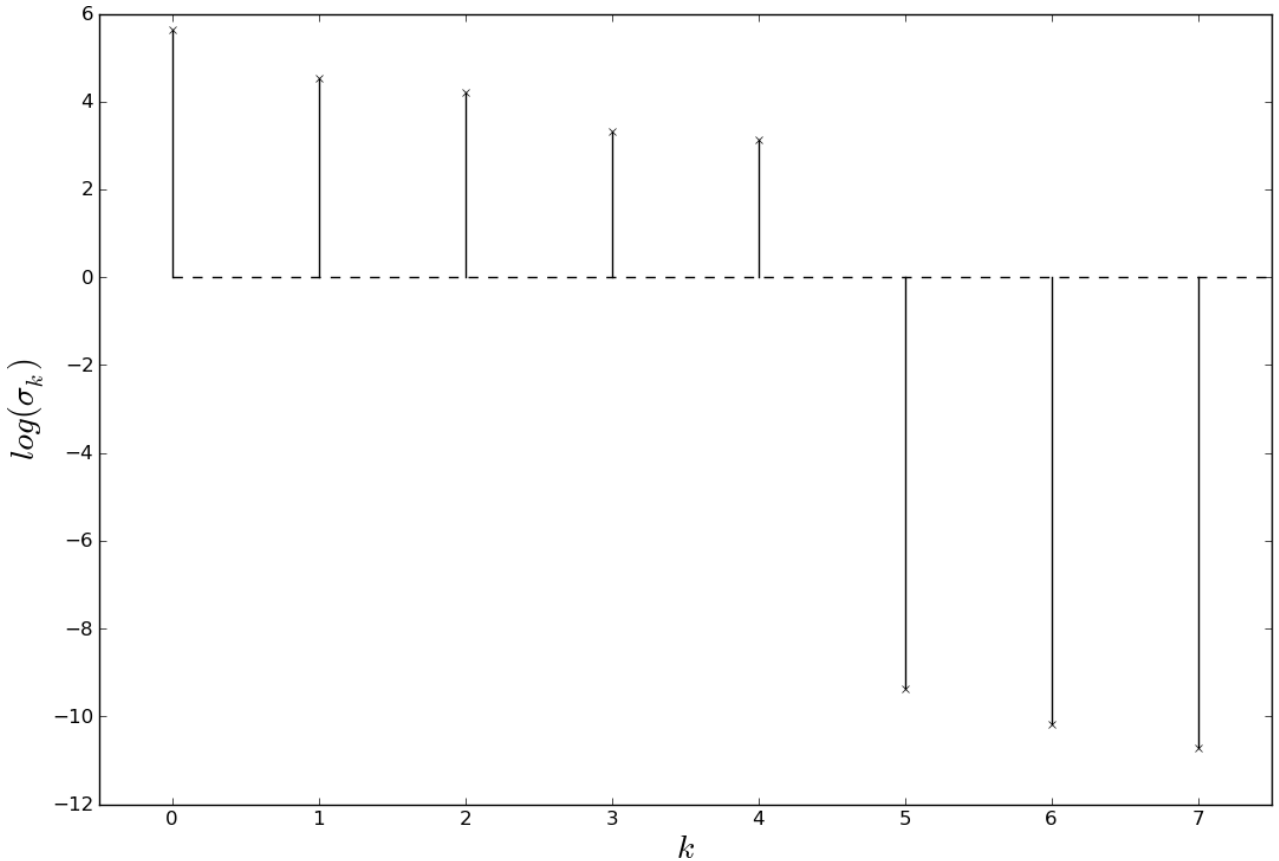


Seperation of Mixed Images

These 8 images are assumed to be a linear mix of unknown sources (via multiplication with an unknown matrix):



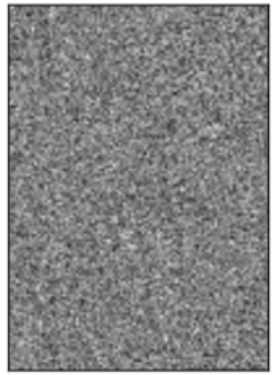
First: we precondition the system by executing PCA (SVD). A stem plot of the singular values, σ, gives:



So from the 8 observed images, there are only 5 significant components detected via PCA/SVD.

Seperation of Mixed Images

The following 5 images were retrieved from the mixed sources using ICA:



Aapo Hyvrinen and Erkki Oja.

A fast Fixed-Point algorithm for independent component analysis.

Neural Computation, 9(7):1483–1492, October 1997.

Fabrizio Esposito, Erich Seifritz, Elia Formisano, Renato Morrone, Tommaso Scarabino, Gioacchino Tedeschi, Sossio Cirillo, Rainer Goebel, and Francesco Di Salle.

Real-time independent component analysis of fMRI time-series.

NeuroImage, 20(4):2209–2224, December 2003.

Pierre Comon.

Independent component analysis, a new concept?

Signal Processing, 36(3):287–314, April 1994.

E. Oja.

Independent component analysis: algorithms and applications, 2000.

Questions?