Introduction to Principal Component Analysis and Independent Component Analysis

Tristan A. Hearn

Bioscience and Technology Branch, NASA Glenn Research Center

May 29, 2010
Table of Contents

Introduction
   Blind Separation 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
Introducion

Principal 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

Principal 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
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:

\[
x_1(t) = a_{11}s_1(t) + \ldots + a_{1N}s_N(t)
\]

\[
x_N(t) = a_{N1}s_1(t) + \ldots + a_{NN}s_N(t)
\]

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

where \( x_i, s_i \in \mathbb{R}^{M \times 1} \)
The Blind Seperation Problem (BSS)

Problem

Consider solving:

\[ x_1(t) = \sum_{i=1}^{N} a_{1i} s_i(t) \]

\[ \vdots \]

\[ x_N(t) = \sum_{i=1}^{N} a_{Ni} s_i(t) \]

\[ \Rightarrow \begin{bmatrix} x_1(t) \\ \vdots \\ x_N(t) \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} s_1(t) \\ \vdots \\ s_N(t) \end{bmatrix} \]

\[ \Rightarrow A S = X \]

where \( x_i, s_i \in \mathbb{R}^{M \times 1} \)
The Blind Seperation Problem (BSS)

Problem

Consider solving:

\[ x_1(t) = a_{11}s_1(t) + \ldots + a_{1N}s_N(t) \]
\[ \vdots \]
\[ x_N(t) = a_{N1}s_1(t) + \ldots + a_{NN}s_N(t) \]

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

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

\[ PX = Y \Rightarrow y_i = \begin{bmatrix} p_i x_i \\ \vdots \\ p_i x_i \end{bmatrix} \]

where \( x_i, y_i \) represent columns of the source and transformed data matrices \( X, Y \) and \( p_i \) represents a row of the transform matrix \( P \). So the rows of \( P \) form a new basis for the columns of \( X \); 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 separate sources, filter data, and reveal 'hidden' dynamics.

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

\[
PX = Y \Rightarrow y_i = \begin{bmatrix} \pi_1x_i \\ \vdots \\ \pi_nx_i \end{bmatrix}
\]

where \( x_i, y_i \) represent columns of the source and transformed data matrices \( X, Y \) and \( p_i \) represents a row of the transform matrix \( P \).

So the rows of \( P \) form a new basis for the columns of \( X \); 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 separate sources, filter data, and reveal 'hidden' dynamics.

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

\[ PX = Y \Rightarrow y_i = \begin{bmatrix} p_1 x_i \\ \vdots \\ p_k x_i \end{bmatrix} \]

where \( x_i, y_i \) represent columns of the source and transformed data matrices \( X, Y \) and \( p_i \) represents a row of the transform matrix \( P \).

So the rows of \( P \) form a new basis for the columns of \( X \); 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 separate sources, filter data, and reveal 'hidden' dynamics.

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

\[ PX = Y \Rightarrow y_i = \begin{bmatrix} p_1 x_i \\ \vdots \\ p_i x_i \end{bmatrix} \]

where \( x_i, y_i \) represent columns of the source and transformed data matrices \( X, Y \) and \( p_i \) represents a row of the transform matrix \( P \).

So the rows of \( P \) form a new basis for the columns of \( X \); 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 separate sources, filter data, and reveal 'hidden' dynamics.

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

\[ PX = Y \Rightarrow y_i = \begin{bmatrix} p_i x_i \\ \vdots \\ p_i x_i \end{bmatrix} \]

where \( x_i, y_i \) represent columns of the source and transformed data matrices \( X, Y \) and \( p_i \) represents a row of the transform matrix \( P \).

So the rows of \( P \) form a new basis for the columns of \( X \); 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 separate sources, filter data, and reveal 'hidden' dynamics.

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

\[ PX = Y \Rightarrow y_i = \begin{bmatrix} p_1 x_i \\ \vdots \\ p_i x_i \end{bmatrix} \]

where \( x_i, y_i \) represent columns of the source and transformed data matrices \( X, Y \) and \( p_i \) represents a row of the transform matrix \( P \).

So the rows of \( P \) form a new basis for the columns of \( X \); 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 $x, y$ are **un-correlated** if their sample covariance is 0:

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

and $\sigma_{x,x}^2 > 0$ is simply the variance of $x$. 
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 \(x, y\) are **un-correlated** if their **sample covariance** is 0:

\[
\sigma_{x,y}^2 = \frac{1}{n-1} (x - \bar{x})(y - \bar{y})^T = 0
\]

and \(\sigma_{x,x}^2 > 0\) is simply the variance of \(x\).
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**:

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

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

To minimize redundancy in the new basis, the sampled data should be \textit{un-correlated} in the new basis.

**Definition**

\textit{n} random samples \( y_1, y_2, \ldots \) are \textit{un-correlated} if their sample covariance matrix is diagonal:

\[
S_Y = \frac{1}{n-1} (Y - \bar{Y}1) (Y - \bar{Y}1)^T = \begin{bmatrix}
a_1 & 0 \\
0 & \ddots \\
0 & a_n
\end{bmatrix}
\]

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

\[
S_Y = \frac{1}{n-1} (Y - \bar{Y}) (Y - \bar{Y})^T = \begin{bmatrix}
a_1 & 0 \\
0 & \ddots \\
0 & a_n
\end{bmatrix}
\]

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

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

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

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

- $S_Y$ is always a square, symmetric matrix.
- Diagonal elements are the individual **variances** of $y_1, y_2, \ldots$.
- Off-diagonal elements are the **covariances** of $y_1, y_2, \ldots$.
- $S_Y$ quantifies the correlation between all possible pairings of $\{y_1, \ldots, y_n\}$.
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 separation problem.
- Large variances in \( X \) correspond to important dynamics in the underlying system.
- The principal components are orthogonal.

**Definition**

A function \( T(x) \) is said to be a **sufficient statistic** for the random variable \( x \) if the conditional probability distribution of \( x \), given \( T(x) \), 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 \textbf{sufficient statistics} for the underlying separation problem.
- Large variances in $X$ correspond to important dynamics in the underlying system.
- The principal components are orthogonal.

\begin{definition}
A function $T(x)$ is said to be a \textbf{sufficient statistic} for the random variable $x$ if the conditional probability distribution of $x$, given $T(x)$, is not a function of any unknown distribution parameters:

$$P(X = x | T(x), \theta \in \Omega) = P(X = x | T(x))$$
\end{definition}
Principal Component Analysis

Summary of assumptions:

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

**Definition**

A function $T(x)$ is said to be a **sufficient statistic** for the random variable $x$ if the conditional probability distribution of $x$, given $T(x)$, 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 separation 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:

\[
P(X = x \mid T(\mathbf{x}), \theta \in \Omega) = P(X = x \mid T(\mathbf{x}))
\]
Principal Component Analysis

Summary of assumptions:

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

Definition

A function $T(x)$ is said to be a sufficient statistic for the random variable $x$ if the conditional probability distribution of $x$, given $T(x)$, 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** $\tilde{X}$ is normalized with zero mean.
- Seek an orthonormal matrix $P$ (where $Y = PX$) such that $S_Y = \frac{1}{n-1} YY^T$ is diagonalized. The rows of $P$ will be the principal components of $X$.

So:

$$S_Y = \frac{1}{n-1} YY^T = P \left( \frac{1}{n-1} XX^T \right) P^T$$

symmetric!
Principal Component Analysis

Solving for the PCs:

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

So:

$$S_Y = \frac{1}{n-1} YY^T = P \left( \frac{1}{n-1} XX^T \right) P^T$$

symmetric!
Principal Component Analysis

Solving for the PCs:

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

So:

$$S_Y = \frac{1}{n-1}YY^T = P \left( \frac{1}{n-1}XX^T \right) P^T$$

symmetric!
Principal Component Analysis

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

**Definition**

The **singular value decomposition** of a real $m \times n$ matrix $X$ is given by:

$$X = U \Sigma V^T$$

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

- The singular values $\sigma$ (elements of $\Sigma$) are 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 $\sigma$s; consider as the principal components only the vectors corresponding to $\sigma$s larger than the chosen threshold.
Principal Component Analysis

**Definition**

The *singular value decomposition* of a real $m \times n$ matrix $X$ is given by:

$$X = U \Sigma V^T$$

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

- The singular values $\sigma$ (elements of $\Sigma$) are 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 $\sigma$s; consider as the principal components only the vectors corresponding to $\sigma$s larger than the chosen threshold.
Principal Component Analysis

Definition

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

\[
\mathbf{X} = \mathbf{U} \Sigma \mathbf{V}^T
\]

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

- The singular values \( \sigma \) (elements of \( \Sigma \)) are ordered from greatest to least, and each correspond to a basis vector in \( \mathbf{U} \) and \( \mathbf{V} \).
- Dimension reduction: choose a minimum acceptable value for the \( \sigma \)s; consider as the principal components only the vectors corresponding to \( \sigma \)s 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:

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

\[ = \frac{1}{n-1} \left( X^T \right)^T X^T = \frac{1}{n-1} XX^T = S_x \]

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

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

For PCA, note that:

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

\[ = \frac{1}{n-1} (X^T)^T X^T = \frac{1}{n-1} XX^T = S_X \]

- So the matrix \( V \) given by the SVD of \( Z \) will give the eigenvectors of \( S_X \), which are the principal components! Therefore \( P = V^T \).

- Once \( P \) is found, the data can be transformed: \( Y = PX \)
Principal Component Analysis

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

For PCA, note that:

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

\[
= \frac{1}{n-1} \left( X^T \right)^T X^T = \frac{1}{n-1} XX^T = S_X
\]

- So the matrix \( V \) given by the SVD of \( Z \) will give the eigenvectors of \( S_X \), which are the principal components! Therefore \( P = V^T \).

- Once \( P \) is found, the data can be transformed: \( Y = PX \)
Principal Component Analysis

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

For PCA, note that:

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

\[
= \frac{1}{n-1} (X^T)^T X^T = \frac{1}{n-1} XX^T = S_X
\]

- So the matrix \( V \) given by the SVD of \( Z \) will give the eigenvectors of \( S_X \), which are the principal components! Therefore \( P = V^T \).

- Once \( P \) is found, the data can be transformed: \( Y = PX \).
Principal Component Analysis

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

For PCA, note that:

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

\[ = \frac{1}{n-1} X^T X = \frac{1}{n-1} \] 

So the matrix \( V \) given by the SVD of \( Z \) will give the eigenvectors of \( S_X \), which are the principal components! Therefore \( P = V^T \).

Once \( P \) is found, the data can be transformed: \( Y = PX \)
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 introduce a new component, so that the data is 3 dimensional: $x_3 = x_1 - x_2$

- $x_3$ 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:

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

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

$$V^T = \begin{bmatrix}
0.61 & 0.77 & -0.16 \\
0.54 & -0.25 & 0.80 \\
-0.577 & 0.577 & 0.577
\end{bmatrix}$$
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
⇒ Projection onto the first two PCs is sufficient to characterize the data!
Principal Component Analysis

2-Source Audio Example
Principal Component Analysis

2-Source Audio Example
In the previous two examples, PCA was not successful in completely separating 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 separate sources, filter data, and reveal 'hidden' dynamics. ICA also begins by assuming that the transformation to the new basis is linear:

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

where $x_i, y_i$ represent columns of the source and transformed data matrices $X, Y$ and $w_i$ represents a row of the transform matrix $W$. So the rows of $W$ form a new basis for the columns of $X$; 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 separate sources, filter data, and reveal 'hidden' dynamics. ICA also begins by assuming that the transformation to the new basis is linear:

\[ WX = Y \Rightarrow y_i = \begin{bmatrix} w_i x_i \\ \vdots \\ w_i x_i \end{bmatrix} \]

where \( x_i, y_i \) represent columns of the source and transformed data matrices \( X, Y \) and \( w_i \) represents a row of the transform matrix \( W \). So the rows of \( W \) form a new basis for the columns of \( X \); 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 separate sources, filter data, and reveal 'hidden' dynamics.

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

\[
WX = Y \Rightarrow y_i = \begin{bmatrix}
    w_i x_i \\
    \vdots \\
    w_i x_i
\end{bmatrix}
\]

where \(x_i, y_i\) represent columns of the source and transformed data matrices \(X, Y\) and \(w_i\) represents a row of the transform matrix \(W\). So the rows of \(W\) form a new basis for the columns of \(X\); 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 separate sources, filter data, and reveal 'hidden' dynamics.
ICA also begins by assuming that the transformation to the new basis is linear:

\[ WX = Y \Rightarrow y_i = \begin{bmatrix} w_i x_i \\ \vdots \\ w_i x_i \end{bmatrix} \]

where \( x_i, y_i \) represent columns of the source and transformed data matrices \( X, Y \) and \( w_i \) represents a row of the transform matrix \( W \). So the rows of \( W \) form a new basis for the columns of \( X \); 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 separate sources, filter data, and reveal 'hidden' dynamics.
ICA also begins by assuming that the transformation to the new basis is linear:

\[ WX = Y \Rightarrow y_i = \begin{bmatrix} w_i x_i \\ \vdots \\ w_i x_i \end{bmatrix} \]

where \( x_i, y_i \) represent columns of the source and transformed data matrices \( X, Y \) and \( w_i \) represents a row of the transform matrix \( W \).

So the rows of \( W \) form a new basis for the columns of \( X \); 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 separate sources, filter data, and reveal 'hidden' dynamics. ICA also begins by assuming that the transformation to the new basis is linear:

\[ WX = Y \Rightarrow y_i = \begin{bmatrix} w_i x_i \\ \vdots \\ w_i x_i \end{bmatrix} \]

where \( x_i, y_i \) represent columns of the source and transformed data matrices \( X, Y \) and \( w_i \) represents a row of the transform matrix \( W \). So the rows of \( W \) form a new basis for the columns of \( X \); 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 characteristic of the underlying dynamics of the system.
- Measures based on higher order statistics ($>2$) are assumed to be necessary to separate 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 characteristic of the underlying dynamics of the system.
- Measures based on higher order statistics ($> 2$) are assumed to be necessary to separate 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 characteristic of the underlying dynamics of the system.
- Measures based on higher order statistics ($> 2$) are assumed to be necessary to separate 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 characteristic of the underlying dynamics of the system.
- Measures based on higher order statistics ($> 2$) are assumed to be necessary to separate 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 characteristic of the underlying dynamics of the system.
- Measures based on higher order statistics ($> 2$) are assumed to be necessary to separate 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 characteristic of the underlying dynamics of the system.
- Measures based on higher order statistics ($>2$) are assumed to be necessary to separate 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 characteristic of the underlying dynamics of the system.
- Measures based on higher order statistics ($> 2$) are assumed to be necessary to separate 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 likelihood
  - 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 $W, Y$ such that $Y = W^{-1}X$ and each row of $Y$ maximizes some high-order measure of independence.

- Typical perspectives:
  - Maximum likelihood
  - 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 $W, Y$ such that $Y = W^{-1}X$ and each row of $Y$ maximizes some high-order measure of independence.

Typical perspectives:
- Maximum likelihood
- 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 \( W, Y \) such that \( Y = W^{-1}X \) and each row of \( Y \) maximizes some high-order measure of independence.

Typical perspectives:
- Maximum likelihood
- 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 $W, Y$ such that $Y = W^{-1}X$ and each row of $Y$ maximizes some high-order measure of independence.

Typical perspectives:
- Maximum likelihood
- 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 $W, Y$ such that $Y = W^{-1}X$ and each row of $Y$ maximizes some high-order measure of independence.

Typical perspectives:
- Maximum likelihood
- 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 likelihood
- 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 $\mu_n$ and variance $\sigma_n^2$. Define $S_N^2 = \sum_{i=1}^{n} \sigma_i^2$. If for some $\delta > 0$ the expectations $E \left[ X_k^{2+\delta} \right]$ are finite for every $k \in \mathbb{N}$ and the condition $\lim_{N \to \infty} \frac{1}{S_N^{2+\delta}} \sum_{i=1}^{N} E \left[ (X_n - \mu_n)^{2+\delta} \right] = 0$ 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 argument:

- The sum of any group of independent random variables is 'more gaussian' than 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 separate the sources in $\mathbf{X}$!
Independent Component Analysis

Heuristic argument:

- The sum of any group of independent random variables is 'more gaussian' than 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 separate the sources in $X$!
Independent Component Analysis

Heuristic argument:

- The sum of any group of independent random variables is 'more gaussian' than 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 separate the sources in $\mathbf{X}$!
Independent Component Analysis

**Definition**

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

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

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

**Definition**

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

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

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

**Definition**

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

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

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

**Definition**

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

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

- Kurtosis is a measure of 'peakedness' and thickness of tails for a distribution.
- Note that if $x$ is gaussian:
  $$\kappa(x) = 3(E[y^2])^2 - 3(E[y^2])^2 = 0$$
- So, simultaneously minimizing $|\kappa(Y_1)|, \ldots, |\kappa(Y_m)|$ or $(\kappa(Y_1))^2, \ldots, (\kappa(Y_m))^2$ 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 criterion:
- 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 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 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 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 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 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 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:

- \( J(X) \approx \frac{1}{12} E[y^3]^2 + \frac{1}{48} \kappa(y)^2 \)
  - Same problems as in the case of just using kurtosis!

- \( J(X) \approx \sum_{i=1}^{n} k_i (E[G_i(y)] - E[G_i(v)])^2 \), 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:

- \( J(X) \approx \frac{1}{12} E[y^3]^2 + \frac{1}{48} \kappa(y)^2 \)
  - Same problems as in the case of just using kurtosis!

- \( J(X) \approx \sum_{i=1}^{n} k_i (E[G_i(y)] - E[G_i(v)])^2 \), 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:

- $J(X) \approx \frac{1}{12} E[y^3]^2 + \frac{1}{48} \kappa(y)^2$
  - Same problems as in the case of just using kurtosis!

- $J(X) \approx \sum_{i=1}^{n} k_i (E[G_i(y)] - E[G_i(v)])^2$, 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 $1 \leq \alpha_1 \leq 2$
  - $G(u) = -\exp (-u^2/2)$
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 $1 \leq \alpha_1 \leq 2$
  - $G(u) = -\exp (-u^2/2)$
Independent Component Analysis

Typically:

- All of the $G_i$ are the same function.
- Very good results have been demonstrated using:
  - $G(u) = \frac{1}{\alpha_1} \log \left[ \cosh (\alpha_1 u) \right]$, 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 \( w_1 \), based on maximizing negentropy: The maxima of the negentropy approximations of \( w_1^T X \) are obtained at certain optima of \( \mathbb{E} \left[ w_1^T X \right] \). KKT conditions: optima of \( \mathbb{E} \left[ w_1^T X \right] \) under the constraint \( \mathbb{E} \left[ (w_1^T X)^2 \right] = \|w_1\| = 1 \) are obtained at points where:

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

So, the Jacobian has the form:

\[
J (w_1) = \mathbb{E} \left[ XX^T g' \left( w_1^T X \right) \right] - \beta I
\]
Independent Component Analysis

If we use the approximation
\[
E \left[ XX^T g' \left( w_1^T X \right) \right] \approx E \left[ XX^T \right] E \left[ g' \left( w_1^T X \right) \right] = E \left[ g' \left( w_1^T X \right) \right] I,
\]
we get the Newton-Raphson iteration:

\[
w_1 = w_1 - \left( E \left[ Xg \left( w_1^T X \right) \right] - \beta w_1 \right) / \left( E \left[ g' \left( w_1^T X \right) \right] - \beta \right)
\]

Dividing both sides by \( \beta - E \left[ g' \left( w_1^T X \right) \right] \) gives:

\[
w_1^+ = E \left[ Xg \left( w_1^T X \right) \right] - E \left[ Xg' \left( w_1^T X \right) \right] w_1
\]

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 $w_1$.
- Compute $w_1^+ = E \left[ X g \left( w_1^T X \right) \right] - E \left[ X g' \left( w_1^T X \right) \right] w_1$
- Set $w_1 = w_1^+ / \|w_1^+\|$.
- Repeat until convergence.
Independent Component Analysis

The computation of a single Independent Component (FastICA):

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

The computation of a single Independent Component (FastICA):

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

The computation of a single Independent Component (FastICA):

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

The computation of a single Independent Component (FastICA):

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

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

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

\[
w_{p+1} = w_{p+1} - \sum_{j=1}^{p} w_{p+1}^T w_j w_j
\]

- Renormalize \( w_{p+1} \):

\[
w_{p+1} = \frac{w_{p+1}}{\sqrt{w_{p+1}^T w_{p+1}}}
\]

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 \( \mathbf{w}_{p+1}^T \mathbf{w}_j \mathbf{w}_j \) for \( j = 1, \ldots, p \):

\[
\mathbf{w}_{p+1} = \mathbf{w}_{p+1} - \sum_{j=1}^{p} \mathbf{w}_{p+1}^T \mathbf{w}_j \mathbf{w}_j
\]

• Renormalize \( \mathbf{w}_{p+1} \):

\[
\mathbf{w}_{p+1} = \frac{\mathbf{w}_{p+1}}{\sqrt{\mathbf{w}_{p+1}^T \mathbf{w}_{p+1}}}
\]

Repeat until \( p = m \)
Independent Component Analysis

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

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

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

- Renormalize $w_{p+1}$:

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

Repeat until $p = m$
Independent Component Analysis

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

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

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

- Renormalize $w_{p+1}$:

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

Repeat until $p = m$
Independent Component Analysis

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

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

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

- Renormalize $w_{p+1}$:

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

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 $X$ 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 $X$ 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 $X$ 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, $\sigma$, gives:

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

The following 5 images were retrieved from the mixed sources using ICA:
Aapo Hyvriinen and Erkki Oja.
A fast Fixed-Point algorithm for independent component analysis.

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.

Pierre Comon.
Independent component analysis, a new concept?

E. Oja.
Questions?