PCA and ICA

Julie Nutini

Machine Learning Reading Group

February 6, 2017

What is PCA?

- Principal Component Analysis (PCA) is a statistical procedure that allows better analysis and interpretation of unstructured data.

What is PCA?

- Principal Component Analysis (PCA) is a statistical procedure that allows better analysis and interpretation of unstructured data.
- Uses an orthogonal linear transformation to convert a set of observations to a new coordinate system that maximizes the variance.

What is PCA?

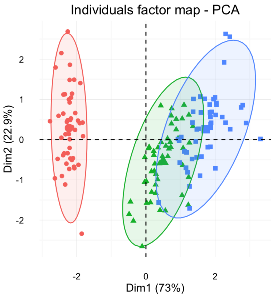
- Principal Component Analysis (PCA) is a statistical procedure that allows better analysis and interpretation of unstructured data.
- Uses an orthogonal linear transformation to convert a set of observations to a new coordinate system that maximizes the variance.
- The new coordinates are called principal components.

What is PCA?

- Principal Component Analysis (PCA) is a statistical procedure that allows better analysis and interpretation of unstructured data.
- Uses an orthogonal linear transformation to convert a set of observations to a new coordinate system that maximizes the variance.
- The new coordinates are called principal components.

Example:

- Fit n-dimensional ellipsoid to data.
- By omitting axis with smallest variance (smallest principal component), we lose smallest amount of info.

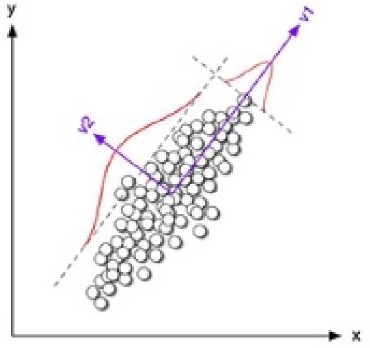


Principal Component Analysis (PCA) aka...

- Signal processing: discrete Kosambi-Karhunen-Loève transform (KLT)
- Multivariate quality control: the Hotelling transform
- Mechanical engineering: proper orthogonal decomposition (POD)
- Linear algebra: singular value decomposition (SVD) of X (Golub and Van Loan, 1983)
- Linear algebra: eigenvalue decomposition (EVD) of $X^T X$
- Psychometrics: factor analysis, Eckart-Young theorem (Harman, 1960), or Schmidt-Mirsky theorem
- Meteorological science: empirical orthogonal functions (EOF)
- Noise and vibration: empirical eigenfunction decomposition (Sirovich, 1987), empirical component analysis (Lorenz, 1956), quasiharmonic modes (Brooks et al., 1988), spectral decomposition
- Structural dynamics: empirical modal analysis
- ...

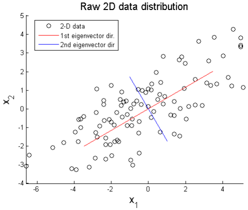
Applications of PCA

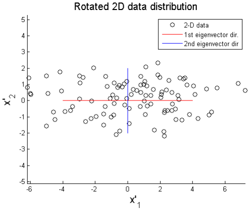
- Dimension construction
- Feature extraction
- Data visualization
- Image compression
- Medical imaging
- Lossy data compression
- ...



Application: 2D Data Analysis

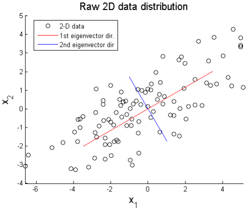
- Data matrix X can be rotated to align principal axes with x and y axis.

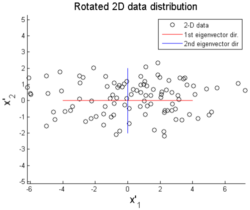




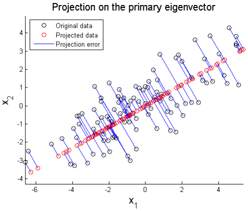
Application: 2D Data Analysis

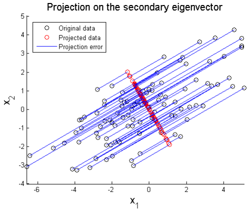
- Data matrix X can be rotated to align principal axes with x and y axis.





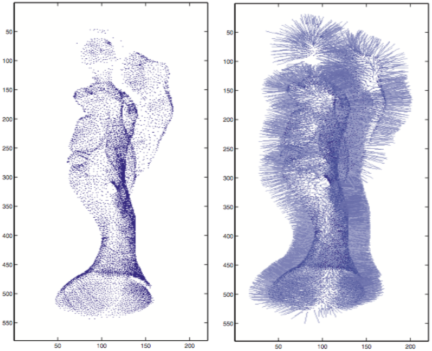
- Project X on the primary and secondary principal direction.





Application: Data Visualization

- Scattered set of points, presumably forms coherent surface.
- Display point cloud data in a pleasing way.



(a) Point cloud. (b) With normals.

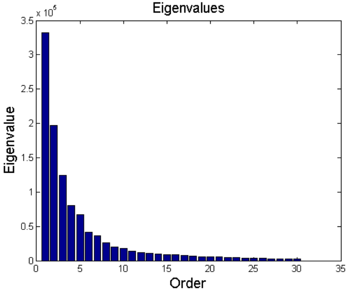
Figure 4.9. *Example* 4.18: *a point cloud representing* (a) *a surface in three-dimensional space, and* (b) *together with its unsigned normals.*

Application: Image Compression

- Effectively represent image with limited number of principal components.

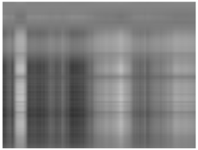
Application: Image Compression

- Effectively represent image with limited number of principal components.



- Do not know # of principal components needed for successful reconstruction.

Application: Image Compression



(a) 1 principal component

(b) 5 principal component

(c) 9 principal component

(d) 13 principal component

(e) 17 principal component

(f) 21 principal component

(g) 25 principal component

(h) 29 principal component

The Problem

Let X be a D-dimensional random vector with covariance matrix S.

The Problem

Let X be a D-dimensional random vector with covariance matrix S.

- **Problem**: Consecutively find the unit vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_D$ such that

$$Y_i = X^T \mathbf{u}_i$$

satisfies:

1. $\text{var}(Y_1)$ is the maximum.
2. $\text{var}(Y_2)$ is the maximum subject to $\text{cov}(Y_2, Y_1) = 0$.
3. $\text{var}(Y_k)$ is the maximum subject to $\text{cov}(Y_k, Y_i) = 0$, where $k = 3, 4, \ldots, D$ and $k > i$.

The Solutions

- Let $(\lambda_i, \mathbf{u}_i)$ be the pairs of eigenvalues and eigenvectors of the covariance matrix S such that

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D (\geq 0)$$

and

$$\|u_i\|_2 = 1, \quad \text{for all } 1 \leq i \leq D.$$

The Solutions

- Let $(\lambda_i, \mathbf{u}_i)$ be the pairs of eigenvalues and eigenvectors of the covariance matrix S such that

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D (\geq 0)$$

and

$$\|u_i\|_2 = 1, \quad \text{for all } 1 \leq i \leq D.$$

- Then $\mathrm{var}(Y_i) = \lambda_i$ for $1 \leq i \leq D$.

→ The principal components of X are the eigenvectors of S.

  → The variance will be a maximum when we set $\mathbf{u}_1$ to the eigenvector having the largest eigenvalue.

The Solutions

- Let $(\lambda_i, \mathbf{u}_i)$ be the pairs of eigenvalues and eigenvectors of the covariance matrix S such that

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D (\geq 0)$$

and

$$\|u_i\|_2 = 1, \quad \text{for all } 1 \leq i \leq D.$$

- Then $\text{var}(Y_i) = \lambda_i$ for $1 \leq i \leq D$.

→ The principal components of X are the eigenvectors of S.

→ The variance will be a maximum when we set $\mathbf{u}_1$ to the eigenvector having the largest eigenvalue.

→ The proportion of variance each eigenvector represents is given by the ratio of the given eigenvalue to the sum of all the eigenvalues.

Restrictions of PCA

- Linear, nonparametric analysis that cannot incorporate prior knowledge.

Restrictions of PCA

- Linear, nonparametric analysis that cannot incorporate prior knowledge.
- Important that variance can be used to differentiate/imply similarity.

Restrictions of PCA

- Linear, nonparametric analysis that cannot incorporate prior knowledge.
- Important that variance can be used to differentiate/imply similarity.
- If the given data set is nonlinear or multimodal distribution, PCA fails to provide meaningful data reduction.

Restrictions of PCA

- Linear, nonparametric analysis that cannot incorporate prior knowledge.
- Important that variance can be used to differentiate/imply similarity.
- If the given data set is nonlinear or multimodal distribution, PCA fails to provide meaningful data reduction.
- To incorporate the prior knowledge of data to PCA, researchers have proposed dimension reduction techniques as extensions of PCA:
  - e.g., kernel PCA, multilinear PCA, and independent component analysis (ICA).

General: How to do PCA?

Goal: Find the axes of the ellipse (i.e., the principal components).

General: How to do PCA?

Goal: Find the axes of the ellipse (i.e., the principal components).

Consider a data matrix X.

1. Subtract the sample mean from each column of X (data has mean 0).

General: How to do PCA?

Goal: Find the axes of the ellipse (i.e., the principal components).

Consider a data matrix X.

1. Subtract the sample mean from each column of X (data has mean 0).
2. Compute covariance matrix of the data.

General: How to do PCA?

Goal: Find the axes of the ellipse (i.e., the principal components).

Consider a data matrix X.

1. Subtract the sample mean from each column of X (data has mean 0).
2. Compute covariance matrix of the data.
3. Calculate the eigenvalues/corresponding eigenvectors of covariance matrix,

$$Xv = \lambda v$$

* Xv does not change direction of v.

General: How to do PCA?

Goal: Find the axes of the ellipse (i.e., the principal components).

Consider a data matrix X.

1. Subtract the sample mean from each column of X (data has mean 0).
2. Compute covariance matrix of the data.
3. Calculate the eigenvalues/corresponding eigenvectors of covariance matrix,

$$Xv = \lambda v$$

   * Xv does not change direction of v.

4. Orthogonalize the set of eigenvectors, normalize each to unit vectors.

There are two main formulations of PCA:

Formulations of PCA

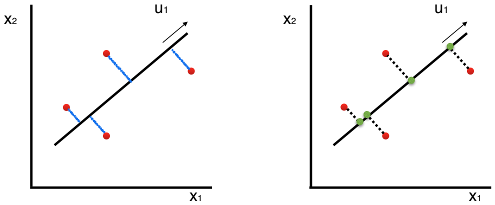
There are two main formulations of PCA:

- **Maximum variance formulation**: The orthogonal projection of the data onto a lower dimensional linear space (principal subspace) such that the variance of the projected data is maximized.

Formulations of PCA

There are two main formulations of PCA:

- **Maximum variance formulation**: The orthogonal projection of the data onto a lower dimensional linear space (principal subspace) such that the variance of the projected data is maximized.
- **Minimum-error formulation**: The linear projection that minimizes the average projection cost, defined as the mean squared distance between the data points and their projections.



Maximum Variance Formulation

Goal: Project data onto space having dimensionality $M < D$ while maximizing variance of projected data.

Maximum Variance Formulation

Goal: Project data onto space having dimensionality $M < D$ while maximizing variance of projected data.

- Consider a data set of observations $\{x_n\}$ where $n = 1, \dots, N$.
- Each x_n is a Euclidean variable with dimensionality D.

Maximum Variance Formulation

Goal: Project data onto space having dimensionality $M < D$ while maximizing variance of projected data.

- Consider a data set of observations $\{x_n\}$ where $n = 1, \ldots, N$.
- Each x_n is a Euclidean variable with dimensionality D.
- Assume projecting onto a one-dimensional space (M = 1).

Maximum Variance Formulation

Goal: Project data onto space having dimensionality $M < D$ while maximizing variance of projected data.

- Consider a data set of observations $\{x_n\}$ where $n = 1, \ldots, N$.
- Each x_n is a Euclidean variable with dimensionality D.
- Assume projecting onto a one-dimensional space (M = 1).
- Define the direction of this space using $\mathbf{u}_1$.
- Assume $\mathbf{u}_1$ is a unit vector ($\mathbf{u}_1^T \mathbf{u}_1 = 1$).

Maximum Variance Formulation

- The mean of the projected data is $\mathbf{u}_1^T \bar{x}$ where $\bar{x}$ is the sample set mean

$$\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n$$

Maximum Variance Formulation

- The mean of the projected data is $\mathbf{u}_1^T \bar{x}$ where $\bar{x}$ is the sample set mean

$$\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n$$

- The variance of the projected data is given by

$$\frac{1}{N} \sum_{n=1}^{N} \left(\mathbf{u}_1^T x_n - \mathbf{u}_1^T \bar{x} \right)^2 = \mathbf{u}_1^T S \mathbf{u}_1$$

where S is the covariance matrix of the data,

$$S = \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^T.$$

Maximum Variance Formulation

- To maximize the variance, we solve the following constrained problem

$$\underset{\mathbf{u}_1}{\text{maximize}} \quad \mathbf{u}_1^T S \mathbf{u}_1 \quad \text{s.t.} \quad \mathbf{u}_1^T \mathbf{u}_1 = 1$$

Maximum Variance Formulation

- To maximize the variance, we solve the following constrained problem

$$\underset{\mathbf{u}_1}{\text{maximize}} \quad \mathbf{u}_1^T S \mathbf{u}_1 \quad \text{s.t.} \quad \mathbf{u}_1^T \mathbf{u}_1 = 1$$

- The Lagrangian of this problem is given by

$$\mathcal{L}(\mathbf{u}_1, \lambda_1) = \mathbf{u}_1^T S \mathbf{u}_1 + \lambda_1 (1 - \mathbf{u}_1^T \mathbf{u}_1).$$

Maximum Variance Formulation

- To maximize the variance, we solve the following constrained problem

$$\underset{\mathbf{u}_1}{\text{maximize}} \quad \mathbf{u}_1^T S \mathbf{u}_1 \quad \text{s.t.} \quad \mathbf{u}_1^T \mathbf{u}_1 = 1$$

- The Lagrangian of this problem is given by

$$\mathcal{L}(\mathbf{u}_1, \lambda_1) = \mathbf{u}_1^T S \mathbf{u}_1 + \lambda_1 (1 - \mathbf{u}_1^T \mathbf{u}_1).$$

- Differentiating with respect to $\mathbf{u}_1$, we have a stationary point when

$$S\mathbf{u}_1 = \lambda_1 \mathbf{u}_1.$$

Maximum Variance Formulation

- By left-multiplying by $\mathbf{u}_1$ and using $\mathbf{u}_1^T \mathbf{u}_1 = 1$, we have

$$\mathbf{u}_1^T S \mathbf{u}_1 = \lambda_1.$$

Maximum Variance Formulation

- By left-multiplying by $\mathbf{u}_1$ and using $\mathbf{u}_1^T\mathbf{u}_1 = 1$, we have

$$\mathbf{u}_1^T S \mathbf{u}_1 = \lambda_1.$$

- Thus, the maximum variance will occur when we set $\mathbf{u}_1$ to the eigenvector having the largest eigenvalue λ_1.

Maximum Variance Formulation

- By left-multiplying by $\mathbf{u}_1$ and using $\mathbf{u}_1^T\mathbf{u}_1 = 1$, we have

$$\mathbf{u}_1^T S \mathbf{u}_1 = \lambda_1.$$

- Thus, the maximum variance will occur when we set $\mathbf{u}_1$ to the eigenvector having the largest eigenvalue λ_1.
- Additional principal components can be defined in an incremental fashion.
- A similar problem can be formed for the minimum error formulation.
    - Solution is in terms of the $D - M$ smallest eigenvalues of the eigenvectors that are orthogonal to the principal subspace.

SVD and PCA

- The singular value decomposition of a matrix $A \in \mathbb{R}^{m \times n}$ is given by

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

where

  - $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices (i.e., $U^T U = U U^T = I$)
  - $D \in \mathbb{R}^{m \times n}$ diagonal matrix with the singular values of A along the diagonal.

SVD and PCA

- The singular value decomposition of a matrix $A \in \mathbb{R}^{m \times n}$ is given by

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

  where

  - $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices (i.e., $U^T U = UU^T = I$)
  - $D \in \mathbb{R}^{m \times n}$ diagonal matrix with the singular values of A along the diagonal.

- The largest variance is in the direction of the first column of U (the first principal component)
- The largest variance on the subspace orthogonal to the first principal component is the direction of the second column of U
- $\ldots$

SVD and PCA

- Therefore,

$$B = U^T A = \Sigma V^T$$

represents a better alignment than the given A in terms of variance differentiation.

SVD and PCA

- Therefore,

$$B = U^T A = \Sigma V^T$$

represents a better alignment than the given A in terms of variance differentiation.

- Covariance matrix of A is a positive semi-definite matrix,

$$C = AA^T = U\Sigma\Sigma^T U^T$$

and the eigenvectors are the columns of U (namely, the singular vectors which are the principal components).

SVD and PCA

- Therefore,

$$B = U^T A = \Sigma V^T$$

  represents a better alignment than the given A in terms of variance differentiation.
- Covariance matrix of A is a positive semi-definite matrix,

$$C = AA^T = U\Sigma\Sigma^T U^T$$

  and the eigenvectors are the columns of U (namely, the singular vectors which are the principal components).
- Application of PCA with respect to SVD:
  - Solving almost singular linear systems
    - If the problem is too ill-conditioned, then regularize it.

Computing the Principal Components

Eigenvalues:

- QR algorithm: costs $O(D^3)$.
- Power Method: Finds first M principal components, costs $O(MD^2)$.

Computing the Principal Components

Eigenvalues:

- QR algorithm: costs $O(D^3)$.
- Power Method: Finds first M principal components, costs $O(MD^2)$.

Singular values:

- SVD costs $O(m^2n + mn^2 + n^3)$ for general matrix A of dimension $m \times n$.

Computing the Principal Components

Eigenvalues:

- QR algorithm: costs $O(D^3)$.
- Power Method: Finds first M principal components, costs $O(MD^2)$.

Singular values:

- SVD costs $O(m^2n + mn^2 + n^3)$ for general matrix A of dimension $m \times n$.

$\rightarrow$ When D is large, a direct application of PCA will be computationally infeasible.

PCA for High-Dimensional Data

Let X be an $(N \times D)$-dimensional centered matrix.

- The nth row is $(x_n - \bar{x})^T$.
- The covariance matrix can be written as $S = \frac{1}{N} X^T X$.

PCA for High-Dimensional Data

- The corresponding eigenvector equation becomes

$$\frac{1}{N} X^T X \mathbf{u}_i = \lambda_i \mathbf{u}_i.$$

PCA for High-Dimensional Data

- The corresponding eigenvector equation becomes
$$\frac{1}{N}X^T X\mathbf{u}_i = \lambda_i \mathbf{u}_i.$$

- Multiply both sides by X,
$$\frac{1}{N}XX^T(X\mathbf{u}_i) = \lambda_i(X\mathbf{u}_i).$$

PCA for High-Dimensional Data

- The corresponding eigenvector equation becomes

$$\frac{1}{N}X^T X\mathbf{u}_i = \lambda_i \mathbf{u}_i.$$

- Multiply both sides by X,

$$\frac{1}{N}XX^T(X\mathbf{u}_i) = \lambda_i(X\mathbf{u}_i).$$

- Let $v_i = X\mathbf{u}_i$ to get

$$\frac{1}{N}XX^T\mathbf{v}_i = \lambda_i\mathbf{v}_i,$$

  which is the eigenvector equation for the $N \times N$ matrix $\frac{1}{N}XX^T$.

PCA for High-Dimensional Data

- The corresponding eigenvector equation becomes

$$\frac{1}{N} X^T X \mathbf{u}_i = \lambda_i \mathbf{u}_i.$$

- Multiply both sides by X,

$$\frac{1}{N} X X^T (X \mathbf{u}_i) = \lambda_i (X \mathbf{u}_i).$$

- Let $v_i = X\mathbf{u}_i$ to get

$$\frac{1}{N} X X^T \mathbf{v}_i = \lambda_i \mathbf{v}_i,$$

  which is the eigenvector equation for the $N \times N$ matrix $\frac{1}{N} X X^T$.
- This has the same $N - 1$ eigenvalues as the original covariance matrix.
- We can solve the eigenvalue problem for cost of $O(N^3)$.

Discussion

- PCA is a statistical procedure that uses an orthogonal linear transformation to reduce the dimension of a dataset while maximizing the variance.

Discussion

- PCA is a statistical procedure that uses an orthogonal linear transformation to reduce the dimension of a dataset while maximizing the variance.
- PCs of a dataset X are the eigenvectors of its covariance matrix.

Discussion

- PCA is a statistical procedure that uses an orthogonal linear transformation to reduce the dimension of a dataset while maximizing the variance.
- PCs of a dataset X are the eigenvectors of its covariance matrix.
- Formulated as a maximum variance problem or a minimum error problem.

Discussion

- PCA is a statistical procedure that uses an orthogonal linear transformation to reduce the dimension of a dataset while maximizing the variance.
- PCs of a dataset X are the eigenvectors of its covariance matrix.
- Formulated as a maximum variance problem or a minimum error problem.
- Transformation for high-dimensional data.
  - Allows you to find principal components in smaller subspace.

Discussion

- PCA is a statistical procedure that uses an orthogonal linear transformation to reduce the dimension of a dataset while maximizing the variance.
- PCs of a dataset X are the eigenvectors of its covariance matrix.
- Formulated as a maximum variance problem or a minimum error problem.
- Transformation for high-dimensional data.
  - Allows you to find principal components in smaller subspace.
- Extensions:
  - Probabilistic PCA
    - Maximum likelihood PCA, EM algorithm for PCA, Bayesian PCA, Factor analysis
  - Kernel PCA

Independent Component Analysis

- PCA focuses on models with latent variables based on linear-Gaussian distributions.
  - The PCs represent a rotation of the coordinate system in data space.
  - Data distribution in the new coordinates is uncorrelated.

Independent Component Analysis

- PCA focuses on models with latent variables based on linear-Gaussian distributions.
    - The PCs represent a rotation of the coordinate system in data space.
    - Data distribution in the new coordinates is uncorrelated.
    - This is a necessary condition for independence, but not a sufficient condition.

Independent Component Analysis

- Independent Component Analysis (ICA):
  - Similar to PCA, finds a new basis to represent data.

Independent Component Analysis

- Independent Component Analysis (ICA):
  - Similar to PCA, finds a new basis to represent data.
  - Computational method for separating multivariate signal into additive subcomponents that are maximally independent.

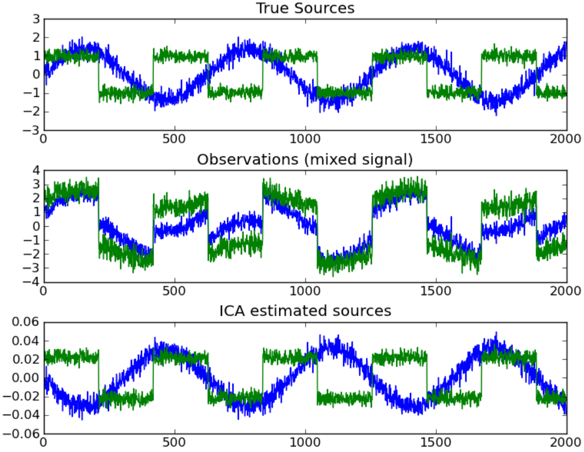
Independent Component Analysis

- Independent Component Analysis (ICA):
  - Similar to PCA, finds a new basis to represent data.
  - Computational method for separating multivariate signal into additive subcomponents that are maximally independent.
  - Observed variables are linear combination of the latent variables.

Independent Component Analysis

- Independent Component Analysis (ICA):
  - Similar to PCA, finds a new basis to represent data.
  - Computational method for separating multivariate signal into additive subcomponents that are maximally independent.
  - Observed variables are linear combination of the latent variables.
  - Assumes subcomponents are non-Gaussian signals and are statistically independent.

Example: blind source separation



- ICA is used to recover the sources.

Example: blind source separation

- Consider some data $s \in \mathbb{R}^n$ that is generated via n independent sources

$$x = As,$$

where A is an unknown matrix (mixing matrix), x received signal.

Example: blind source separation

- Consider some data $s \in \mathbb{R}^n$ that is generated via n independent sources

$$x = As,$$

where A is an unknown matrix (mixing matrix), x received signal.

- Repeated observations gives a data set $\{x^{(i)}, i = 1, \ldots, m\}$.

Example: blind source separation

- Consider some data $s \in \mathbb{R}^n$ that is generated via n independent sources

$$x = As,$$

where A is an unknown matrix (mixing matrix), x received signal.

- Repeated observations gives a data set $\{x^{(i)}, i = 1, \ldots, m\}$.
- **Goal**: Recover $s^{(i)}$.

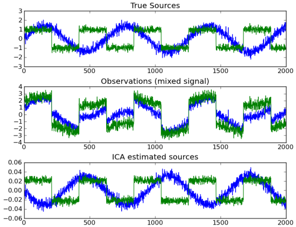
- Given no prior knowledge about the sources or the mixing matrix, some inherent ambiguities in A are impossible to recover.

- Permutation of the sources is ambiguous.

ICA

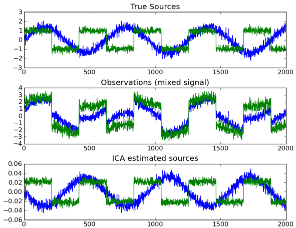
- Given no prior knowledge about the sources or the mixing matrix, some inherent ambiguities in A are impossible to recover.

- Permutation of the sources is ambiguous.
- Scalings of $W = A^{-1}$ cannot be recovered.
  - Might not matter depending on the application.
- We cannot determine the order of the independent components.



- Given no prior knowledge about the sources or the mixing matrix, some inherent ambiguities in A are impossible to recover.

- Permutation of the sources is ambiguous.
- Scalings of $W = A^{-1}$ cannot be recovered.
    - Might not matter depending on the application.
- We cannot determine the order of the independent components.



- These are the ONLY ambiguities assuming the sources s_i are non-Gaussian.
- As long as the data is non-Gaussian, we can recover the n independent sources.

ICA Algorithm (Bell and Sejnowski)

- Suppose the distribution of each source s_i is given by a density p_s.
- The joint distribution of the sources s is given by

$$p(s) = \prod_{i=1}^{n} p_s(s_i).$$

ICA Algorithm (Bell and Sejnowski)

- Suppose the distribution of each source s_i is given by a density p_s.
- The joint distribution of the sources s is given by

$$p(s) = \prod_{i=1}^{n} p_s(s_i).$$

→ By modelling the joint distribution as a product of the marginal, we capture the assumption that the sources are independent.

ICA Algorithm (Bell and Sejnowski)

- Suppose the distribution of each source s_i is given by a density p_s.
- The joint distribution of the sources s is given by

$$p(s) = \prod_{i=1}^{n} p_s(s_i).$$

→ By modelling the joint distribution as a product of the marginal, we capture the assumption that the sources are independent.

- This implies the following density on $x = As = W^{-1}s$:

$$p(x) = \prod_{i=1}^{n} p_s(w_i^T x) \cdot |W|.$$

- Need to specify a density for the individual sources p_s.

ICA Algorithm

- We need to specify a cdf for it that slowly increases from 0 to 1.
- Reasonable default: the sigmoid function

$$g(s) = \frac{1}{(1 + e^{-s})}.$$

- This yields $p_s(s) = g'(s)$.

ICA Algorithm

- We need to specify a cdf for it that slowly increases from 0 to 1.
- Reasonable default: the sigmoid function

$$g(s) = \frac{1}{(1 + e^{-s})}.$$

- This yields $p_s(s) = g'(s)$.
- Given a training set $\{x^{(i)}, i = 1, \ldots, m\}$, the log likelihood for our parameter matrix W is

$$\ell(W) = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \log g'(w_j^T x^{(i)}) + \log |W| \right).$$

ICA Algorithm

- Maximizing this in terms of W, we derive a stochastic gradient ascent learning rule for training example $x^{(i)}$:

$$W := W + \alpha \left(\begin{bmatrix} 1 - 2g(w_1^T x^{(i)}) \\ 1 - 2g(w_2^T x^{(i)}) \\ \vdots \\ 1 - 2g(w_n^T x^{(i)}) \end{bmatrix} {x^{(i)}}^T + (W^T)^{-1} \right)$$

where α is the learning rate.

ICA Algorithm

- Maximizing this in terms of W, we derive a stochastic gradient ascent learning rule for training example $x^{(i)}$:

$$W := W + \alpha \left(\begin{bmatrix} 1 - 2g(w_1^T x^{(i)}) \\ 1 - 2g(w_2^T x^{(i)}) \\ \vdots \\ 1 - 2g(w_n^T x^{(i)}) \end{bmatrix} {x^{(i)}}^T + (W^T)^{-1} \right)$$

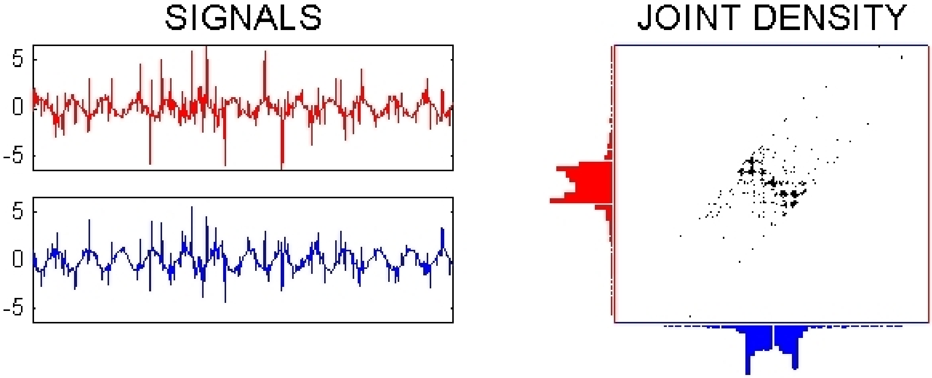
  where α is the learning rate.

- After the algorithm converges, we compute $s^{(i)} = Wx^{(i)}$ to recover the original sources.

FastICA Algorithm

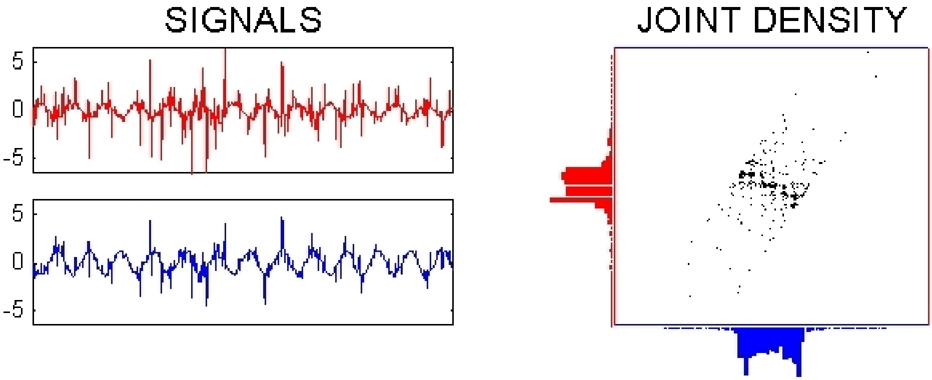
- FastICA [http://research.ics.aalto.fi/ica/fastica/]
- Implements the fast fixed-point algorithm for ICA and projection pursuit.
- Can download (for R, C++, Python and Matlab)

FastICA Algorithm



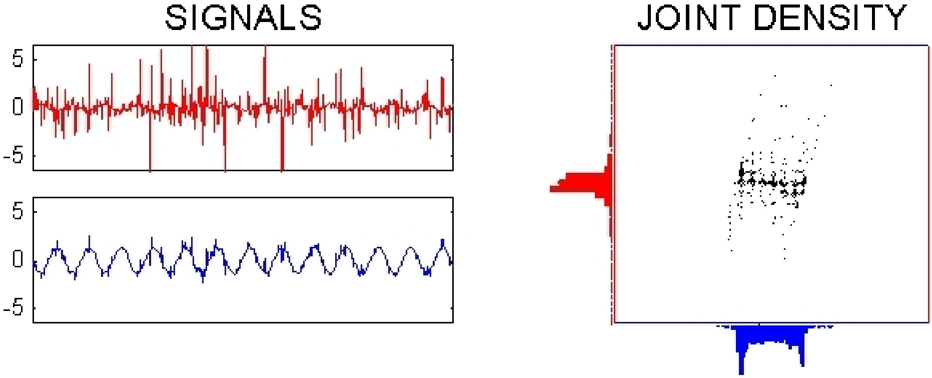
Separated signals after 1 step of FastICA

FastICA Algorithm



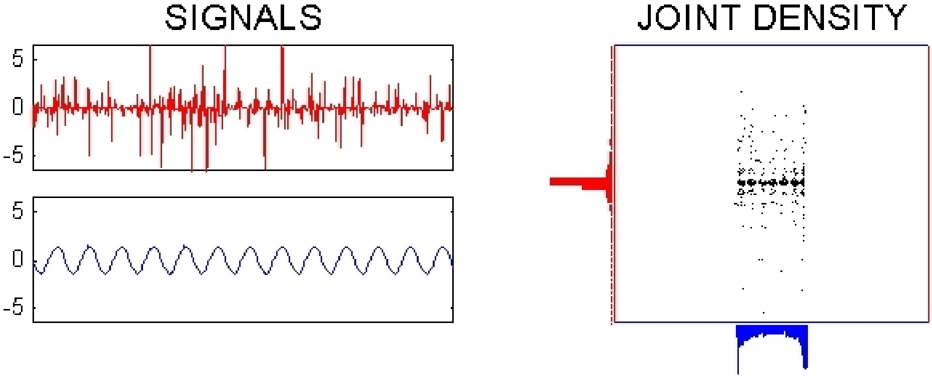
Separated signals after 2 steps of FastICA

FastICA Algorithm

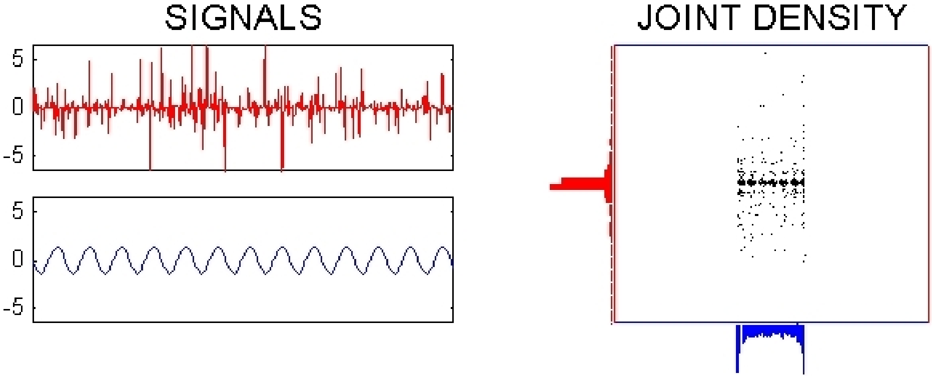


Separated signals after 3 steps of FastICA

FastICA Algorithm



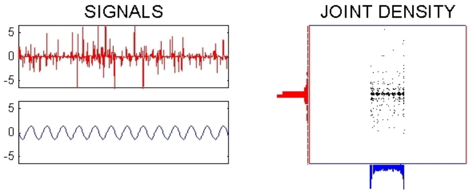
Separated signals after 4 steps of FastICA



Separated signals after 5 steps of FastICA

FastICA Algorithm

- The source signals were sinusoidal and impulsive noise.
- The joint density is the product of the marginal densities.
  - Definition of independence.



Separated signals after 5 steps of FastICA

Discussion

- ICA is a statistical and computational technique used to reveal hidden factors that underlie sets of random variables, measurements, or signals.

Discussion

- ICA is a statistical and computational technique used to reveal hidden factors that underlie sets of random variables, measurements, or signals.
- Data assumed to be linear combinations of some unknown latent variables.
- Latent variables are assumed to be non-Gaussian and independent.

Discussion

- ICA is a statistical and computational technique used to reveal hidden factors that underlie sets of random variables, measurements, or signals.
- Data assumed to be linear combinations of some unknown latent variables.
- Latent variables are assumed to be non-Gaussian and independent.
- ICA finds these independent components.

Discussion

- ICA is a statistical and computational technique used to reveal hidden factors that underlie sets of random variables, measurements, or signals.
- Data assumed to be linear combinations of some unknown latent variables.
- Latent variables are assumed to be non-Gaussian and independent.
- ICA finds these independent components.
- Stochastic gradient ascent learning rule for training example $x^{(i)}$.
- FastICA
- ...

Thank you!

- U. M. Ascher and C. Greif, *A First Course in Numerical Methods*, SIAM, 2011.
- C. M. Bishop. *Pattern Recognition and Machine Learning*, Springer, 2006.
- A. Hyvärinen. What is Independent Component Analysis? `https://www.cs.helsinki.fi/u/ahyvarin/whatisica.shtml`
- S. Jang. Basics and Examples of Principal Component Analysis (PCA), slecture, 2014 `https://www.projectrhea.org/rhea/index.php/PCA_Theory_Examples`.
- A. Ng. Independent Component Analysis, CS299 Lecture Notes.
- FastICA, `http://research.ics.aalto.fi/ica/fastica/`.