Unsupervised learning I – Dimensionality reduction

Sebastian Stich

2022-06-27

CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

About me:
Working on *Algorithms* for Machine Learning, Distributed and Decentralized Optimization, Federated Learning, etc.
We study *aspects* such as efficiency, optimalitly, robustness, privacy, fairness, etc.
Master theses or research internships available.

Outline

Main references

- Bishop - Chapter 12
- EML - Chapter 14
- KPCA - B. Schölkopf, A. Smola and K.R. Müller(1998). Non linear component analysis as kernel eigenvalue problem. Neural Computation 10, 1299–1319.
- ICA - A. Hyvärinen and E. Oja (2000). Independent component analysis: algorithms and applications. Neural Networks 13, 411-430.

Outline

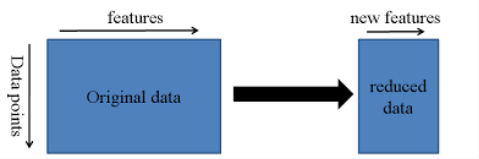
Dimensionality reduction I

Dimensionality reduction methods allow us to reduce the dimensionality of the input space while at the same time:

- Use only relevant data, i.e, remove irrelevant/noisy/correlated components minimizing the loss of **relevant** information
- Discover good combinations of input variables (features).

These characteristics allow us to simplify the ML stage since:

- We reduce the number of parameters in the prediction model (alleviating the **curse of dimensionality**)
- We obtain a compact data representation (crucial for large datasets)



Motivational example

$$\mathbf{X}^\top = \overbrace{\left[\mathbf{x}_1, \ldots, \underbrace{\mathbf{x}_n}_{D \text{ features}}, \ldots \mathbf{x}_N\right]}^{N \text{ samples}} \qquad (D \times N)$$

Example Dataset

- $N = 2389$ municipalities in Switzerland
- $D = 245$ votes Jan 1981– Dec 2011
- $(\mathbf{x}_n)_d \in [0, 1]$ percentage of 'yes' votes

- How we find the most relevant representation of this data in 1 (or 2) dimensions?
- For voting data, we could try to (manually) classify each vote as 'left'/'right' and to estimate the general tendency for each city. For other datasets we might not have such a good understanding of the data. Is there an unsupervised method?

Dimensionality reduction II

Dimensionality Reduction: Construction of a mapping $\phi : \mathcal{X} \to \mathbb{R}^K$, where the dimensionality K of the target space is usually much smaller than that of the input space $\mathcal{X}$. Generally, the mapping should preserve properties of the input space $\mathcal{X}$ e.g. distances.

Why should we do dimensionality reduction?

- **Manifold assumption:** The internal degrees of freedom are much smaller than the number of measured features. That is, data lies along a low-dimensional structure in feature space.
- **Visualization:** interpretation of data in high dimensions is difficult—embeddings in two or three dimensions can provide visual insight.
- **Data compression:** compress the data but retain most of the information.

Examples of approaches

Supervised dimensionality reduction:

- Linear discriminant analysis (LDA)

Unsupervised dimensionality reduction:

- Principal Components Analysis (PCA),
- Kernel PCA,
- Independent Component Analysis (ICA).

All the last methods correspond to eigenvalue problems.

Outline

Principal component Analysis (PCA) I

Assumptions: Consider we have a dataset with N observations where each data, $\mathbf{x}^{(n)} \in \mathbb{R}^D$, belongs to a D dimensional space. For the sake of simplicity, let's consider that these data have **zero mean**.

Goal: find a new representation over a low dimensional space ($K < D$) by applying a linear transformation of the data of the form:

$$\mathbf{X}' = \mathbf{X}\mathbf{U}$$

where

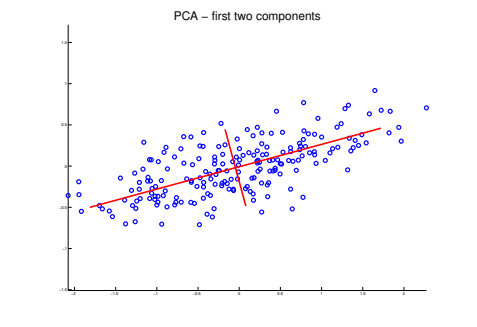
$$\mathbf{U} = \begin{bmatrix} u_{1,1} \dots u_{1,K} \\ \vdots \ddots \vdots \\ u_{D,1} \dots u_{D,K} \end{bmatrix}$$

Each column of $\mathbf{U}$, called as **principal component**, will provide a new dimension of the data in the projected space.

PCA II

PCA aims to find **U** such that (2 interpretations):

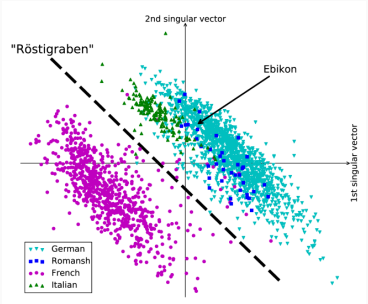
1. **Variance interpretation**: the first K principal components of the data contain most of the variance.
2. **Approximation interpretation**: the principal K-components span the K-dimensional affine subspace which yields the best approximation of the data.

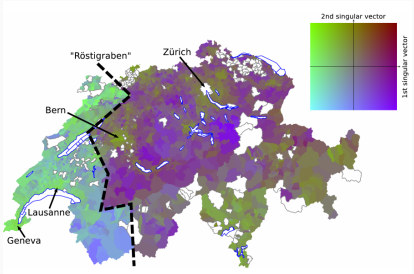


(Image by Prof. Hein)

Motivational example revisited

Example with voting data, $K = 2$.





[Etter et al, Mining Democracy, 2014.]

Mathematical formulation I

We aim at maximizing the variance of the projected data $\mathbf{X}'$, which has covariance $\mathbf{C}_{\mathbf{X}'\mathbf{X}'} = \mathbf{X}'^{\top}\mathbf{X} = \mathbf{U}^{\top}\mathbf{X}^{\top}\mathbf{X}\mathbf{U} = \mathbf{U}^{\top}\mathbf{C}_{\mathbf{X}\mathbf{X}}\mathbf{U}$. This yields to the following optimization problem:

$$\mathbf{U} = \underset{\mathbf{U}}{\operatorname{argmax}} \operatorname{Tr}\left\{\mathbf{U}^{\top}\mathbf{C}_{\mathbf{X}\mathbf{X}}\mathbf{U}\right\}$$

$$\text{s.t.} \quad \mathbf{U}^{\top}\mathbf{U} = \mathbf{I}$$

For simplicity, we work with matrix notation, i.e.,

- Input data is denoted by $\mathbf{X}$ ($N \times D$), which is assumed to be centered (i.e, to have **zero mean**).
- Data covariance is given by $\mathbf{C}_{\mathbf{X}\mathbf{X}} = \mathbf{X}^{\top}\mathbf{X}$. Note that this definition of the covariance matrix is assuming that the data have zero mean.
- The transformed data is denoted by $\mathbf{X}' = (\mathbf{U}^{\top}\mathbf{X}^{\top})^{\top}$ ($N \times K$).

Mathematical formulation II

Solution: Let's start considering $K = 1$ and solve:

$$\mathbf{u}_1 = \underset{\mathbf{u}_1}{\operatorname{argmax}}\ \mathbf{u}_1^\top \mathbf{C_{XX}} \mathbf{u}_1$$

$$\text{s.t.}\ \ \mathbf{u}_1^\top \mathbf{u}_1 = 1$$

If we apply Lagrange multipliers, we can include the constraint into functional by means of the lagrange multiplier λ_1 and we arrive to an unconstraint problem

$$\mathbf{u}_1 = \underset{\mathbf{u}_1}{\operatorname{argmax}}\ \mathbf{u}_1^\top \mathbf{C_{XX}} \mathbf{u}_1 + \lambda_1 \left(1 - \mathbf{u}_1^\top \mathbf{u}_1\right)$$

by making its derivate equal to zero, we obtain that the optimum solution has to satisfiy

$$\mathbf{C_{XX}} \mathbf{u}_1 = \lambda_1 \mathbf{u}_1$$

which indicates that $\mathbf{u}_1$ must be an eigenvector of the covariance matrix $\mathbf{C_{XX}}$ and λ_1 is its associated eigenvalue.

Mathematical formulation III

If we left-multiply the expression $\mathbf{C_{XX}}\mathbf{u}_1 = \lambda_1\mathbf{u}_1$ by $\mathbf{u}_1^\top$ and make use of $\mathbf{u}_1^\top\mathbf{u}_1 = 1$, we obtain that

$$\mathbf{u}_1^\top\mathbf{C_{XX}}\mathbf{u}_1 = \lambda_1$$

that is, the variance of the projected data by the eigenventor $\mathbf{u}_1$ is equal to its associated eigenvalue λ_1.

Observations:

- We obtain the maximum projected variance with a single projection, if we set this first principal component $\mathbf{u}_1$ as the eigenvector with the largest eigenvalue λ_1.
- We can define additional principal components by choosing each new direction in an incremental fashion by chosing the eigenvector with the next highest eigenvalue so that the projected variance is maximized.

Mathematical formulation IV

In general, we aim to obtain the first K projections of the PCA algorithm satisfying that

$$\mathbf{C_{XX}}\mathbf{u}_i = \lambda_i \mathbf{u_i}$$

for $i = 1, \ldots, K$, and $\lambda_i \mathbf{u_i}$ correspond to the top eigenvalues. That is, the solution to such a problem is the projection matrix $\mathbf{U}$, which consists of the first eigenvectors of $\mathbf{C_{XX}}$ (i.e., those associated with largest eigenvalues). So, the PCA algorithm works as follows:

- First, find the first principal componente, $\mathbf{u}_1$, so that $\mathbf{u}_1^\top \mathbf{x}^\top$ projects the maximum variance of the data. Next, find $\mathbf{u}_2$ so that $\mathbf{u}_2^\top \mathbf{x}^\top$ is able to obtain the projection with maximum variance, etc.
- So, computing $K < D$ new features, we remove the directions with less variance.

We recall that the eigenvectors, $\{\mathbf{u}_i\}_{i=1}^K$, of $\mathbf{C_{XX}}$ determine an uncorrelated orthonormal basis, i.e.,

$$\langle \mathbf{u}_i, \mathbf{C_{XX}}\mathbf{u}_j \rangle = \lambda_i \delta_{ij}, \quad i, j = 1, \ldots, K.$$

Approximation interpretation

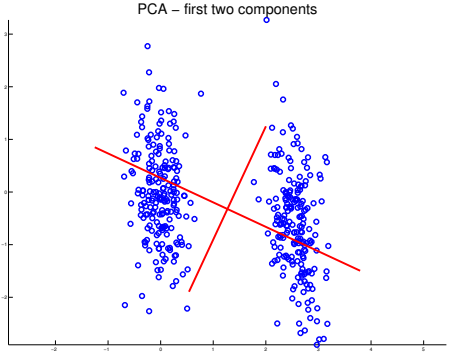
PCA can alternatively seen as a method that minimized the squared error between the lower rank projection of the data $\mathbf{Z}$ ($N \times D$) and the original data $\mathbf{X}$, i.e.,

$$\sum_{n=1}^{N} \|\mathbf{x}_n - \mathbf{z}_n\|^2 = \|\mathbf{X} - \mathbf{Z}\|^2$$

where notice that $rank(\mathbf{Z}) \leq K$.

Excersise! Show that the projection $\mathbf{Z}$ with $rank(\mathbf{Z}) \leq K$ that minimizes the above objecive is $Z = \mathbf{XUU}^\top$ with $\mathbf{U}$ being the solution to the optimization problem in Slide 11 (i.e., to maximize the variance of the low-dimensional representation of the data).
Refer to Chapter 12.1.2 of Bishop for further details, Chapter 14.5 of EML, and/or additional notes in CMS.

Illustration



Red lines represent the 2 principal directions in the data. The length of the red lines: $4\sqrt{\lambda_i}$, where λ_i is the i-th eigenvalue of $\mathbf{C_{xx}}$.

Whitening I

Note that the eigenvectors, $\{\mathbf{u}_i\}_{i=1}^K$, of $\mathbf{C_{XX}}$ determine an uncorrelated orthonormal basis, i.e.,

$$\langle \mathbf{u}_i, \mathbf{C_{XX}} \mathbf{u}_j \rangle = \lambda_i \delta_{ij}, \quad i, j = 1, \ldots, K.$$

For Gaussian data: $p(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\det \mathbf{C_{XX}}|^{\frac{1}{2}}} e^{-\frac{1}{2}\mathbf{x}^T \mathbf{C_{XX}}^{-1}\mathbf{x}}$,
we get in new coordinates $\mathbf{x}'$ defined as,

$$\mathbf{x}' = \mathbf{C_{XX}}^{-\frac{1}{2}}\mathbf{x} = \sum_{i=1}^{K} \frac{1}{\sqrt{\lambda_i}} \mathbf{u}_i \mathbf{u}_i^T \mathbf{x},$$

which implies that the dimentions of $\mathbf{x}$ are independent and equally distributed, i.e., $\mathbf{C_{X'X'}} = \mathbf{I}$. This process is called **whitening** and if often used as a preprocessing step in ML.

Whitening II

Whitening are three concatenated operations:

1. **centering** the data to enforce zero mean—equivalent to a translation in $\mathbb{R}^D$;
2. **projection onto (all) principal components** (rotation)—equivalent to a change from the initial basis to the basis spanned by the eigenvectors of $\mathbf{C_{XX}}$; and,
3. **rescaling**—one rescales each axis by the square-root of the corresponding eigenvalue such that $\mathbf{C_{X'X'}} = \mathbf{I}$.

Observations

PCA is an unsupervised algorithm! Thus if we use it as a pre-procesisng step prior to train a prediction model, we may wonder:

- Which direction will PCA consider as the most relevant features for the prediction task?
- Clearly, when dealing with supervised problems, we should consider the labels to obtain good features.

Solution: we can find in the literature supervised feature extraction approaches, such as, Partial Least Squares (PLS), Orthogonal Partial Least Squares (OPLS) and Canonical Correlation Analysis (CCA). These methods try to maximize either the covariance or correlation between the projected data and the labels, thus finding discriminative proyections.

PCA is linear method! Simple, easy to understand, robust to overfitting problems. However, it lacks expressive power.

Solution: non-linear extension by means of kernel methods!

Outline

Kernel PCA

Kernel PCA aims to find the projections that maximize the variance of the data in the **feature space** resulting from applying a basis function $\phi(\cdot)$, i.e., ,

$$\mathbf{U} = \underset{\mathbf{U}}{\operatorname{argmax}} \operatorname{Tr}\left\{\mathbf{U}^\top \mathbf{\Phi}^\top \mathbf{\Phi} \mathbf{U}\right\} \qquad \text{s.t.} \quad \mathbf{U}^\top \mathbf{U} = \mathbf{I}$$

Kernel PCA

Kernel PCA aims to find the projections that maximize the variance of the data in the **feature space** resulting from applying a basis function $\phi(\cdot)$, i.e., ,

$$\mathbf{U} = \underset{\mathbf{U}}{\operatorname{argmax}} \operatorname{Tr}\left\{\mathbf{U}^\top \mathbf{\Phi}^\top \mathbf{\Phi} \mathbf{U}\right\} \qquad \text{s.t.} \quad \mathbf{U}^\top \mathbf{U} = \mathbf{I}$$

Applying:

1. The Representer Theorem ($\mathbf{U} = \mathbf{\Phi}^\top \mathbf{A}$)

$$\mathbf{A} = \underset{\mathbf{A}}{\operatorname{argmax}} \operatorname{Tr}\left\{\mathbf{A}^\top \mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{A}\right\} \qquad \text{s.t.} \quad \mathbf{A}^\top \mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{A} = \mathbf{I}$$

2. The kernel trick, so we replace the dot products in the feature space, $\mathbf{\Phi}\mathbf{\Phi}^\top$, by the kernel matrix

$$\mathbf{A} = \underset{\mathbf{A}}{\operatorname{argmax}} \operatorname{Tr}\left\{\mathbf{A}^\top \mathbf{K}\mathbf{K}\mathbf{A}\right\} \qquad \text{s.t.} \quad \mathbf{A}^\top \mathbf{K}\mathbf{A} = \mathbf{I}$$

Kernel PCA

Kernel PCA aims to find the projections that maximize the variance of the data in the **feature space** resulting from applying a basis function $\phi(\cdot)$, i.e., ,

$$\mathbf{U} = \underset{\mathbf{U}}{\operatorname{argmax}} \operatorname{Tr}\left\{\mathbf{U}^\top \mathbf{\Phi}^\top \mathbf{\Phi} \mathbf{U}\right\} \qquad \text{s.t.} \;\; \mathbf{U}^\top \mathbf{U} = \mathbf{I}$$

Applying:

1. The Representer Theorem ($\mathbf{U} = \mathbf{\Phi}^\top \mathbf{A}$)

$$\mathbf{A} = \underset{\mathbf{A}}{\operatorname{argmax}} \operatorname{Tr}\left\{\mathbf{A}^\top \mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{A}\right\} \qquad \text{s.t.} \;\; \mathbf{A}^\top \mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{A} = \mathbf{I}$$

2. The kernel trick, so we replace the dot products in the feature space, $\mathbf{\Phi}\mathbf{\Phi}^\top$, by the kernel matrix

$$\mathbf{A} = \underset{\mathbf{A}}{\operatorname{argmax}} \operatorname{Tr}\left\{\mathbf{A}^\top \mathbf{K}\mathbf{K}\mathbf{A}\right\} \qquad \text{s.t.} \;\; \mathbf{A}^\top \mathbf{K}\mathbf{A} = \mathbf{I}$$

For a positive semidefinite kernel matrix, we can solve the above optimization problem via the following eigenvalue problem $\mathbf{K}\mathbf{a} = \lambda \mathbf{a}$, where $\mathbf{A}$ consist of the first eigenvectors of $\mathbf{K}$ (i.e., those associated with largest eigenvalues).

Illustration

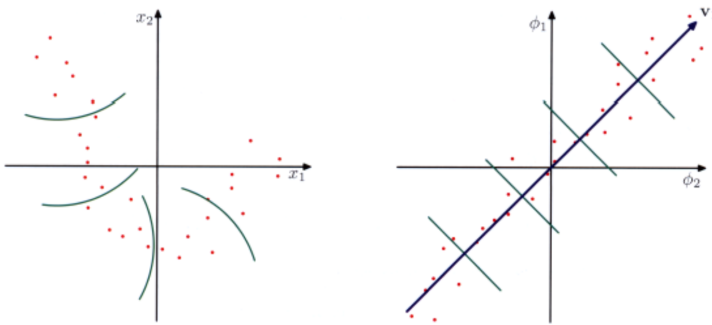


Figure 12.16 Schematic illustration of kernel PCA. A data set in the original data space (left-hand plot) is projected by a nonlinear transformation $\phi(\mathbf{x})$ into a feature space (right-hand plot). By performing PCA in the feature space, we obtain the principal components, of which the first is shown in blue and is denoted by the vector $\mathbf{v}_1$. The green lines in feature space indicate the linear projections onto the first principal component, which correspond to nonlinear projections in the original data space. Note that in general it is not possible to represent the nonlinear principal component by a vector in $\mathbf{x}$ space.

Interpretation

Standard-PCA:

$$\mathbf{C_{XX}u} = \lambda \mathbf{u}, \quad \implies \quad \frac{1}{n}\sum_{i=1}^{n} \langle \mathbf{x}_i, \mathbf{u} \rangle \ \mathbf{x}_i = \lambda \, \mathbf{u}.$$

I.e., all eigenvectors lie in the span of the data points.

Kernel-PCA: there exist a mapping function $\phi : \mathcal{X} \to \mathcal{H}_k$ such that

$$\mathbf{C_{\Phi\Phi}} = \frac{1}{n}\sum_{j=1}^{n} \phi(\mathbf{x}_j)\phi(\mathbf{x}_j)^\top.$$

As in PCA we want to find the eigenvectors of $\mathbf{C_{\Phi\Phi}}$,

$$\mathbf{C_{\Phi\Phi}} v = \lambda \mathbf{u} \quad \implies \quad \frac{1}{n}\sum_{i=1}^{n} \langle \phi(\mathbf{x}_i), \mathbf{u} \rangle_{\mathcal{H}_k} \ \phi(\mathbf{x}_i) = \lambda \, \mathbf{u}.$$

I.e., all eigenvectors lie in the span of the **mapped** data points.
Recall that KPCA is a linear method in the Hilbert space!

Outline

Independent Component Analysis (ICA) I

Independent Component Analysis (ICA)

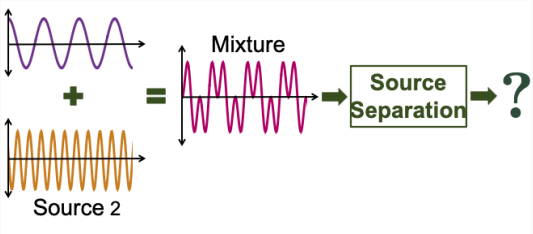
Motivation: cocktail party problem—blind source separation

- k different speakers (sources),

$$s_1(t), \ldots, s_k(t).$$

- d microphones (sensors),

$$x_1(t), \ldots, x_d(t).$$



Assumption: measured signal is linear superposition of sources.

Goal: having only the signal of the microphones, find the sources—determine the **mixing matrix A**, such that

$$\mathbf{x}(t) = \mathbf{A}\,\mathbf{s}(t).$$

Independent Component Analysis (ICA) II

Motivation for ICA

- speakers (sources) are independent of each other.

$$s_1(t), \ldots, s_k(t),$$

in the stochastic sense (source signals are independent random variables),

$$p_s\Big(s_1(t), \ldots, s_k(t)\Big) = \prod_{i=1}^{k} p_{s_i}\big(s_i(t)\big).$$

Find new representation such that components are maximally independent (not only uncorrelacted)!

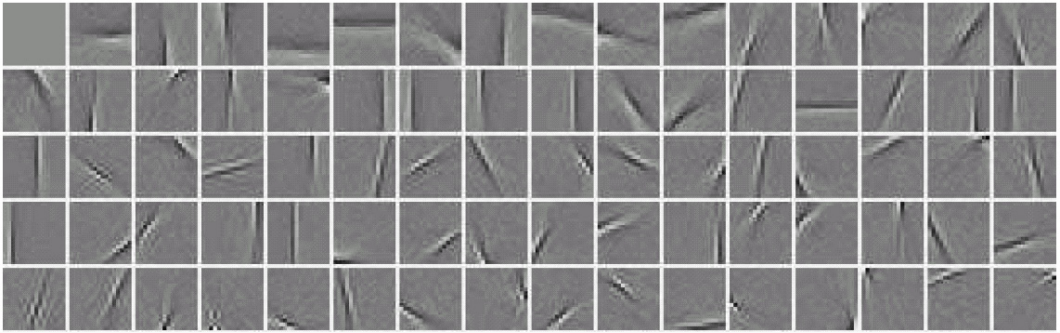
Independent Component Analysis (ICA) III

Application scenarios

- sound (speech, music,...) signals,
- EEG signals,
- natural images (patches),
- financial data,
- ...

Example on Natural Images

ICA for natural images - 16 $\times$ 16 - patches



Basis functions in ICA of natural images. The input window size was 16 $\times$ 16 pixels. These basis functions can be considered as the independent features of images.

(Image by A. Hyvärinen and E. Oja, 2000)

Formulation I

What kind of independent components can we seek for?

- **Non-Gaussian sources:** suppose that $s(t) \in \mathbb{R}^k$ is Gaussian distributed $\implies \mathbf{x} = \mathbf{A}\mathbf{s}$ is again Gaussian distributed,

$$\mathbb{E}[\mathbf{x}\mathbf{x}^\top] = \mathbb{E}[\mathbf{A}\,\mathbf{s}\mathbf{s}^\top\mathbf{A}^\top] = \mathbf{A}\mathbb{E}[\mathbf{s}\,\mathbf{s}^\top]\mathbf{A}^\top = \mathbf{A}\mathbf{I}_k\mathbf{A}^\top = \mathbf{A}\mathbf{A}^\top.$$

  Whitening yields independent components—but not necessarily the true sources $\mathbf{s}(t)$.

- **Sources can be identified only up to rescaling:**

$$\mathbf{x}(t) = \mathbf{A}\,\mathbf{s}(t) = (\mathbf{A}\,\mathbf{D}^{-1})\,(\mathbf{D}\,\mathbf{s}(t)),$$

  where $\mathbf{D}$ is a diagonal matrix—$\mathbf{D}\,\mathbf{s}(t)$ is also independent. W.l.o.g.,

$$\mathbb{E}[\mathbf{s}(t)\,\mathbf{s}(t)^\top] = \mathbf{I}_k.$$

- **Sources cannot be ordered:** Let $\mathbf{P}$ be a permutation matrix, then $\mathbf{P}\mathbf{s}(t)$ is independent, $\mathbf{x}(t) = \mathbf{A}\,\mathbf{s}(t) = (\mathbf{A}\,\mathbf{P}^{-1})\,(\mathbf{P}\mathbf{s}(t))$.

Formulation II

We use whitening as a pre-processing step for ICA to transforms the signal $\mathbf{x}(t)$,

$$\mathbf{x}'(t) \;=\; \mathbf{W}\,\mathbf{x}(t) \;=\; \mathbf{W}\,\mathbf{A}\,\mathbf{s}(t),$$

such it becomes **uncorrelated**,

$$\mathbf{I}_k = \mathbb{E}[\mathbf{x}'(t)\,\mathbf{x}'(t)^\top] = \mathbf{W}\,\mathbf{A}\,\mathbb{E}[\mathbf{s}\,\mathbf{s}^\top]\mathbf{A}^\top\mathbf{W}^\top = \mathbf{W}\,\mathbf{A}\,\mathbf{A}^\top\mathbf{W}^\top,$$

thus, whitening simplifies the problem since the mixing matrix $\mathbf{W}\,\mathbf{A}$ for $\mathbf{x}'(t)$ is orthogonal.

New problem: find the **orthogonal mixing matrix** $\mathbf{B} = \mathbf{W}\,\mathbf{A}$

$$\mathbf{x}'(t) = \mathbf{B}\,\mathbf{s}(t),$$

resp. $\mathbf{B}^\top$ such that $\mathbf{B}^\top\mathbf{x}'(t) = \mathbf{B}^\top\mathbf{B}\mathbf{s}(t) = \mathbf{s}(t)$ is maximally independent.

Formulation II

Steps for ICA:

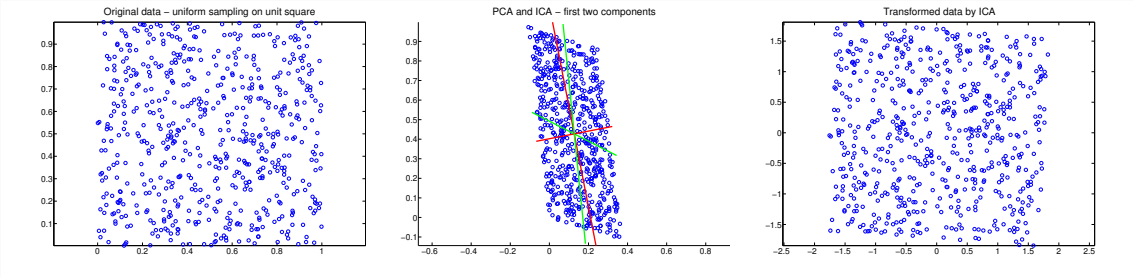
- apply whitening to the data: $\mathbf{x}'(t) = \mathbf{B}\,\mathbf{s}(t)$.
- find orthogonal de-mixing matrix $\mathbf{B}$ such that $\mathbf{s}(t) = \mathbf{B}^\top \mathbf{x}'(t)$ is maximally independent.
  Different criteria:
  - maximize non-gaussianity of $\mathbf{s}(t)$,
  - minimize mutual information $I\big(\{s_k(t)\}_{i=1}^k\ \mathbf{s}(t)\big)$ - mutual information is zero if and only if joint density of $\mathbf{s}(t)$ factorizes into the product of the marginal densities, i.e., is sources are independent.

Problems:

- Joint density of $\mathbf{s}(t)$, as well as mutual information are hard to estimate.
- Instead, one often minimizes higher order correlations e.g. **kurtosis**:

$$\mathrm{kurt}(\mathbf{x}') = \mathbb{E}[\mathbf{x}'^4] - 3\Big(\mathbb{E}[\mathbf{x}'^2]\Big)^2.$$

Illustration



- **Left:** Original sources - individual features are independent $p(x_1, x_2) = p(x_1)p(x_2)$.
- **Middle:** Measured signal—directions of PCA (eigenvectors of covariance matrix) and directions of ICA (columns of estimated mixing matrix) are shown—note that the directions of ICA are **not** orthogonal.
- **Right:** Source signal estimated by ICA—coincides up to rescaling with the original signal.