Principal & Independent Component Analysis

PCA material based on: Section 6.3 of Pattern Recognition of Theodoridis & Koutroumbas available here: http://www.sciencedirect.com.ezproxy.is.ed.ac.uk/science/article/pii/B9781597492720500086
ICA material based on (in part): Section 6.5 of the above and also *Independent Component Analysis: a Tutorial* by Aapo Hyvarinen and Erkki Oja available through the internet.

Principal & Independent Component Analysis

Our final topic concerns decomposing signals into useful low dimensional subsets:

- For feature space selection in classification
- For redundancy reduction
- To avoid overfitting
- For signal separation

The key aim is to find a linear transform of the data that better represents the underlying information

e.g. Fourier transform of an image concentrates information into low frequencies

Both can be cast as approximation problems of the form Y~CX, where Y is the data (in columns) and C and X are matrices, optimizing a Frobenius norm (L2) criterion $|Y-CX|_F$. We saw this in k-means and here different constraints are posed on C.

PCA & Subspace Projections

Introduced by Pearson in 1901 "*On lines and planes of closest fit to a system of points in space.*" (a.k.a. Karhunen-Loéve transform (KLT), or the Hotelling transform,…).

Suppose our data consists of d-dimensional vectors $\mathbf{x}^{(n)} \in R^d$ and we want a low-dimensional approximation for the data.

Let $\boldsymbol{u}_i$ be an orthonormal basis for R^d (i.e. $U^T U = I$) then we can approximate x by:

$$\tilde{\mathbf{x}} = \sum_{i=1}^{M} (\mathbf{u}_i^T \mathbf{x}) \mathbf{u}_i + \sum_{j=M+1}^{d} b_j \mathbf{u}_i$$

b_j is constant

let $z_i = \mathbf{u}_i^T \mathbf{x}$

Subspace Projections

x has d degrees of freedom while $\widetilde{\mathbf{x}}$ has M deg. of freedom.

Principal Component Analysis – choose $\boldsymbol{u}_i$ and b_j to best approximate x in the LSE sense, i,e, minimize E_M:

$$E_M = \frac{1}{2}\sum_{n=1}^{N}\left\|\mathbf{x}^{(\mathbf{n})} - \widetilde{\mathbf{x}}^{(\mathbf{n})}\right\|^2 = \frac{1}{2}\sum_{n=1}^{N}\left\|\mathbf{U}^T\mathbf{x}^{(\mathbf{n})} - \mathbf{U}^T\widetilde{\mathbf{x}}^{(\mathbf{n})}\right\|^2 = \frac{1}{2}\sum_{n=1}^{N}\sum_{j=M+1}^{d}\left(z_j^{(n)} - b_j\right)^2$$

Taking the derivative with respect to b_j *gives:*

$$\sum_{n=1}^{N}\left(z_j^{(n)} - b_j\right) = 0 \quad \Rightarrow \quad b_j = \frac{1}{N}\sum_{n=1}^{N} z_j^{(n)} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{u}_j^T\mathbf{x}^{(\mathbf{n})} = \mathbf{u}_j^T\overline{\mathbf{x}}$$

Subspace Projections (cont.)

So we can write:

$$E_M = \frac{1}{2}\sum_{n=1}^{N}\sum_{j=M+1}^{d}\left(\mathbf{u}_j^T\left(\mathbf{x}^{(n)}-\bar{\mathbf{x}}\right)\right)^2$$

$$= \frac{1}{2}\sum_{j=M+1}^{d}\sum_{n=1}^{N}\mathbf{u}_j^T\left(\mathbf{x}^{(n)}-\bar{\mathbf{x}}\right)\left(\mathbf{x}^{(n)}-\bar{\mathbf{x}}\right)^T\mathbf{u}_j = \frac{1}{2}\sum_{j=M+1}^{d}\mathbf{u}_j^T\mathbf{R}_{\mathbf{x}}\mathbf{u}_j$$

where $\mathbf{R}_{\mathbf{x}}$ is the sample covariance matrix for $\{\mathbf{x}^{(n)}\}$

Minimizing E_M with respect to $\boldsymbol{u}_j$ is satisfied by:

$$\mathbf{R}_{\mathbf{x}}\,\mathbf{u}_j = \lambda\mathbf{u}_j, \text{ for } \mathrm{j}=1,...,\mathrm{M}$$

i.e. the M basis vectors are the principal eigenvectors of the sample covariance matrix.

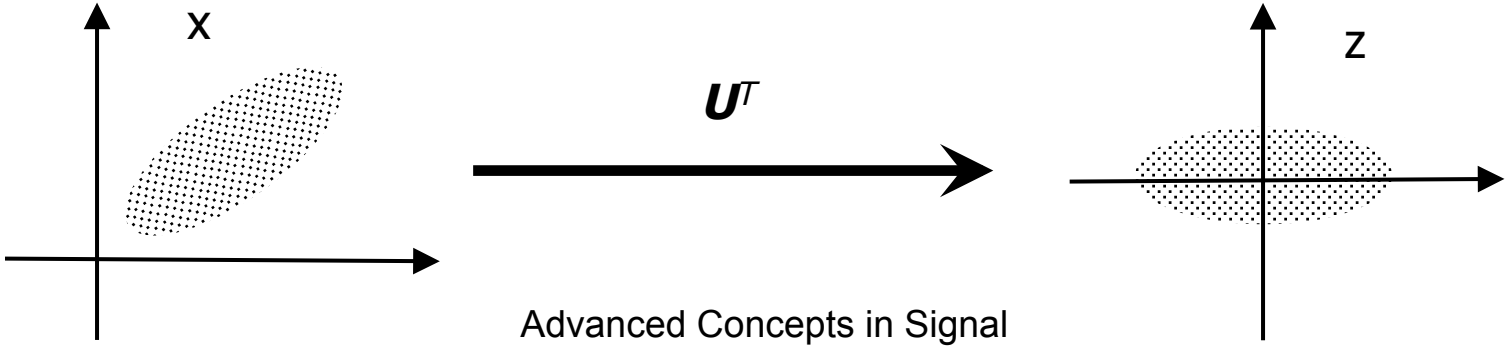
PCA and Spatial Whitening

Transforming with eigenvector basis decorrelates data:

If $\mathbf{R}_x = \mathbf{U}\Lambda\mathbf{U}^T$ is the eigenvalue decomposition of $\mathbf{R}_x$, where Λ is a diagonal matrix. Then $\mathbf{U}$ is a decorrelating matrix :

Let $\mathbf{z} = \mathbf{U}^T(\mathbf{x} - \bar{\mathbf{x}})$ then

$$\mathbf{R}_\mathbf{z} = E\{\mathbf{z}\mathbf{z}^T\} = E\{\mathbf{U}^T(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T\mathbf{U}\} = \mathbf{U}^T\mathbf{R}_x\mathbf{U} = \Lambda$$



PCA example: eigenfaces

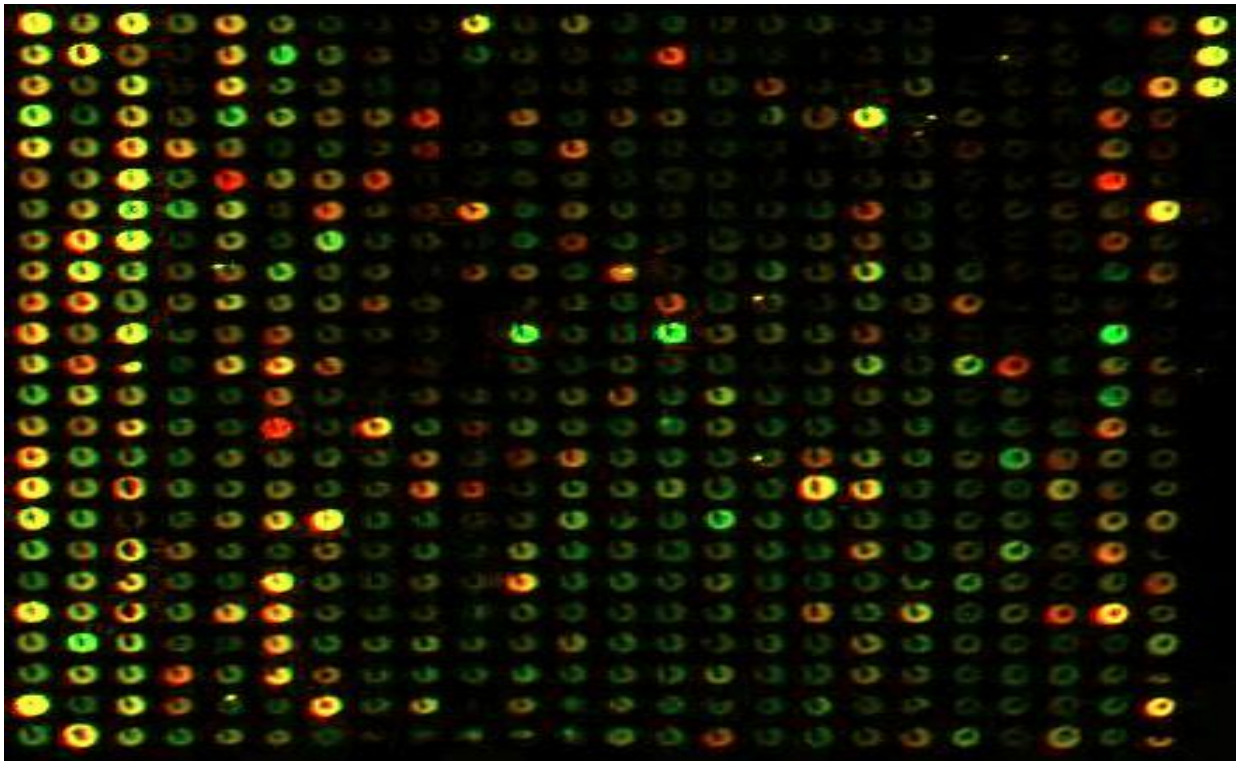
In face recognition a common practice is to first project data (after alignment) onto a low dimensional PCA space,

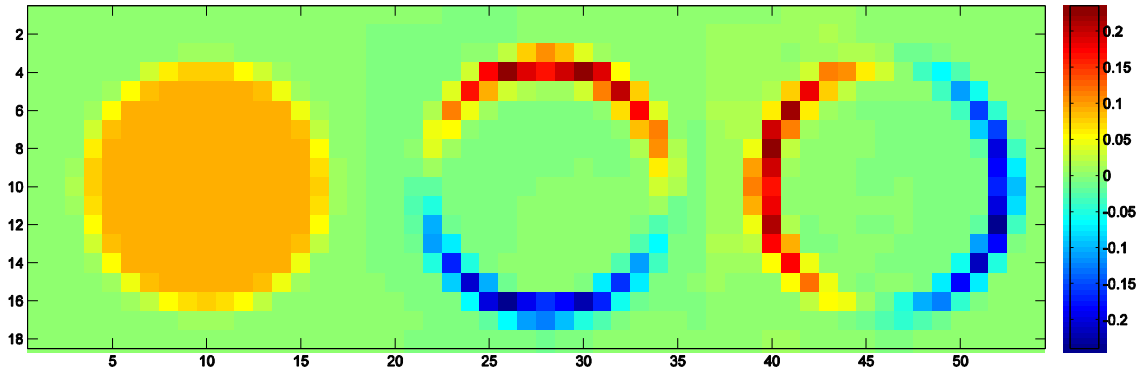
e.g. images from images AT&T Laboratories Cambridge.

u_1 u_2 u_3 u_4

Eigenfaces capture appearance and lighting conditions quite well.

Examples EigenSpots





PCA Algorithm Summary

The PCA algorithm is summarized as follows:

1. subtract mean $\bar{\mathbf{x}}$ from data
2. Calculate sample covariance matrix, $\mathbf{R}_{\mathbf{x}}$ for $\{\mathbf{x}_n - \bar{\mathbf{x}}\}$
3. Perform eigenvalue decomposition: $\mathbf{R}_x = \mathbf{U}\Lambda\mathbf{U}^T$
4. Approximate data by the first M components that have the largest eigenvalue:

$$\mathbf{x}_n \approx \bar{\mathbf{x}} + \sum_{i=1}^{m} (\mathbf{u}_i^T (\mathbf{x}_n - \bar{\mathbf{x}}))\mathbf{u}_i$$

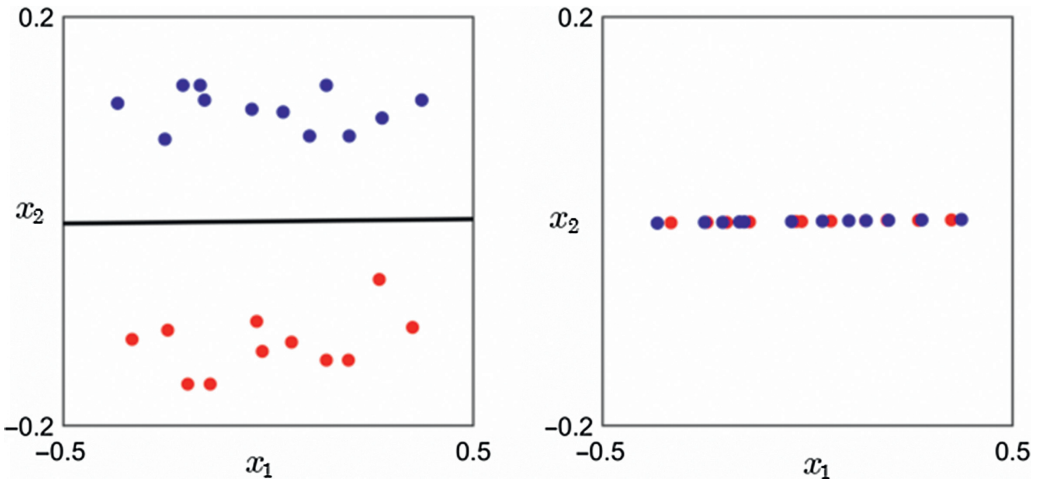
Independent Component Analysis

Limitations of PCA:

- PCA only considers 2nd order statistics (there is a Gaussian interpretation).
- Principal Directions may not be meaningful for classification

Why:

PCA aims to retain the max variance in the components. (This is easy to see from the spatial whitening. The variance of the transformed variable is maximal. Graphically it means it retains a direction that the projection has highest variance)



(left) A toy classification dataset consisting of two linearly separable classes. The ideal subspace produced via PCA is shown in black. (right) Projecting the data onto this subspace (in other words reducing the feature space dimension via PCA) destroys completely the original separability of the data.

Independent Component Analysis

Limitations of PCA:

- PCA only considers 2nd order statistics (there is a Gaussian interpretation).
- Principal Directions may not be meaningful

Can we find a more meaningful decomposition?

Independent Component Analysis (ICA): decompose a data vector into *independent* 1-d components (assuming such a representation exists).

N.B. "independent" is much stronger than "uncorrelated"
ie we now put even more stringent requirements

A motivating example

- Blind source separation
- Two speakers John and Jane talk s1, s2
- I have two microphones and record

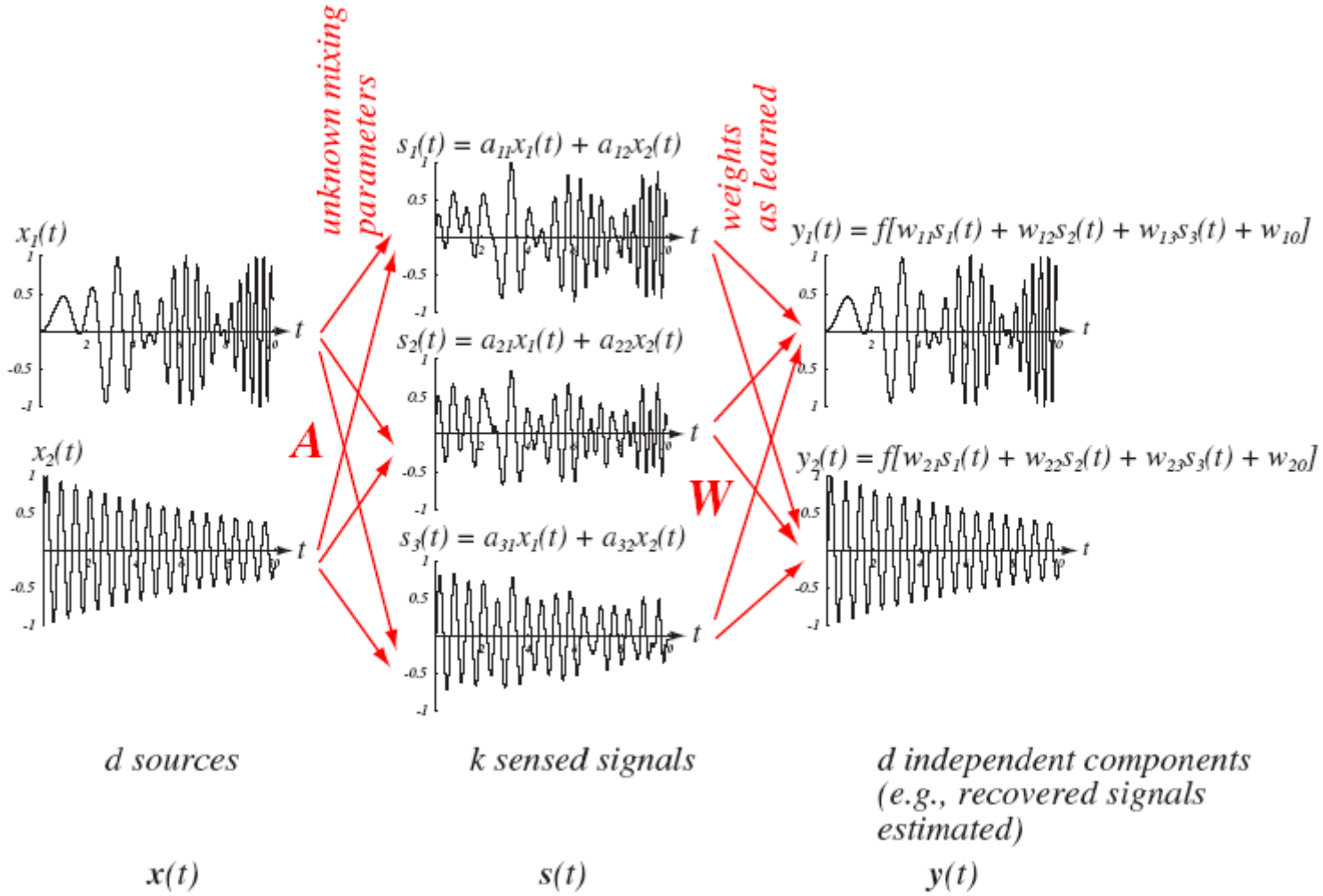
$y1 = a_{11}s1+a_{12}s2$

$y2 = a_{21}s1 + a_{22}s2$

$\mathbf{y} = \mathbf{As}$

Known as the blind source separation problem..

Example: Audio Source Separation



At a grand scale AudioScope and Norsonic acoustic camera

- http://www.youtube.com/watch?v=bgz7Cx-qSFw

The ICA Model

In the ICA model we model the data, $\boldsymbol{x}^{(n)}$, (assumed zero mean) as independent random variables transformed through a linear 'mixing' operation:

$$\mathbf{x}^{(n)} = \mathbf{A}\mathbf{s}^{(n)}$$

where A is an unknown invertible transform, and the components of $\boldsymbol{s}^{(n)}$ are element-wise independent:

$$p(\mathbf{s}^{(n)}) = \prod_i p(s_i^{(n)})$$

ICA attempts to learn A (and hence $\boldsymbol{s}^{(n)}$) from $\boldsymbol{x}^{(n)}$.

An important motivation for ICA is it is equivalent to solving the "*Blind Source Separation"* problem (this has many applications)

Spatial whitening and ICA

Decorrelation or 'spatial' whitening (as in PCA) is not enough. If

$$\mathbf{R}_x = E\{\mathbf{x}\mathbf{x}^T\} = \mathbf{V}\mathbf{D}\mathbf{V}^T$$

(eigenvalue decomposition) then

$$\mathbf{z} = \mathbf{Q}\mathbf{D}^{-1/2}\mathbf{V}^T\mathbf{x}$$

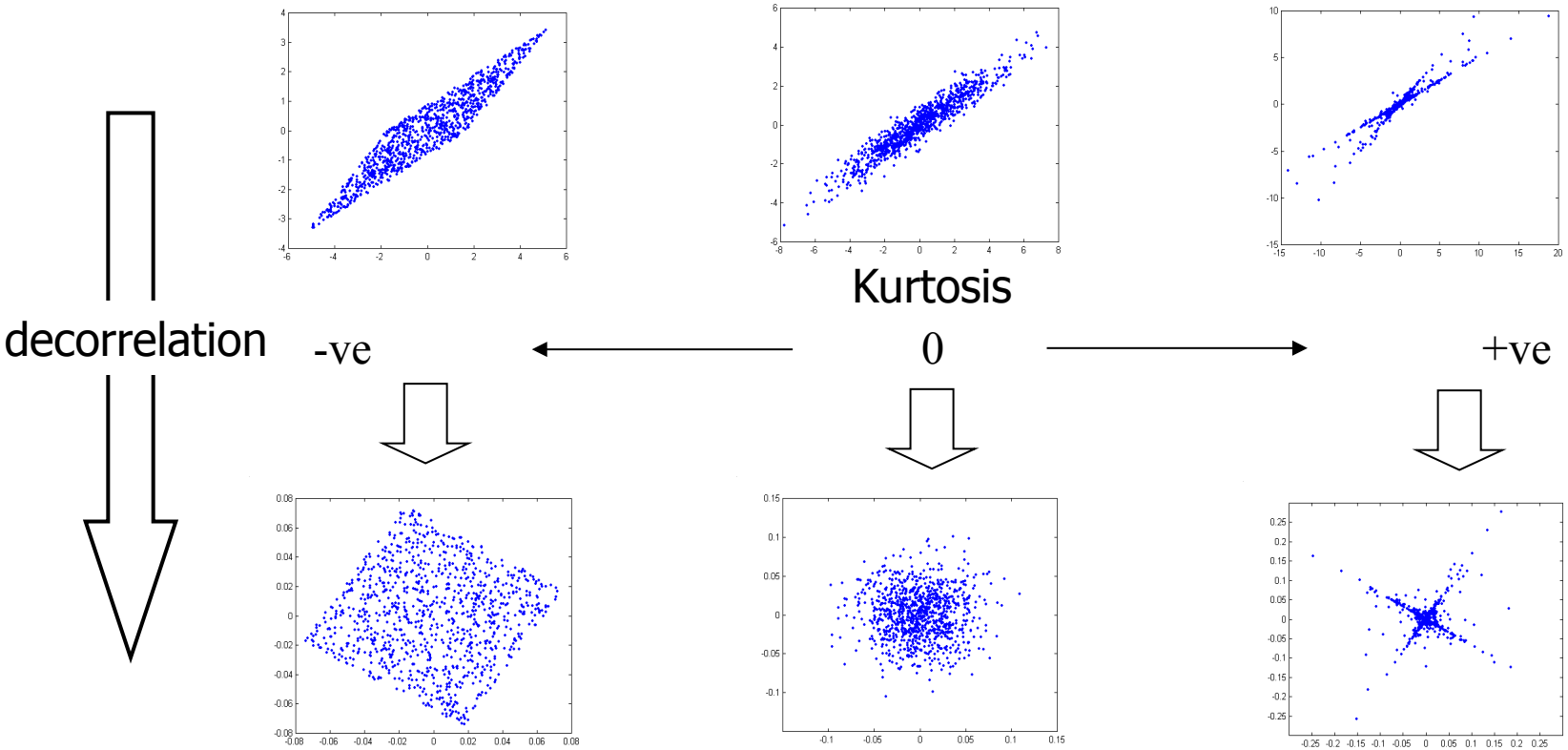
Is also spatially white, where ***Q*** is any rotation matrix (orthogonal $\mathbf{Q}^T\mathbf{Q}=\mathbf{I}$). To see this write:

$$\mathbf{R}_z = E\{\mathbf{z}\mathbf{z}^T\} = \mathbf{Q}\mathbf{D}^{-1/2}\mathbf{V}^T E\{\mathbf{x}\mathbf{x}^T\}\mathbf{V}\mathbf{D}^{-1/2}\mathbf{Q}^T = \mathbf{I}$$

Therefore spatial whitening only does 'half' the job – we still need to find some unknown rotation matrix.

Pre-whitening (in pictures)

All decorrelated but not necessarily independent



Note independent solution identifiable when data is not Gaussian

Understanding the previous slide

- Kurtosis: a measure of non-Gaussianity [the more symmetric the smaller; the more tails the bigger]
- Top row shows 2D pdfs of two sources
  - Middle is a gaussian
- By prewhitening you can see that:
  - the left one the pdf resembles a uniform (with some rotation) but you can easily identify the major components (which are also orthogonal) [uniform distributions have negative kurtosis]
  - the rightmost one after the rotation it still contains lots of points in the center but also some towards the end; you see also that variation happens across two orthogonal axis and as such you can identify the components
  - the middle one, looks like a nice spherical blob. It is hard to see the components and in fact you can pick any of them ➔ Gaussian sources are bad for ICA!

Maximum Likelihood ICA

Assume that we have prior knowledge of the pdf for $\mathbf{s}^{(n)}$:

$$p_{s_i}(s_i) = q(s_i) \text{ and let } \varphi(s_i) = -\frac{\partial}{\partial s_i}\log q(s_i)$$

It is also easier to work with the 'unmixing' matrix $\mathbf{W}=\mathbf{A}^{-1}$. Then the *negative log likelihood* can be written as:

$$\begin{aligned} \mathrm{J}_{\mathrm{ML}}[\mathbf{W}] &= -\sum_{n=1}^{N} \log \mathrm{p}(\mathbf{x}^{(n)} \mid \mathbf{W}, \mathrm{q}) \\ &= -\sum_{n=1}^{N} \left(\log \mathrm{p}(\mathbf{s}^{(n)} \mid \mathbf{W}, \mathrm{q}) - \log |\det \mathbf{W}| \right) \\ &= -\sum_{n=1}^{N} \left(\sum_{i} \log \mathrm{q}([\mathbf{W}\mathbf{x}^{(n)}]_i) - \log |\det \mathbf{W}| \right) \end{aligned}$$

This can be interpreted as fitting the product of q distributions to the data.

More generally, if s is a vector-valued distribution with density p_s, and $x = As$ for a square, invertible matrix A, then the density of x is given by

$$p_x(x) = p_s(Wx) \cdot |W|,$$

where $W = A^{-1}$.

Maximizing the Likelihood

Simple differentiation gives:

$$-\frac{\partial J_{ML}}{\partial W} = \sum_{n=1}^{N}\left(\boldsymbol{\varphi}(\mathbf{s}^{(n)})\mathbf{x}^{(n)^T} - [\mathbf{W}^{-1}]^T\right)$$

Where $\boldsymbol{\varphi}(\mathbf{y}) = [\varphi(s_1), \varphi(s_2), \ldots, \varphi(s_d)]^T$ is an entry-wise nonlinear function defined by fixed functions. Equating to zero gives:

$$-\frac{\partial J_{ML}}{\partial W} = 0 \quad \Rightarrow \quad \sum_{n=1}^{N}\left(\boldsymbol{\varphi}(\mathbf{s}^{(n)})\mathbf{x}^{(n)^T} - [\mathbf{W}^{-1}]^T\right) = 0$$

$$\Rightarrow \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\varphi}(\mathbf{s}^{(n)})\mathbf{s}^{(n)^T} = I$$

This is a form of nonlinear decorrelation (solved through gradient descent). It looks like we need to know the pdf of the sources to make this work but in reality we don't have to be exact!

Component Analysis Summary

- We have looked at two different decompositions for multidimensional data:
  - Principal Component Analysis
  - Independent Component Analysis
- Both provide “explanations” for the data in terms of low dimensional descriptions.
- In Machine Learning they are often used to pre-process data or generate feature spaces
- Other applications include: source separation, noise reduction, complexity reduction

ACSP Course Summary

What have we learned?

- Machine Learning and pattern recognition use a variety of different models to process data
- Algorithms can be:
  - Discriminative (e.g. SVMs, MLPs) or
  - Generative (e.g. Mixture models, HMMs, ICA)
- The notion of “Learning” can be viewed as parameter optimization
  - learning probabilistic models is closely aligned with Maximum Likelihood methods.
- Learning can be supervised or unsupervised