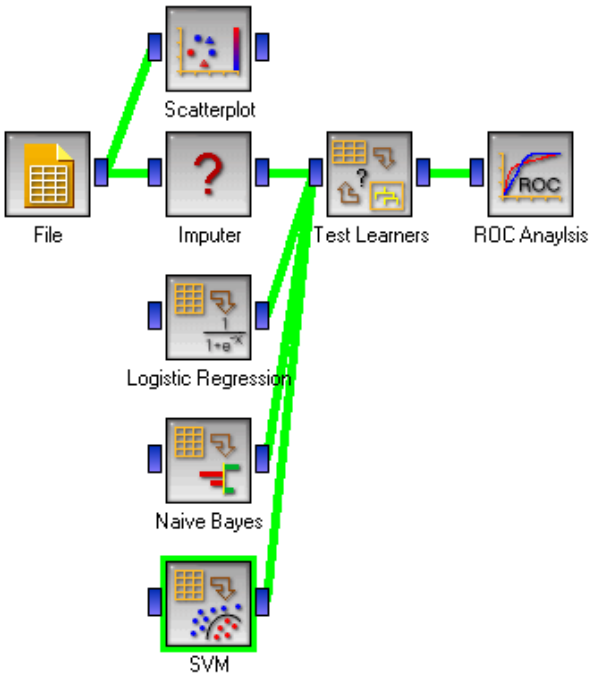
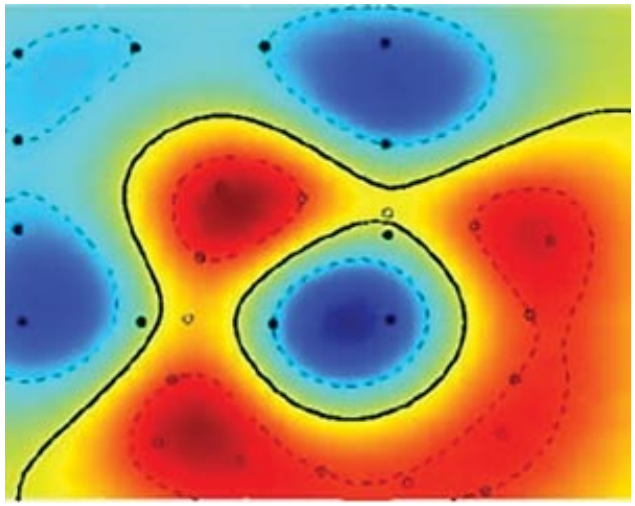
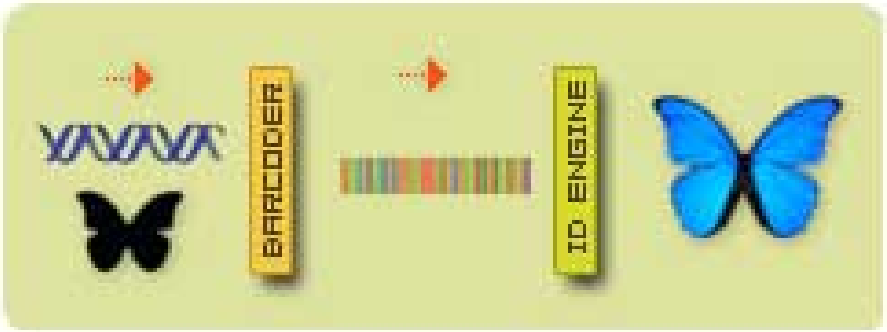
Unsupervised Learning
Dimensionality Reduction - Latent Variable Analysis







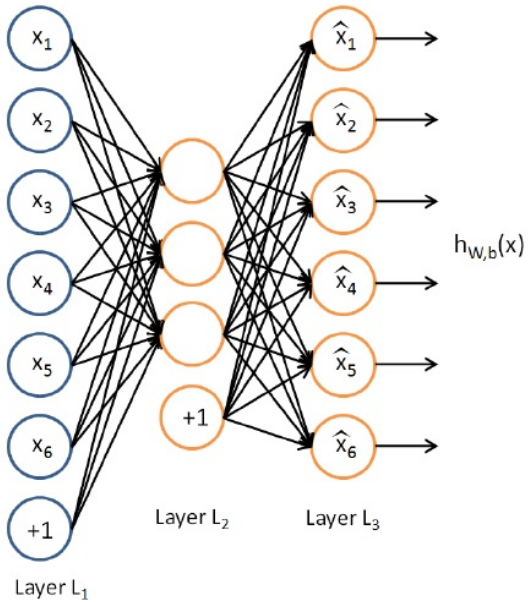
A wide variety of techniques

- Autoencoders
- Nonlinear PCA, kernel PCA
- Sparse PCA
- Independent Component Analysis

Dimensionality Reduction Neural Networks

Autoencoders

An autoencoder is a network with one input layer, one or more hidden layers and one output layer. This type of network aims at providing an internal representation (the layer in the middle) by learning how to predict the input from itself: $x \approx g(x)$.



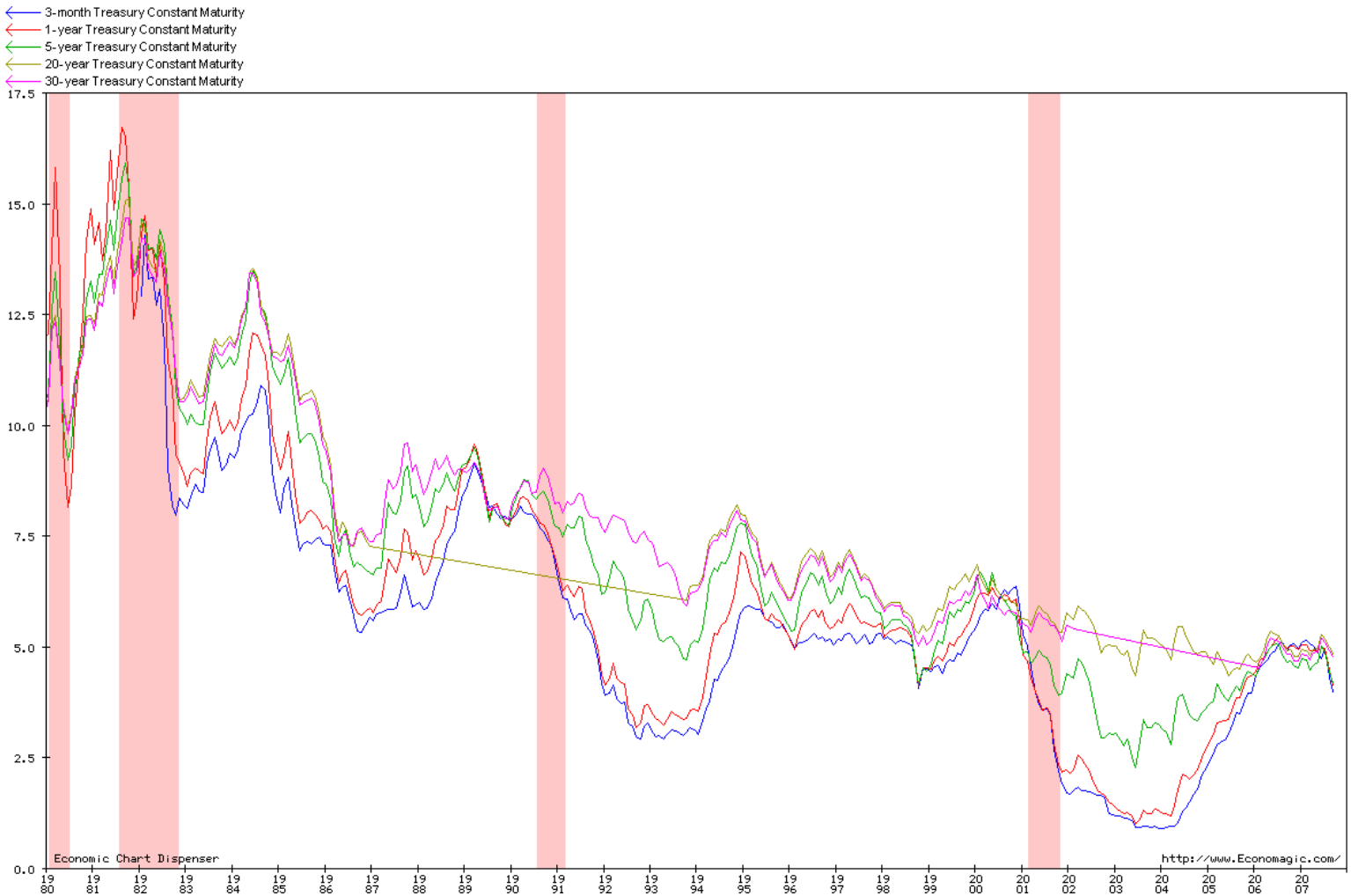
Learning autoencoders

- An auto-encoder is learned by backpropagation of the gradient. When the autoencoder is sparse, the loss considered in general is quadratic and penalized by an additive regularisation term, so that the activity of each unit of the hidden layer is limited in average.
- The autoencoder learns internal representations of complex data

Alternative Techniques for Dimensionality Reduction

Example 1 - Finance

Analysis of interest rates



Example 1 - Finance (2)

- Variables = 18 maturities
  = 1M, 3M, 6M, 9M, 1Y, 2Y, ..., 30Y

- Observations = History (monthly) over 8 years
  = 96 bonds

Example 2 - Web 2.0

Last-FM - collaborative webradio

Example 2 - Web 2.0 (2)

- 28302 artists and their "tags"
- Variables = 735 tags
  = trance, techno, ambient, alternative,
  rap metal, rock, ...
- Observations = 2840 users

Example 3 - Face Identification

Example 3 - Face Identification (2)

- Variables = 256 x 256 pixels
- Observations = 64 images

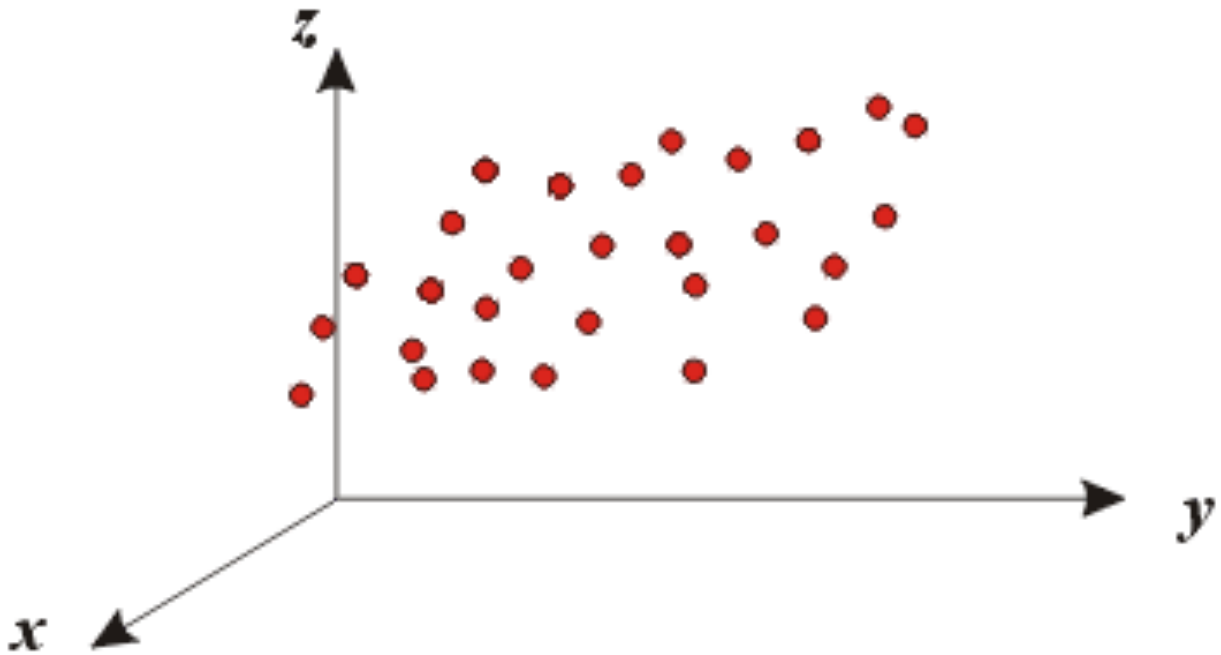
Common features

- Multivariate data
- Need for interpretation
- Variability explained by combinations of the original variables

Data

- Dimension = number of variables = p
- Sample size = number of observations = n
- Table $n \times p$ of quantitative variables

Graphical representation



$\Rightarrow$ Cloud of n points in $\mathbb{R}^p$

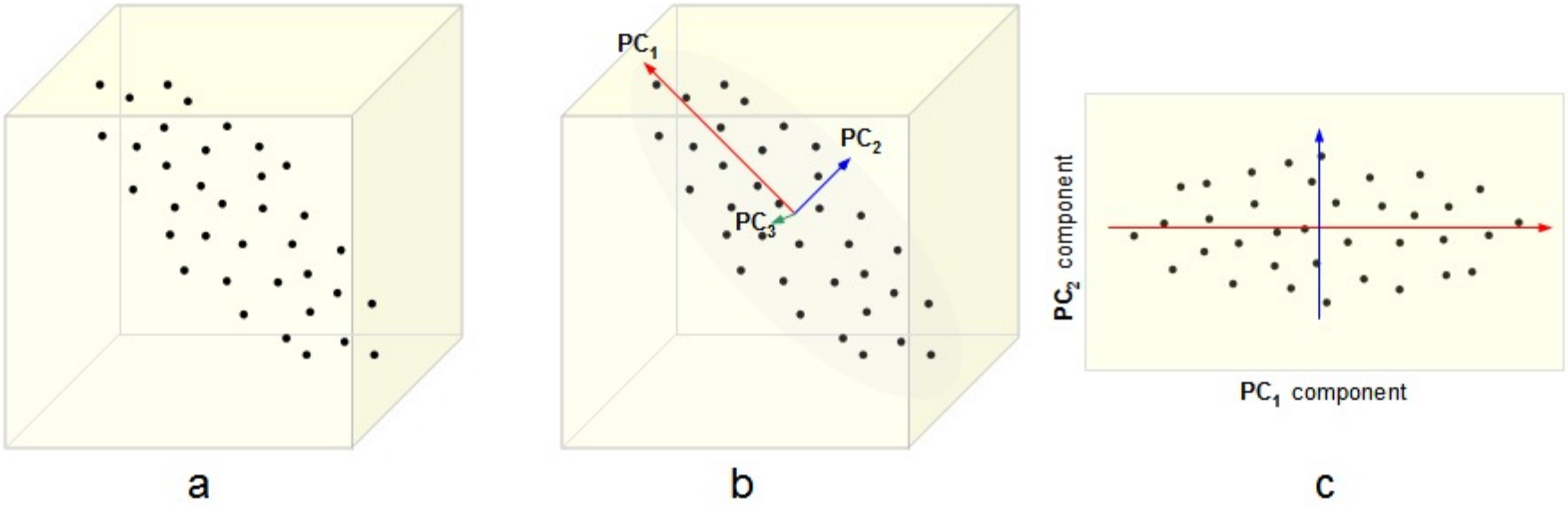
Objectives

- Reduction of the dimension
- Visualization of the cloud in 2D or 3D
- Explaining the variability

Principal Component Analysis (PCA)

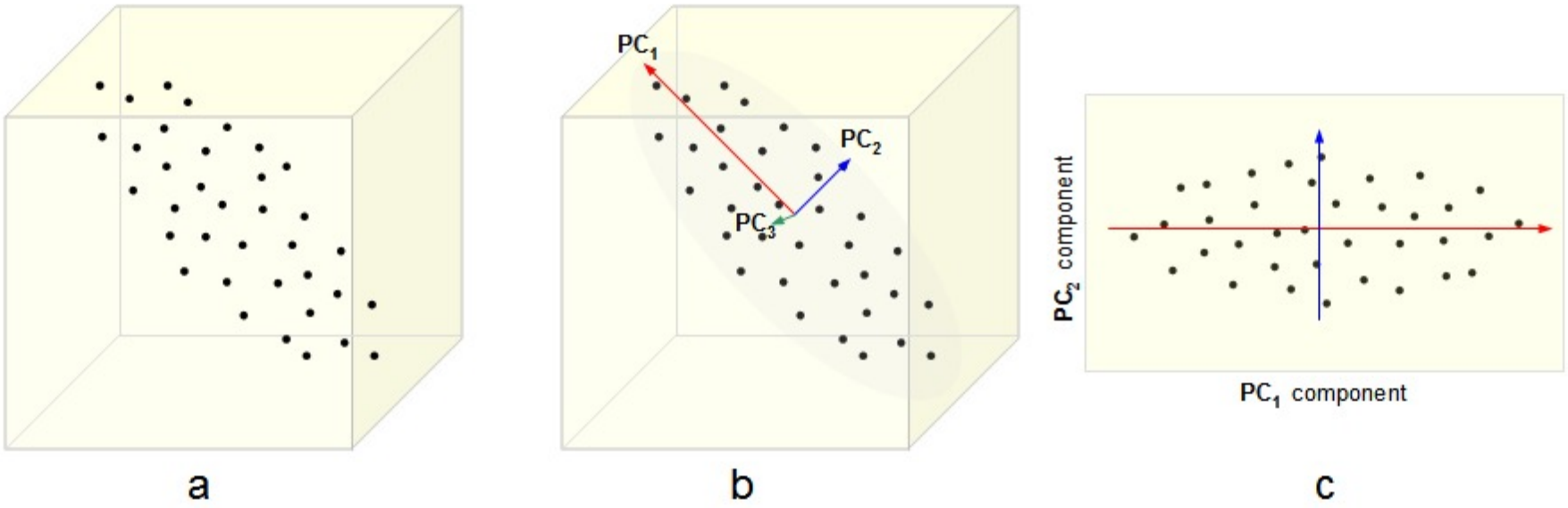
Rationale of PCA

$\rightarrow$ Project the cloud onto the "right" axis



Rationale of PCA

$\rightarrow$ Project the cloud onto the "right" axis



Idée : maximize the dispersion

Statistical framework : data arrays

- Observations : $X_i \in \mathbb{R}^p \quad , \quad 1 \leq i \leq n$
- Variable j : $X_{1j}, \ldots, X_{nj}$
- Matrix $n \times p$ of data $X = (X_1, \ldots, X_n)^T$

$$X = (X_{ij})_{i,j} = \begin{pmatrix} X_{11} & \ldots & X_{1p} \\ \vdots & \ddots & \vdots \\ X_{n1} & \ldots & X_{np} \end{pmatrix}$$

Empirical covariance matrix

- Barycenter

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \in \mathbb{R}^p$$

- Empirical covariance matrix $(p \times p)$

$$S = (s_{kj})_{k,j} = \frac{1}{n}\sum_{i=1}^{n} X_i X_i^T - \bar{X}\bar{X}^T$$

Best direction

- Direction of the projection $a \in \mathbb{R}^p$
- Sample (1D) $= (a^T X_1, \ldots, a^T X_n)$
- Maximize the empirical variance in a :

$$s_a^2 = a^T S a$$

- Solution (linear algebra):

  eigenvector $g_{(1)}$ of the largest eigenvalue l_1

Diagonalization of S PSD

- Eigenvalues : $l_1 \geq \ldots \geq l_p$
- Orthonormal basis of eigenvectors $g_{(1)}, \ldots, g_{(p)}$
- Reduction of the matrix $S = GLG^T$ where
  - $L = diag(l_1, \ldots, l_p)$ diagonal matrix $p \times p$
  - G orthogonal matrix $p \times p$

$$G = (g_{(1)}, \ldots, g_{(p)}) = (g_{kj})_{k,j}$$

Principal components (PC's)

- Principal components : for all vector $z \in \mathbb{R}^p$

$$y_j(z) = g_{(j)}^T(z - \bar{X}) \ , \quad 1 \leq j \leq p$$

- The matrix $n \times p$

$$Y = (y_j(X_i))_{1 \leq i \leq n,\ 1 \leq j \leq p}$$

replaces the matrix X with the original data.

Empirical linear correlation "Variable vs. PC"

- Empirical linear correlation between variable k and the PC y_j :

$$\tilde{r}_{kj} = g_{kj}\sqrt{\frac{l_j}{s_{kk}}} \qquad \text{(définition)}$$

- Property:

$$\sum_{j=1}^{p} \tilde{r}_{kj}^2 = 1$$

Empirical variance of the k-th variable

- Part of the empirical variance of the k-th variable explained by the first 2 PC's (y_1, y_2) :

$$\tilde{r}_{k1}^2 + \tilde{r}_{k2}^2$$

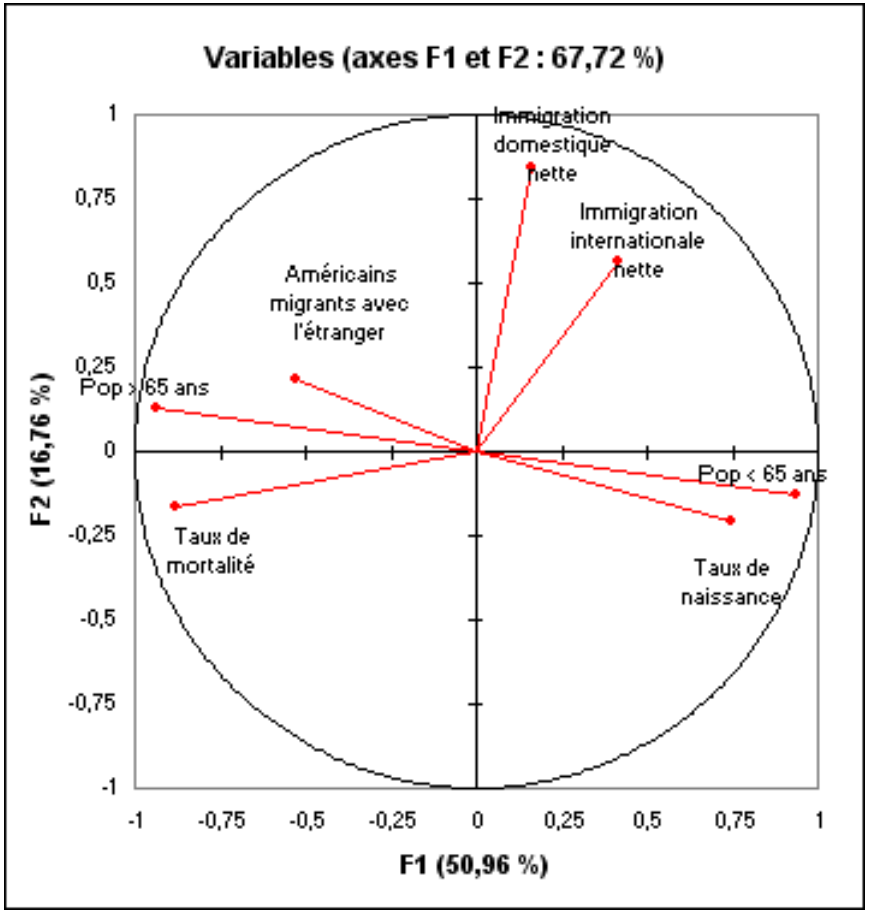
- We have :

$$l_1 + l_2 = \sum_{k=1}^{p} s_{kk}(\tilde{r}_{k1}^2 + \tilde{r}_{k2}^2)$$

- 2D Visualization : correlation disk

Correlation disk

- Point $(\tilde{r}_{k1}, \tilde{r}_{k2})$ corresponds to variable k



Empirical variance

- Part of the empirical variance of the cloud of points explained by the PC y_j:

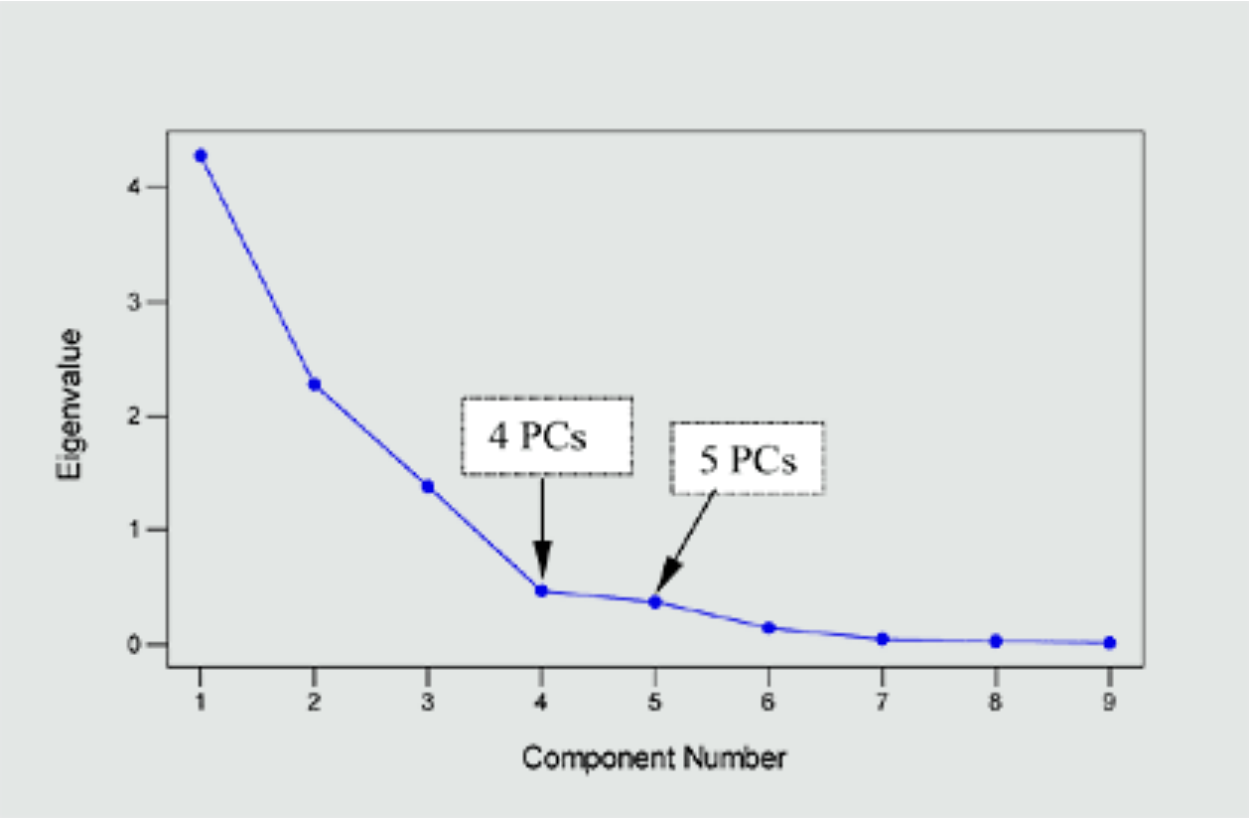
$$v_j = \frac{l_j}{Tr(S)}$$

où $Tr(S) = \sum_{j=1}^{p} l_j$.

- Visualization : scree-graph

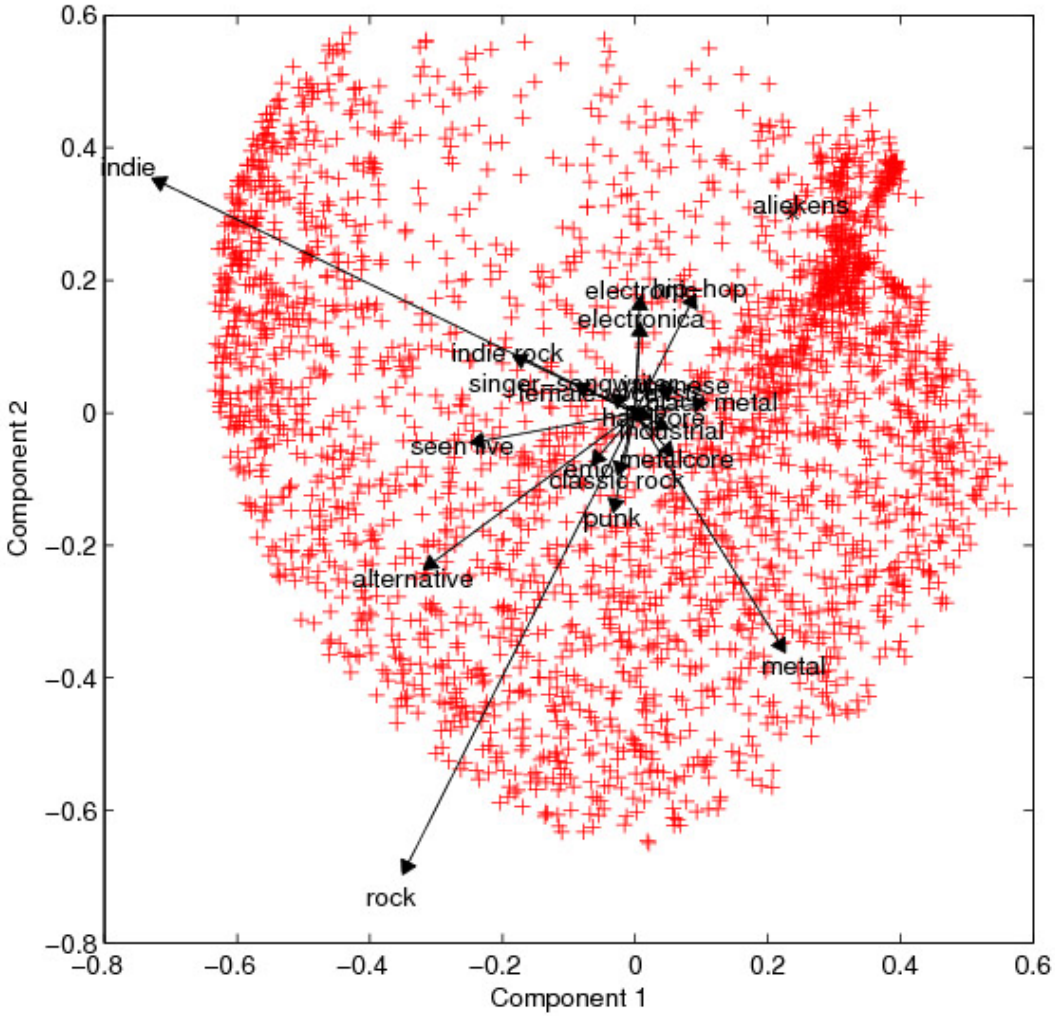
Scree-graph

- Axes = index j of the PC and part of the variance v_j



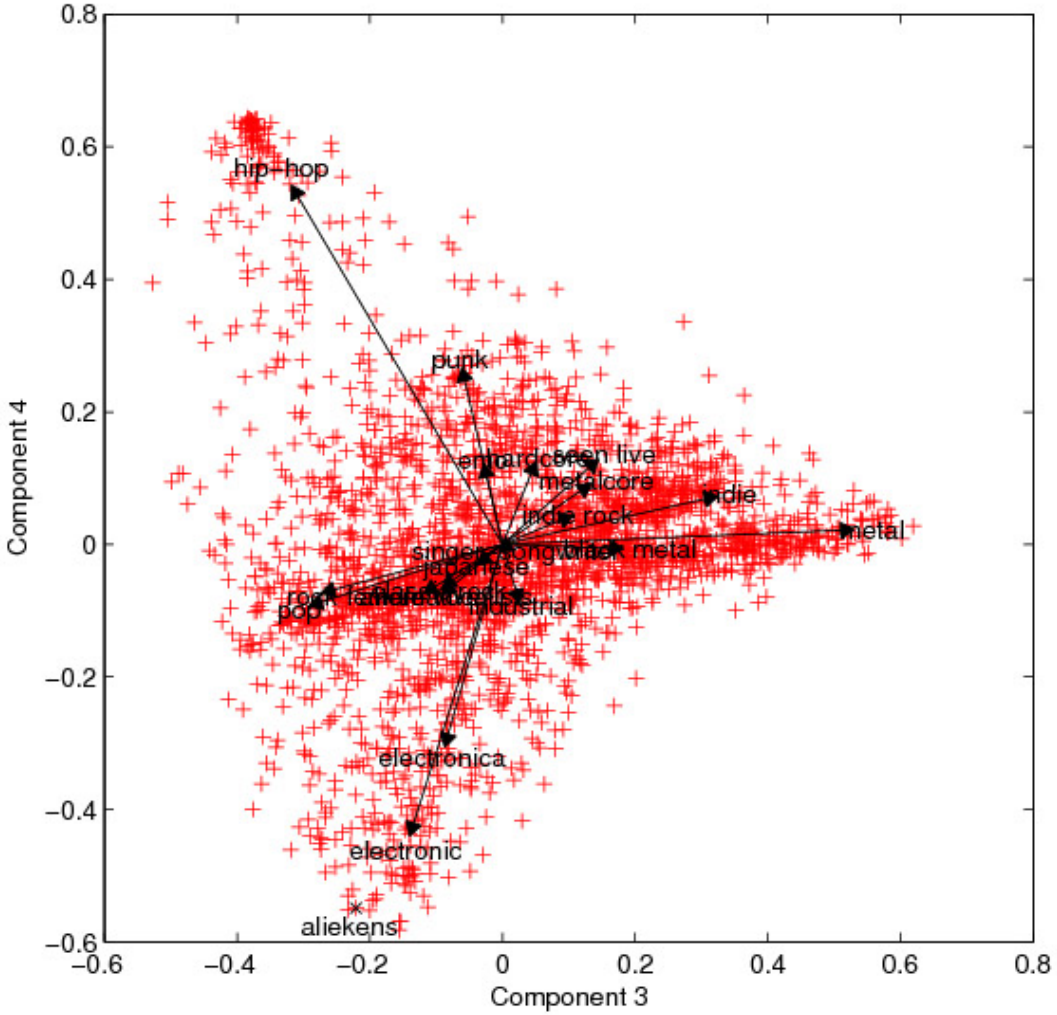
Results of the PCA - Last-FM (1)

Projection of the cloud of points onto (PC1, PC2)



Results of the PCA - Last-FM (2)

Projection of the cloud of points onto (CP3, CP4)

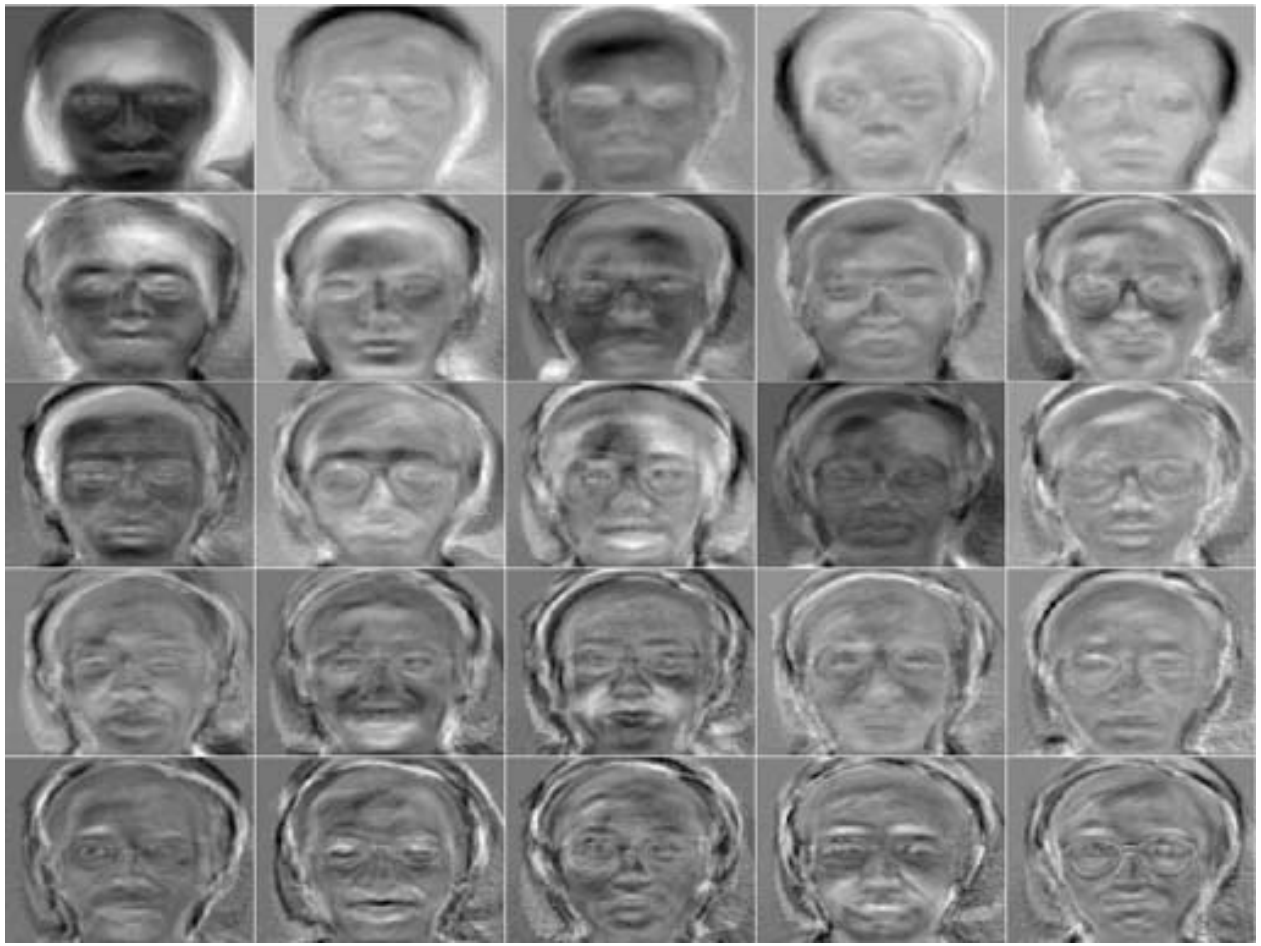


Results of the PCA - Faces (1)

Data

Results of the PCA - Faces (2)

"Clean faces"



Results of the PCA - Faces (3)

Partial reconstruction (sub-column of the matrix Y)

Results of the PCA - Faces (4)

Projection of other images

A few remarks

- PCA = linear tool
- Orthogonality of principal components
- **In practice :**

  Reduction of the matrix $R = (r_{kj})_{k,j}$ of correlations

$$r_{kj} = \frac{s_{kj}}{\sqrt{s_{kk} s_{jj}}}$$

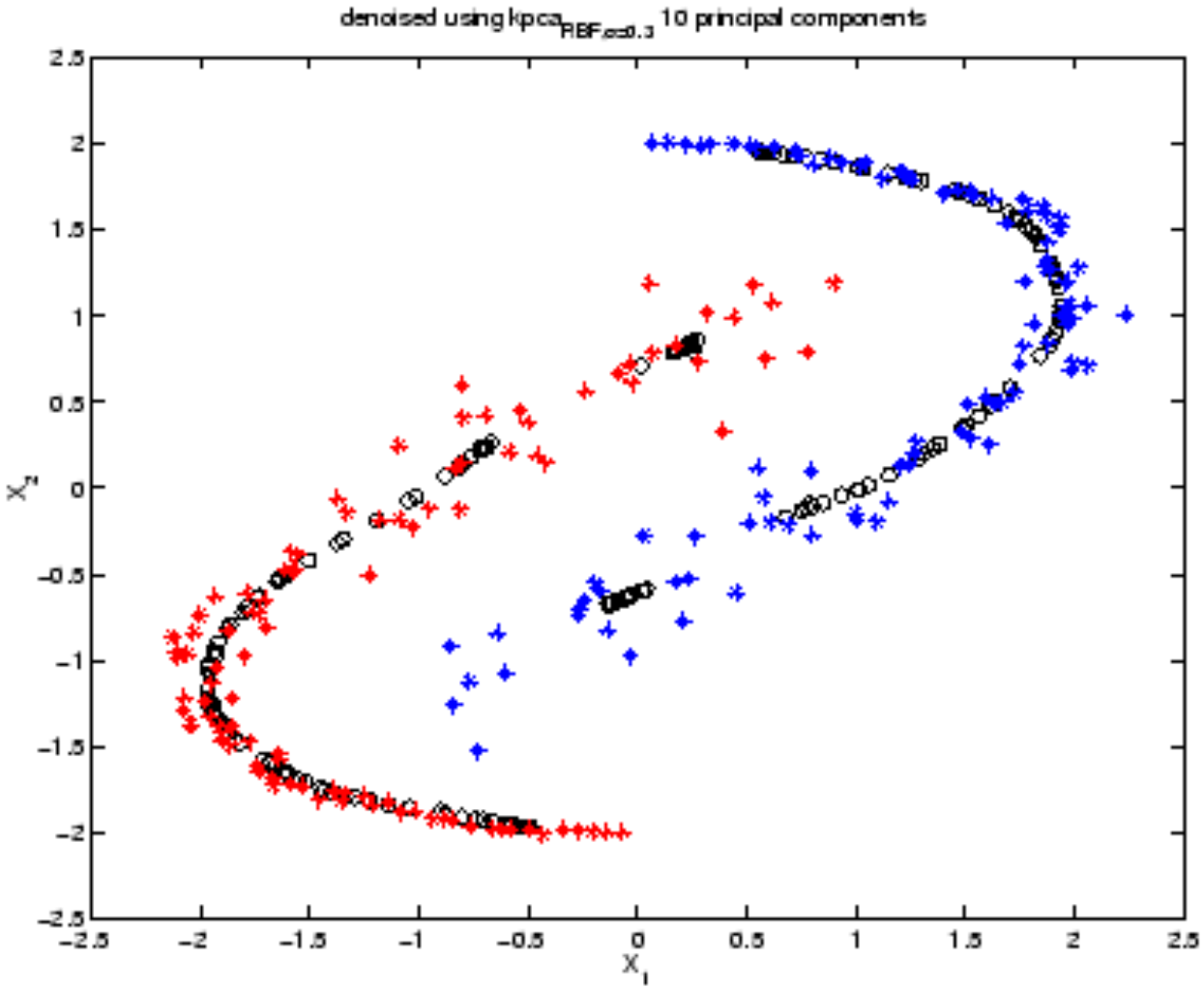
- **Computational difficulties :**

  Reduction of S in very high dimension

When does it work ?

- When the clouds are of the form of ellipsoïds
- Implicit model = gaussian model
- Information carried by order statistics
- Absence of outliers

Failure of PCA



$\Rightarrow$ **Extension :** non-linear PCA (with kernel)

Positive kernels

Definition

Let $\mathcal{X}$ be the space of the observations.

Positive kernel

A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive kernel iff

1. k is symmetric: $k(x, x') = k(x', x)$, $\forall x, x' \in \mathcal{X}$
2. k is positive:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0, \quad \forall c_i \in \mathbb{R}, \quad \forall x_i \in \mathcal{X}, \quad \forall n \geq 1$$

A theorem in Functional Analysis

Theorem of Mercer

For all positive kernel k on $\mathcal{X}$ there exists a Hilbert space $\mathcal{H}$ and a mapping Φ ('feature map') such that:

$$k(x, x') = < \Phi(x), \Phi(x') >, \quad \forall x, x' \in \mathcal{X}$$

where $< , >$ represents the inner product on $\mathcal{H}$.

Comments

- The theorem of Mercer is not constructive: it does not provide $\mathcal{H}$, nor Φ

Comments

- The theorem of Mercer is not constructive: it does not provide $\mathcal{H}$, nor Φ
- In practice:

Comments

- The theorem of Mercer is not constructive: it does not provide $\mathcal{H}$, nor Φ
- In practice:
  - $\mathcal{H}$ is a space of high (possibly infinite) dimension

Comments

- The theorem of Mercer is not constructive: it does not provide $\mathcal{H}$, nor Φ
- In practice:
  - $\mathcal{H}$ is a space of high (possibly infinite) dimension
  - Φ is a non-linear mapping

Comments

- The theorem of Mercer is not constructive: it does not provide $\mathcal{H}$, nor Φ
- In practice:
  - $\mathcal{H}$ is a space of high (possibly infinite) dimension
  - Φ is a non-linear mapping
- $\mathcal{H}$ is a space used for representing the data, referred to as "feature space"

Comments

- The theorem of Mercer is not constructive: it does not provide $\mathcal{H}$, nor Φ
- In practice:
  - $\mathcal{H}$ is a space of high (possibly infinite) dimension
  - Φ is a non-linear mapping
- $\mathcal{H}$ is a space used for representing the data, referred to as "feature space"
- The kernel trick consists in avoiding to specify $\mathcal{H}$ and Φ when we know they do exist!

Image distances

- Euclidean norm on $\mathbb{R}^m$: $\forall u \in \mathbb{R}^m,\ \|u\| = \sqrt{< u, u >}$ where $< , >$ inner product on $\mathbb{R}^m$

Image distances

- Euclidean norm on $\mathbb{R}^m$: $\forall u \in \mathbb{R}^m,\ \|u\| = \sqrt{< u, u >}$ where $< , >$ inner product on $\mathbb{R}^m$
- Euclidean distance:
  $d(u, v) = \|u - v\| = \sqrt{< u, u > + < v, v > -2 < u, v >}$

Image distances

- Euclidean norm on $\mathbb{R}^m$: $\forall u \in \mathbb{R}^m,\ \|u\| = \sqrt{< u, u >}$ where $< , >$ inner product on $\mathbb{R}^m$
- Euclidean distance:
  $d(u, v) = \|u - v\| = \sqrt{< u, u > + < v, v > -2 < u, v >}$
- Non-linear transform : $\Phi\ :\ \mathbb{R}^d \to \mathbb{R}^m$ avec $m > d$

Image distances

- Euclidean norm on $\mathbb{R}^m$: $\forall u \in \mathbb{R}^m,\ \|u\| = \sqrt{< u, u >}$ where $< , >$ inner product on $\mathbb{R}^m$
- Euclidean distance:
  $d(u, v) = \|u - v\| = \sqrt{< u, u > + < v, v > -2 < u, v >}$
- Non-linear transform : $\Phi\ :\ \mathbb{R}^d \to \mathbb{R}^m$ avec $m > d$
- Kernel: $k(x, x') =< \Phi(x), \Phi(x') >$

Image distances

- Euclidean norm on $\mathbb{R}^m$: $\forall u \in \mathbb{R}^m,\ \|u\| = \sqrt{< u, u >}$ where $< , >$ inner product on $\mathbb{R}^m$
- Euclidean distance:
  $d(u, v) = \|u - v\| = \sqrt{< u, u > + < v, v > -2 < u, v >}$
- Non-linear transform : $\Phi\ :\ \mathbb{R}^d \to \mathbb{R}^m$ avec $m > d$
- Kernel: $k(x, x') =< \Phi(x), \Phi(x') >$
- Image distance:

$$d_\Phi(x, x') = \|\Phi(x) - \Phi(x')\| = \sqrt{k(x, x) + k(x', x') - 2k(x, x')}$$

$\Rightarrow$ the distance induced by Φ involves the kernel only

Gain

- No algorithmic complication when replacing the original inner product by another similarity measure

Gain

- No algorithmic complication when replacing the original inner product by another similarity measure
- Turn a problem initially non-linear into a linear problem by sending the data to a space of higher dimension

Gain

- No algorithmic complication when replacing the original inner product by another similarity measure
- Turn a problem initially non-linear into a linear problem by sending the data to a space of higher dimension

Gain

- No algorithmic complication when replacing the original inner product by another similarity measure
- Turn a problem initially non-linear into a linear problem by sending the data to a space of higher dimension

Example

Let $f(x, y) = ax^2 + bx + c - y = 0$ be a quadratic surface decision (parabolic in $\mathbb{R}^2$).

Gain

- No algorithmic complication when replacing the original inner product by another similarity measure
- Turn a problem initially non-linear into a linear problem by sending the data to a space of higher dimension

Example

Let $f(x, y) = ax^2 + bx + c - y = 0$ be a quadratic surface decision (parabolic in $\mathbb{R}^2$).

Key role of the transformation:

$$\begin{array}{rccc} \Phi & : & \mathbb{R}^2 & \rightarrow & \mathbb{R}^4 \\ & & x & \mapsto & (x^2, x, 1, y)^T \end{array}$$

From non-linear to linear

Example (continued)

One may write:

$$g(x^2, x, 1, y) = ax^2 + bx + c - y = 0$$

où $g(u, v, w, y) = au + bv + cw - y$.

The equation $g(u, v, w, y) = 0$ defines a decision surface that is linear in $\mathbb{R}^4$.

From non-linear to linear

Example (continued)

One may write:

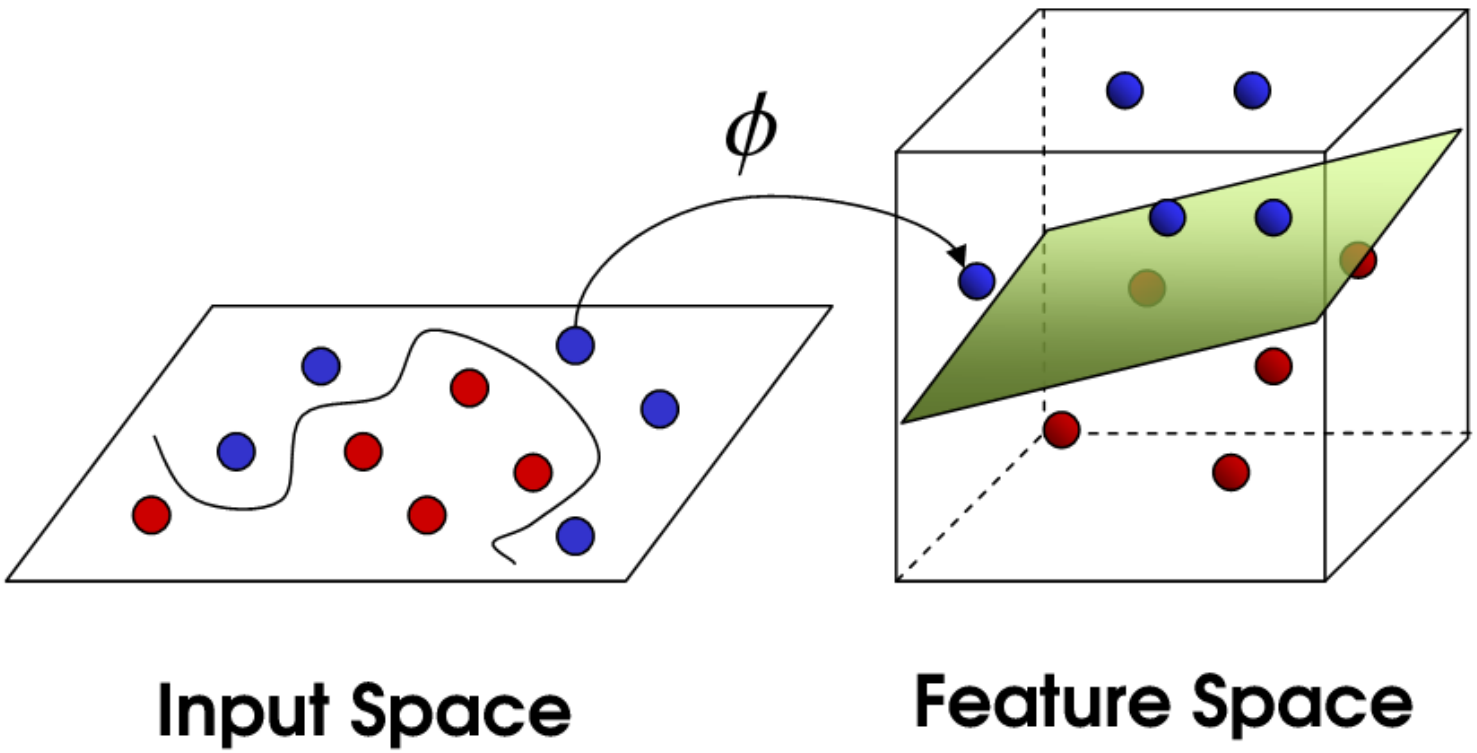
$$g(x^2, x, 1, y) = ax^2 + bx + c - y = 0$$

où $g(u, v, w, y) = au + bv + cw - y$.

The equation $g(u, v, w, y) = 0$ defines a decision surface that is linear in $\mathbb{R}^4$.

A non-linear problem in a certain space can be formulated sometimes as a linear problem in a 'larger' space.

From non-linear to linear



Kernel PCA

Classic PCA

Consider a cloud of points $x_1, \ldots, x_n$ centered at the origin.

Classic PCA

Consider a cloud of points $x_1, \ldots, x_n$ centered at the origin.

Goals of PCA

- Method for visualizing the data
- Effective reduction of the dimension of the data

Classic PCA

Consider a cloud of points $x_1, \ldots, x_n$ centered at the origin.

Goals of PCA

- Method for visualizing the data
- Effective reduction of the dimension of the data

PCA consists in identifying the principal components of the data formed by

1. the best direction for projecting the cloud of points
   i.e. that with maximal variance

Classic PCA

Consider a cloud of points $x_1, \ldots, x_n$ centered at the origin.

Goals of PCA

- Method for visualizing the data
- Effective reduction of the dimension of the data

PCA consists in identifying the principal components of the data formed by

1. the best direction for projecting the cloud of points
   i.e. that with maximal variance
2. next, the best direction for projecting that is orthogonal to the first

Classic PCA

Consider a cloud of points $x_1, \ldots, x_n$ centered at the origin.

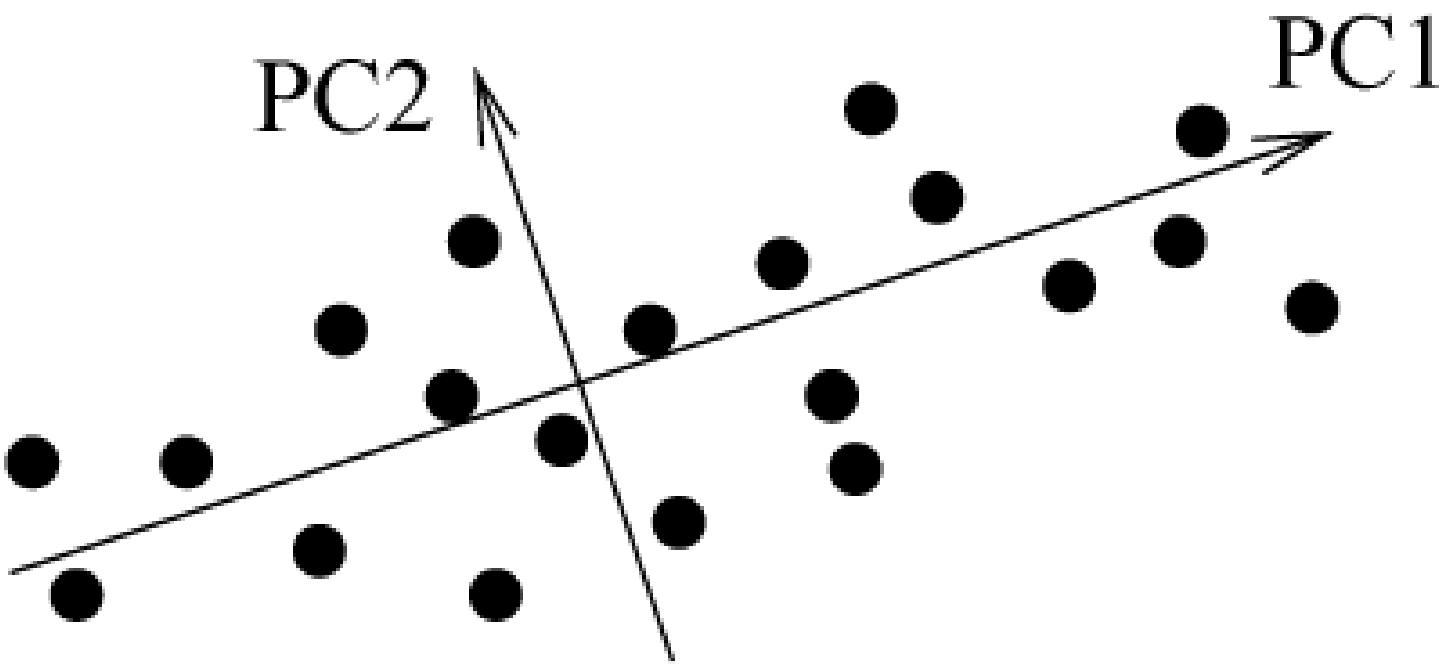
Goals of PCA

- Method for visualizing the data
- Effective reduction of the dimension of the data

PCA consists in identifying the principal components of the data formed by

1. the best direction for projecting the cloud of points
   i.e. that with maximal variance
2. next, the best direction for projecting that is orthogonal to the first
3. and, so on so forth, until the n-th

PCA



PCA (continued)

- Orthogonal projection of a vector x onto direction $w \in \mathbb{R}^d$:

$$p_w(x) = \frac{< x, w >}{\|w\|}$$

PCA (continued)

- Orthogonal projection of a vector x onto direction $w \in \mathbb{R}^d$:

$$p_w(x) = \frac{< x, w >}{\|w\|}$$

- Empirical variance of the cloud of points along direction w:

$$\mathbb{V}(p_w) = \frac{1}{n} \sum_{i=1}^{n} \frac{< x_i, w >^2}{\|w\|^2}$$

PCA (continued)

- Orthogonal projection of a vector x onto direction $w \in \mathbb{R}^d$:

$$p_w(x) = \frac{< x, w >}{\|w\|}$$

- Empirical variance of the cloud of points along direction w:

$$\mathbb{V}(p_w) = \frac{1}{n} \sum_{i=1}^{n} \frac{< x_i, w >^2}{\|w\|^2}$$

- Empirical covariance matrix $\Sigma = \frac{1}{n} \sum_{i=1}^{n} x_i \; x_i^T$

PCA (continued)

- Orthogonal projection of a vector x onto direction $w \in \mathbb{R}^d$:

$$p_w(x) = \frac{< x, w >}{\|w\|}$$

- Empirical variance of the cloud of points along direction w:

$$\mathbb{V}(p_w) = \frac{1}{n} \sum_{i=1}^{n} \frac{< x_i, w >^2}{\|w\|^2}$$

- Empirical covariance matrix $\Sigma = \frac{1}{n} \sum_{i=1}^{n} x_i \; x_i^T$
- We then have :

$$\mathbb{V}(p_w) = \frac{w^T \Sigma w}{\|w\|^2}$$

Optimization problem

First principal component

$$\arg\max_{w} \mathbb{V}(p_w) = \frac{w^T \Sigma w}{\|w\|^2}$$

Optimization problem

First principal component

$$\arg\max_{w} \mathbb{V}(p_w) = \frac{w^T \Sigma w}{\|w\|^2}$$

Solution

The principal components are the eigenvectors of Σ sorted by decreasing order of magnitude of the corresponding eigenvalues.

Optimization problem

First principal component

$$\arg\max_{w} \mathbb{V}(p_w) = \frac{w^T \Sigma w}{\|w\|^2}$$

Solution

The principal components are the eigenvectors of Σ sorted by decreasing order of magnitude of the corresponding eigenvalues.

Remark : the matrix Σ is PSD, hence diagonalizable in an orthonormal basis.

PCA (continued)

One searches for a vector v and a real number λ such that:

$$\Sigma v = \lambda v$$

Observe that we have:

$$\Sigma v = \frac{1}{n} \sum_{i=1}^{n} < x_i, v > \ x_i$$

Thus:

$$v = \sum_{i=1}^{n} \left(\frac{< x_i, v >}{n\lambda} \right) x_i = \sum_{i=1}^{n} \alpha_i x_i$$

PCA (continued)

One searches for a vector v and a real number λ such that:

$$\Sigma v = \lambda v$$

Observe that we have:

$$\Sigma v = \frac{1}{n} \sum_{i=1}^{n} < x_i, v > \ x_i$$

Thus:

$$v = \sum_{i=1}^{n} \left(\frac{< x_i, v >}{n\lambda} \right) x_i = \sum_{i=1}^{n} \alpha_i x_i$$

One uses

$$x_j^T \Sigma v = \lambda < x_j, v >, \quad \forall j$$

and one substitutes the expressions for Σ and v:

$$\frac{1}{n} \sum_{i=1}^{n} \alpha_i \left\langle x_j, \sum_{k=1}^{n} < x_k, x_i > \ x_k \right\rangle = \lambda \sum_{i=1}^{n} \alpha_i < x_j, x_i >$$

PCA (continued)

- Denote by $K = (< x_i, x_j >)_{i,j}$ the Gram matrix

PCA (continued)

- Denote by $K = (< x_i, x_j >)_{i,j}$ the Gram matrix
- One may then write the system:

$$K^2\alpha = n\lambda K\alpha$$

PCA (continued)

- Denote by $K = (< x_i, x_j >)_{i,j}$ the Gram matrix
- One may then write the system:

$$K^2\alpha = n\lambda K\alpha$$

- To find α, one thus solves the problem

$$K\alpha = n\lambda\alpha$$

Limitations of classic PCA

- it is essentially tailored to the case of multivariate Gaussian data

Limitations of classic PCA

- it is essentially tailored to the case of multivariate Gaussian data
  - in general, non-correlation does not imply independence of the principal directions

Limitations of classic PCA

- it is essentially tailored to the case of multivariate Gaussian data
  - in general, non-correlation does not imply independence of the principal directions
  - alternative : Independent Component Analysis

Limitations of classic PCA

- it is essentially tailored to the case of multivariate Gaussian data
  - in general, non-correlation does not imply independence of the principal directions
  - alternative : Independent Component Analysis
- it is tailored to detect linear structures

Limitations of classic PCA

- it is essentially tailored to the case of multivariate Gaussian data
  - in general, non-correlation does not imply independence of the principal directions
  - alternative : Independent Component Analysis
- it is tailored to detect linear structures
  - Not all clouds of points are ellipsoïds!!

Limitations of classic PCA

- it is essentially tailored to the case of multivariate Gaussian data
  - in general, non-correlation does not imply independence of the principal directions
  - alternative : Independent Component Analysis
- it is tailored to detect linear structures
  - Not all clouds of points are ellipsoïds!!
  - alternative : Kernel PCA

Kernel PCA

- Apply a transformation Φ that sends the cloud of points X to a space where the structure is linear

Kernel PCA

- Apply a transformation Φ that sends the cloud of points X to a space where the structure is linear
- The covariance matrix of $\Phi(X) = (\Phi(x_1), \ldots, \Phi(x_n))^T$ is then

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \Phi(x_i)^T$$

Kernel PCA

- Apply a transformation Φ that sends the cloud of points X to a space where the structure is linear
- The covariance matrix of $\Phi(X) = (\Phi(x_1), \ldots, \Phi(x_n))^T$ is then

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \Phi(x_i)^T$$

- **Kernel trick :** $K = \left(k(x_i, x_j)\right)_{i,j} = \left(\Phi(x_i)^T \Phi(x_j)\right)_{i,j}$

Kernel PCA (continued)

- Principal "Directions" of the form:

$$p_i(x) = \sum_{j=1}^{n} \alpha_{i,j} k(x_j, x)$$

Kernel PCA (continued)

- Principal "Directions" of the form:

$$p_i(x) = \sum_{j=1}^{n} \alpha_{i,j} k(x_j, x)$$

- the vector $\alpha_i = (\alpha_{i,1}, \ldots, \alpha_{i,n})$ is solution to the optimisation problem:

$$\max_{\alpha} \frac{\alpha^T K^2 \alpha}{\alpha^T K \alpha}$$

under the constraints: $\alpha_i^T K \alpha_j$ pour $j = 1, \ldots, i-1$

Kernel PCA (continued)

- Principal "Directions" of the form:

$$p_i(x) = \sum_{j=1}^{n} \alpha_{i,j} k(x_j, x)$$

- the vector $\alpha_i = (\alpha_{i,1}, \ldots, \alpha_{i,n})$ is solution to the optimisation problem:

$$\max_{\alpha} \frac{\alpha^T K^2 \alpha}{\alpha^T K \alpha}$$

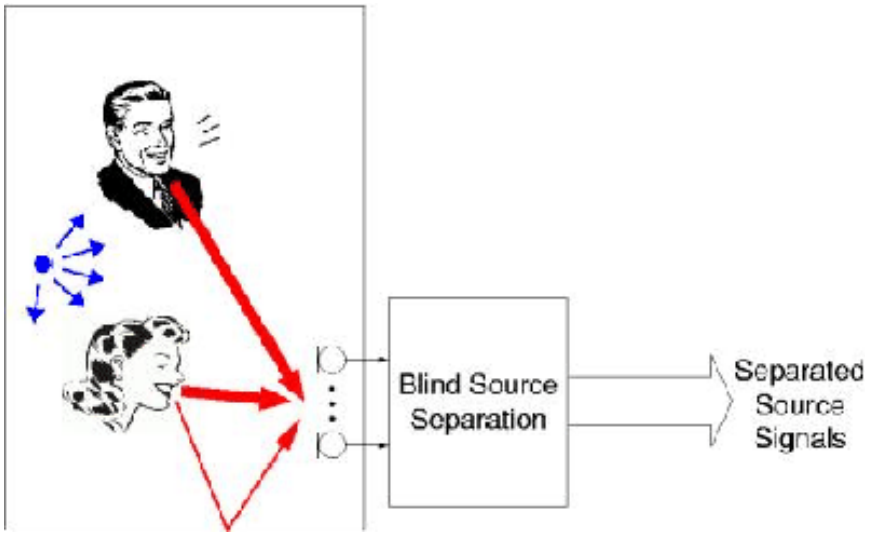
  under the constraints: $\alpha_i^T K \alpha_j$ pour $j = 1, \ldots, i-1$

- one solves the problem:

$$K\alpha = n\lambda\alpha$$

Independent Component Analysis (ICA)

The "cocktail-party" problem



- PCA = based on the notion of correlation
- Appropriate notion = notion of independence

Correlation vs. Independence

- Notice that : X and Y independent $\Rightarrow \mathrm{cov}(X, Y) = 0$
- Reciprocal false in general, except in the Gaussian case..
- From PCA to ICA... (much more difficult !)

Formulation of the problem

- $S = (S_1, \ldots, S_d)^T$ unknown independent and non-Gaussian sources
- $\mathbf{A}$ mixing matrix $d \times d$ unknown
- $X = (X_1, \ldots, X_d)^T$ observations (sensors), one assumes $\mathrm{Cov}(X) = \mathbf{I}$
- One has the system : $X = \mathbf{A}S$
- One searches for $\mathbf{A}$ orthogonal such that:

$$S = \mathbf{A}^T X \qquad \text{has independent components}$$

Information theory

- Entropy of a r.v. $Z \sim p(z)$:

$$H(Z) = -\mathbb{E}(\log(p(Z)))$$

- Consider the r.v. T with variance v, then

$$Z \sim \mathcal{N}(0,1) \quad \rightarrow \quad \max_{T} H(T)$$

- Mutual information for $S = (S_1, \ldots, S_d)^T$:

$$I(S) = \sum_{i=1}^{d} H(S_i) - H(S)$$

ICA via entropy method

- Property of the entropy : if $S = \mathbf{A}^{\mathbf{T}} X$

$$H(S) = H(X) + \log(|\det(\mathbf{A})|)$$

- One then has the following optimization problem:

$$\rightarrow \quad \min_{\mathbf{A}:\mathbf{A}^T\mathbf{A}=\mathbf{I}} I(\mathbf{A}^T X) = \sum_{i=1}^{d} H(S_i) - H(X)$$

- Interpretation : deviation from the Gaussian behavior (minimization of the entropy of the components)