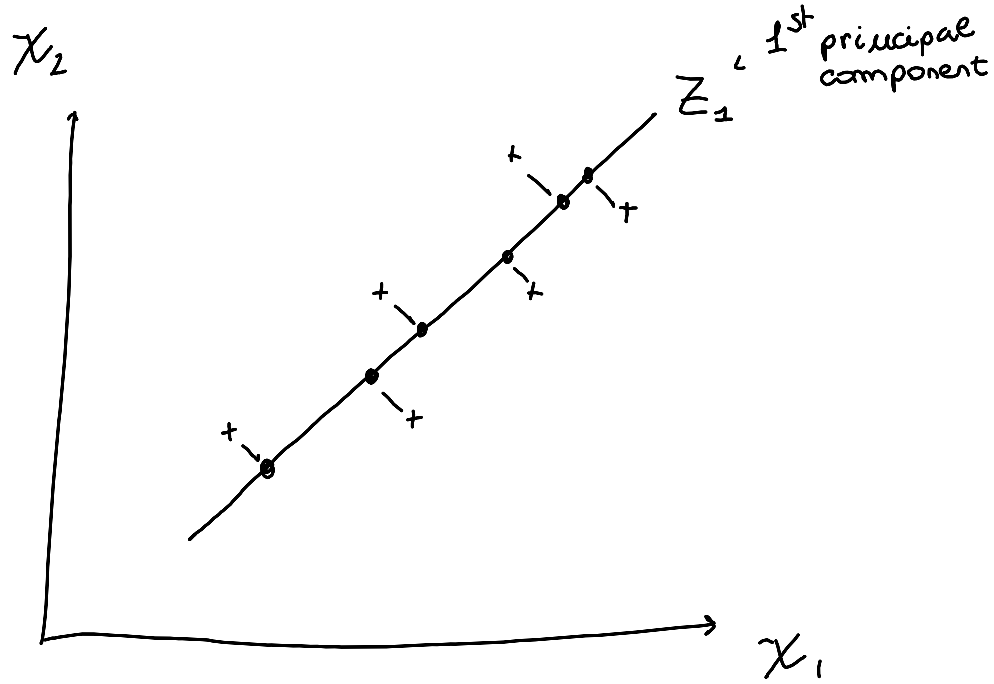
PCA:

projecting the data onto a low-dimensional space, without losing most of the information within the data



$$Z_1 := \varphi_1^T \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$$

$\Rightarrow$ direction of the feature space along which the data are <u>most</u> variable

In General:

$\mathbb{R}^n = (X_1, \dots, X_n)$ feature space

$\mathbb{R}^p = (Z_1, \dots, Z_p)$ principal component space $(p<n)$

- m training instances $\in \mathbb{R}^n$

$$\text{score} \rightarrow Z_{i1} = \varphi_1^T x_1 = \varphi_1^T \begin{pmatrix} x_{11} \\ \vdots \\ x_{n1} \end{pmatrix}$$

$X \in \mathbb{R}^{m \times n}$ data matrix

$\Phi \in \mathbb{R}^{n \times p}$ PC matrix

$$\Big[\quad = \overbrace{\begin{bmatrix} \text{---} \; x_1 \; \text{---} \\ \text{---} \; x_2 \; \text{---} \\ \\ \text{---} \; x_m \; \text{---} \end{bmatrix}}^{n \text{ features}} \overbrace{\begin{bmatrix} \varphi_1^T \\ \varphi_2^T \\ \vdots \\ \varphi_n^T \end{bmatrix}}^{p \text{ princ. comp}}$$

PCA:

1^{st} interpretation: Directions of highest variance

$X \in \mathbb{R}^{n \times p}$ data set ; p features.

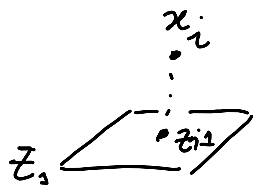
$E[X_i] = 0 \quad \forall i = 1, \dots, p$ (centered features)

then:

$$z_{i1} = \phi_{11} x_{i1} + \phi_{21} x_{i2} + \dots + \phi_{p1} x_{ip} =$$

SCORE

$$= \phi_1^T \begin{pmatrix} x_{i1} \\ \vdots \\ x_{ip} \end{pmatrix} =$$

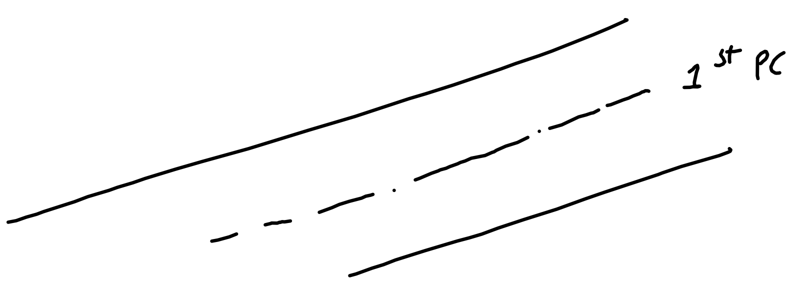


projection of the i-th training instance onto the 1^{st} princ. comp.

GOAL:

$$\max_{\phi_1^T} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \text{ with } \|\phi_1\|^2 = 1$$

$$\underbrace{\max_{\phi_1^T} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\phi_1^T x_i \right)^2 \right\}}_{\text{searching the direction upon which the project points have highest variance.}} = \underbrace{\max_{\phi_1^T} \mathbb{V}[\phi_1^T X]}_{\text{projection variance}}$$



PCA:

1st interpretation, geometric mess.

Repetat until you find the needed number of PC, adding the constraint the every new PC must be ⊥ others (⊥⊥). Aftere we have found k principal components the whole dataset can be projected upon the lower-dimensional princ. component space.

all projections on the 1st component.

$\boxed{k < p}$

$$\underbrace{\begin{array}{c} X_1 \quad \cdots \quad X_p \\ \begin{pmatrix} x_{11}, \dots, x_{1p} \\ x_{21}, \dots, x_{2p} \\ \vdots \qquad \vdots \\ x_{n1}, \dots, x_{np} \end{pmatrix} \end{array}}_{\text{original space}} \underbrace{\begin{pmatrix} \varphi_1 \ \varphi_2 \sim \varphi_k \end{pmatrix}}_{\text{feature weight matrix}} = \underbrace{\begin{pmatrix} z_{11}, \dots, z_{1k} \\ z_{21}, \dots, z_{2k} \\ \vdots \\ z_{n1}, \dots, z_{nk} \end{pmatrix}}_{\text{principal component space}}$$

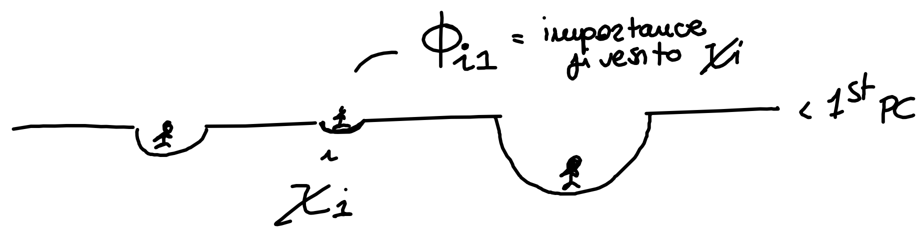
$z_{nk} \rightarrow$ score

$$\begin{pmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1k} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2k} \\ \vdots & \vdots & & \vdots \\ \varphi_{p1} & \varphi_{p2} & \cdots & \varphi_{pk} \end{pmatrix}$$

loading vector of the 1st component

Tells the importance given from that PC to the original p features.

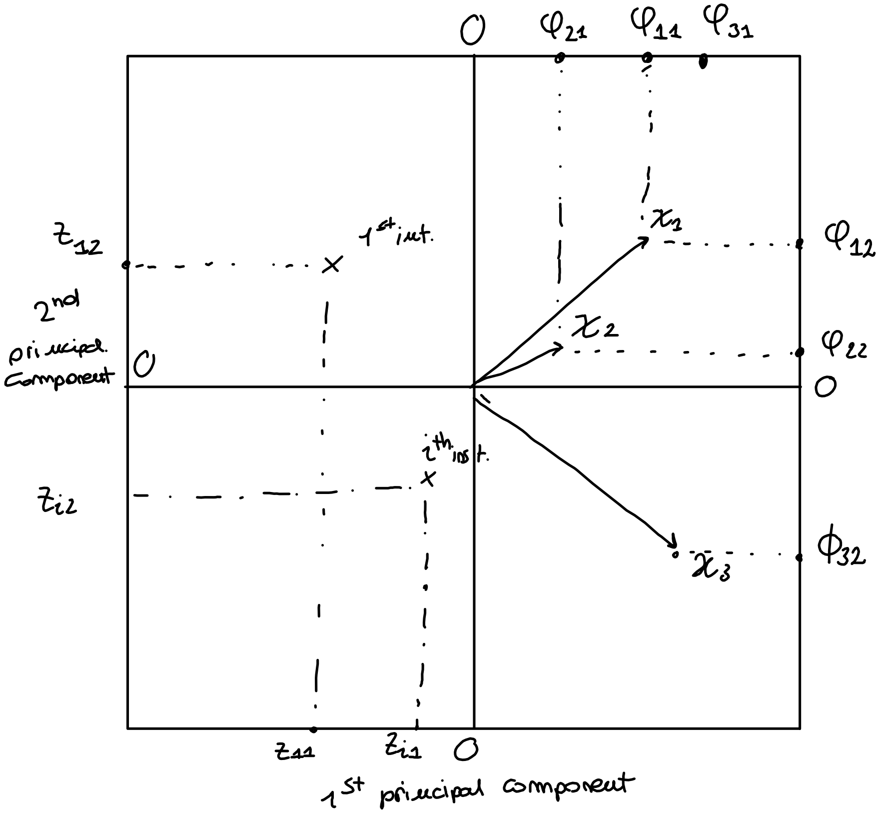
p-th feature weight (X_p) for the k-th PC.



PCA

1st interpretation: plotting.

The BIPLOT is a plot that shows the original data wrt the 1st and 2nd princ comp. It also shows the loadings for each feature.



$$\overbrace{\begin{pmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & \vdots & \vdots \\ \vdots & & \\ X_{n1} & X_{n2} & X_{n3} \end{pmatrix}}^{\text{original data}} \overbrace{\begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \\ \phi_{31} & \phi_{32} \end{pmatrix}}^{\text{load. matrix}} = \overbrace{\begin{pmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \\ \vdots & \\ z_{n1} & z_{n2} \end{pmatrix}}^{\text{projected point}}$$

projection of x_1 onto the 2nd component

new coordinates of the x_n train. instance

ϕ_1 loading vector of the first component

PCA:

2nd int: the principal components can be seen to be also the directions in the feature space that best approximate the data.

Specifically the same loading $(\phi_1, \dots, \phi_K)$ vectors can be shown to be the solution of the following optimization problem:

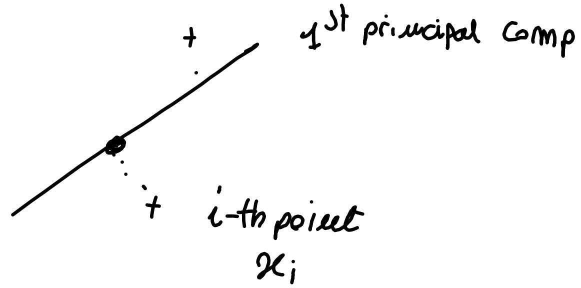
$$\min_{A,B} \left\{ \sum_{j=1}^{p} \sum_{i=1}^{n} \left(x_{ij} - \sum_{m=1}^{M} a_{im} b_{im} \right)^2 \right\}$$

IDEA:

each point of the training set x_i can be approximated by M products $(a_{im} \cdot b_{im})$

SOLUTION

The first M principal components scores $(a_{im} = z_{im})$ and loadings $(\phi_{im} = b_{im})$ solve the problem.



$$x_{ij} \simeq z_{i1}\phi_{i1} + z_{i2}\phi_{i2} + \cdots$$

Meaning that the projected point (z_1) onto the principal components space is the best least square approx. of x_i

PCA

% of variance explained

Each feature X_j can be seen as a random variable:

$\tilde{X}_j$ with $E[X_j] = \emptyset$ (if centered) and

a variance $V[\tilde{X}_j] = E[\tilde{x}_j]^2 - (\cancel{E[X_j]})^2$ 0

So the ~~total~~ variance in the data is:

$$\sum_{j=1}^{p} \mathrm{Var}(\tilde{X}_j) = \frac{1}{n} \sum_{j=1}^{p} \sum_{i=1}^{n} x_{ij}^2$$

Also a princ. comp. can be seen as a rand. var. derived from $\tilde{X}_1, \dots, \tilde{X}_p$.

$$\tilde{Z}_m = \Phi_m^T \begin{pmatrix} \tilde{X}_1 \\ \vdots \\ \tilde{X}_p \end{pmatrix} \quad \text{with realization:} \quad Z_{im} = \Phi_m^T \begin{pmatrix} x_{i1} \\ \vdots \\ x_{ip} \end{pmatrix}$$

So we have: $V[\tilde{Z}_m] = E[\tilde{Z}_m^2] - (E[\tilde{Z}])^2$

$$= \frac{1}{n} \sum_{i=1}^{n} z_{im}^2 - \left(\cancel{E[\Phi_m^T \begin{pmatrix} \tilde{X}_1 \\ \vdots \\ \tilde{X}_p \end{pmatrix}]} \right)^2 \quad 0$$

So the % of variance explained by the m-th principal comp is:

$$\frac{V[\tilde{Z}_m]}{\sum_{j=1}^{p} V[\tilde{X}_j]} = \frac{\sum_{i=1}^{n} z_{im}^2}{\sum_{j=1}^{p} \sum_{i=1}^{n} x_{ij}^2}$$

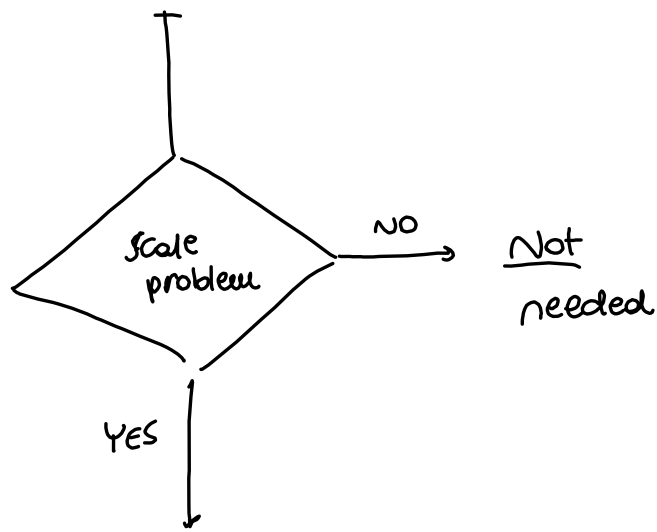
PCA

data preparation

1. Centering each feature at Ø.

Fundamental to make everything work

2. Feature scaling



Then it can be that $\mathbb{V}[\tilde{X}_j] \geq \mathbb{V}[\tilde{X}_i]$ $(i \neq j)$ for scale and not for any other reason. So an idea could be standardizing each feature: so that

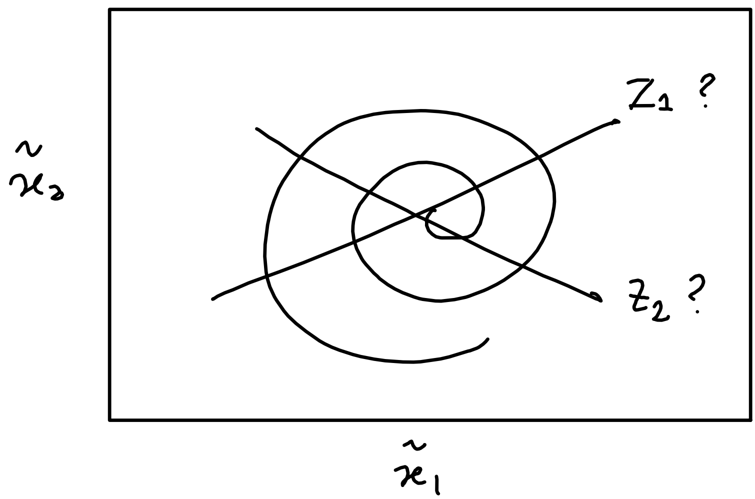
$$\tilde{X}_j \sim \left(E[\tilde{X}_j] = \emptyset;\ \mathbb{V}[\tilde{X}_j] = 1 \right)$$

Unless the 1st p.c. could capture the most variance just of some features, cause of their scales and not for some interesting phenomena

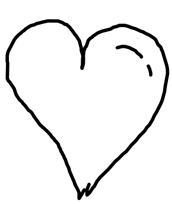
Kernel PCA

Normal PCA assumes that the variance within the data can be explained and viewed also on a lower-dimensional hyperplane.

What if the variance of data could be best understood on a non-linear lower-dimensional surface within the feature space?



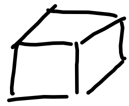
We could apply a kernel trick, hoping that in an higher-dim. space than the original feature space, the data variance could be best understood and explained on a linear surface. (look SVM)

possible  kernels

- → Linear
- → RBF Gaussian
- → Sigmoid [...]

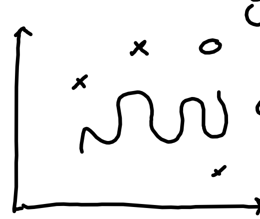
t - SNE : dim reduction algorithm widely used for imaging

high dimensional



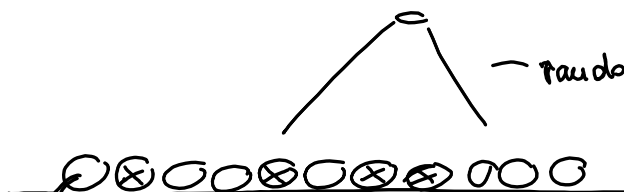
data

lower dimensional graph.



PROJECTION

preserves clusters



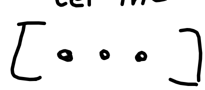
— random proj. onto LD

< lower dim space (LDS)

point in the LDS are:

repelled by points far from... ⇝ ○ ⇝ attracted : points near in the original space

let the



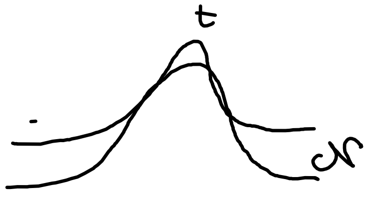
points move

clustered points in LDS

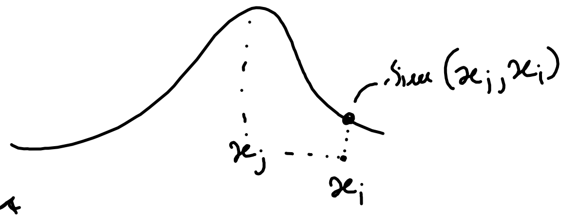
t-SNE: math behind

$\forall$ space:

point near/far in the scatter plot $\Longleftrightarrow$ $Sim(x_j, x_i)$ using:

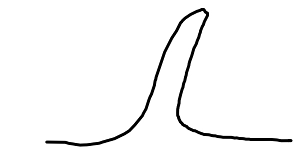


t is wider than N in general.

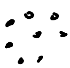


$$\mathcal{N}\left(x_j, \sigma^2\left(1/f_{\underline{X}}(\underline{x})\right)\right) \text{ or } t$$

high-density

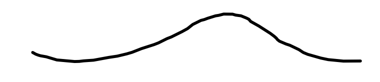


$\sigma^2 \ll$

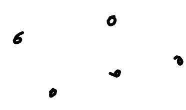


overall higher similarities

low-density



$\sigma^2 \gg$



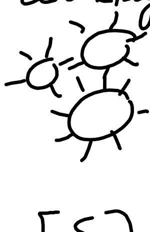
overall lower similarities.

similarity matrix

[S]

points moving to make [S] or. sp $\approx$ [S] LDS.

working:



$[S]_{\text{orig. space}} \triangleright [S]_{\text{low-dim space}}$

NOTE [S] LDS computed $\sim t$ to avoid clust clumping in LDS