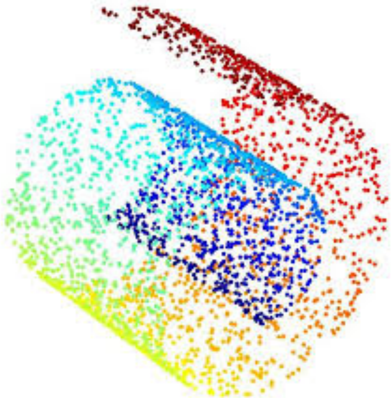
Data analysis

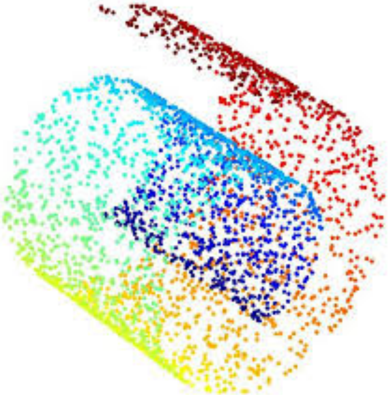
Multidimensional Scaling & Isomap

BENAZHA Hamed

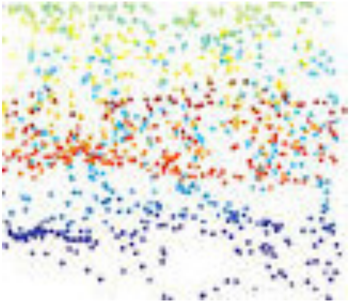
PCA - not always enough



PCA - not always enough

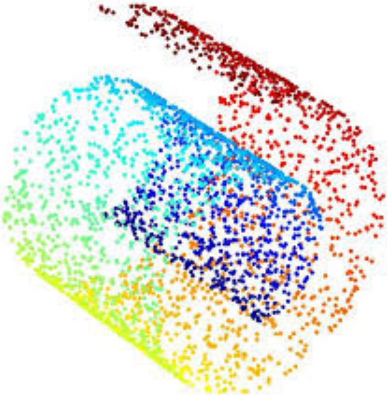


$\rightarrow$



PCA

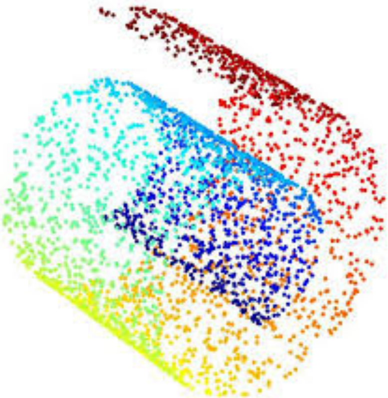
PCA - not always enough



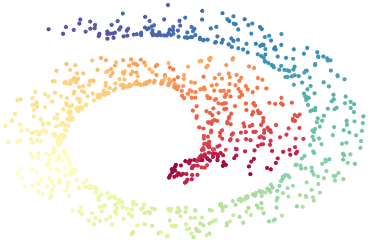
$\rightarrow$

"correct" solution?

PCA - not always enough



$\rightarrow$



"correct" solution?

Multidimensional scaling (MDS) - idea

"An MDS algorithm aims to place each object in N-dimensional space such that the between-object distances are preserved as well as possible."

Distances:

Data?

	x0	x1	x2	x3	x4
x0	0.	3.	2.24	3.8	1.8
x1	3.	0.	2.8	6.7	2.5
x2	2.24	2.8	0	4.5	3.5
x3	3.8	6.7	4.5	0	5.4
x4	1.8	2.5	3.5	5.4	0

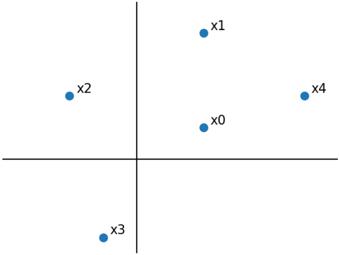
Multidimensional scaling (MDS) - idea

"An MDS algorithm aims to place each object in N-dimensional space such that the between-object distances are preserved as well as possible."

Distances:

	x0	x1	x2	x3	x4
x0	0.	3.	2.24	3.8	1.8
x1	3.	0.	2.8	6.7	2.5
x2	2.24	2.8	0	4.5	3.5
x3	3.8	6.7	4.5	0	5.4
x4	1.8	2.5	3.5	5.4	0

Data?



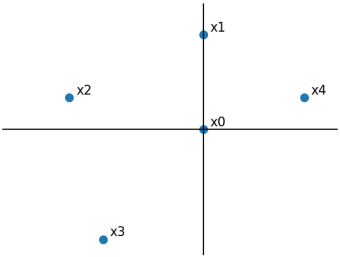
Multidimensional scaling (MDS) - idea

"An MDS algorithm aims to place each object in N-dimensional space such that the between-object distances are preserved as well as possible."

Distances:

	x0	x1	x2	x3	x4
x0	0.	3.	2.24	3.8	1.8
x1	3.	0.	2.8	6.7	2.5
x2	2.24	2.8	0	4.5	3.5
x3	3.8	6.7	4.5	0	5.4
x4	1.8	2.5	3.5	5.4	0

Data?



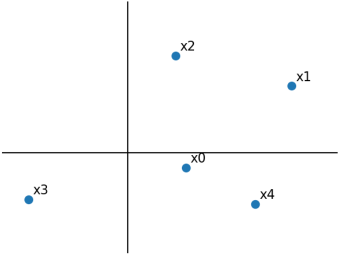
Multidimensional scaling (MDS) - idea

"An MDS algorithm aims to place each object in N-dimensional space such that the between-object distances are preserved as well as possible."

Distances:

	x0	x1	x2	x3	x4
x0	0.	3.	2.24	3.8	1.8
x1	3.	0.	2.8	6.7	2.5
x2	2.24	2.8	0	4.5	3.5
x3	3.8	6.7	4.5	0	5.4
x4	1.8	2.5	3.5	5.4	0

Data?



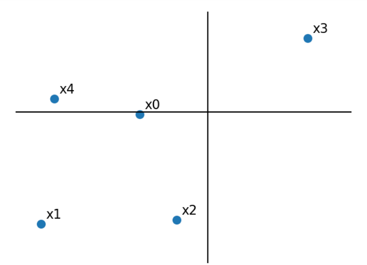
Multidimensional scaling (MDS) - idea

"An MDS algorithm aims to place each object in N-dimensional space such that the between-object distances are preserved as well as possible."

Distances:

	x0	x1	x2	x3	x4
x0	0.	3.	2.24	3.8	1.8
x1	3.	0.	2.8	6.7	2.5
x2	2.24	2.8	0	4.5	3.5
x3	3.8	6.7	4.5	0	5.4
x4	1.8	2.5	3.5	5.4	0

Data?



MDS - maths

- Remember squared euclidian distance between x and z:

$$d^2 = \sum_k (x_k - z_k)^2 = \sum_k (x_k^2 + z_k^2 - 2x_k z_k).$$

- If we subtract $x_k^2 + z_k^2$, divide it by two and multiply with 1 we are left with $\sum_k x_k z_k = x^\top z$.
- Matrix containing this type of data can be written as $\mathbf{XX^T}$ with data samples on rows of $\mathbf{X}$

MDS - maths

- Assume distance matrix Δ is given (Euclidian distance)
- If we double-center Δ^2 (subtract column and row means, add overall mean) we get a *positive semi-definite matrix* that we know has been generated with some $\mathbf{Z}\mathbf{Z}^\top$ (property of psd matrices).
- Eigendecomposition of a psd matrix is $\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T = \mathbf{V}\boldsymbol{\Lambda}^{1/2}\boldsymbol{\Lambda}^{1/2}\mathbf{V}^T = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$

MDS - algo

Classical MDA process:

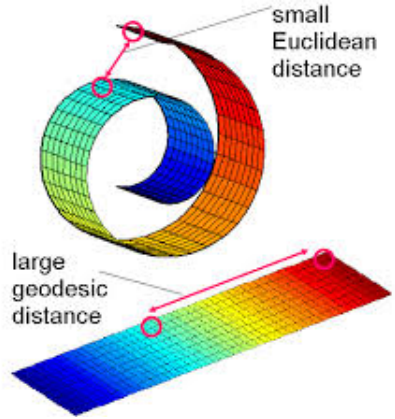
1. Form distance matrix Δ and calculate elementwise square
2. double-center this matrix $\Delta^* = -0.5\mathbf{J}\boldsymbol{\Delta}^{(2)}\mathbf{J}$ where $\mathbf{J}$ is centering matrix, $\mathbf{J} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ (and $\mathbf{I}$ is identity matrix, $\mathbf{1}$ is vector of ones).
3. Perform eigendecomposition of Δ^* to recover low-dimensional new data $\mathbf{X}$ as

$$\underbrace{\mathbf{X}}_{n\times p} = \underbrace{\mathbf{V}_p}_{n\times p}\underbrace{\Lambda_p^{1/2}}_{p\times p}.$$

   Here p largest eigenvalues are in diagonal of matrix Λ_p, and the corresponding eigenvectors on columns of $\mathbf{V}_p$ (so again, as in PCA, the largest eigenvalues are the most important ones).

Isomap - extension of MDS

Geodesic distances:



Excercises

- Again at www.celestium.eu/dana
- Open with:
  - google colab: go to colab.research.google.com and upload there the notebook you downloaded from my website
  - or with jupyter notebook from your own computer if you have python 3, jupyter notebook and necessary libraries like sklearn and matplotlib