Tutorial on Dimensionality Reduction

Some recents approches in statistics and Machine Learning

Julien Chiquet

UMR MIA Paris-Saclay, AgroParisTech, INRAE

May 25, 2023

https://jchiquet.github.io/

MIA
PARIS-SACLAY

INRAE
la science pour la vie, l'humain, la terre

General Introduction

Exploratory analysis of (modern) data sets

Assume a table with n individuals described by p features/variables

Questions

Look for **patterns** or **structures** to summarize the data by

- Finding groups of "similar" individuals
- Finding variables important for these data
- Performing visualization

Challenges

- Size data may be large ("big data": large n large p)
- Dimension data may be high dimensional (more variables than individual or $n \ll p$)
- Redundancy many variables may carry the same information
- Unsupervised we don't necessary know what we are looking after

An example in genetics: ‘snp’

Genetics variant in European population

Description: *medium/large data, high-dimensional*}

500, 000 Genetics variants (SNP – Single Nucleotide Polymorphism) for 3000 individuals (1 meter × 166 meter (height × width)

- SNP : 90 % of human genetic variations
- coded as 0, 1 or 2 (10, 1 or 2 allel different against the population reference)

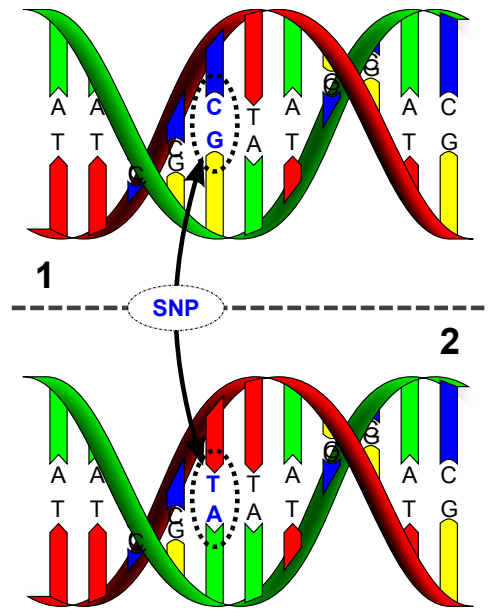


Figure 1: SNP (wikipedia)

Summarize 500,000 variables with 2 features

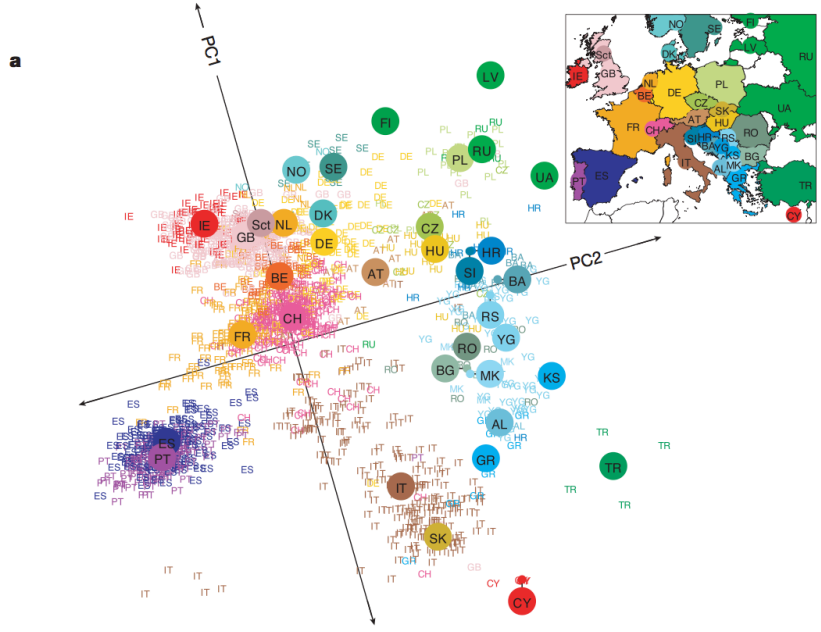


Figure 2: Dimension reduction + labels {source: Nature "Gene Mirror Geography Within Europe", 2008}

In the original messy $3,000 \times 500,000$ table, we may find - an extremely strong structure between individuals (**"clustering"**) - a very simple subspace where it is obvious (**"dimension reduction"**)

Theoretical argument: dimensionality Curse

Theorem (Folks theorem)

If $\mathbf{x}_1, \dots, \mathbf{x}_n$ in the hypercube of dimension p such that their coordinates are i.i.d then

$$p^{-1/2}\left(\max \|\mathbf{x}_i - \mathbf{x}_{i'}\|_2 - \min \|\mathbf{x}_i - \mathbf{x}_{i'}\|_2\right) = 0 + O\left(\sqrt{\frac{\log n}{p}}\right)$$

$$\frac{\max \|\mathbf{x}_i - \mathbf{x}_{i'}\|_2}{\min \|\mathbf{x}_i - \mathbf{x}_{i'}\|_2} = 1 + O\left(\sqrt{\frac{\log n}{p}}\right).$$

$\rightsquigarrow$ When p is large, all the points are almost equidistant\

Hopefully, the data **are not really leaving in p** dimension (think of the SNP example)

Dimension reduction: general goals

Main objective:

find a **low-dimensional representation** that captures the "essence" of (high-dimensional) data

Application in Machine Learning

Preprocessing, Regularization

- Compression, denoising, anomaly detection
- Reduce overfitting in supervised learning

Application in Statistics/Data analysis}

Better understanding of the data

- descriptive/exploratory methods
- visualization (difficult to plot and interpret $> 3d$!)

Dimension reduction: problem setup

Settings

- Training data : $\mathscr{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \in \mathbb{R}^p$, (i.i.d.)
- Space $\mathbb{R}^p$ of possibly high dimension ($n \ll p$)

Dimension Reduction Map

Construct a map Φ from the space $\mathbb{R}^p$ into a space $\mathbb{R}^q$ of smaller dimension:

$$\begin{aligned} \Phi : \quad & \mathbb{R}^p \to \mathbb{R}^q, q \ll p \\ & \mathbf{x} \mapsto \Phi(\mathbf{x}) \end{aligned}$$

How should we design/construct Φ?

Criterion

- Geometrical approach
- Reconstruction error
- Relationship preservation

Form of the map Φ

- **Linear** or non-linear ?
- tradeoff between **interpretability** and versatility ?
- tradeoff between high or **low** computational resource

Background: Principal Component Analysis

Outline

1 Geometric approach to PCA

2 Principal axes and variance maximization

3 Representation and interpretation

4 Additional tools and Complements

Cloud of observation in $\mathbb{R}^p$

Individuals can be represented in the variable space $\mathbb{R}^p$ as a point cloud

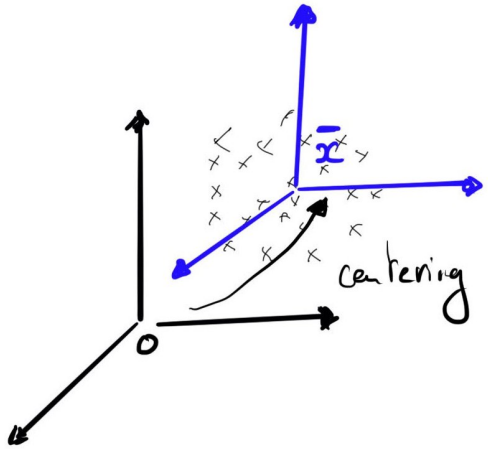


Figure 3: Example in $\mathbb{R}^3$

Center of Inertia

(or barycentrum, or empirical mean)

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i = \begin{pmatrix} \sum_{i=1}^{n} x_{i1}/n \\ \sum_{i=1}^{n} x_{i2}/n \\ \vdots \\ \sum_{i=1}^{n} x_{ip}/n \end{pmatrix}$$

We center the cloud $\mathbf{X}$ around $\mathbf{x}$ denote this by $\mathbf{X}^c$

$$\mathbf{X}^c = \begin{pmatrix} x_{11} - \bar{x}_1 & \dots & x_{1j} - \bar{x}_j & \dots & x_{1p} - \bar{x}_p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{i1} - \bar{x}_1 & \dots & x_{ij} - \bar{x}_j & \dots & x_{ip} - \bar{x}_p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n1} - \bar{x}_1 & \dots & x_{nj} - \bar{x}_j & \dots & x_{np} - \bar{x}_p \end{pmatrix}$$

Inertia and Variance

Total Inertia:

distance of the individuals to the center of the cloud

$$I_T = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p}(x_{ij} - \bar{x}_j)^2 = \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i - \bar{\mathbf{x}}\|^2 = \frac{1}{n}\sum_{i=1}^{n}\text{dist}^2(\mathbf{x}_i, \bar{\mathbf{x}})$$

Proportional to the total variance

Let $\hat{\mathbf{\Sigma}}$ be the empirical variance-covariance matrix

$$I_T = \frac{1}{n}\sum_{j=1}^{p}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2 = \sum_{j=1}^{p}\frac{1}{n}\|\mathbf{x}^j - \bar{x}_j\|^2 = \sum_{j=1}^{p}\mathbb{V}(\mathbf{x}^j) = \text{trace}\left(\hat{\mathbf{\Sigma}}\right)$$

⇝ Good representation has large inertia (much variability)

⇝ Large dispertion ∼ Large distances between points

Inertia with respect to an axis

The Inertia of the cloud wrt axe Δ is the sum of the distances between all points and their orthogonal projection on Δ.

$$I_\Delta = \frac{1}{n} \sum_{i=1}^{n} \text{dist}^2(\mathbf{x}_i, \Delta)$$

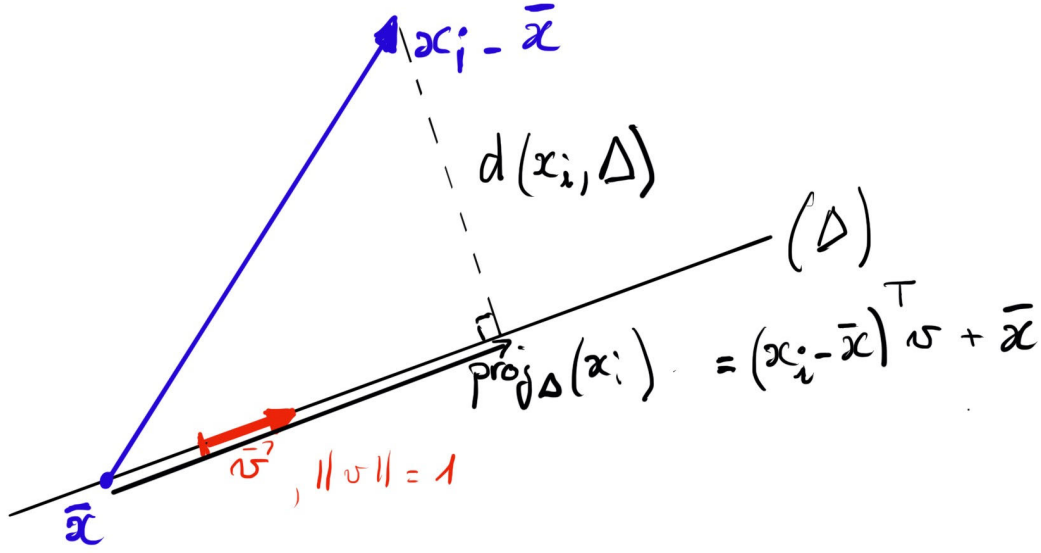
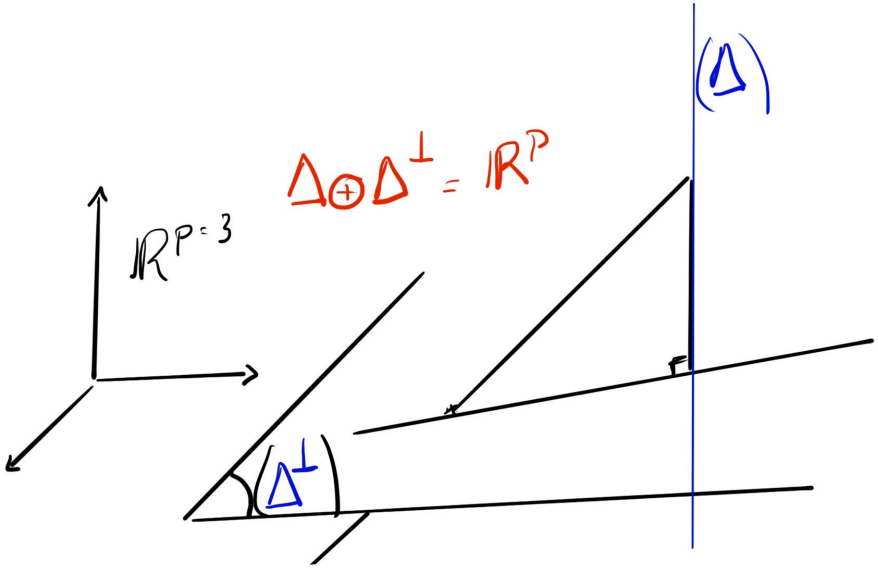


Figure 4: Projection of $\mathbf{x}_i$ onto a line Δ passing through $\bar{\mathbf{x}}$

Decomposition of total Inertia (1)

Let $\Delta^\perp$ be the orthogonal subspace of Δ in $\mathbb{R}^p$



Theorem (Huygens)

A consequence of the above (Pythagoras Theorem) is the decomposition of the following total inertia:

$$I_T = I_\Delta + I_{\Delta^\perp}$$

By projecting the cloud $\mathbf{X}$ *onto* Δ*, with loss the inertia measured by* $\Delta^\perp$

Decomposition of total Inertia (2)

Consider only subspaces with dimension 1 (that is, lines or axes). We can decompose $\mathbb{R}^p$ as the sum of p othogonal axis.

$$\mathbb{R}^p = \Delta_1 \oplus \Delta_2 \oplus \cdots \oplus \Delta_p$$

⇝ These axes form a new basis for representing the point cloud.

Theorem (Huygens)

$$I_T = I_{\Delta_1} + I_{\Delta_2} + \cdots + I_{\Delta_p}$$

Outline

1 Geometric approach to PCA

2 Principal axes and variance maximization

3 Representation and interpretation

4 Additional tools and Complements

Finding the best axis (1)

Definition of the problem

- The best axis Δ_1 is the "closest" to the point cloud
- Inertia of Δ_1 measures the distance between the data and Δ_1
- Δ_1 is defined by the director vector $\mathbf{u}_1$, such as $\|\mathbf{u}_1\| = 1$
- $\Delta_1^\perp$ is defined by the normal vector $\mathbf{u}_1$, such as $\|\mathbf{u}_1\| = 1$

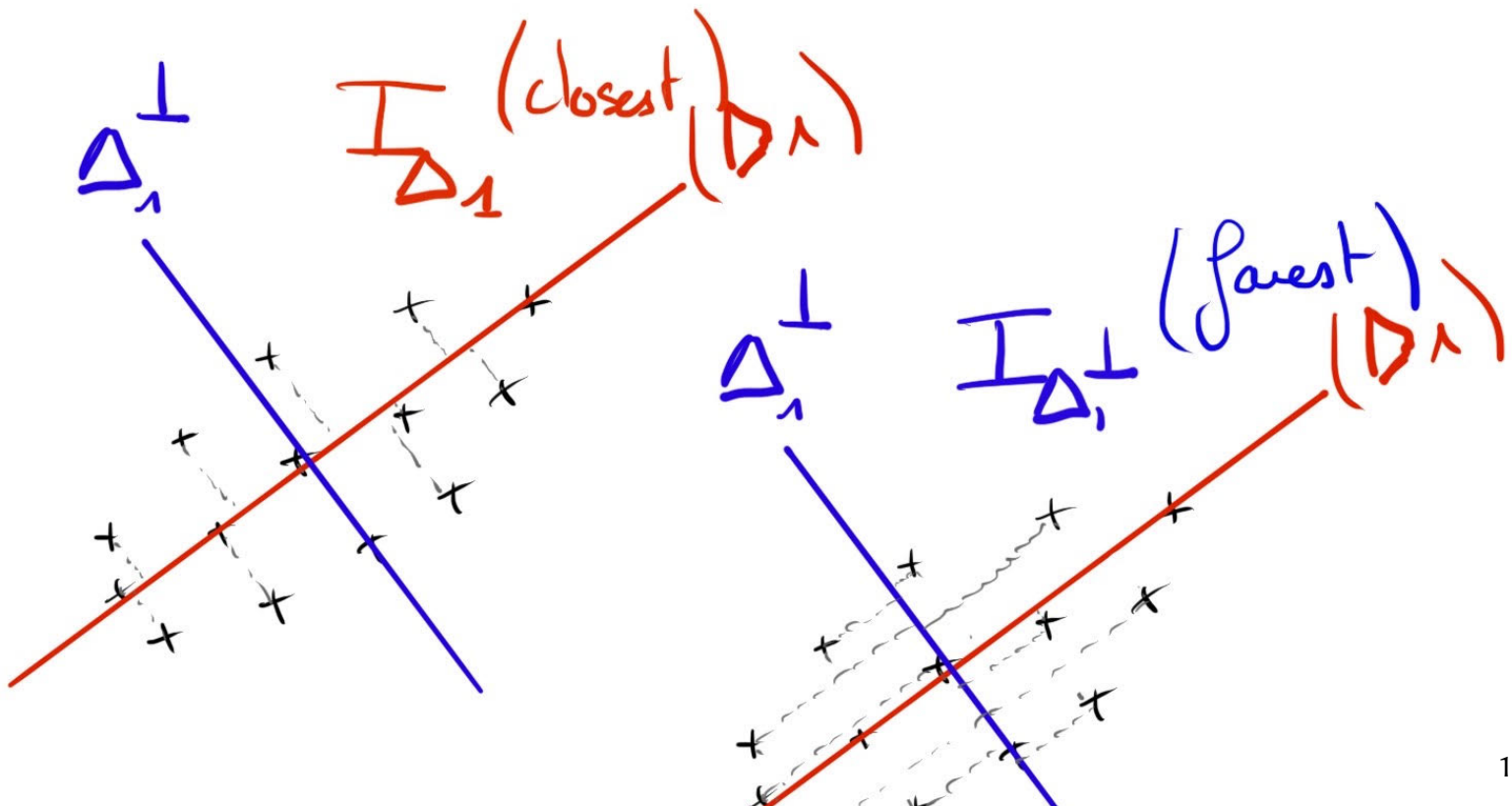
$\rightsquigarrow$ The best axis Δ_1 is the one with the minimal Inertia.

Finding the best axis (2)

Stating the optimization problem

Since $\Delta_1 \oplus \Delta_1^\perp = \mathbb{R}^p$ and $I_T = I_{\Delta_1} + I_{\Delta_1^\perp}$, then

$$\underset{\mathbf{u}\in\mathbb{R}^p : \|\mathbf{u}\|=1}{\text{minimize}} \; I_{\Delta_1} \;\Leftrightarrow\; \underset{\mathbf{u}\in\mathbb{R}^p : \|\mathbf{u}\|=1}{\text{maximize}} \; I_{\Delta_1^\perp}$$



Finding the best axis (3)

Stating the problem (algebraically)

Find $\mathbf{u}_1; \|\mathbf{u}_1\| = 1$ that maximizes

$$
\begin{aligned}
I_{\Delta_1^\perp} &= \frac{1}{n} \sum_{i=1}^{n} \text{dist}(\mathbf{x}_i, \Delta_1^\perp)^2 \\
&= \frac{1}{n} \sum_{i=1}^{n} \mathbf{u}_1^\top (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top \mathbf{u}_1 \\
&= \mathbf{u}_1^\top \left(\sum_{i=1}^{n} \frac{1}{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top \right) \mathbf{u}_1 \\
&= \mathbf{u}_1^\top \hat{\mathbf{\Sigma}} \mathbf{u}_1
\end{aligned}
$$

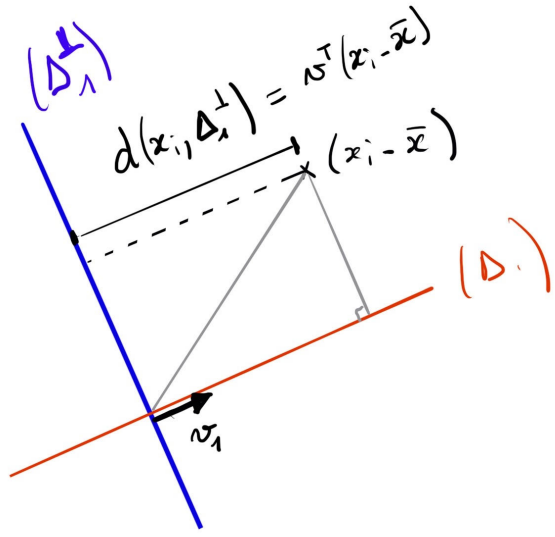


Figure 5: Geometrical insight

Finding the best axis (4)

We solve a simple constraint maximization problem with the method of Lagrange multipliers:

$$\underset{\mathbf{u}_1 : \|\mathbf{u}_1\|=1}{\text{maximize}}\, \mathbf{u}_1^\top \hat{\mathbf{\Sigma}} \mathbf{u}_1 \Leftrightarrow \underset{\mathbf{u}_1 \in \mathbb{R}^p, \lambda_1 > 0}{\text{maximize}}\, \mathbf{u}_1^\top \hat{\mathbf{\Sigma}} \mathbf{u}_1 - \lambda_1 (\|\mathbf{u}_1\|^2 - 1)$$

By straightforward (vector) differentiation, an using that $\mathbf{u}_1^\top \mathbf{u}_1 = 1$

$$\begin{cases} 2\hat{\mathbf{\Sigma}}\mathbf{u}_1 - 2\lambda_1 \mathbf{u}_1 = 0 \\ \mathbf{u}_1^\top \mathbf{u}_1 - 1 = 0 \end{cases} \Leftrightarrow \begin{cases} \hat{\mathbf{\Sigma}}\mathbf{u}_1 = \lambda_1 \mathbf{u}_1 \\ \mathbf{u}_1^\top \hat{\mathbf{\Sigma}} \mathbf{u}_1 = \lambda_1 \mathbf{u}_1^\top \mathbf{u}_1 = \lambda_1 = I_{\Delta_1}^\perp \end{cases}$$

- $\mathbf{u}_1$ is the first (normalized) eigen vector of $\hat{\mathbf{\Sigma}}$
- λ_1 is the first eigen value of $\hat{\mathbf{\Sigma}}$

$\boxtimes$ **Δ_1 is defined by the first eigen vector of $\hat{\Sigma}$**

$\boxtimes$ **Variance "carried" by Δ_1 is equal to the largest eigen value of $\hat{\Sigma}$**

Finding the following axes

Second best axis

Find Δ_2 with dimension 1, director vector $\mathbf{u}_2$ orthogonal to Δ_1 solving

$$\underset{\mathbf{u}_2 \in \mathbb{R}^p}{\text{maximize}}\, I_{\Delta_2^\perp} = \mathbf{u}_2^\top \hat{\mathbf{\Sigma}} \mathbf{u}_2, \quad \text{with } \|\mathbf{u}_2\| = 1, \mathbf{u}_1^\top \mathbf{u}_2 = 0.$$

$\rightsquigarrow$ $\mathbf{u}_2$ is the second eigen vector of $\hat{\mathbf{\Sigma}}$ with eigen value λ_2

And so on!

PCA is roughly a matrix factorisation problem

$$\hat{\mathbf{\Sigma}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top, \quad \mathbf{U} = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2, & \dots & \mathbf{u}_p \end{pmatrix}, \quad \mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_p)$$

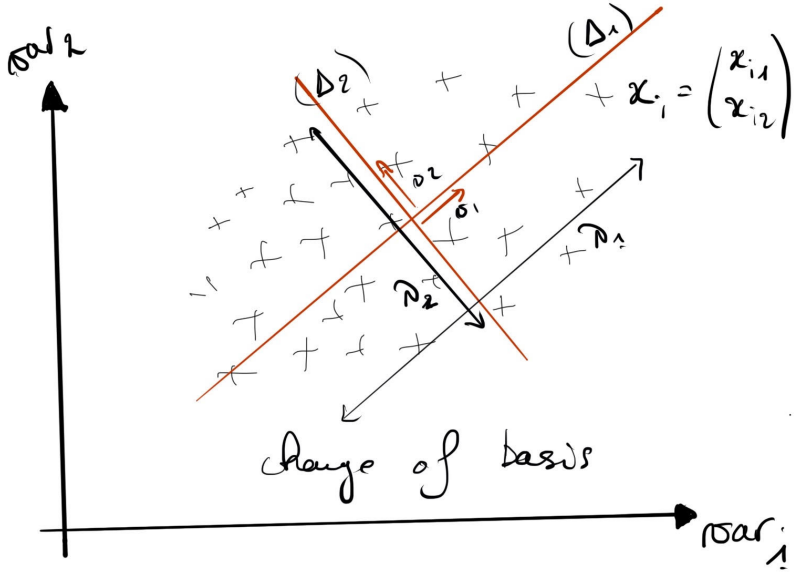
- $\mathbf{U}$ is an orthogonal matrix of normalized eigen vectors.
- $\mathbf{\Lambda}$ is diagonal matrix of ordered eigen values.

Interpretation in $\mathbb{R}^p$

$\mathbf{U}$ describes a new orthogonal basis and a rotation of data in this basis

$\rightsquigarrow$ PCA is an appropriate rotation on axes that maximizes the variance

$$\begin{cases} \Delta_1 & \oplus & \dots & \oplus & \Delta_p \\ \mathbf{u}_1 & \perp & \dots & \perp & \mathbf{u}_p \\ \lambda_1 & > & \dots & > & \lambda_p \\ I_{\Delta_1^\perp} & > & \dots & > & I_{\Delta_p^\perp} \end{cases}$$



Outline

1 Geometric approach to PCA

2 Principal axes and variance maximization

3 Representation and interpretation

4 Additional tools and Complements

Contribution of each axis and quality of the representation}

Δ_k is carrying inertia/variance defined by its orthogonal, thus

$$I_T = I_{\Delta_1^\perp} + \cdots + I_{\Delta_p^\perp} = \lambda_1 + \cdots + \lambda_p$$

Relative contribution of axis k

$$\text{contrib}(\Delta_k) = \frac{\lambda_k}{\sum_{k=1}^{p} \lambda_j} = \frac{\lambda_k}{\text{trace}\left(\hat{\Sigma}\right)} \times 100$$

$\rightsquigarrow$ Percentage of explained inertia/variance explained

Global quality of the representation on the first k axes

$$\text{contrib}(\Delta_1, \ldots, \Delta_k) = \frac{\lambda_1 + \cdots + \lambda_k}{\text{trace}\left(\hat{\Sigma}\right)} \times 100$$

A few axes may explain a large proportion of the total variance.

$\rightsquigarrow$ This paves the way for dimension reduction

Contribution of each axis and quality of the representation}

Δ_k is carrying inertia/variance defined by its orthogonal, thus

$$I_T = I_{\Delta_1^\perp} + \cdots + I_{\Delta_p^\perp} = \lambda_1 + \cdots + \lambda_p$$

Relative contribution of axis k

$$\text{contrib}(\Delta_k) = \frac{\lambda_k}{\sum_{k=1}^{p} \lambda_j} = \frac{\lambda_k}{\text{trace}\left(\hat{\Sigma}\right)} \times 100$$

$\rightsquigarrow$ Percentage of explained inertia/variance explained

Global quality of the representation on the first k axes

$$\text{contrib}(\Delta_1, \ldots, \Delta_k) = \frac{\lambda_1 + \cdots + \lambda_k}{\text{trace}\left(\hat{\Sigma}\right)} \times 100$$

A few axes may explain a large proportion of the total variance.

$\rightsquigarrow$ This paves the way for dimension reduction

Individuals: representation in the new basis

Projection

The projection of $\mathbf{x}_i$ onto axis Δ_k is $c_{ik}\mathbf{u}_k$, with

$$c_{ik} = \mathbf{u}_k^\top(\mathbf{x}_i - \bar{\mathbf{x}}),$$

the coordinate of i in the basis $\mathbf{u}_k$ (along axis Δ_k).

Coordinates

Coordinates of i in the new basis $\{\mathbf{u}_1, \dots, \mathbf{u}_p\}$ is thus

$$\mathbf{c}_i = (\mathbf{U}^\top(\mathbf{x}_i - \bar{\mathbf{x}}))^\top = (\mathbf{x}_i - \bar{\mathbf{x}})^\top\mathbf{U} = \mathbf{X}_i^c\mathbf{U}, \quad \mathbf{c}_i \in \mathbb{R}^p.$$

- $\mathbf{U}$ are often the called the **loadings**, or **weights**
- $\mathbf{c}_i$ are the **scores** or **coordinates** in the new space for the individuals

Warning: about distances after projection

Close projection doesn't mean close individuals!

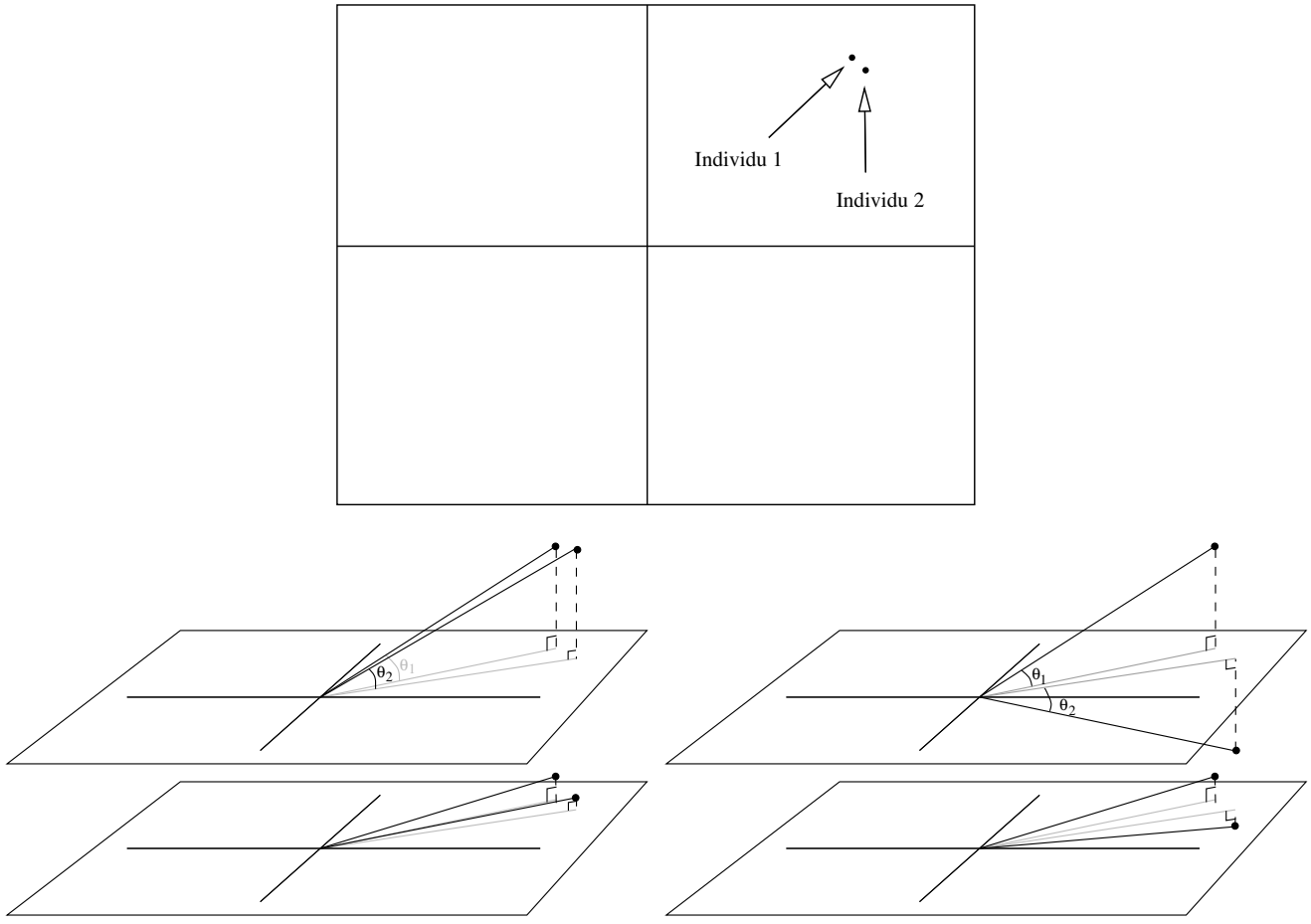


Figure 6: Same projections but different situations (source: E. Matzner)

⇝ Only work when individuals are well represented in the lower space

Individual: representation

Quality

- An individual i is well represented by Δ_k if it is close to this axis.
- In other word, vector $\mathbf{x}_i - \bar{\mathbf{x}}$ and $\mathbf{u}_k$ are close to collinear

Use the cosine of the angle between $\mathbf{x}_i - \bar{\mathbf{x}}$ and $\mathbf{u}_k$ to measure collinearity:

$$\cos^2(\theta_{ik}) = \frac{\left(\mathbf{u}_k^\top(\mathbf{x}_i - \bar{\mathbf{x}})\right)^2}{\|\mathbf{x}_i - \bar{\mathbf{x}}\|^2 \|\mathbf{u}_k\|^2}$$

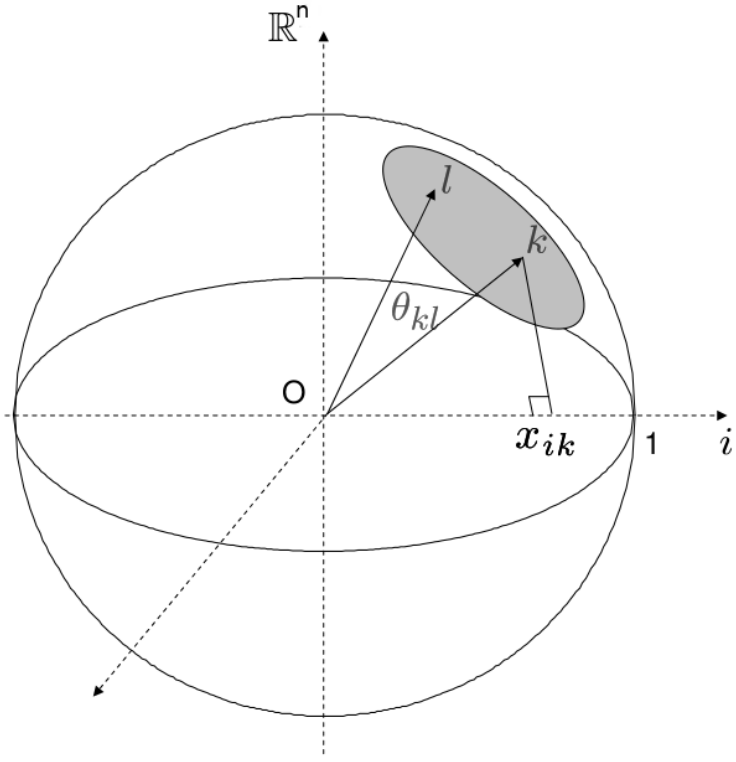
Contribution

- Inertia "explained" by Δ_k is inertia of $\Delta_k^\perp$
- $I_{\Delta_k^\perp} = n^{-1} \sum_{i=1}^n \text{dist}^2(\Delta_k^\perp, \mathbf{x}_i)$

Contribution is the proportion of variance/inertia carried by individual i:

$$n^{-1}\text{dist}^2(\Delta_k^\perp, \mathbf{x}_i) \qquad \left(\mathbf{u}_k^\top(\mathbf{x}_i - \bar{\mathbf{x}})\right)^2$$

Cloud of variables



Direct equivalence between geometry and statistics (collinearity ≡ correlation)

$$\cos(\theta_{kl}) = \frac{\langle \mathbf{x}^k, \mathbf{x}^\ell \rangle}{\|\mathbf{x}^k\| \|\mathbf{x}^\ell\|} = \rho(\mathbf{x}^k, \mathbf{x}^\ell)$$

Principal Components

Dual representation

A symmetric reasoning can be made in $\mathbb{R}^n$ for the variables, like with the individuals in $\mathbb{R}^p$.
$\rightsquigarrow$ New axes are linear combinaison of the original variables, which can be seen as **new variables** in the new latent space

Principal component

It is the linear combinasion formed by the orginal variables with weights given by the loadings $\mathbf{u}_k = (u_{k1}, \dots, u_{kj}, \dots, u_{kp})$

$$\mathbf{f}_k = \sum_{j=1}^{p} u_{kj}(\mathbf{x}^j - \bar{x}_j) = \mathbf{X}^c \mathbf{u}_k, \quad \mathbf{f}_k \in \mathbb{R}^n$$

Sometimes called "factors" in factor analysis, as latent (hidden) variables.

Variable representation in the new space

Connection with original variables

- essential for interpretation
- answer to the question: how to read the axes of the individual map
- use correlation to measure connection to original variable

$$\mathbb{V}(\mathbf{f}_k) = \frac{1}{n}\mathbb{V}(\mathbf{X}^c\mathbf{u}_k) = \mathbf{u}_k^\top \frac{1}{n}(\mathbf{X}^c)^\top \mathbf{X}^c\mathbf{u}_k = \mathbf{u}_k^\top \hat{\mathbf{\Sigma}}\mathbf{u}_k = \lambda_k$$

$$\text{cov}(\mathbf{f}_k, (\mathbf{x}^j - \bar{x}_j)) = \mathbf{u}_k \top \mathbf{X}^{c\top}\mathbf{X}^c e_j = \mathbf{u}_k \lambda_k e_j = \lambda_k \mathbf{u}_{kj}$$

$$\text{cor}(\mathbf{f}_k, (\mathbf{x}^j - \bar{x}_j)) = \sqrt{\frac{\lambda_k}{\mathbb{V}(\mathbf{x}^j)}}\mathbf{u}_{kj}$$

Warning: about angle after projection

Close projection doesn't mean close variable!

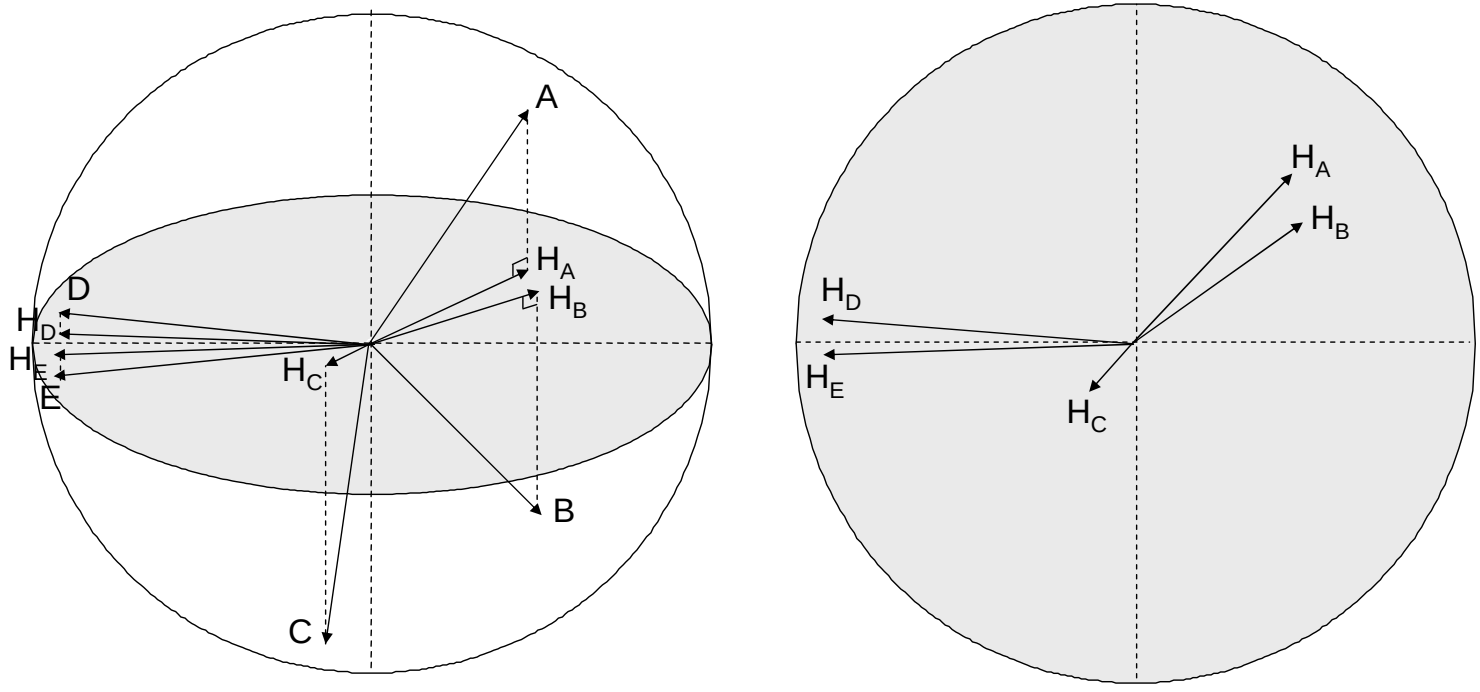


Figure 7: Same angle but different situations {(source: J. Josse)}

⇝ Only work when variables are well represented in the latent space

Variable representation

Quality

- An variable j is well represented by Δ_k if its projection is close to $\mathbf{f}_k$.
- High collinearity means high absolute correlation and high cosine.
- use cosine to the square of the angle between the original and new variables.

$\rightsquigarrow$ The projection of j must be close to the boundary of the correlation circle

Contribution

Similarly to individuals, we can measure the contribution of the original variables to the construction of the new ones.

Outline

1 Geometric approach to PCA

2 Principal axes and variance maximization

3 Representation and interpretation

4 Additional tools and Complements

Unifying view of variables and individuals

Principal components

The full matrix of principal component connects individual coordinates to latent factors:

$$\text{PC} = \mathbf{X}^c\mathbf{U} = \begin{pmatrix} \mathbf{f}_1 & \mathbf{f}_2 & \dots & \mathbf{f}_p \end{pmatrix} = \begin{pmatrix} \mathbf{c}_1^\top \\ \mathbf{c}_2^\top \\ \dots \\ \mathbf{c}_n^\top \end{pmatrix}$$

- new variables (latent factor) are seen column-wise
- new coordinates are seen row-wise

⇝ Everything can be interpreted on a single plot, called the biplot

Reconstruction formula

Recall that $\mathbf{F} = (\mathbf{f}_1, \ldots, \mathbf{f}_p)$ is the matrix of Principal components. Then,

- $\mathbf{f}_k = \mathbf{X}^c \mathbf{u}_k$ for projection on axis k
- $\mathbf{F} = \mathbf{X}^c \mathbf{U}$ for all axis.

Using orthogonality of $\mathbf{U}$, we get back the original data as follows, without loss ($\mathbf{U}^T$ performs the inverse rotation of $\mathbf{U}$):

$$\mathbf{X}^c = \mathbf{F}\mathbf{U}^\top$$

We obtain an approximation $\tilde{\mathbf{X}}^c$ (compression) of the data $\mathbf{X}^c$ by considering a subset $\mathcal{S}$ of PC, typically $\mathcal{S} = 1, \ldots, q$ with $q \ll p$.

$$\tilde{\mathbf{X}}^c = \mathbf{F}_{\mathcal{S}} \mathbf{U}_{\mathcal{S}}^\top = \mathbf{X}^c \mathbf{U}_{\mathcal{S}} \mathbf{U}_{\mathcal{S}}^\top$$

$\rightsquigarrow$ This is a rank-q approximation of $\mathbf{X}$ (information captured by the first q axes).

PCA (and linear methods) limitations

Do not account for ‘complex’ data distribution

- PCA is tied to a hidden **Gaussian assumption**
- Fails with **Count data**
- Fails with **Skew data**
- Linear methods like PCA are robust but badly shaped for complex geometries
- High-dim. datas are characterized by multiscale properties (local / global structures)

Possible solutions

- Probabilistic (non Gaussian) models
- Need transformed (non-linear) input space (preserving local characteristics of distances)

Dimension reduction: revisiting the problem setup

Settings

- Training data : $\mathscr{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \in \mathbb{R}^p$, (i.i.d.)
- Space $\mathbb{R}^p$ of possibly high dimension ($n \ll p$)

Dimension Reduction Map

Construct a map Φ from the space $\mathbb{R}^p$ into a space $\mathbb{R}^q$ of smaller dimension:

$$\begin{aligned} \Phi : \quad & \mathbb{R}^p \to \mathbb{R}^q, q \ll p \\ & \mathbf{x} \mapsto \Phi(\mathbf{x}) \end{aligned}$$

How should we design/construct Φ?

Geometrical approach

(**see slides on PCA**)

Idea to go beyond linear approaches

- Modify the model by amending the **reconstruction error**
- Focus on **Relationship preservation**

Form of the map Φ

- Linear or **non-linear ?**
- tradeoff between interpretability and **versatility ?**
- tradeoff between **high** or low computational resource

Reconstruction error approach

1. Construct a map Φ from the space $\mathbb{R}^p$ into a space $\mathbb{R}^q$ of smaller dimension:

$$\begin{aligned} \Phi : \quad & \mathbb{R}^p \to \mathbb{R}^q, q \ll p \\ & \mathbf{x} \mapsto \Phi(\mathbf{x}) = \tilde{\mathbf{x}} \end{aligned}$$

2. Construct $\tilde{\Phi}$ from $\mathbb{R}^q$ to $\mathbb{R}^p$ (reconstruction formula)
3. Control an error ϵ between $\mathbf{x}$ and its reconstruction $\hat{\mathbf{x}} = \tilde{\Phi}(\Phi(\mathbf{x}))$

For instance, the error measured with the Frobenius between the original data matrix $\mathbf{X}$ and its approximation:

$$\epsilon(\mathbf{X}, \hat{\mathbf{X}}) = \left\|\mathbf{X} - \hat{\mathbf{X}}\right\|_F^2 = \sum_{i=1}^{n} \left\|\mathbf{x}_i - \tilde{\Phi}(\Phi(\mathbf{x}_i))\right\|^2$$

Reinterpretation of PCA

PCA model

Let $\mathbf{V}$ be a $p \times q$ matrix whose columns are of q orthonormal vectors.

$$\Phi(\mathbf{x}) = \mathbf{V}^\top(\mathbf{x} - \boldsymbol{\mu}) = \tilde{\mathbf{x}}$$

$$\mathbf{x} \simeq \tilde{\Phi}(\tilde{\mathbf{x}}) = \boldsymbol{\mu} + \mathbf{V}\tilde{\mathbf{x}}$$

↝ Model with **Linear assumption + ortho-normality constraints**

PCA reconstruction error

$$\underset{\boldsymbol{\mu} \in \mathbb{R}^p, \mathbf{V} \in \mathcal{O}_{p,q}}{\text{minimize}} \sum_{i=1}^{n} \left\| (\mathbf{x}_i - \boldsymbol{\mu}) - \mathbf{V}\mathbf{V}^\top(\mathbf{x}_i - \boldsymbol{\mu}) \right\|^2$$

Solution (explicit)

- $\boldsymbol{\mu} = \bar{\mathbf{x}}$ the empirical mean
- $\mathbf{V}$ an orthonormal basis of the space spanned by the q first eigenvectors of the empirical covariance matrix

Important digression: SVD

Singular Value Decomposition (SVD)

The SVD of $\mathbf{M}$ a $n \times p$ matrix is the factorization given by

$$\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^\top,$$

where $r = \min(n, p)$ and

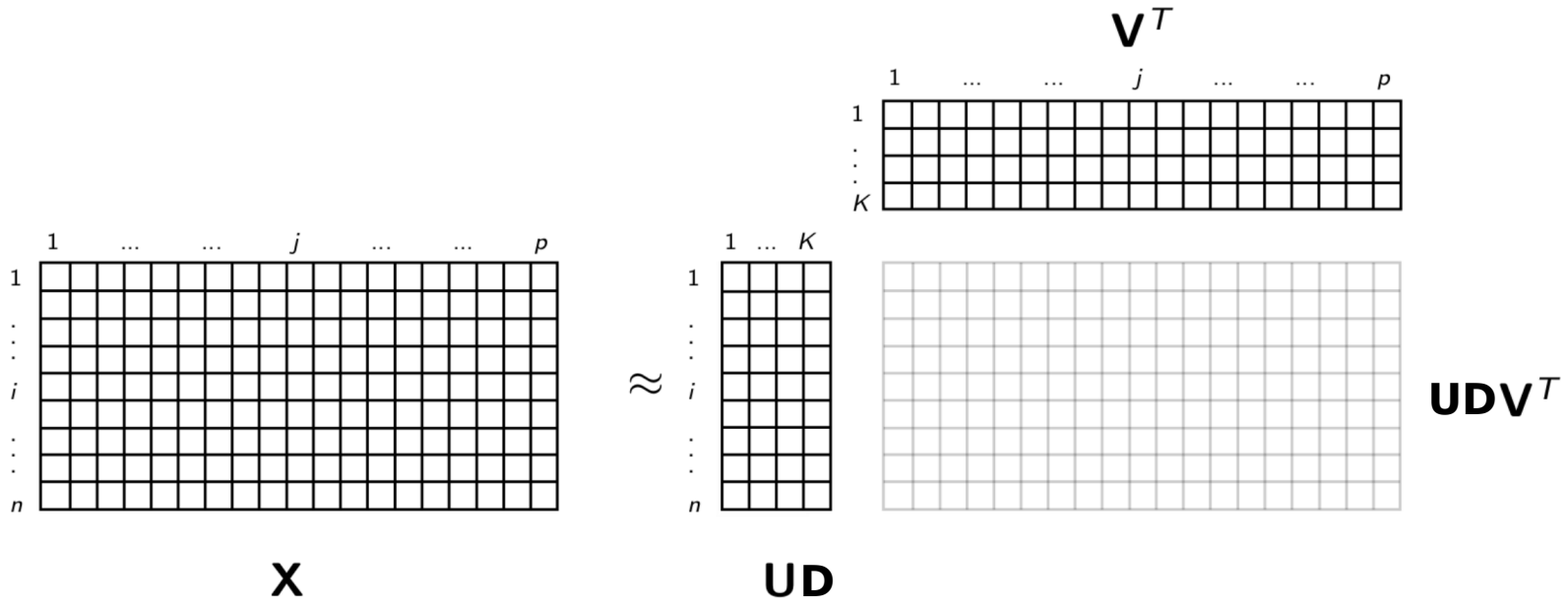
- $\mathbf{D}_{r \times r} = \text{diag}(\delta_1, ...\delta_r)$ is the diagonal matrix of singular values.
- $\mathbf{U}$ is orthonormal, whose columns are eigen vectors of $(\mathbf{M}\mathbf{M}^T)$
- $\mathbf{V}$ is orthonormal whose columns are eigen vectors of $(\mathbf{M}^T\mathbf{M})$

  ⇝ Time complexity in $\mathcal{O}(npqr)$ (less when $k \ll r$ components are required)

Connection with eigen decomposition of the covariance matrix

$$\begin{aligned}\mathbf{M}^\top\mathbf{M} &= \mathbf{V}\mathbf{D}\mathbf{U}^\top\mathbf{U}\mathbf{D}\mathbf{V}^\top \\ &= \mathbf{V}\mathbf{D}^2\mathbf{V}^\top = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top\end{aligned}$$

PCA solution is given by SVD of the centered data matrix



Since $\tilde{\mathbf{X}} = \mathbf{X}^c\mathbf{V} = \mathbf{U}\mathbf{D}\mathbf{V}^\top\mathbf{V} = \mathbf{U}\mathbf{D}$, PCA can be rephrased as

$$\hat{\mathbf{X}}^c = \mathbf{F}\mathbf{V}^\top = \underset{\mathbf{F}\in\mathscr{M}_{n,q},\mathbf{V}\in\mathcal{O}_{p,q}}{\arg\min} \left\|\mathbf{X}^c - \mathbf{F}\mathbf{V}^\top\right\|_F^2 \text{ with } \|\mathbf{A}\|_F^2 = \sum_{ij} a_{ij}^2,$$

$\left.\tilde{\mathbf{X}} \in \mathbb{R}^{n\times q}, \mathbf{V} \in \mathbb{R}^{p\times q}\right\}$ Best linear low-rank representation of $\mathbf{X}$

Non-negative Matrix Factorization – NMF

Setup

Assume that $\mathbf{X}$ contains only non-negative entries (i.e. ≥ 0).

Model

Linear assumption + non-negativity constraints on both $\mathbf{V}$ and $\tilde{\mathbf{x}}$

$$\Phi(\mathbf{x}) = \mathbf{V}^\top \mathbf{x} = \tilde{\mathbf{x}}$$

$$\mathbf{x} \simeq \tilde{\Phi}(\tilde{\mathbf{x}}) = \mathbf{V}\tilde{\mathbf{x}}$$

For the whole data matrix $\mathbf{X}$,

$$\hat{\mathbf{X}} = \underbrace{\tilde{\mathbf{X}}}_{\mathbf{F}\text{, the factors}} \mathbf{V}^\top$$

NMF reconstruction errors

Build $\hat{\mathbf{X}} = \mathbf{F}\mathbf{V}^\top$ to minimize a distance $D(\hat{\mathbf{X}}, \mathbf{X})$. Several choice, e.g:

- Least-square loss (distance measured by Frobenius norm)

$$\hat{\mathbf{X}}^{\text{ls}} = \underset{\substack{\mathbf{F}\in\mathscr{M}(\mathbb{R}_+)_{n,q} \\ \mathbf{V}\in\mathscr{M}(\mathbb{R}_+)_{p,q}}}{\arg\min} \left\|\mathbf{X} - \mathbf{F}\mathbf{V}^\top\right\|_F^2,$$

- Generalized Kullback-Leibler divergence ("distance" for distributions)

$$\begin{aligned}\hat{\mathbf{X}}^{\text{kl}} &= \underset{\substack{\mathbf{F}\in\mathscr{M}(\mathbb{R}_+)_{n,q} \\ \mathbf{V}\in\mathscr{M}(\mathbb{R}_+)_{p,q}}}{\arg\min} \sum_{i,j} x_{ij}\log\left(\frac{x_{ij}}{(\mathbf{F}\mathbf{V}^\top)_{ij}}\right) + (\mathbf{F}\mathbf{V}^\top)_{ij} \\ &= \underset{\substack{\mathbf{F}\in\mathscr{M}(\mathbb{R}_+)_{n,q} \\ \mathbf{V}\in\mathscr{M}(\mathbb{R}_+)_{p,q}}}{\arg\max} \sum_{i,j} x_{ij}\log((\mathbf{F}\mathbf{V}^\top)_{ij}) - (\mathbf{F}\mathbf{V}^\top)_{ij},\end{aligned}$$

NMF: limitations

Caveats

- Basis $\mathbf{V}$ formed by standard NMF is not orthogonal!
- Visualization is questionable ...
- Used to performed matrix factorization rather than exploratory analysis

Other model-based approaches

Use a probabilistic-based model to better described non-negative data

- Look for models handling**surdispersion** \{multivariate Poisson-lognormal model, Poisson-Gamma, etc.}
- Look for **zero-inflated** distributions

$$\mathbb{P}(\mathbf{x}_i) = \pi_0 \, \delta_0 + (1 - \pi_0) f(\mathbf{x}_i)$$

Kernel-PCA

Principle: non linear transformation of $\mathbf{x}$ prior to linear PCA

1. Project the data into a higher space where it is linearly separable
2. Apply PCA to the transformed data

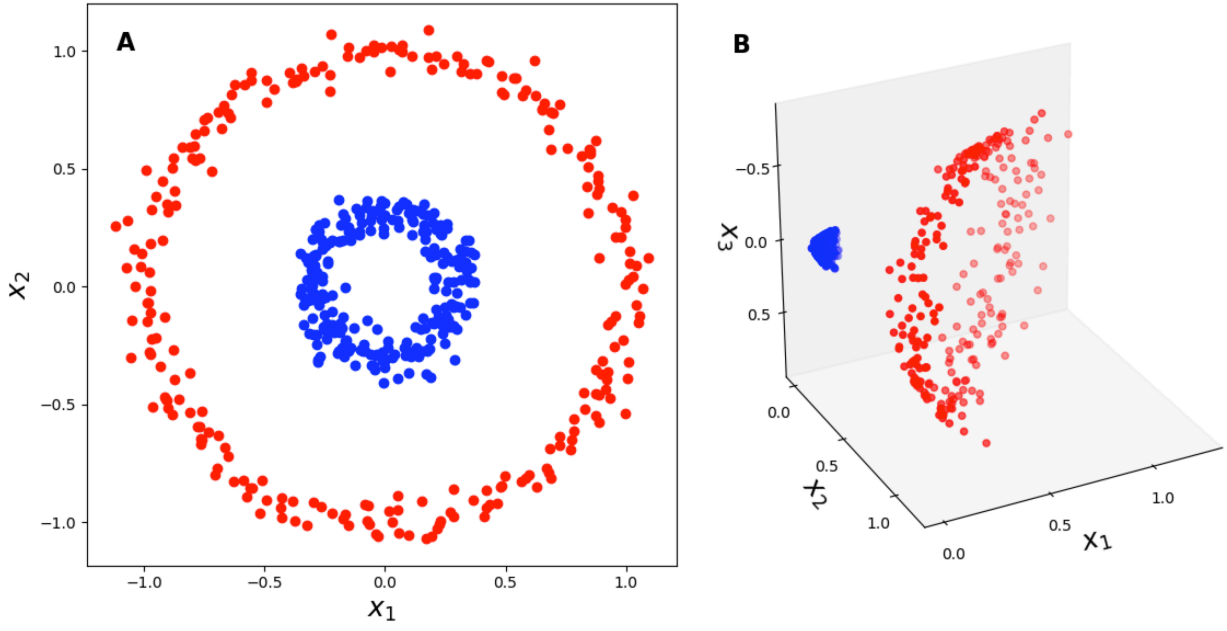


Figure 8: Transformation $\Psi : \mathbf{x} \rightarrow \Psi(\mathbf{x})$ (illustration in presence of existing labels)

Kernel-PCA

Kernel PCA Model

Assume a non linear transformation $\Psi(\mathbf{x}_i)$ where $\Psi : \mathbb{R}^p \to \mathbb{R}^n$, then perform linear PCA, with $\mathbf{U}$ a $n \times q$ orthonormal matrix

$$\Phi(\mathbf{x}) = \mathbf{U}^\top \Psi(\mathbf{x} - \boldsymbol{\mu}) = \tilde{\mathbf{x}}$$

Kernel trick

Never calculate $\Psi(\mathbf{x}_i)$ thanks to the kernel trick:

$$K = k(\mathbf{x}, \mathbf{y}) = (\Psi(\mathbf{x}), \Psi(\mathbf{y})) = \Psi(\mathbf{x})^T \Psi(\mathbf{y})$$

Solution

Eigen-decomposition of the doubly centered kernel matrix $\mathbf{K} = k(\mathbf{x}_i, \mathbf{x}_{i'})$

$$\tilde{\mathbf{K}} = (\mathbf{I} - \mathbf{1}\mathbf{1}^\top/n)\mathbf{K}(\mathbf{I} - \mathbf{1}\mathbf{1}^\top/n) = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$$

Choice of a kernel

A symmetric positive definite function $k(\mathbf{x}, \mathbf{y}) \in \mathbb{R}$, which depends on the kind of **similarity** assumed

Some common kernels

- **Polynormial Kernel**

$$k(\mathbf{x}_i, \mathbf{x}_{i'}) = (\mathbf{x}_i^\top \mathbf{x}_{i'} + c)^d$$

- **Gaussian (radial) kernel**

$$k(\mathbf{x}_i, \mathbf{x}_{i'}) = \exp \frac{-\|\mathbf{x}_i - \mathbf{x}_{i'}\|^2}{2\sigma^2}$$

- **Laplacian kernel**

$$k(\mathbf{x}_i, \mathbf{x}_{i'}) = \exp \frac{-\|\mathbf{x}_i - \mathbf{x}_{i'}\|}{\sigma}$$

↝ Kernel PCA suffers from the choice of the Kernel

(Variational) Auto-encoders

Highly non-linear model

Find Φ and $\tilde{\Phi}$ with **two** neural-networks, controlling the error.

$$\epsilon(\mathbf{X}, \hat{\mathbf{X}}) = \sum_{i=1}^{n} \left\| \mathbf{x}_i - \tilde{\Phi}(\Phi(\mathbf{x}_i)) \right\|^2 + \text{regularization}(\Phi, \tilde{\Phi})$$

- # layers and neurons determine the **model complexity**
- Need regularization to avoid **overfitting**
- Fitted with optimization tools like stochastic gradient descent
- Require more data and more computational resources
- **Interpretation questionable**

Manifold learning (preserving pairwise relations)

Pairwise Relation

Focus on pairwise relation $\mathscr{R}(\mathbf{x}_i, \mathbf{x}_{i'})$.

Distance Preservation

Construct a map Φ from the space $\mathbb{R}^p$ into a space $\mathbb{R}^q$ of smaller dimension:

$$\begin{aligned} \Phi : \quad & \mathbb{R}^p \to \mathbb{R}^q, q \ll p \\ & \mathbf{x} \mapsto \Phi(\mathbf{x}) \end{aligned}$$

$$\text{such that} \quad \mathscr{R}(\mathbf{x}_i, \mathbf{x}_{i'}) \sim \mathscr{R}'(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_{i'})$$

Multidimensional scaling

A.k.a PCoA (Principal Coordinates Analysis)
Try to preserve inner product related to the distance (e.g. Euclidean)

t-SNE – Stochastic Neighborhood Embedding

Try to preserve relations with close neighbors with Gaussian kernel

Multidimensional scaling

a.k.a Principale Coordinates Analysis

Problem setup

Consider a collection of points $\mathbf{x}_i \in \mathbb{R}^p$ and assume either

- $D = d_{ii'}$ a $n \times n$ dissimilarity matrix, or
- $S = s_{ii'}$ a $n \times n$ similarity matrix, or

Goal: find $\tilde{\mathbf{x}}_i \in \mathbb{R}^q$ while preserving S/D in the latent space

↝ Don't need access to the position in $\mathbb{R}^p$ (only D or S ↝ 'kernel').

Classical MDS model

Measure similarities with the (centered) **inner product** and minimize

$$\sum_{i \neq i'} \left((\mathbf{x}_i - \boldsymbol{\mu})^\top (\mathbf{x}_i - \boldsymbol{\mu}) - \tilde{\mathbf{x}}_i^\top \tilde{\mathbf{x}}_{i'} \right)^2,$$

assuming a linear model $\tilde{\mathbf{x}} = \mathbf{V}^\top (\mathbf{x}_i - \boldsymbol{\mu})$, with $\mathbf{V} \in \mathcal{O}_{p \times q}$.

Classical MDS: solution

With the linear model $\tilde{\mathbf{x}} = \Phi(\mathbf{x}) = \mathbf{V}^\top(\mathbf{x}_i - \boldsymbol{\mu})$, we aim at minimizing

$$\begin{aligned}\text{Stress}^{cMDS} &= \sum_{i \neq i'} \left((\mathbf{x}_i - \boldsymbol{\mu})^\top (\mathbf{x}_{i'} - \boldsymbol{\mu}) - \tilde{\mathbf{x}}_i^\top \tilde{\mathbf{x}}_{i'} \right)^2, \\ &= \sum_{i \neq i'} \left((\mathbf{x}_i - \boldsymbol{\mu})^\top (\mathbf{x}_{i'} - \boldsymbol{\mu}) - (\mathbf{x}_i - \boldsymbol{\mu})^\top \mathbf{V}\mathbf{V}^\top (\mathbf{x}_{i'} - \boldsymbol{\mu}) \right)^2,\end{aligned}$$

It can be showed that $\underset{\boldsymbol{\mu} \in \mathbb{R}^p, \mathbf{V} \in \mathcal{O}_{pq}}{\text{minimize}}\ \text{Stress}^{cMDS}(\tilde{\mathbf{x}}_i)$ is dual to principal component analysis and leads to

$$\tilde{\mathbf{x}} = \mathbf{X}^c\mathbf{V} = \mathbf{U}\mathbf{D}\mathbf{V}^\top\mathbf{V} = \mathbf{U}\mathbf{D}.$$

↝ The principal coordinates in $\mathbb{R}^q$ correspond to the scores of the n individuals projected on the first q principal components.

Metric Multidimensional Scalings

Idea to generalize classical MDS:

preserving similarities in term of **inner product** amounts to preserve dissimilarity in terms of Euclidean distance

Least-squares/Kruskal-Shephard scaling

Use a distance base formulation with the following loss (Stress) function:

$$\text{Stress}^{SK} = \sum_{i \neq i'} \left(d_{ii'} - \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_{i'}\|\right)^2,$$

- Almost equivalent to classical MDS when d is the Euclidean distance
- Generalize to any **quantitative** dissimilarity/distance d

Sammong mapping - Variant of the loss (Stress) function

$$\text{Stress}^{SM} = \sum_{i \neq i'} \frac{\left(d_{ii'} - \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_{i'}\|\right)^2}{d_{ii'}}.$$

Stochastic Neighbor Embedding (SNE)

Let $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ be the original points in $\mathbb{R}^p$, and measure similarities by

$$p_{ij} = (p_{j|i} + p_{i|j})/2n$$

where

$$\begin{aligned} p_{j|i} &= \frac{\exp(-\|\mathbf{x}_j - \mathbf{x}_i\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_k - \mathbf{x}_i\|^2/2\sigma_i^2)}, \\ &= \frac{\exp(-d_{ij}^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-d_{ki}^2/2\sigma_i^2)} \end{aligned}$$

- SNE preserves relations with **close neighbors** with Gaussian kernels
- σ smooths the data (linked to the regularity of the target manifold)

The perplexity parameter

The variance σ_i^2 should adjust to local densities (neighborhood of point i)

Perplexity: a smoothed effective number of neighbors

The perplexity is defined by

$$Perp(p_i) = 2^{H(p_i)}, \qquad H(p_i) = -\sum_{j=1}^{n} p_{j|i} \log_2 p_{j|i}$$

where H is the Shannon entropy of $p_i = (p_{1|i}, \dots, p_{n|i})$.\
↝ SNE performs a binary search for the value of σ_i that produces a p_i with a fixed perplexity that is specified by the user.

tSNE and Student / Cauchy kernels

Consider $(\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_n)$ are points in the low dimensional space $\mathbb{R}^{q=2}$

- Consider a similarity between points in the new representation:

$$q_{i|j} = \frac{\exp(-\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|^2)}{\sum_{k \neq i} \exp(-\|\tilde{\mathbf{x}}_k - \tilde{\mathbf{x}}_j\|^2)}$$

- Robustify this kernel by using Student(1) kernels (ie Cauchy)

$$q_{i|j} = \frac{(1 + \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_k\|^2)^{-1}}$$

Optimizing tSNE

- Minimize the KL between p and q so that the data representation minimizes:

$$C(y) = \sum_{ij} KL(p_{ij}, q_{ij})$$

- The cost function is not convex

$$\left[\frac{\partial C(y)}{\partial y}\right]_i = \sum_j (p_{ij} - q_{ij})(y_i - y_j)$$

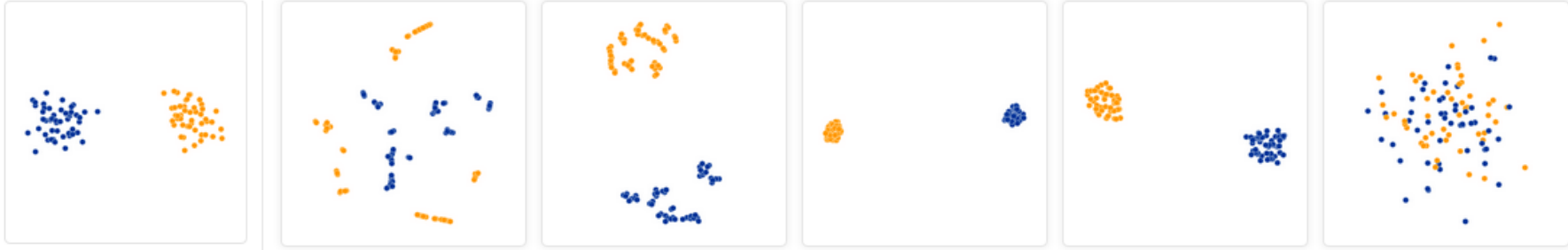
- Gradient update (adaptive learning rate η) with momentum $\alpha(t)$

$$Z^{(t)} = Z^{(t-1)} + \eta \frac{\partial C(Z)}{\partial Z} + \alpha(t)(Z^{(t-1)} - Z^{(t-2)})$$

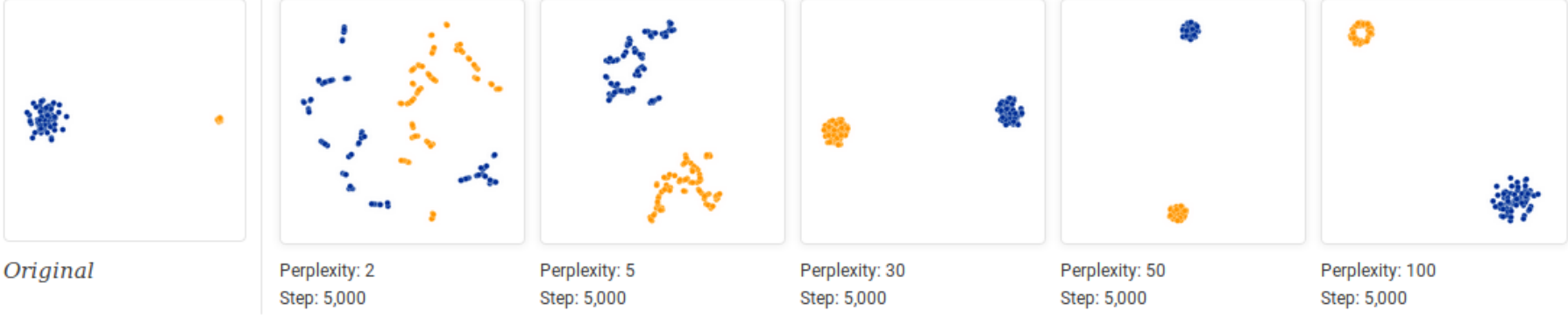
- Initialization $Z_i^{(0)} \sim \mathcal{N}(0, \delta I)$, δ small.

Empirical properties of tSNE (1)

Effect of Hyperparameters : Perplexity



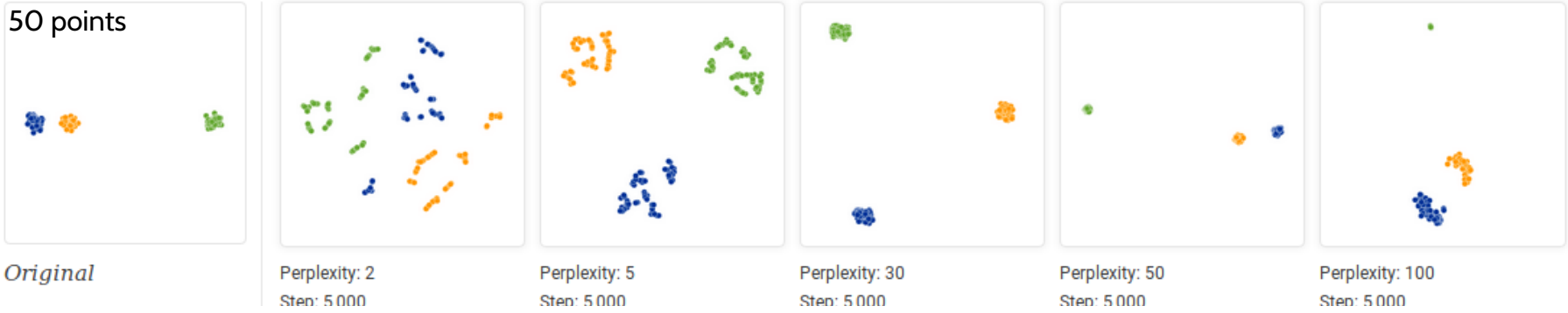
tSNE does not account for heteroscedasticity

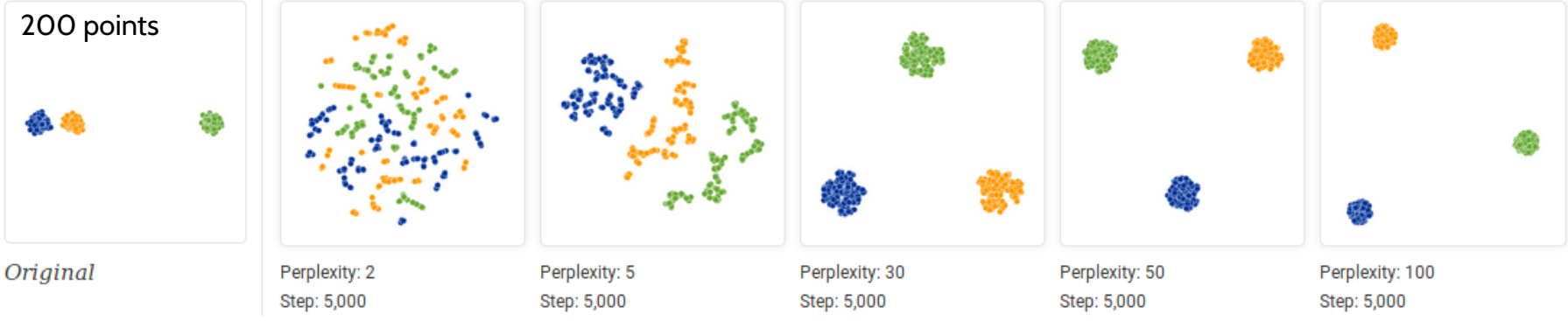


https://distill.pub/2016/misread-tsne/

Empirical properties of tSNE (2)

tSNE does not account for between-cluster distance



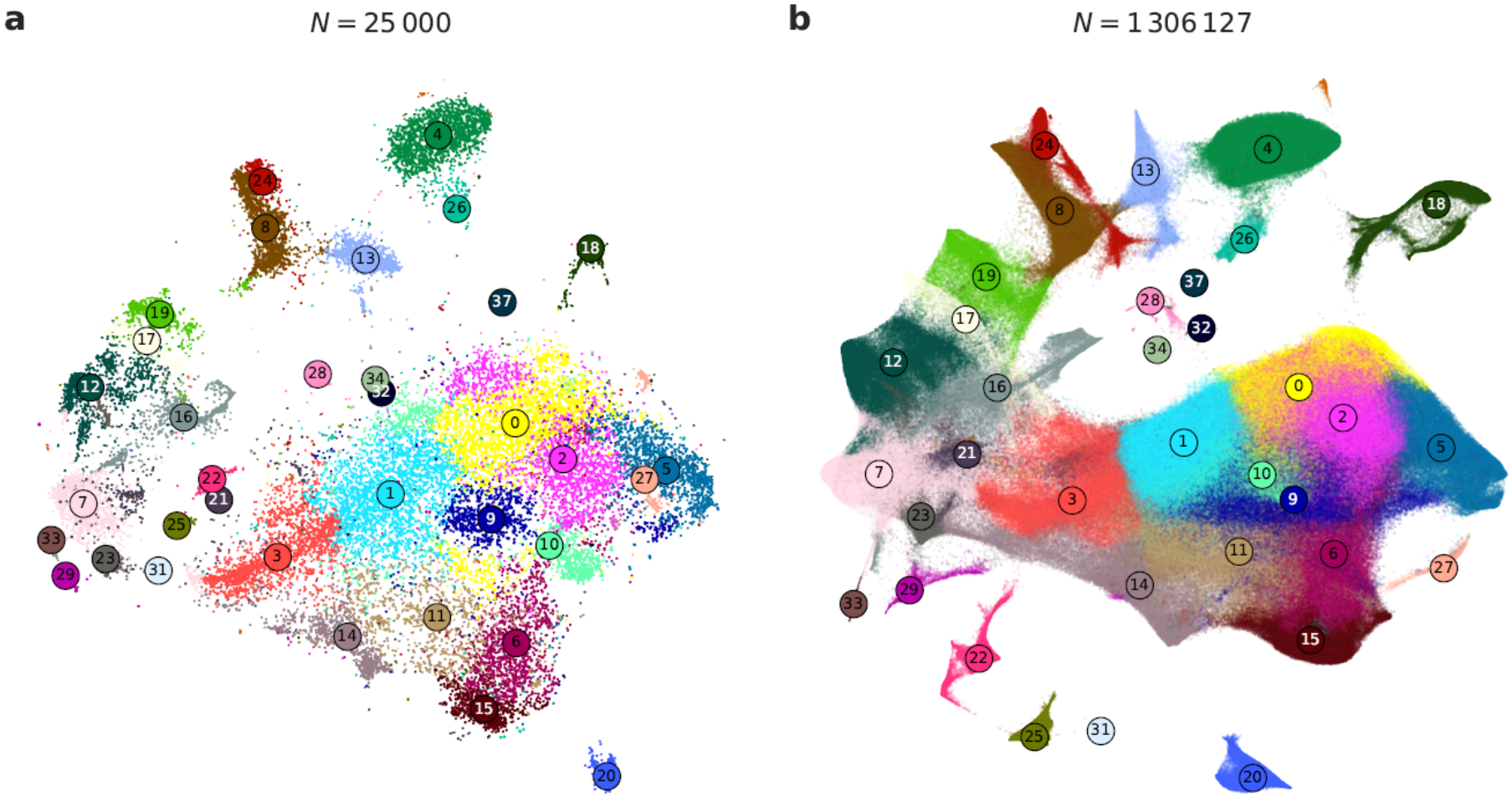


What about random noise ?

Empirical properties of tSNE (3)

Catching Complex Geometries

tSNE on single cell Gene Expression data [1]



t-SNE: pros/cons

Properties

- good at preserving local distances (intra-cluster variance)
- not so good for global representation (inter-cluster variance)
- good at creating clusters of close points, bad at positioning clusters wrt each other

Limitations

- importance of preprocessing: initialize with PCA and feature selection plus log transform (non linear transform)
- percent of explained variance ? interpretation of the q distribution ?
- Lack of reproducibility due to stochastic optimization

Uniform Manifold Approximation and Projection [2]

For j in the k-neighborhood of i, define the conditional distribution

$$p_{j|i} = \exp\left(-\frac{\|X_i - X_j\|_2^2 - \rho_i}{\sigma_i}\right) \quad \text{with } \rho_i = \min_{j \neq i} \|X_i - X_j\|^2$$

and its symmetrized version

$$p_{ij} = p_{j|i} + p_{i|j} - p_{j|i} p_{i|j}.$$

Rely on a generalized Student-distribution with a, b fitted on the data:

$$q_{ij} = \left(1 + a\|Z_i - Z_j\|_2^{2b}\right)^{-1}$$

UMAP solves the following problem:

$$\min_{Z \in \mathbb{R}^{n \times d}} \quad -\sum_{i<j} p_{ij} \log q_{ij} + (1 - p_{ij}) \log(1 - q_{ij})$$

References

[1] Kobak, D. and Berens, P. (2018). The art of using t-SNE for single-cell transcriptomics. *bioRxiv*.

[2] McInnes, L., Healy, J. and Melville, J. (2018). UMAP: Uniform manifold approximation and projection for dimension reduction. *Arxiv* 1–63.