Fundamentals of AI

Dimensionality reduction

Linear methods of dimred:

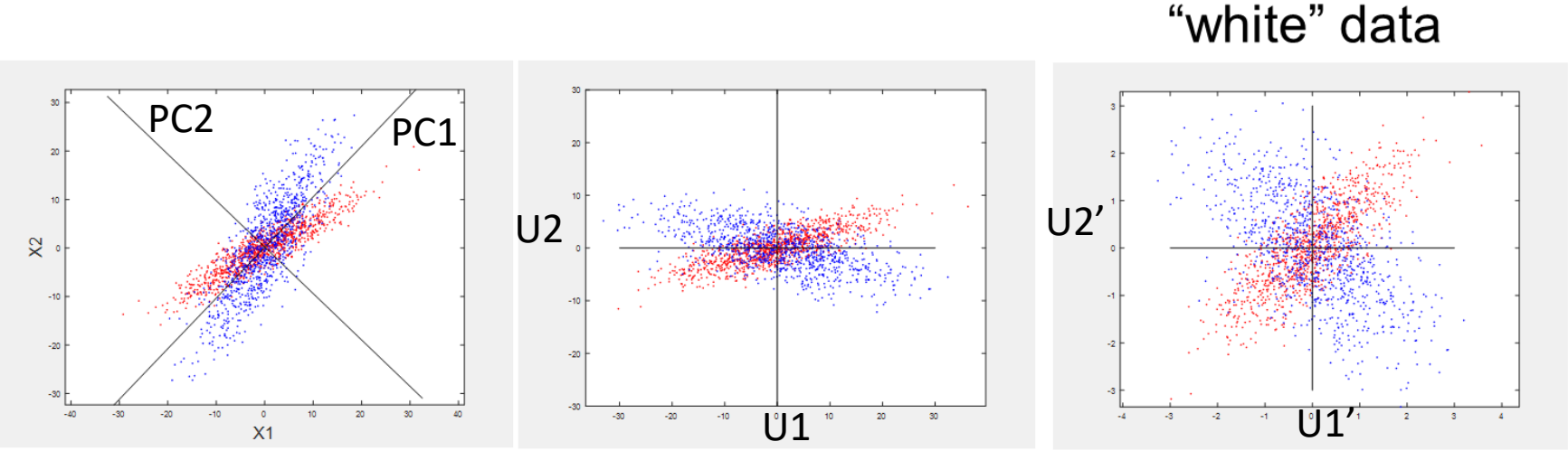
- Independent Component Analysis (ICA)
- Non-negative Matrix Factorization (NMF)
- Factor analysis (FA)

Independent Component Analysis (ICA)

Data whitening:

when PCA can not be applied

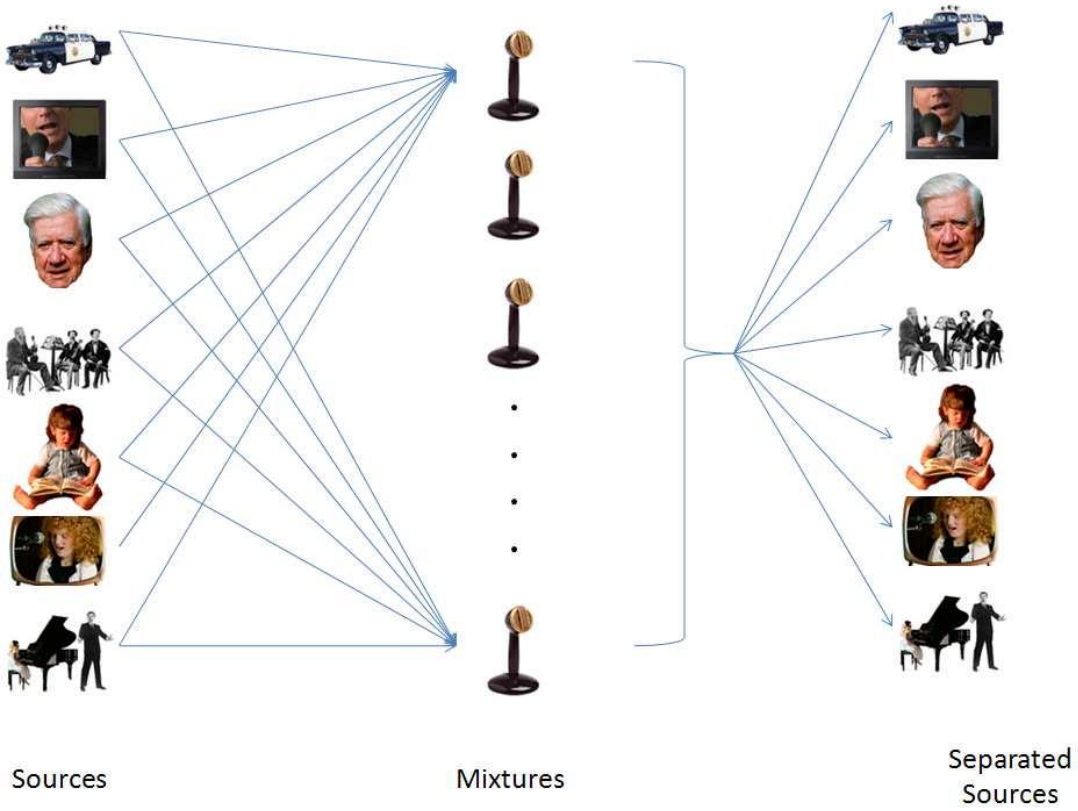
- Select m first principal components
- Compute principal components X = UV
- Normalize U → U’ such that $U'U'^{T}=I$
- U’ – is new whitened dataset
- It’s empirical covariance is unity, all eigenvalues equal 1
- PCA can not be applied to U anymore
- But higher moments of U are not zero if U is not Gaussian!
- Can we use them?



In other words…

- Imagine that we reduced the dimensionality of the data point cloud to *m* components by PCA
- Any rotation of this subspace won't change the total amount variance explained by the reduced R^m
- Can we choose an orthogonal rotation of axes such that the axes would reflect something more informative than variance?

Cocktail party problem: ‘blind source separation’



Can be solved if sources are statistically independent!

Check https://www.di.ens.fr/~fbach/kernel-ica/sound-demos.htm

Linear data transformation

- S = WX, W is orthonormal matrix (rotation), $WW^T = I$
- Probability Density Function of X is $p(X) = p(x_1, x_2, \ldots, x_m)$
- Imagine that for new variables *S*, we can factorize $p(s_1, s_2, \ldots, s_m) = p_1(s_1)\, p_2(s_2) \ldots\, p_m(s_m)$
- In this case we say that $s_1, s_2, \ldots, s_p$ are **independent!**
- But how to find such W that S would be as independent as possible?

But how to find such W in S=WX that S would be as independent as possible?

- Best W must minimize mutual information : $I(s_1, s_2, \ldots, s_m) = \sum_{i=1}^{m} H(s_i) - H(\mathbf{s})$

where $H(x) = -\int p_x(\xi) \log p_x(\xi) d\xi$ - **entropy!** (measure of 'disorder')

Can be solved using InfoMax algorithm suggested by Bell and Sejnowski in 1995

However, today another approach, fastICA, is more popular

FastICA principle

Entropy is maximal for the standard (with unit variance) Gaussian distribution

$$J(\mathbf{x}) = H(\mathbf{x}_{gauss}) - H(\mathbf{x})$$

Negentropy is a information-based measure of deviation from ‘Gaussianity’

Some calculations for the mutual information, *assuming that the data is whitened and the matrix W is orthogonal*, gives:

$$I(s_1, s_2, \ldots, s_m) = const - \sum_{i=1}^{m} J(s_i)$$

Minimizing mutual information = maximizing non-Gaussianity of signals!

FastICA principle

- Negentropy is difficult to estimate from finite datasets (without knowing PDF), one needs to approximate
- One of the approximations is through using kurtosis (normalized fourth moment of data distribution):

$$J(y) \approx \frac{1}{12} E\{y^3\}^2 + \frac{1}{48} \operatorname{kurt}(y)^2$$

- Aapo Hyvärinen suggested more general form:

$$J(y) \propto [E\{G(y)\} - E\{G(\nu)\}]^2$$

where G() is some non-quadratic function and ν is a standardized normal distribution

FastICA algorithm

1. Center the data to make its mean zero.
2. Whiten the data to give $\mathbf{z}$.
3. Choose an initial (e.g., random) vector $\mathbf{w}$ of unit norm.
4. Let $\mathbf{w} \leftarrow E\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\} - E\{g'(\mathbf{w}^T\mathbf{z})\}\mathbf{w}$
5. Let $\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}||$.
6. If not converged, go back to step 4.

$g(y) = \tanh(y)$

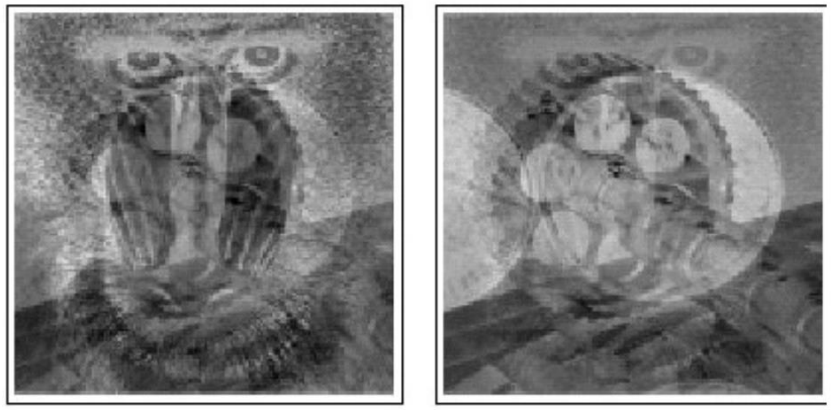
$g(y) = y \exp(-y^2/2)$

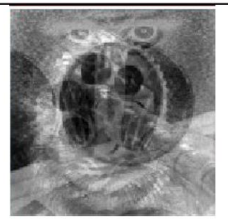
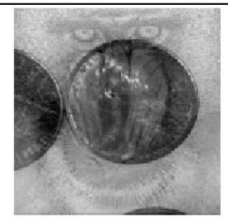
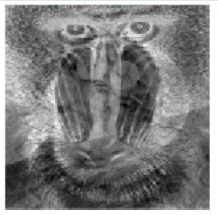
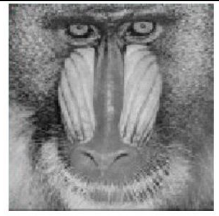
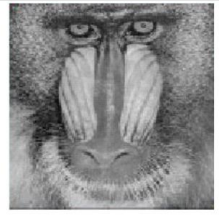
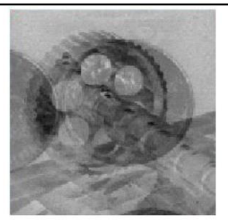
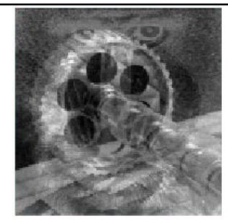
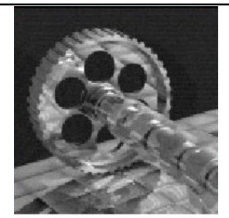
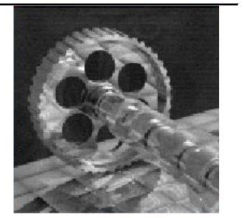
$g(y) = y^3$

In order to find other components, deflation approach

Example of ICA work

Mixtures



	PCA	InfoMax super	InfoMax sub	FastICA
Extracted component 1				
Extracted component 2				
Extracted component 3				

Ambiguities of ICA

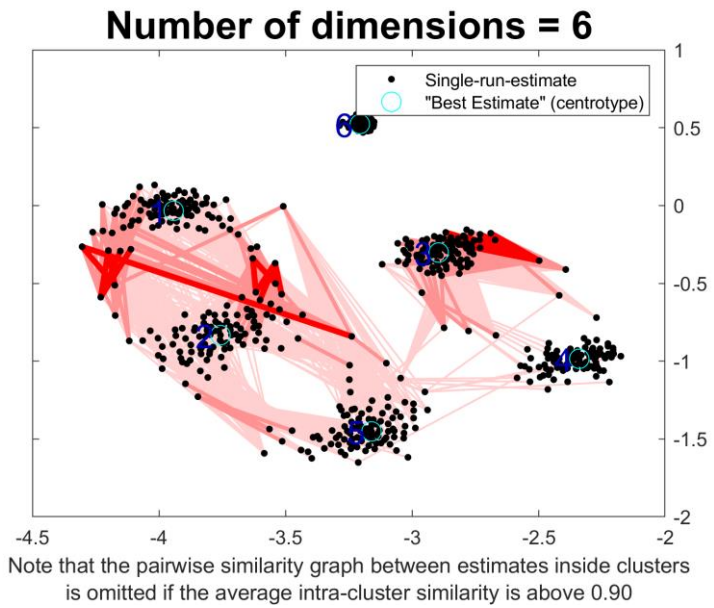
- We can not determine the variances (energies) of the independent components
- We can not determine the order of the independent components
- However, we can apply bootstrap and estimate the component’s stability

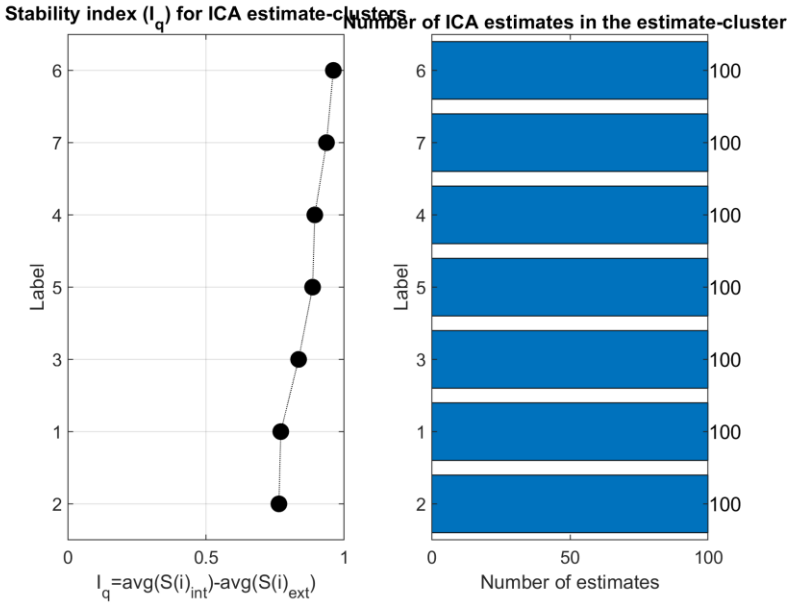
Icasso stabilization of independent components

$$I_q(C_k) = \frac{1}{|C_k|^2} \sum_{i,j \in C_k} | r_{ij} | - \frac{1}{|C_k| \sum_{l \neq k} |C_l|} \sum_{i \in C_k} \sum_{j \notin C_k} | r_{ij} |$$

r_{ij} – Pearson correlation between component *i* and *j*

Clustering quality criterion!
But here we cluster components!

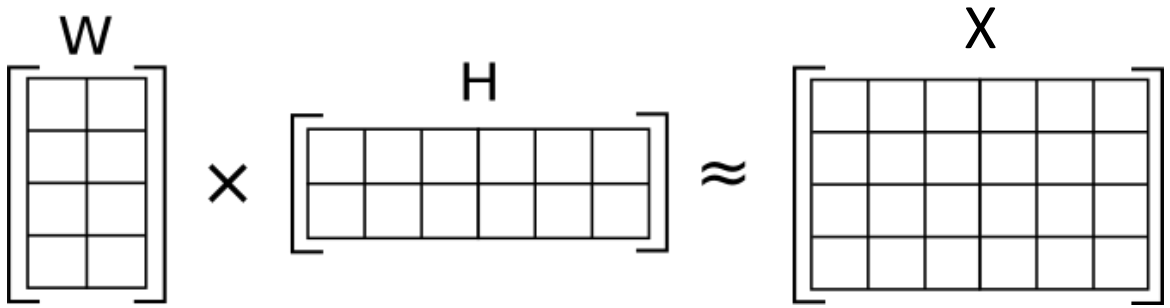




Non-negative matrix factorization (NMF)

Non-negative matrix factorization

- Group of algorithms solving the problem X = WH

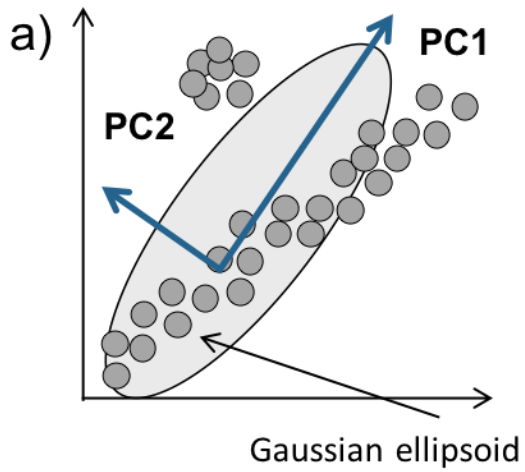


where W and H contain only **non-negative values**

Non-negative matrix factorization: geometric view

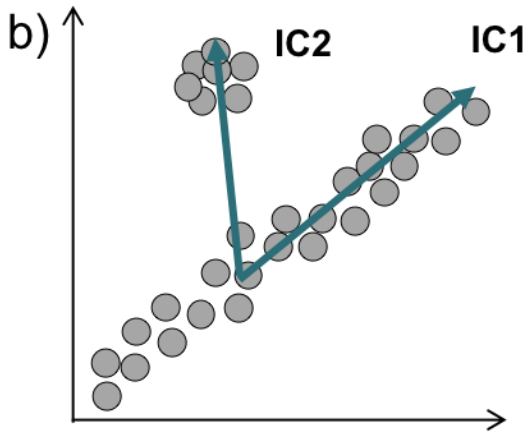
Principal Component Analysis

PCA



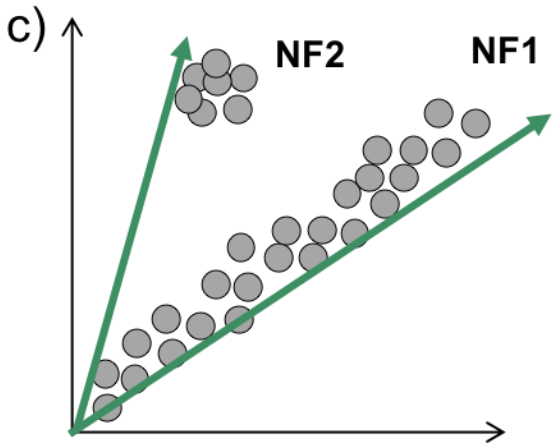
Independent Component Analysis

ICA



Non-negative Matrix Factorization

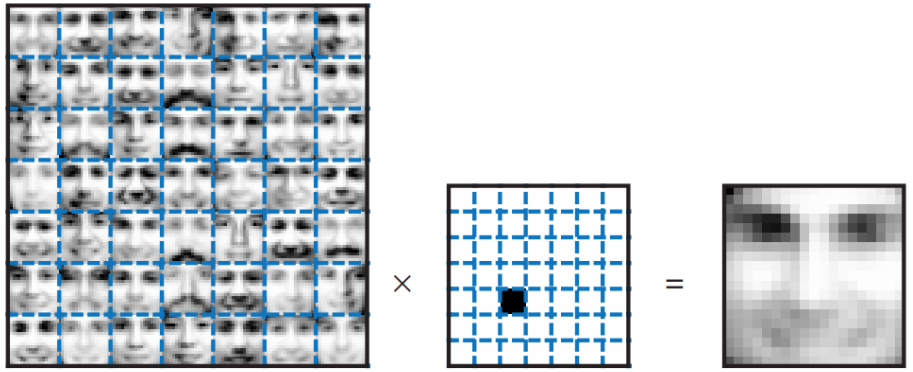
NMF



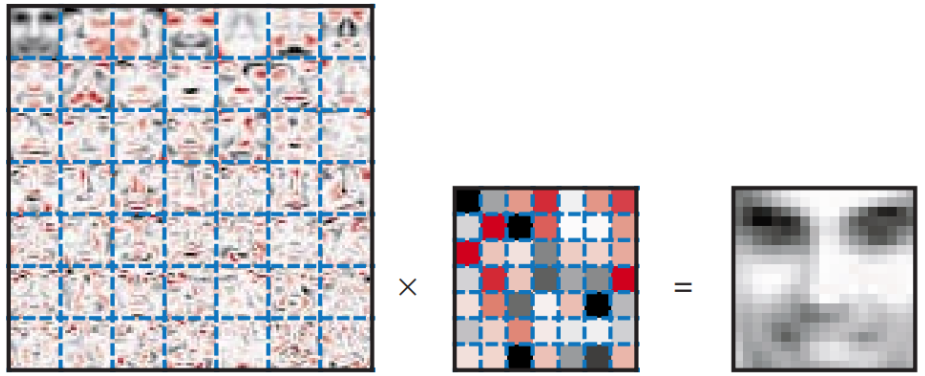
Learning the parts of objects by non-negative matrix factorization (Lee and Seung, Nature, 1999)

Original

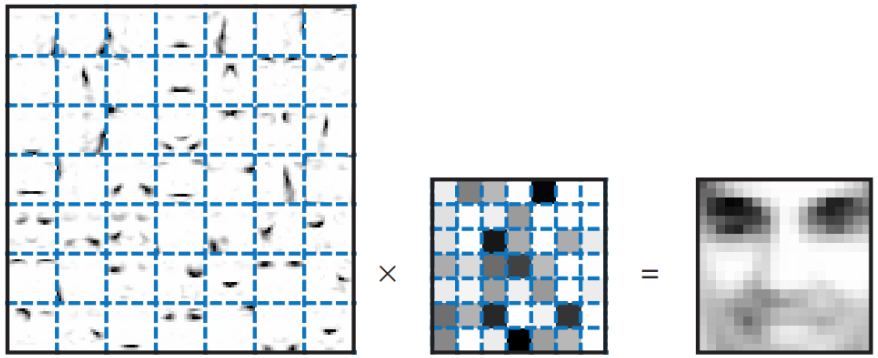
K-MEANS



PCA



NMF



Most popular algorithm: Lee and Seung's update rule

We solve the problem $|X - WH|^2 \rightarrow$ min, subject to $W \geq 0$, $H \geq 0$

initialize: $\mathbf{W}$ and $\mathbf{H}$ non negative.

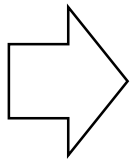
Then update the values in $\mathbf{W}$ and $\mathbf{H}$ by computing the following, with n as an index of the iteration.

$$\mathbf{H}^{n+1}_{[i,j]} \leftarrow \mathbf{H}^{n}_{[i,j]} \frac{((\mathbf{W}^n)^T \mathbf{V})_{[i,j]}}{((\mathbf{W}^n)^T \mathbf{W}^n \mathbf{H}^n)_{[i,j]}}$$

and

$$\mathbf{W}^{n+1}_{[i,j]} \leftarrow \mathbf{W}^{n}_{[i,j]} \frac{(\mathbf{V}(\mathbf{H}^{n+1})^T)_{[i,j]}}{(\mathbf{W}^n \mathbf{H}^{n+1}(\mathbf{H}^{n+1})^T)_{[i,j]}}$$

Until $\mathbf{W}$ and $\mathbf{H}$ are stable.



Iterative application of non-negative least square regression

Non-negative matrix factorization as a clustering method

$$||X - WH||^2 \rightarrow min, subject\ to\ W \geq 0, H \geq 0$$

Theorem: if ***H*** is orthogonal (***HH***^{T} is diagonal) then the solution to the problem is given by K-means clustering of columns in ***X***. ***W*** are cluster centroids in this case.

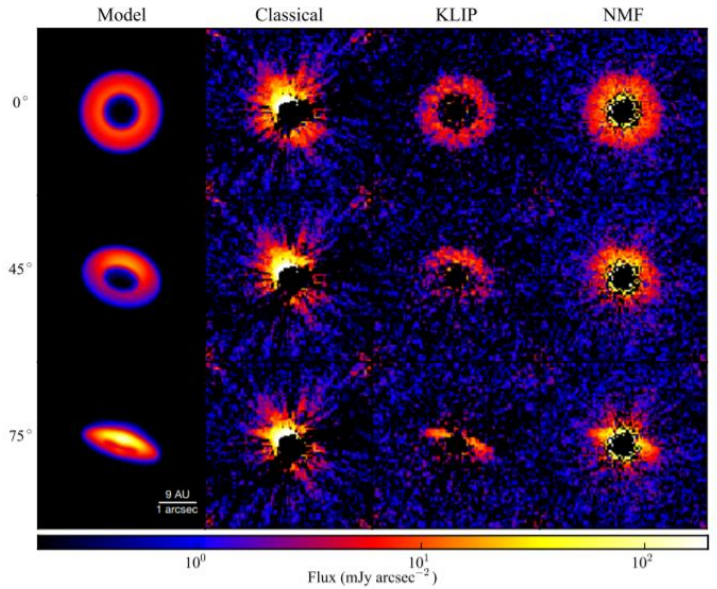
Remark: if H is only approximately orthogonal (frequently the case), the clustering property still holds

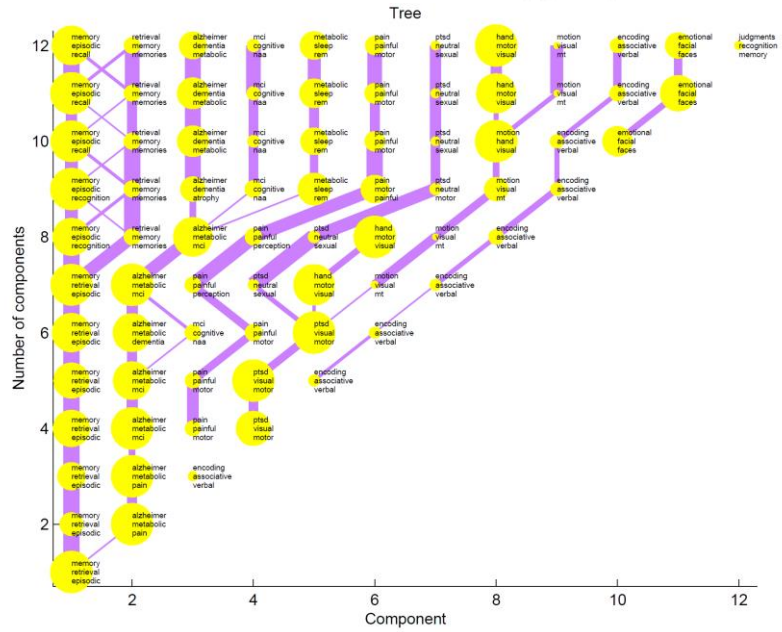
Exercise: Prove this theorem*

*hint: read http://ranger.uta.edu/~chqding/papers/NMF-SDM2005.pdf

Most of the NMF applications are for clustering

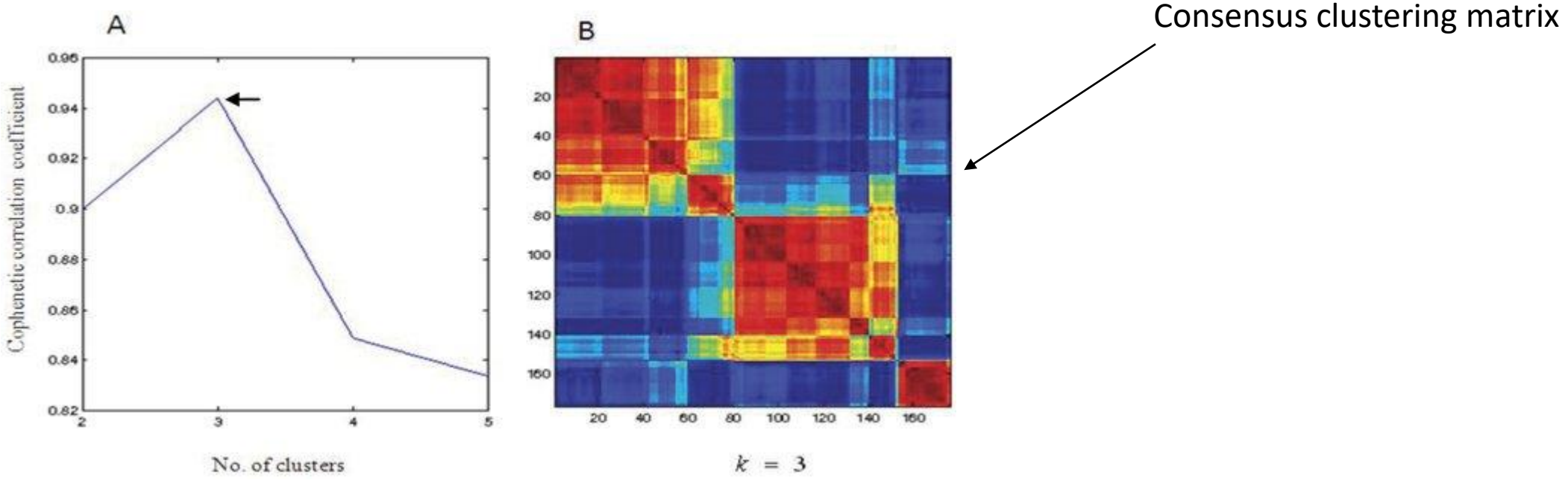
- **Astronomy** (astrophysical signals are non-negative, e.g. spectra)
- **Text mining** (word frequencies are non-negative)
- **Bioinformatics** (clustering gene expression and DNA methylation)
- **Nuclear imaging** (SPECT and PET medical imaging)





Number of components in NMF

- Cophenetic correlation coefficient (measure of how faithfully a dendrogram preserves the pairwise distances in R^m)



Factor Analysis (FA)

FA: Probabilistic linear dimred technique

- The goal is to find **latent random variables** explaining the data as a **linear superposition** of them
- We assume that **the data is centered** (mean of all variables equals to zero)
- x_i are columns of the data matrix (variables), i=1…p

$$x_i = l_{i1}F_1 + \cdots + l_{ik}F_k + \varepsilon_i .$$

F_1… F_k are ‘latent factors’ (random variables)

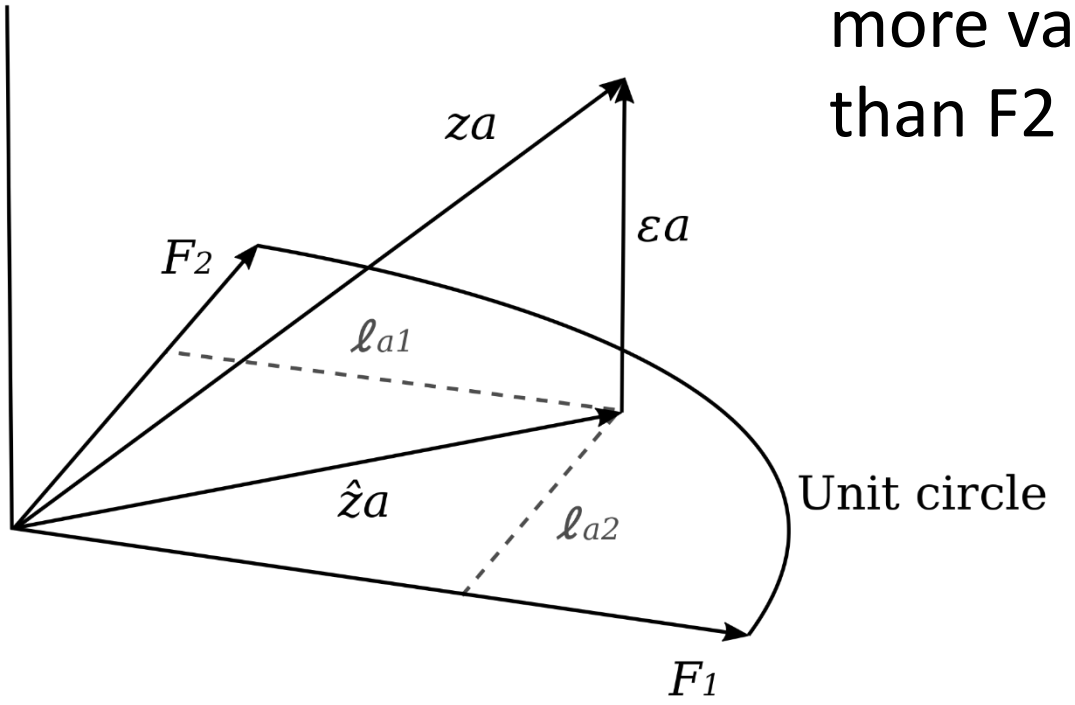
We assume that the factors are **uncorrelated** : $Cov(F) = I$
We assume that the factors are **centered** (mean of F_i is zero)
We assume that the noise ε_i and factors are **independent**
Usually we assume that each F_i has standartized **Gaussian distribution**

Matrix formulation

$X = LF + \varepsilon$, subject to $Cov(F)=I$

Very similar to the objective of PCA! But F1 does not have to ‘explain’ more variance than F2

Geometric image



Rotation of factors

$\boldsymbol{X} = \boldsymbol{LF} + \varepsilon$, subject to $Cov(\boldsymbol{F}) = \boldsymbol{I}$

Take an orthonormal matrix $\boldsymbol{R}$ (rotation matrix): $\boldsymbol{R}\boldsymbol{R}^{\mathrm{T}} = \boldsymbol{I}$

$\boldsymbol{X} = \boldsymbol{LF} + \varepsilon = \boldsymbol{L}\boldsymbol{R}\boldsymbol{R}^{\mathrm{T}}\boldsymbol{F} + \varepsilon = \boldsymbol{L'F'} + \varepsilon$,
$Cov(\boldsymbol{F'}) = \boldsymbol{I}$, $\boldsymbol{F'}$ is standardized Gaussian

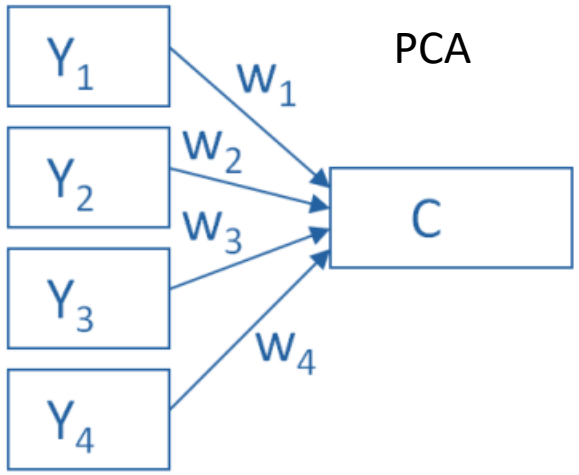
We can rotate factors without changing the model!

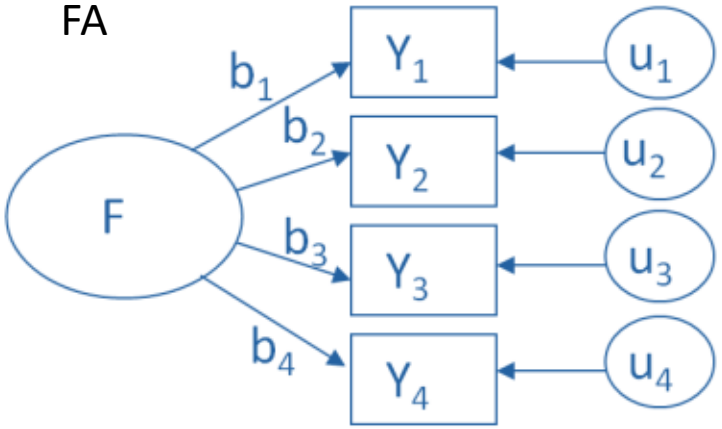
We can rotate them to achieve *sparsity* or other desired properties

Difference between FA and PCA: a confusing discussion

For example, https://www.theanalysisfactor.com/the-fundamental-difference-between-principal-component-analysis-and-factor-analysis/

- 'PCA looks for a linear combination of variables'
- 'Factor Analysis is a measurement model of a latent variable'
- "As you can probably guess, this fundamental difference has many, many implications."





Difference between FA and PCA: a confusing discussion

1) FA is a method from probabilistic approach to data mining, PCA is a geometric method

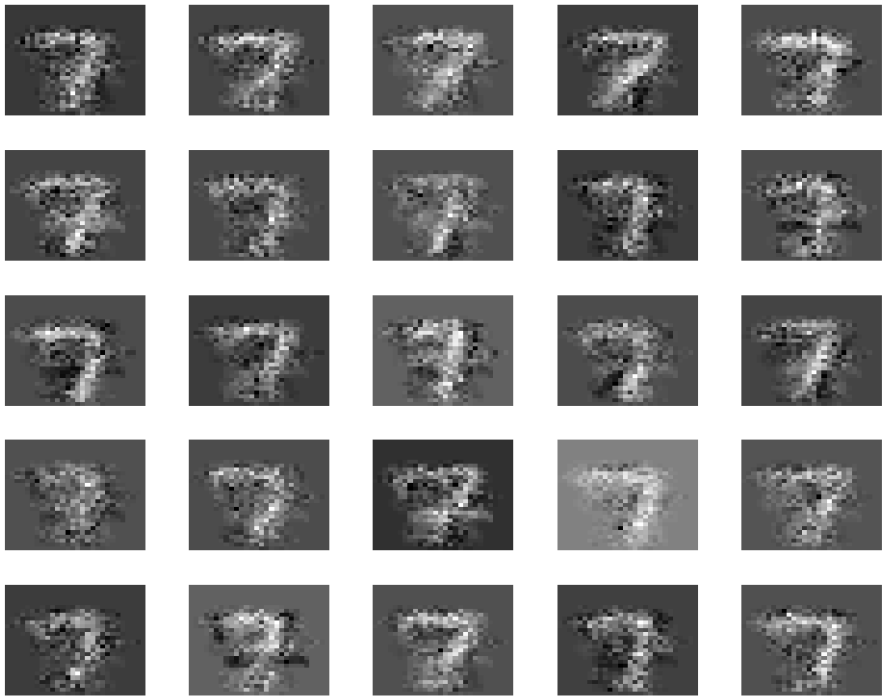
2) PCA is one possibility to solve the problem of FA in some simple cases

3) After application of PCA, the axes can be rotated to optimize something

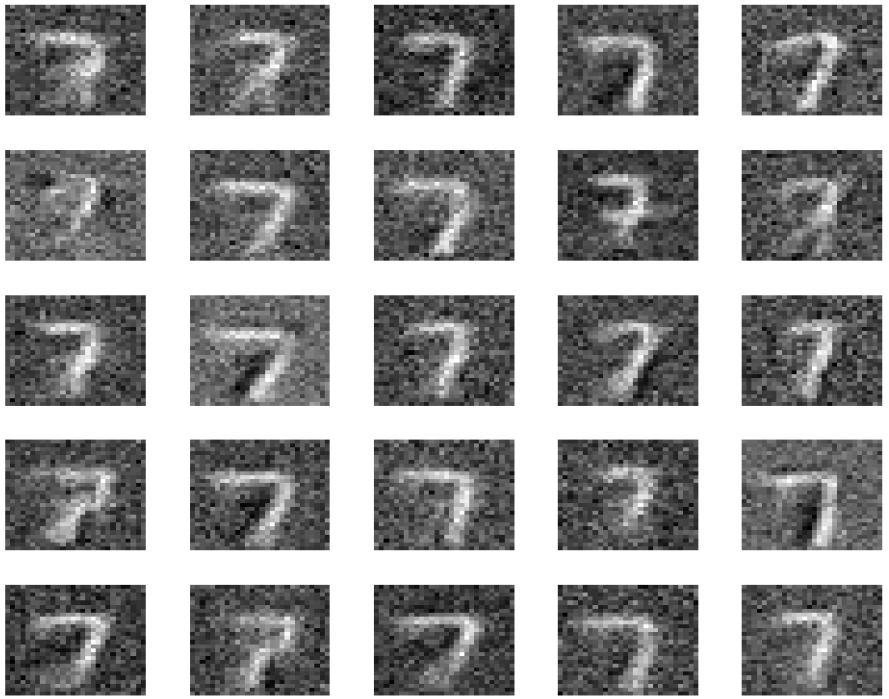
4) FA can be made more general than PCA (e.g., assume non-Gaussian factors)

5) *The noise model can be different in PCA and FA*

$$\boldsymbol{X} = \boldsymbol{LF} + \varepsilon \qquad \epsilon \sim \begin{cases} \mathcal{N}(\epsilon; \mathbf{0}; \hat{\boldsymbol{\Psi}}) & \text{for FA} \\ \mathcal{N}(\epsilon; \mathbf{0}; \hat{\sigma}^2 \mathbf{I}) & \text{for PPCA} \end{cases}$$



(a) Factor Analysis



(b) PPCA

Example taken from : https://www.inf.ed.ac.uk/teaching/courses/pmr/17-18/assets/slides/slides10.pdf

What you have to take

- Besides standard PCA, many other linear methods for dimensionality reduction, also called matrix factorization methods
- ICA is usually a step after PCA application: remove Gaussian signal, find rotation of coordinate axes maximizing non-Gaussianity
- NMF works as a clustering method with data matrices without negative values
- FA is a wide family of probabilistic methods for dimensionality reduction