Welcome to CME 250 Introduction to Machine Learning!

Spring 2020 – Online version

April 16th 2020

ICME

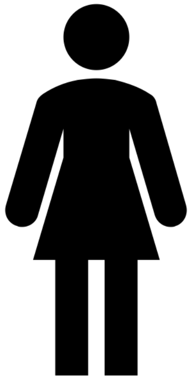
Stanford | Institute for Computational & Mathematical Engineering

Today's schedule

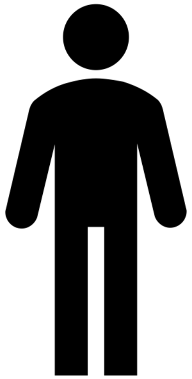
- Dimensionality reduction: Why is it useful?
- Reduce number of features
  - PCA as Maximal Variance Projection
  - Other (really useful) matrix decompositions
- Creating features out of similarity
  - Spectral clustering

Let's get to know each other…

Breakout room



You



Another student

Name

Location

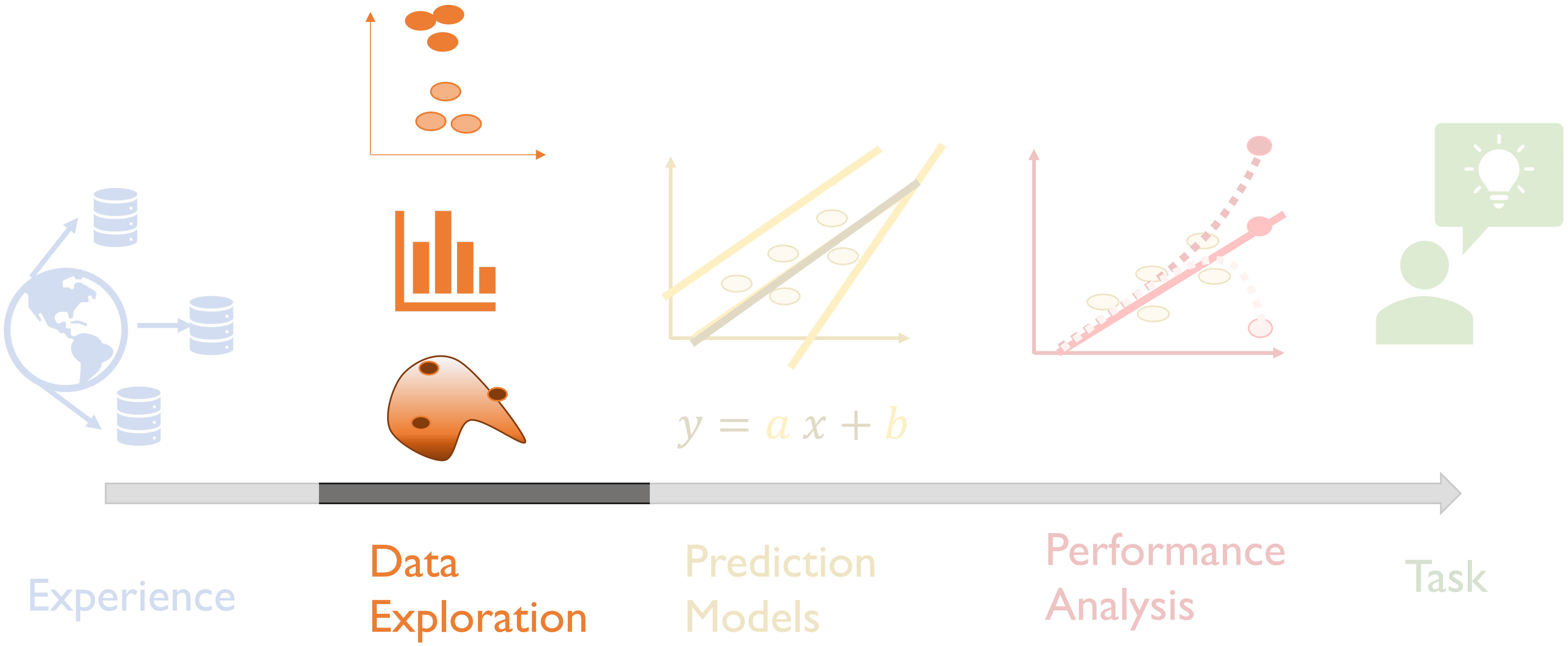
Department

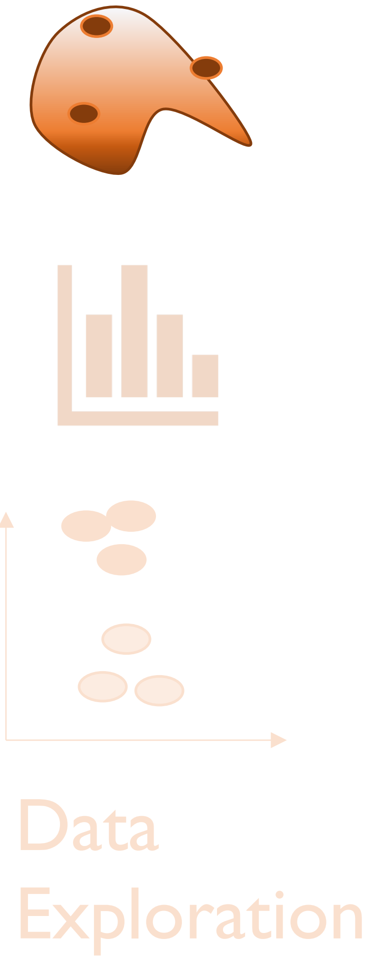
Year

Describe where you are: kitchen, car, coffee shop …

3 mins

Chat/Audio/Video


y = a x + b
Experience
Data Exploration
Prediction Models
Performance Analysis
Task



Unsupervised Learning Part II: Dimensionality Reduction

Introduction to Statistical Learning

Chapter 10.2: Principal Component Analysis,

10.4: Practical Lab in R

Elements Statistical Learning

Chapter 14.5 – 14.9: Different methods of DR

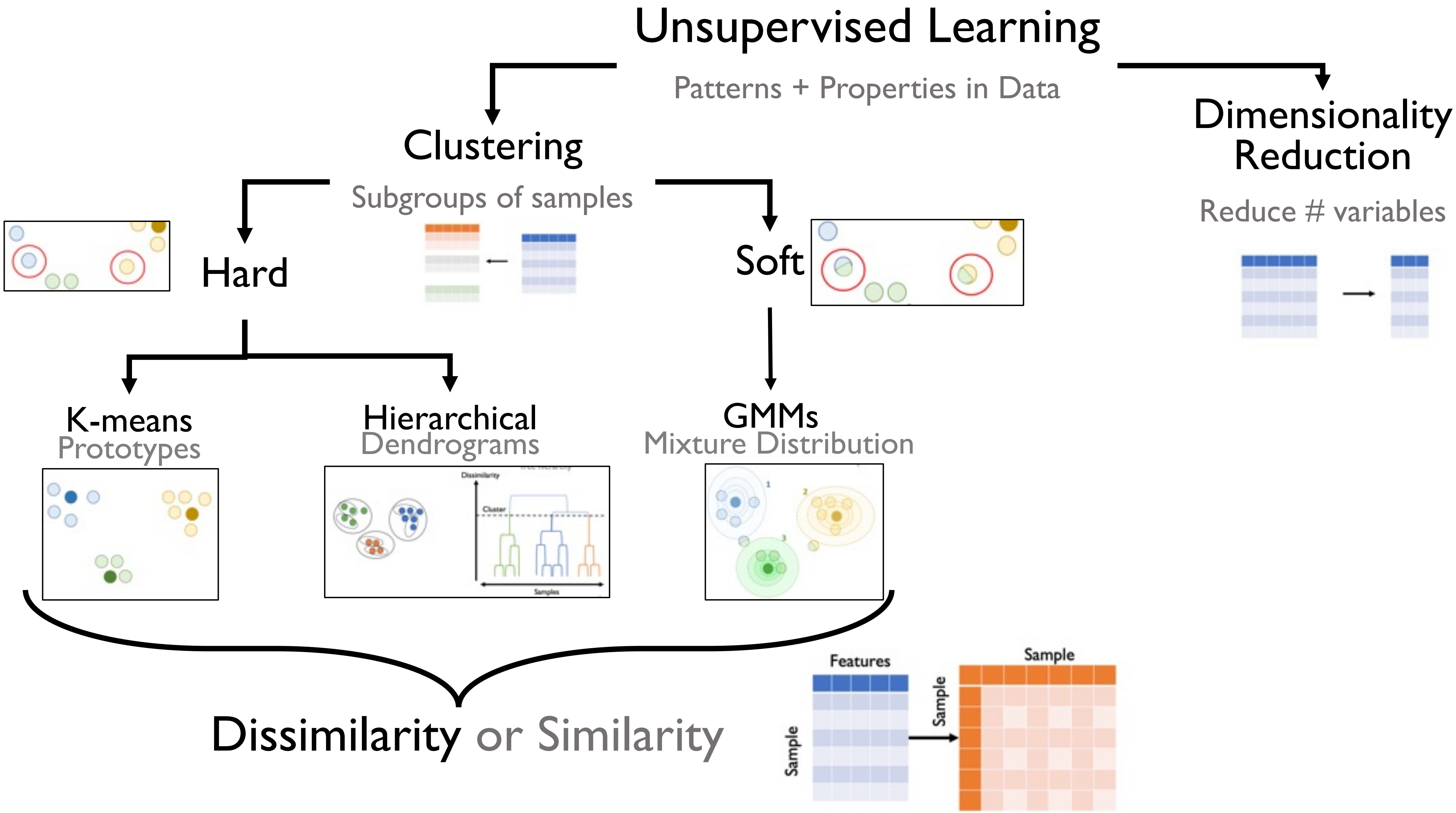
Recommended:

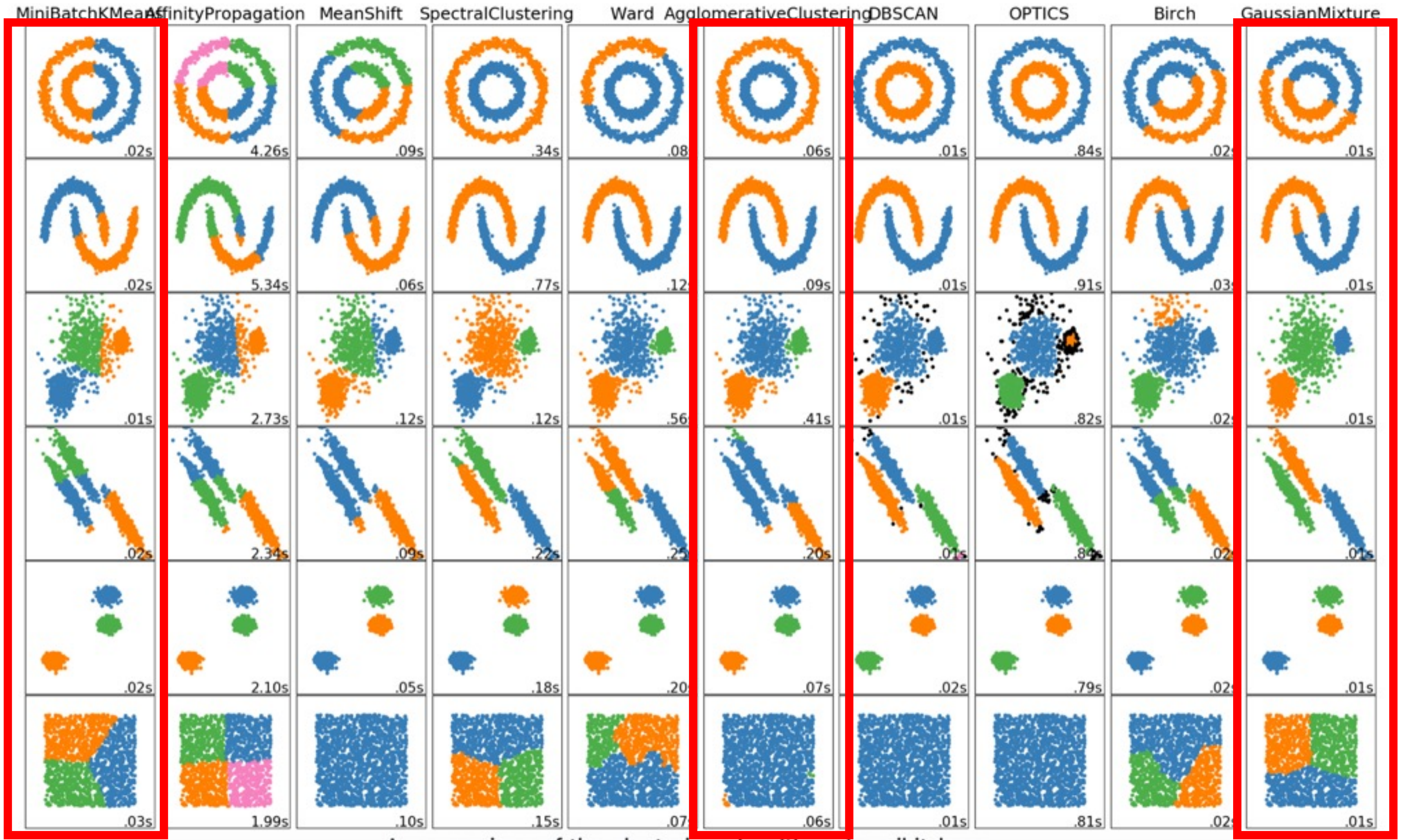
Ten quick tips for effective dimensionality reduction

Lan Huong Nguyen, Susan Holmes.(2019)

https://doi.org/10.1371/journal.pcbi.1006907

Last class recap



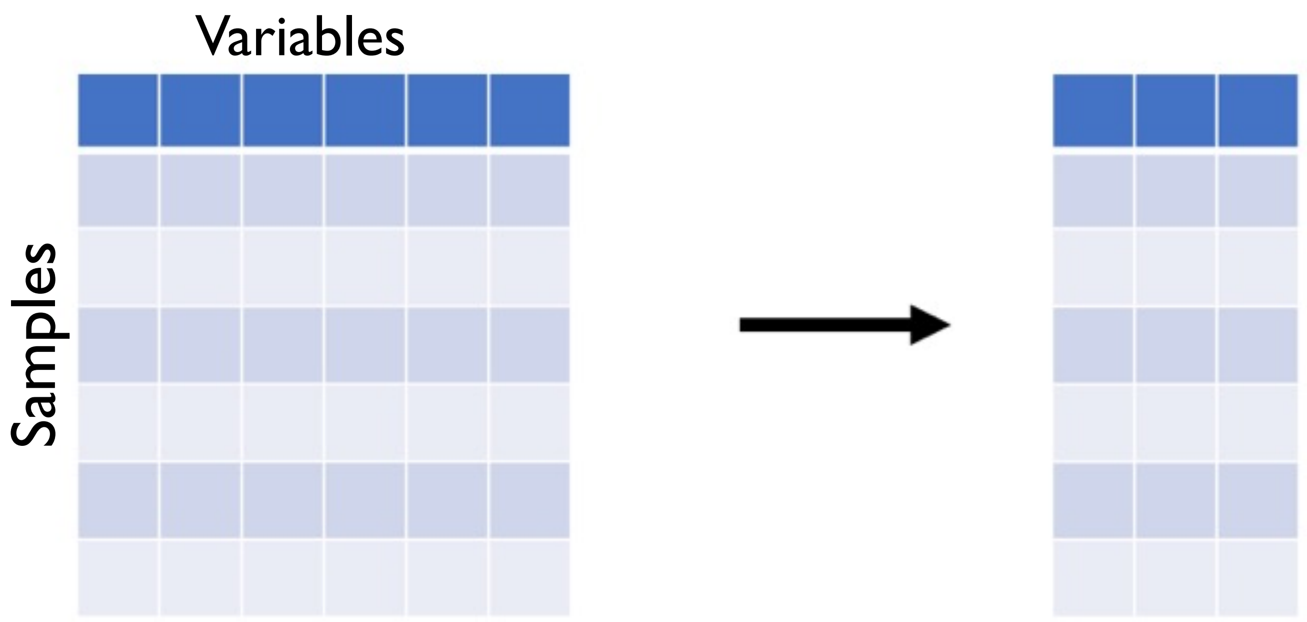


A comparison of the clustering algorithms in scikit-learn

Code Available https://scikit-learn.org/stable/modules/clustering.html

What is dimensionality reduction?

Reduce # of variables preserving most of the information



Latent variables

Visualization

Data compression

Noise reduction

Algorithm performance

* Focus on qualitative variables
For Categorical: Adaptations using dummy variables
More in Nguyen et al.

PCA: Principal Component Analysis

The most common DR method

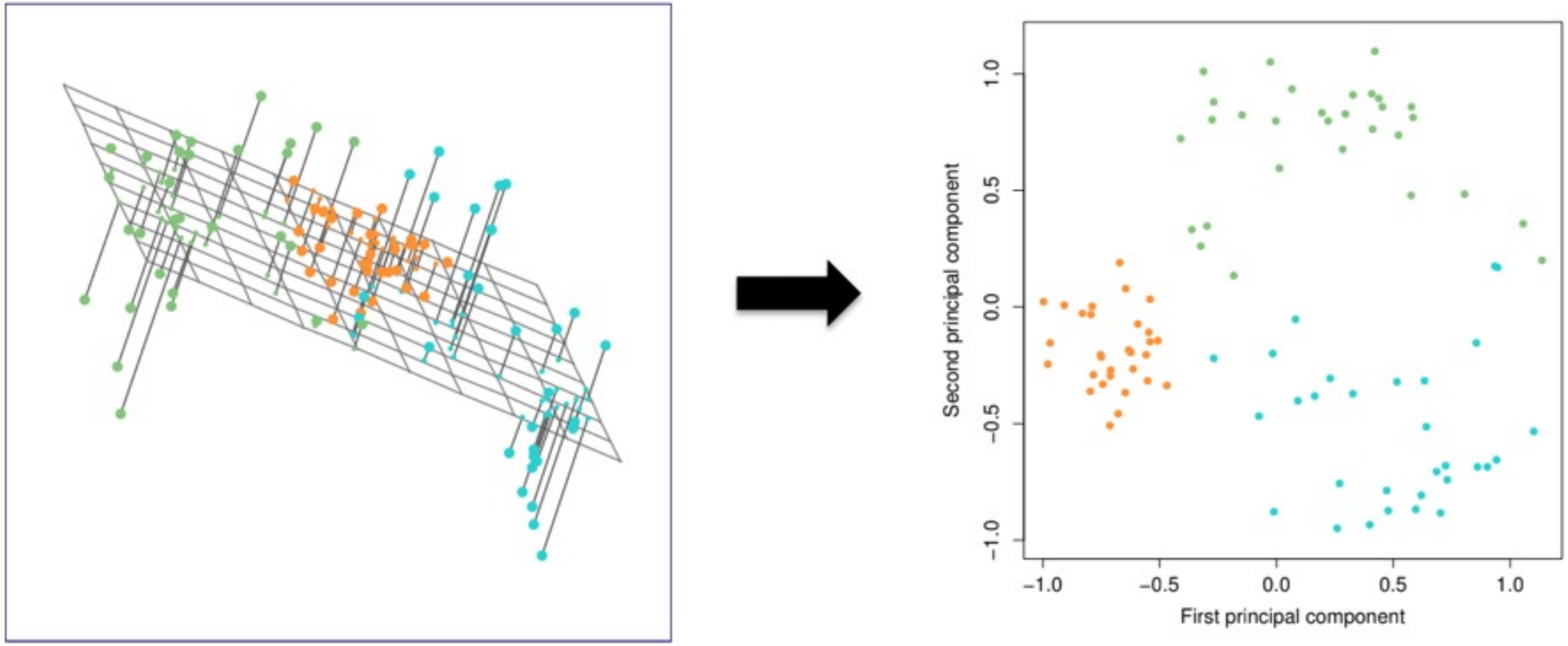


Figure 10.2 ISL (2013)

Goal: Find the projection that maximizes variation

PCA: What is 1st principal component?

Variables: $X_1, X_2, \dots, X_p$

1st principal component	$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{p1}X_p$	the largest variance

PCA: Computing 1st principal component

For each sample: $x_1^{(i)}, x_2^{(i)}, \dots, x_p^{(i)}$

1) Center observations $\tilde{x}_j^{(i)} = x_j^{(i)} - \overline{x_j}$

2) We look for $z_1^{(i)} = \phi_{11}\tilde{x}_1^{(i)} + \phi_{21}\tilde{x}_2^{(i)} + \cdots + \phi_{p1}\tilde{x}_p^{(i)}$

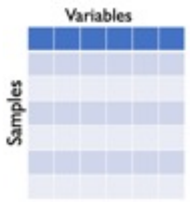
by solving

$$\max_{\phi_{11},\dots,\phi_{21}} \frac{1}{N}\sum_{i=1}^{N}\left(z_1^{(i)}\right)^2 \quad \text{s.t.} \quad \sum_{j=1}^{p}\phi_{p1}^2 = 1$$

variance normalized

PCA: Computing 1st principal component (In matrix form)

Observation vs. feature $\boldsymbol{X}$



1) Center observations $\widetilde{\boldsymbol{X}} = \left(\boldsymbol{I} - \frac{\boldsymbol{1}}{\boldsymbol{n}}\boldsymbol{1}\boldsymbol{1}^{\boldsymbol{T}}\right)\boldsymbol{X}$

2) We look for $z_1 = \widetilde{\boldsymbol{X}}\phi_1$

by solving

$$\max_{\phi_1} \phi_1^T(\widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}})\phi_1 \qquad \text{s.t.} \qquad \|\phi_1\|_2 = 1$$

normalized

variance

Solution = Largest **eigenvector** of $\widetilde{X^T}\,\widetilde{X}$ **normalized**

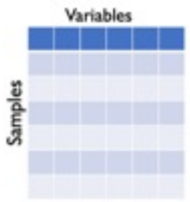
PCA: What are first k principal components?

Variables: $X_1, X_2, \dots, X_p$

k principal components	$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{p1}X_p$ $\dots$ $Z_k = \phi_{1k}X_1 + \phi_{2k}X_2 + \cdots + \phi_{pk}X_p$	the largest variance and uncorrelated

PCA: Computing k principal components (In matrix form)

Observation vs. feature $\boldsymbol{X}$



1) Center observations $\widetilde{\boldsymbol{X}} = \left(\boldsymbol{I} - \frac{\boldsymbol{1}}{\boldsymbol{n}}\boldsymbol{1}\boldsymbol{1}^{\boldsymbol{T}}\right)\boldsymbol{X}$

2) We look for $\boldsymbol{Z_k} = \widetilde{\boldsymbol{X}}\boldsymbol{V_k}$

by solving

$$\max_{V_k} tr\left(V_k^T \widetilde{\boldsymbol{X}}^T \widetilde{\boldsymbol{X}} V_k\right) \quad \text{s.t.} \quad \boldsymbol{V_k^T V_k} = \boldsymbol{I}$$

variance

normalized
and
uncorrelated

Solution = **orthogonal** projection into k largest **eigenvectors** of $\widetilde{X^T}\,\widetilde{X}$

PCA: How to compute eigenvectors?

Eigenvalue Decomposition

$$\boldsymbol{A} = \boldsymbol{B\Lambda B^{-1}}$$ $\boldsymbol{\Lambda}$: diagonal

For $\widetilde{X}^T\widetilde{X}$: $$\widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}} = \boldsymbol{VD^2V^T}$$ s.p.d: symmetric positive definite

Singular Value Decomposition

$$\widetilde{\boldsymbol{X}} = \boldsymbol{UDV^T}$$

$$d_{11} \geq d_{22} \geq \cdots \geq d_{NN} \geq 0$$

$$\boldsymbol{U^TU = I, V^TV = I}$$

PCA: The SVD and the principal components

We look for $\boldsymbol{Z_k} = \boldsymbol{\widetilde{X}V_k}$ uncorrelated and with largest variance

SVD: $\boldsymbol{\widetilde{X}} = \boldsymbol{UDV^T}$

K principal components

- Take $\boldsymbol{V_k}$ = first k columns of V
- $\boldsymbol{Z_k} = \boldsymbol{U_k D_k}$

Truncated SVD

PCA = SVD applied to centered matrix $\boldsymbol{\widetilde{X}} = \left(\boldsymbol{I} - \frac{\boldsymbol{1}}{\boldsymbol{n}}\boldsymbol{1}\boldsymbol{1}^{\boldsymbol{T}}\right)\boldsymbol{X}$

The SVD and data compression

For any A, truncated SVD minimizes

$$\min_{U_k, D_k, V_k} \left\| A - U_k D_k V_k^T \right\|_2$$

$U_k \in R^{N \times k}, D_k \in R^{\times k}, V_k \in R^{p \times k}$

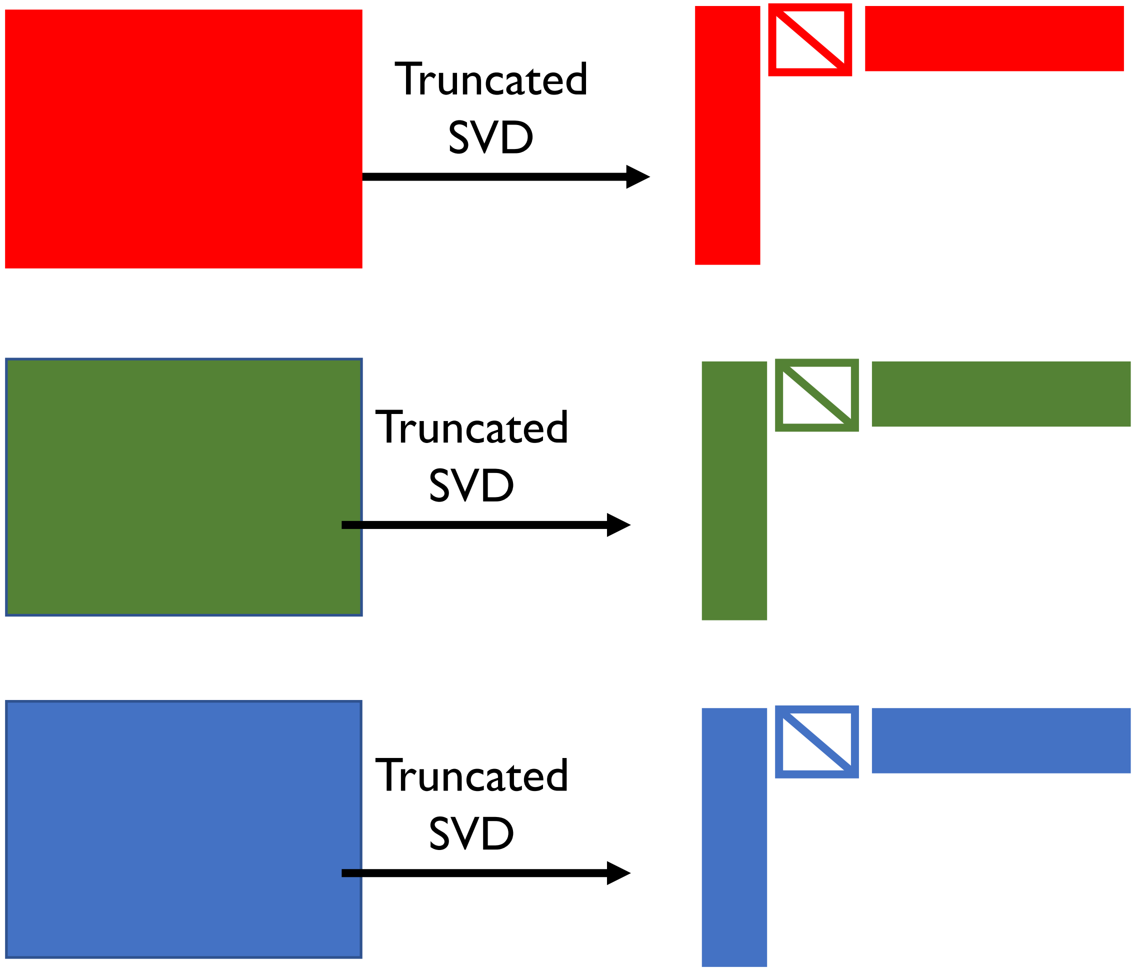
Orthogonal Diagonal Orthogonal

Truncated SVD →

12 times smaller

http://timbaumann.info/svd-image-compression-demo/

The SVD and data compression

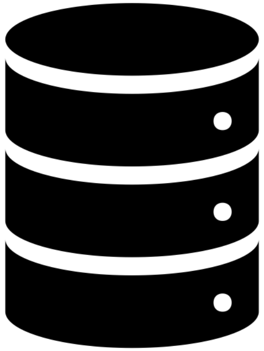


12 times smaller

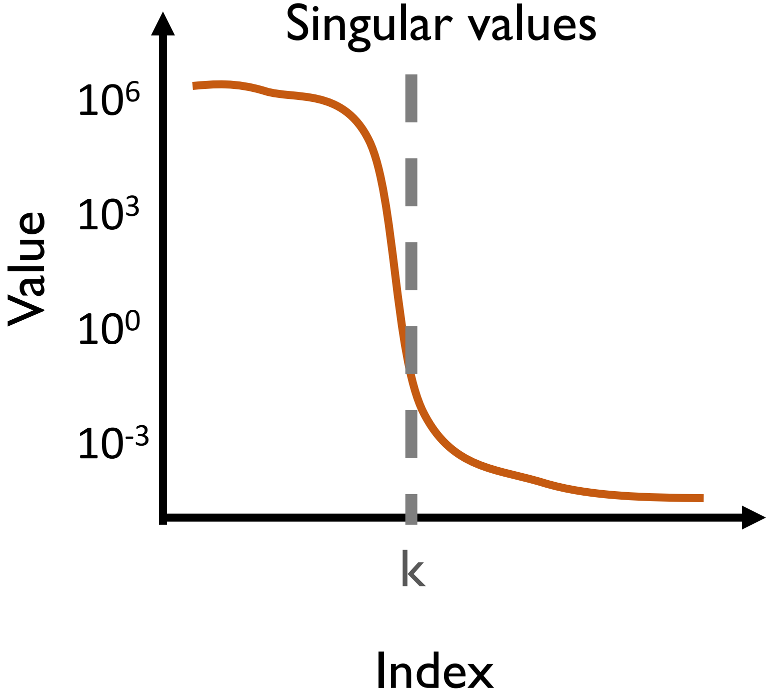
http://timbaumann.info/svd-image-compression-demo/

PCA: How many k do we pick?

Space available



"Clear" cut

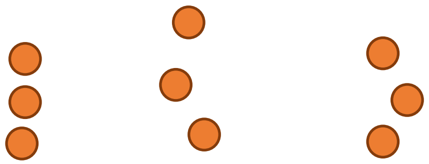


% Variance

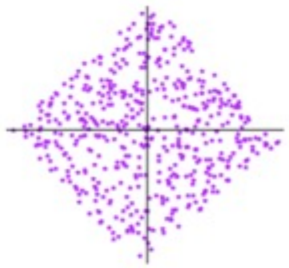
$$\frac{\sum_{i=1}^{k} d_i^2}{\sum_{i=1}^{p} d_i^2} = \frac{\text{Variance Explained}}{\text{Total Variance}}$$

PCA: SVD Challenges

Not all variables have the same influence

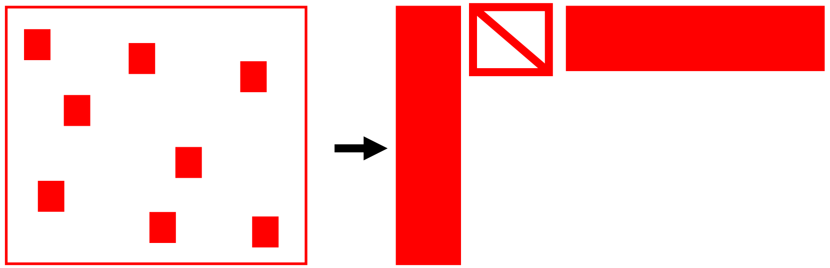


Uncorrelation, not independence

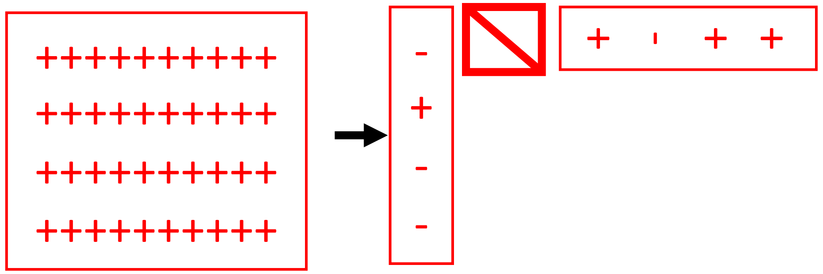


Non robust

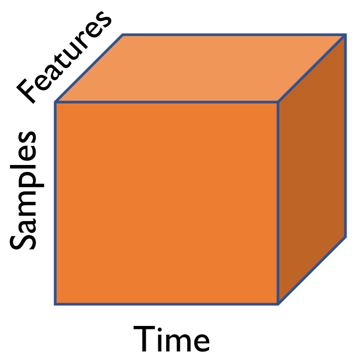
Non suitable for sparse data, expensive to compute



Does not preserve non negativity



Only 2D arrays



PCA: SVD Challenges

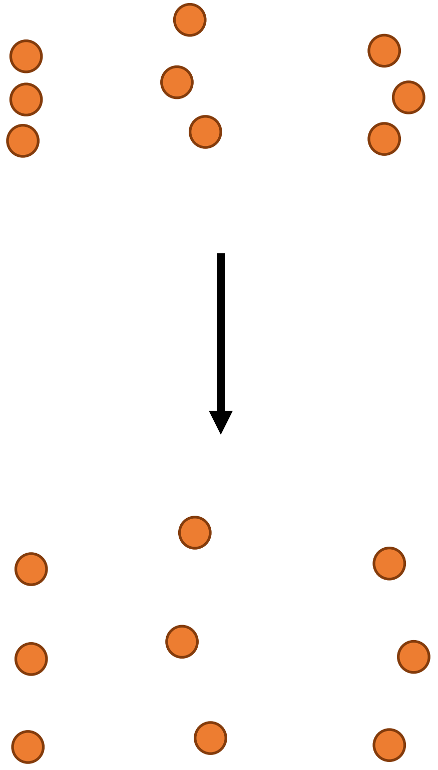
$$\min_{U_k, D_k, V_k} \left\| A - U_k D_k V_k^T \right\|_2 \qquad U_k \in R^{N \times k}, D_k \in R^{\times k}, V_k \in R^{p \times k}$$

Orthogonal Diagonal Orthogonal

Solution to Challenges → Additional optimization constraints → Other matrix factorizations

Makes optimization more challenging

PCA: Not all variables have the same influence

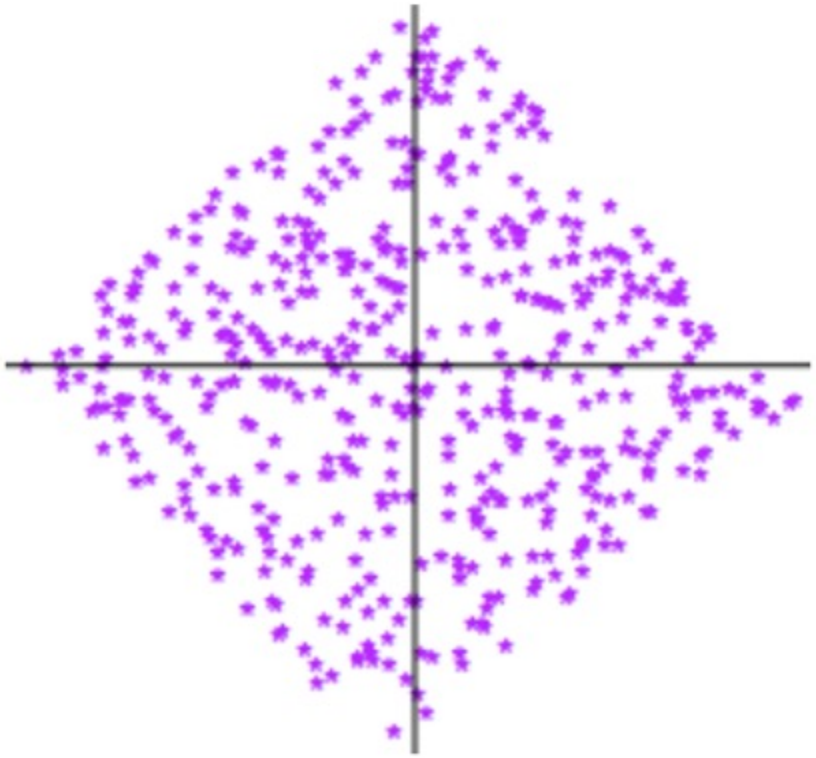


Weighted PCA

$$\boldsymbol{\tilde{X}W \approx U_k D_k V_k^T}$$

e.g. W diagonal $w_i = \frac{1}{\sqrt{var(x_i)}}$

PCA: Components are uncorrelated, but not independent

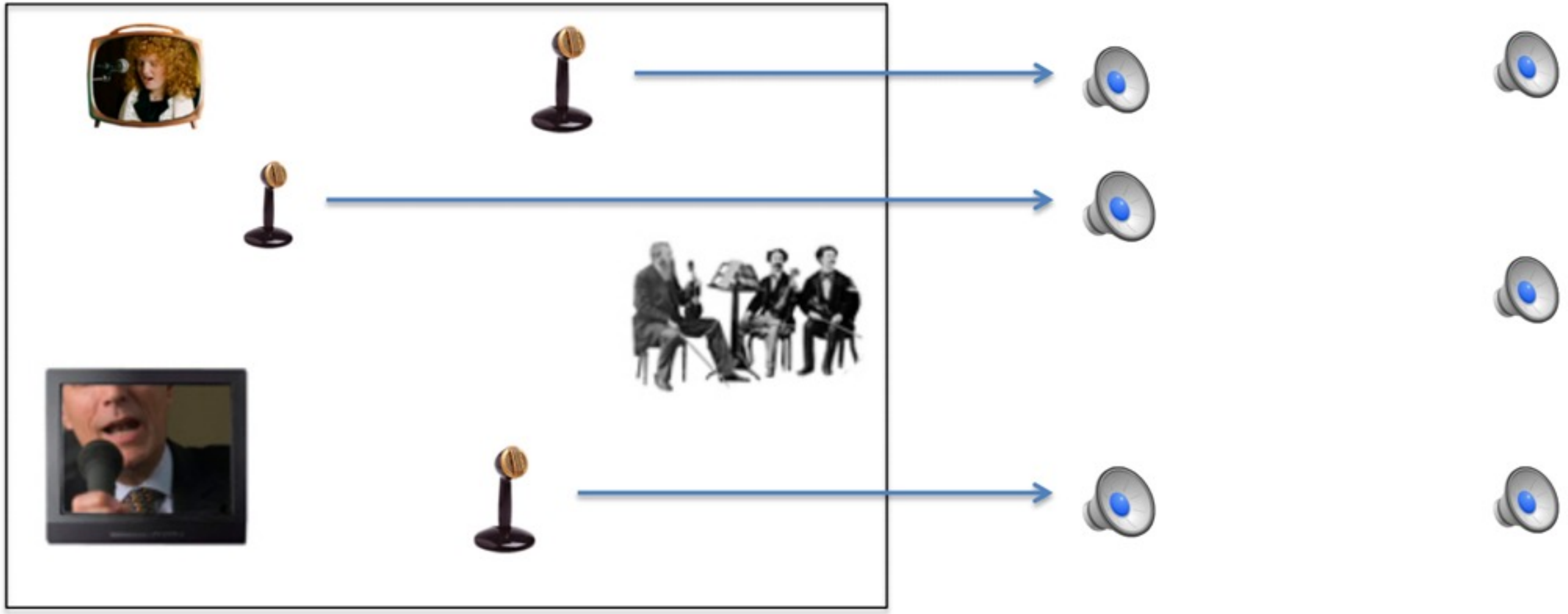


Independent Component Analysis ICA

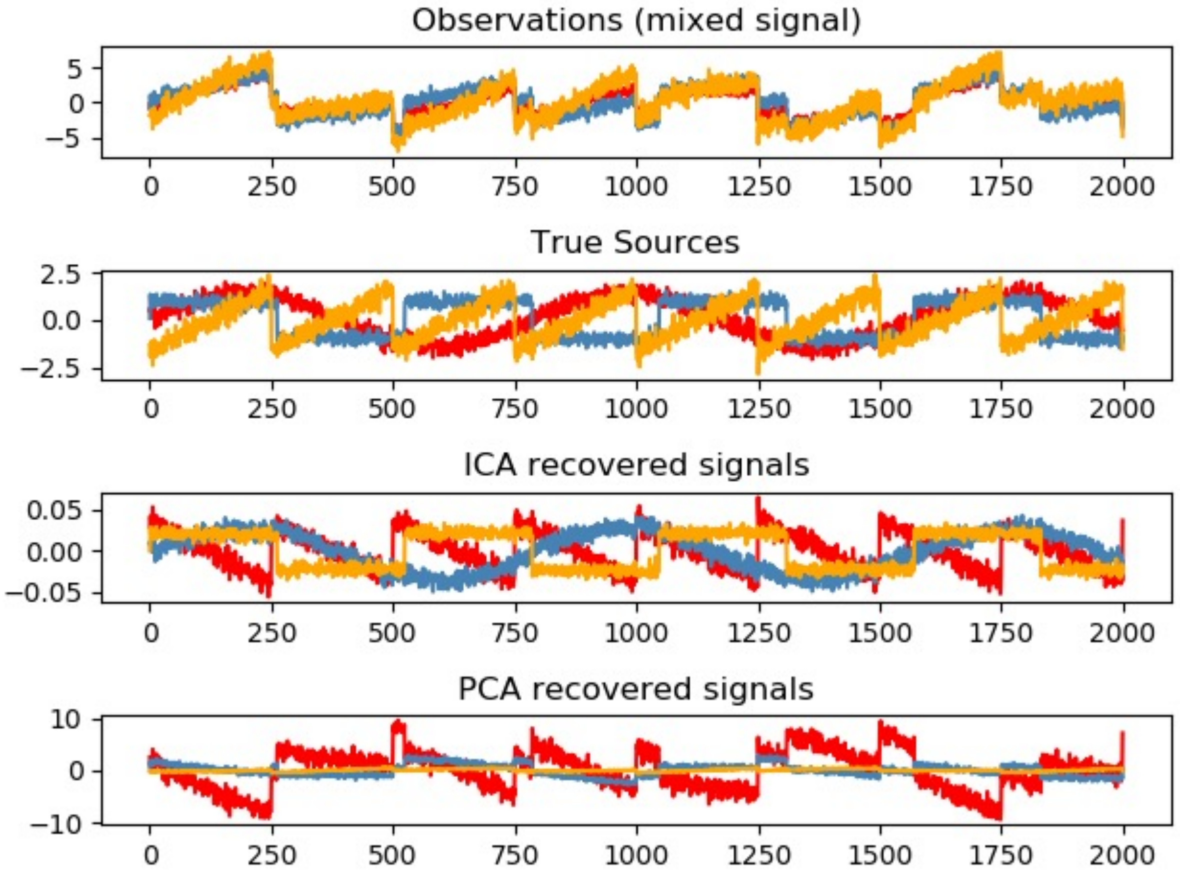
$$\tilde{\boldsymbol{X}}\boldsymbol{\Sigma}^{-1/2} \approx \boldsymbol{U}\boldsymbol{A}$$

$\boldsymbol{U}$ low entropy = non-Gaussian projection

PCA: Components are uncorrelated, but not independent



PCA: Components are uncorrelated, but not independent



https://scikit-learn.org/stable/auto_examples/decomposition/plot_ica_blind_source_separation.html

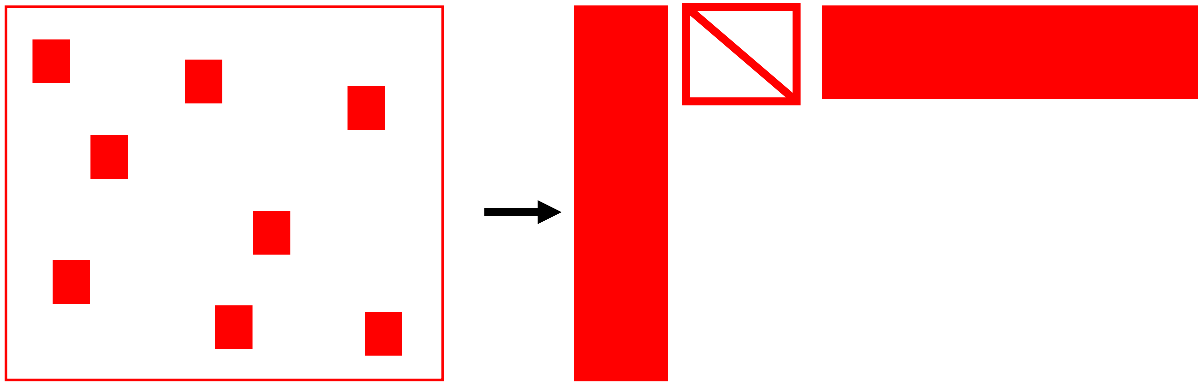
PCA: Non robust to outliers

Robust PCA

$$\widetilde{\boldsymbol{X}} \approx \boldsymbol{U}_k \boldsymbol{D}_k \boldsymbol{V}_k^T + \boldsymbol{S}$$

$\boldsymbol{S}$ sparse

PCA: Non suitable for sparse data, expensive to compute



Sparse PCA

$$\widetilde{X} \approx U_k D_k V_k^T$$

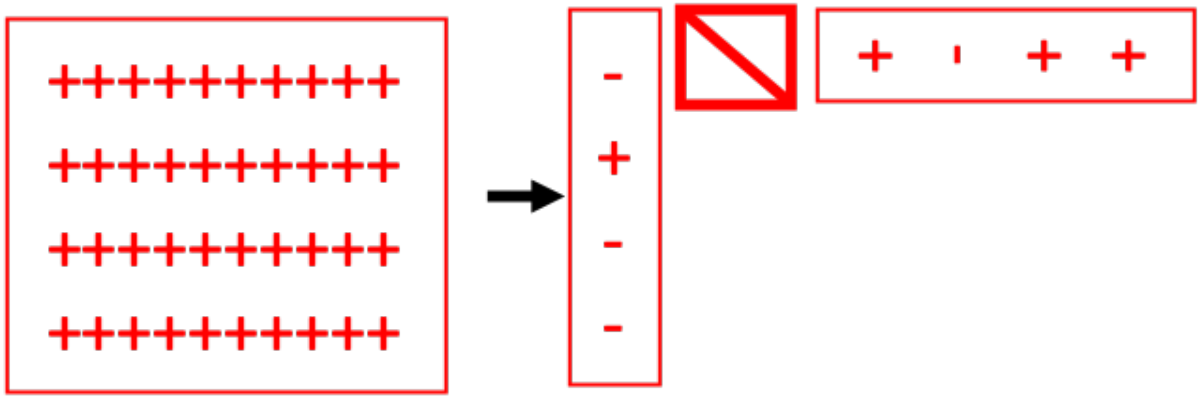
V_k sparse

CUR

$$\widetilde{X} \approx CUR$$

C columns
R rows

PCA: Non suitable for sparse data, expensive to compute

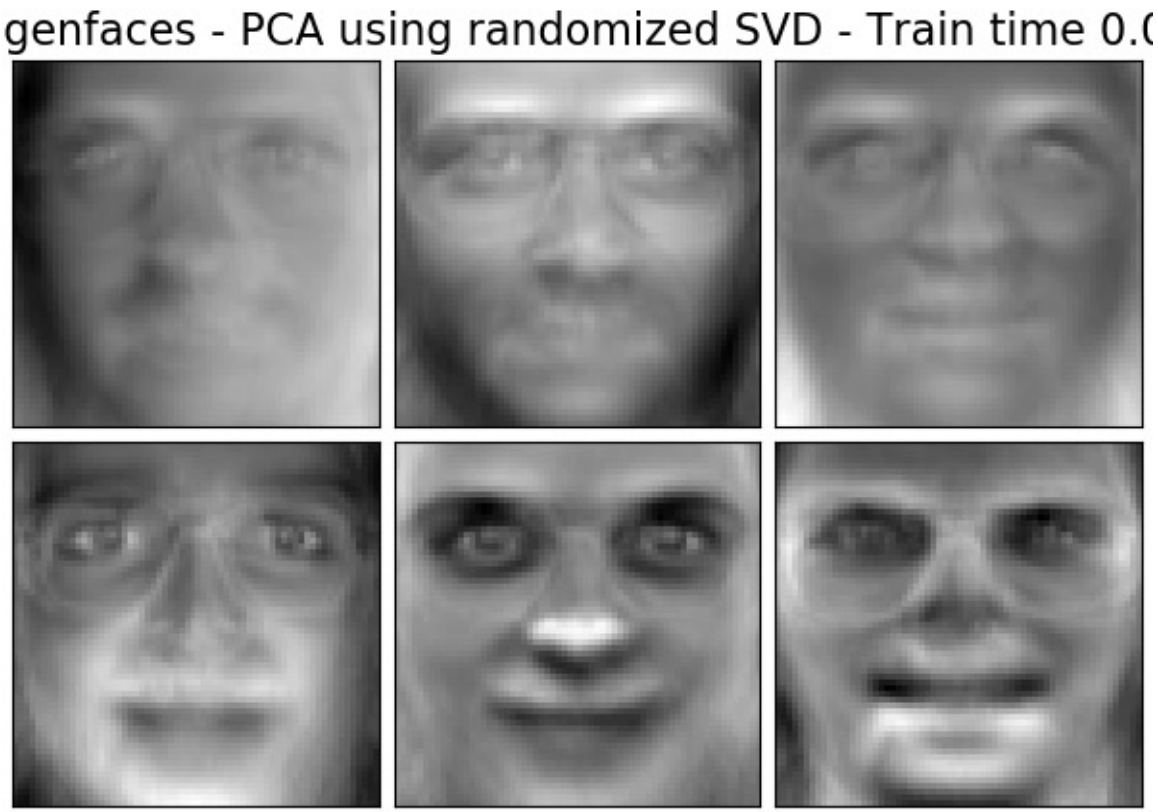


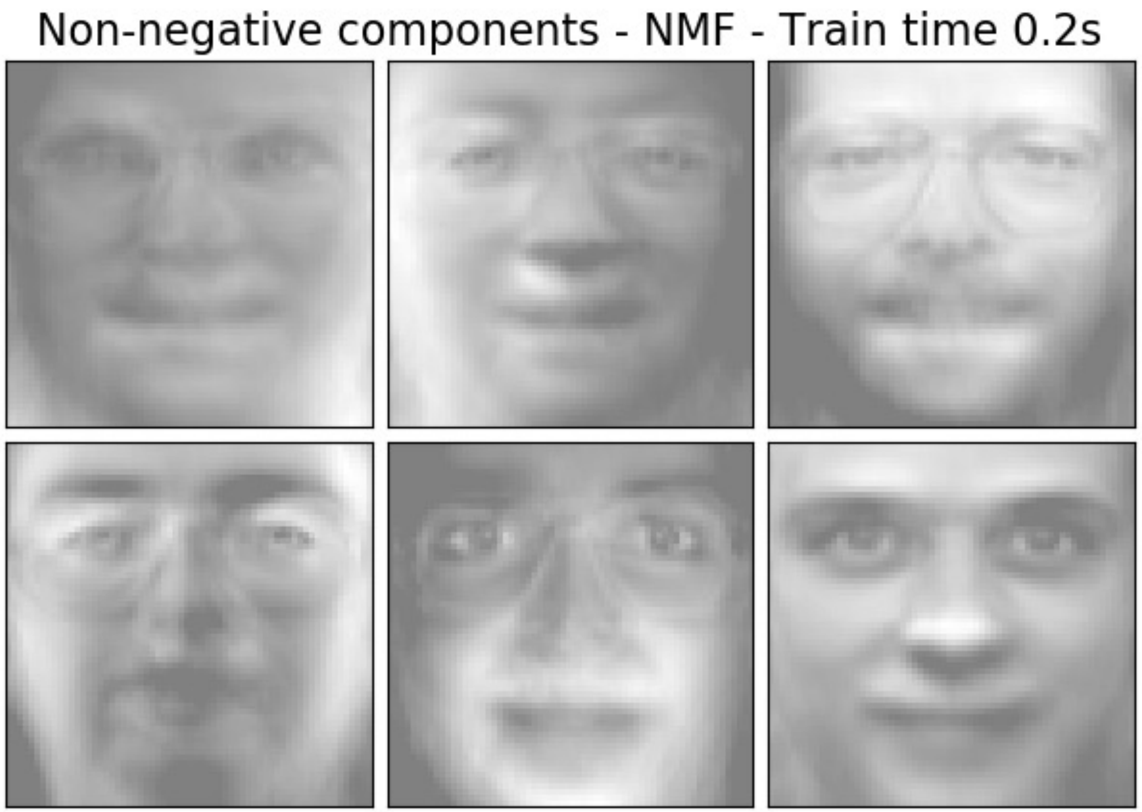
Non-Negative Factorization

$$\tilde{X} \approx WH$$

$$w_{i,k}, h_{k,j} \geq 0$$

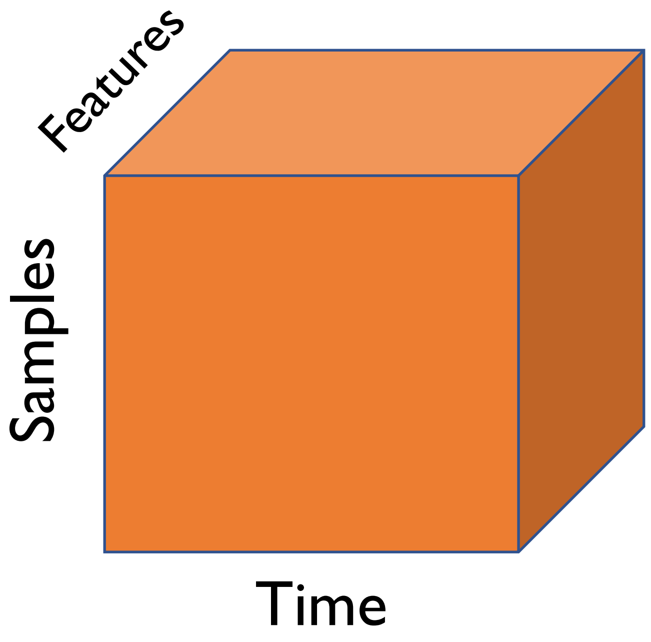
PCA: Non suitable for sparse data, expensive to compute



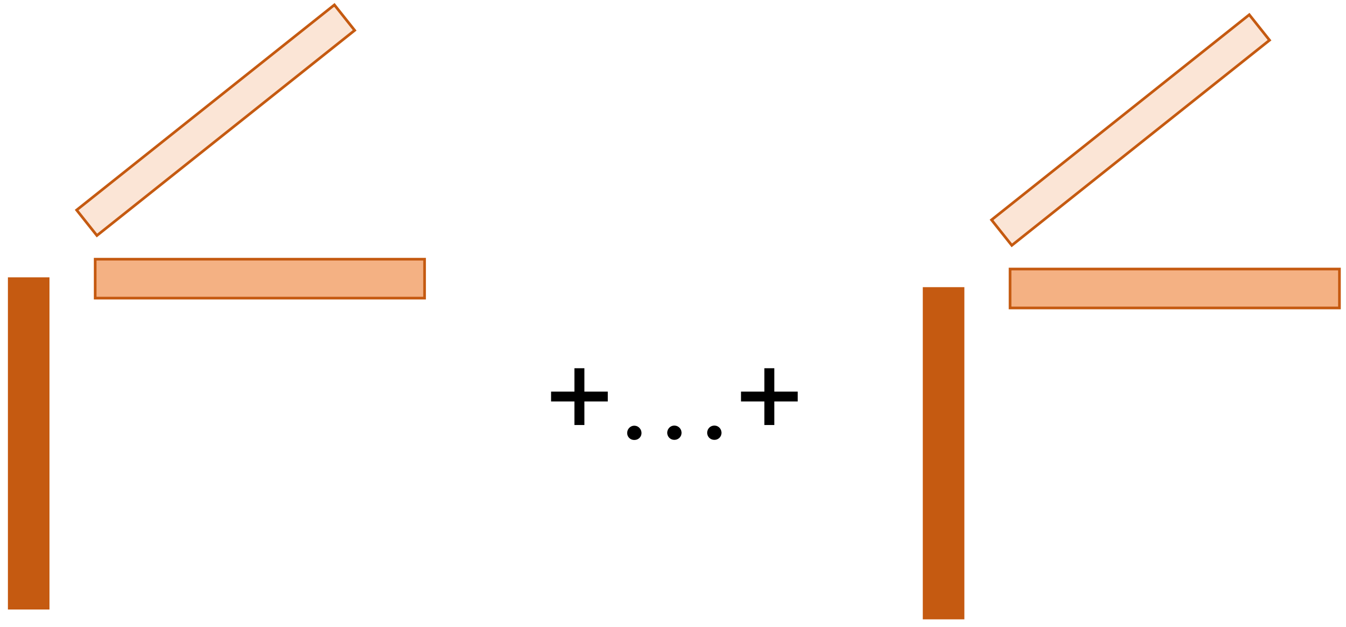


https://scikit-learn.org/stable/auto_examples/decomposition/plot_faces_decomposition.html

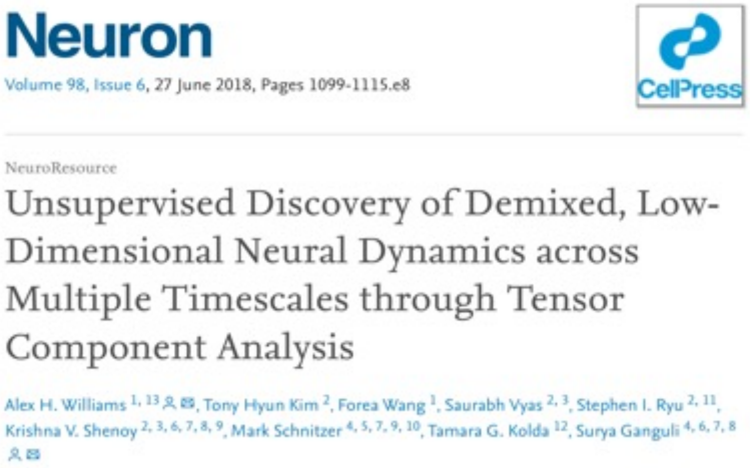
PCA: Only 2D arrays



Tensor Decomposition
CP decomposition

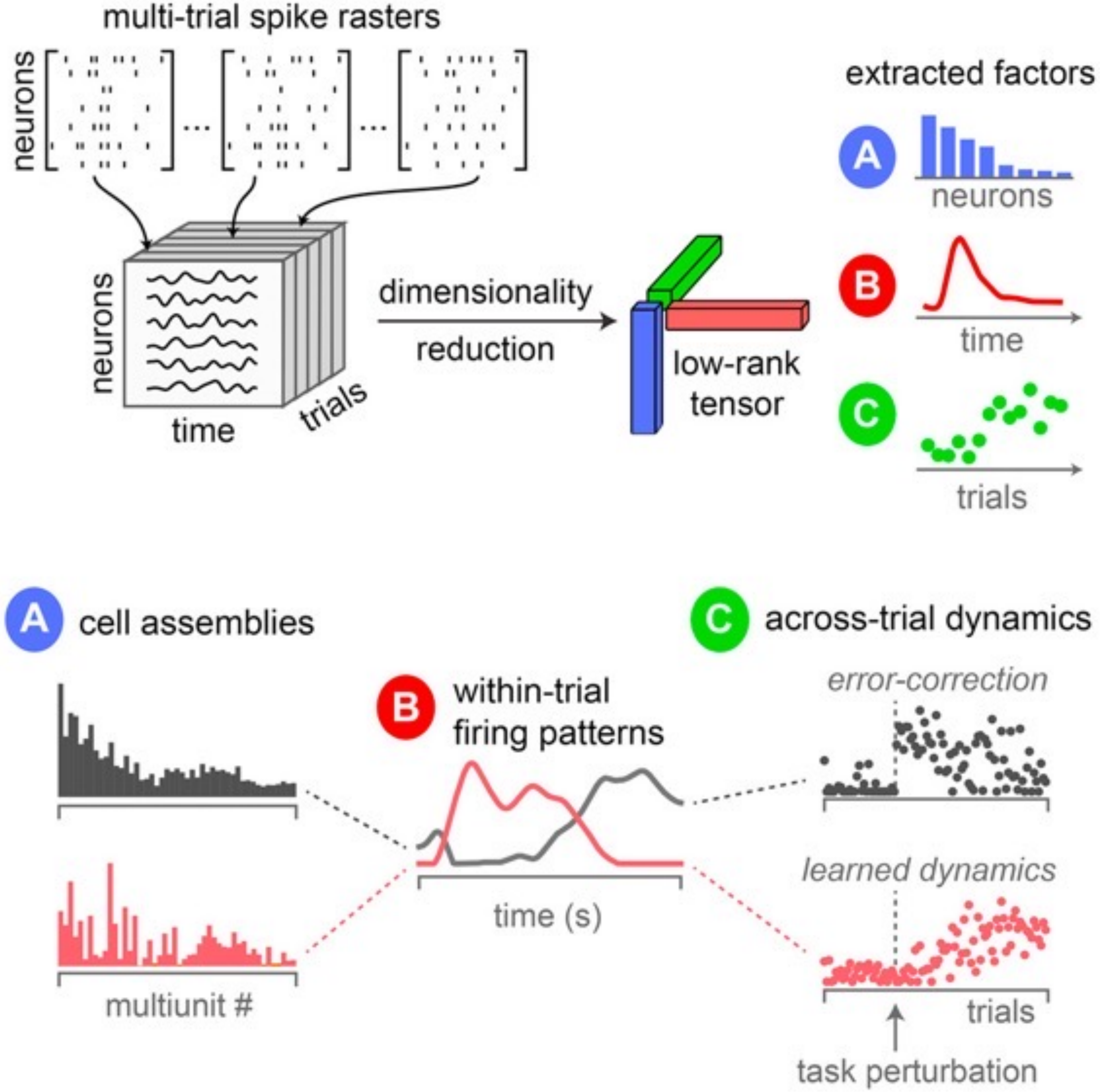


PCA: Only 2D arrays


Neuron
Volume 98, Issue 6, 27 June 2018, Pages 1099-1115.e8
CellPress
NeuroResource

Unsupervised Discovery of Demixed, Low-Dimensional Neural Dynamics across Multiple Timescales through Tensor Component Analysis

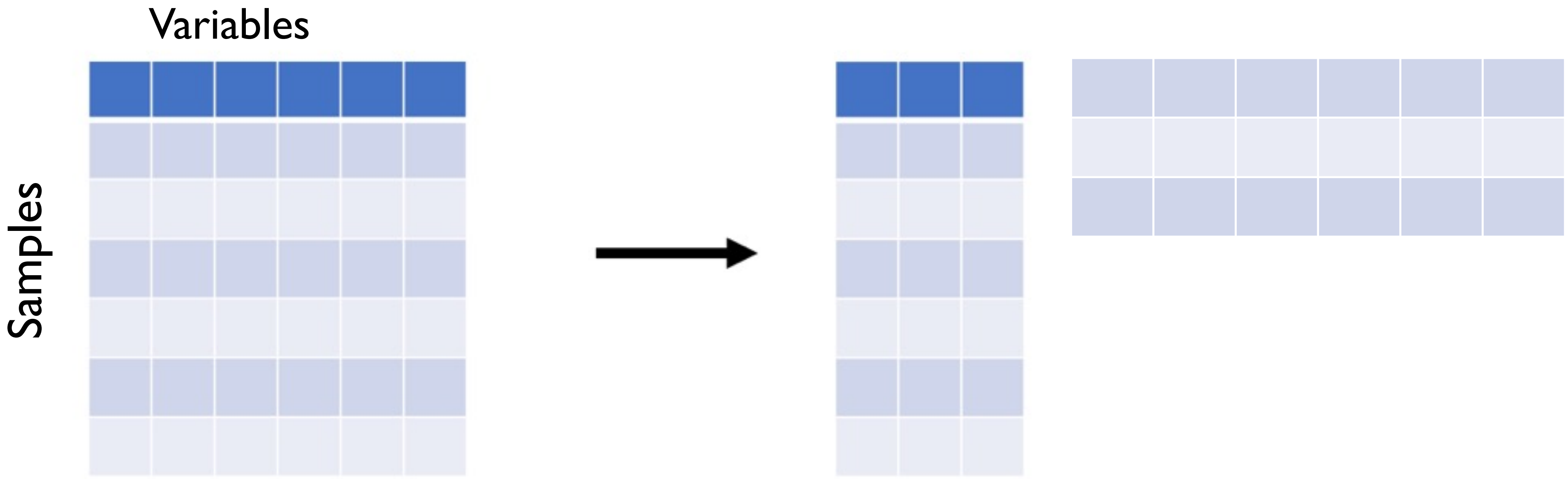
Alex H. Williams [1, 13], Tony Hyun Kim [2], Forea Wang [1], Saurabh Vyas [2, 3], Stephen I. Ryu [2, 11], Krishna V. Shenoy [2, 3, 6, 7, 8, 9], Mark Schnitzer [4, 5, 7, 9, 10], Tamara G. Kolda [12], Surya Ganguli [4, 6, 7, 8]



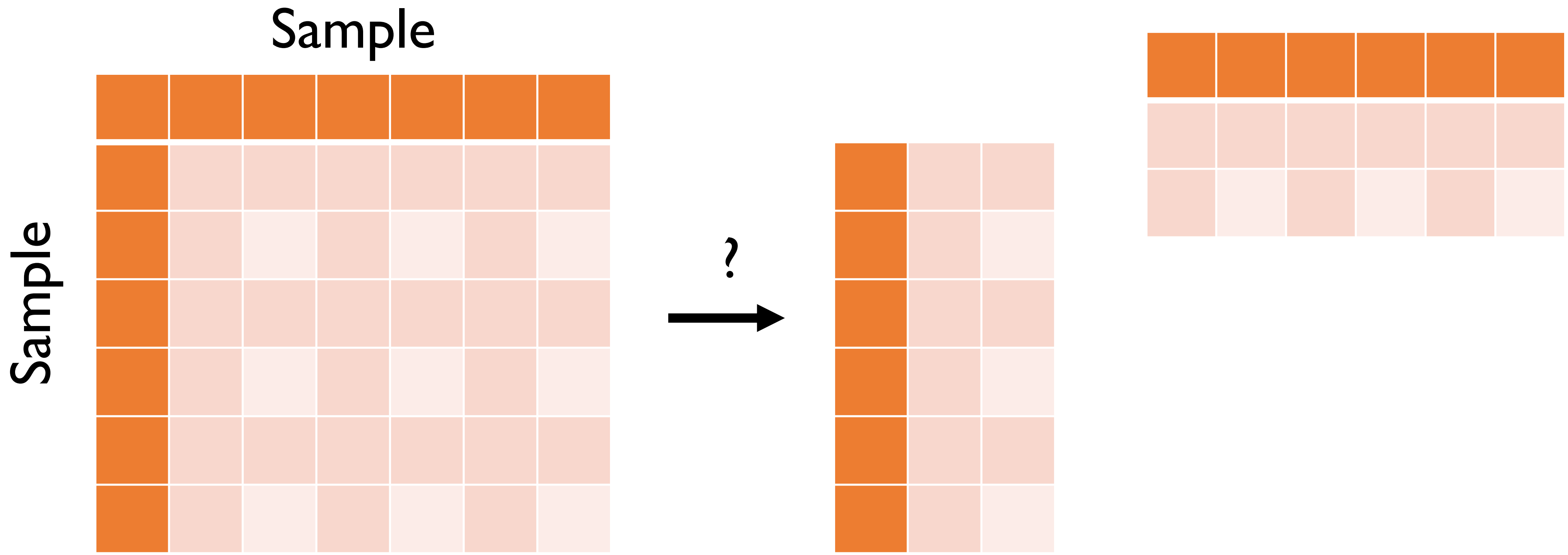
https://doi.org/10.1016/j.neuron.2018.05.015

PCA + Other matrix factorizations

$$\tilde{\boldsymbol{X}} \approx \boldsymbol{U}\boldsymbol{A}$$



What about similarity matrix?



Correlation as similarity

Pattern

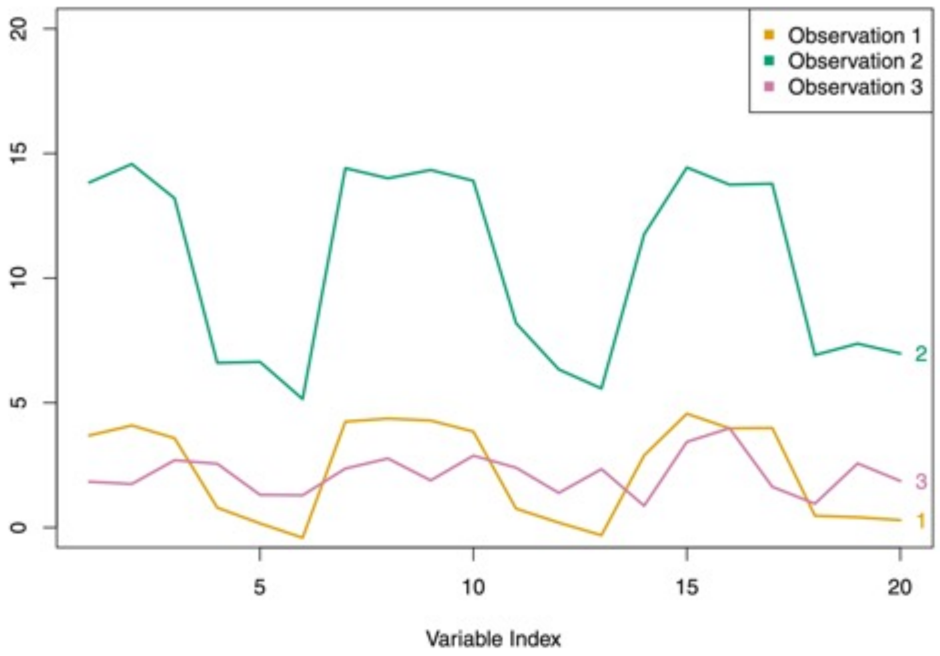


Figure 10.13 ISL (2013)

Correlation

$$d\left(x^{(1)}, x^{(2)}\right) = \sum_{k=1}^{P} \frac{\left(x_k^{(1)} - \bar{x}^{(1)}\right)\left(x_k^{(2)} - \bar{x}^{(2)}\right)}{\sqrt{\sum_{k=1}^{P}\left(x_k^{(1)} - \bar{x}^{(1)}\right)^2} \sqrt{\sum_{k=1}^{P}\left(x_k^{(2)} - \bar{x}^{(2)}\right)^2}}$$

In matrix notation, $\widetilde{\boldsymbol{K}}$ similarity matrix

$$\widetilde{\boldsymbol{K}} = \widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{X}}^{\boldsymbol{T}}$$

SVD and Similarity matrix

In matrix notation, K similarity matrix

$$\tilde{\boldsymbol{K}} = \tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^T$$

In terms of cosine similarity

$$\tilde{\boldsymbol{K}} = \left(\boldsymbol{I} - \frac{\boldsymbol{1}}{\boldsymbol{n}}\boldsymbol{1}\boldsymbol{1}^T\right)\boldsymbol{X}\boldsymbol{X}^T\left(\boldsymbol{I} - \frac{\boldsymbol{1}}{\boldsymbol{n}}\boldsymbol{1}\boldsymbol{1}^T\right)$$

Using SVD of $\tilde{\boldsymbol{X}}$ (Principal Components)

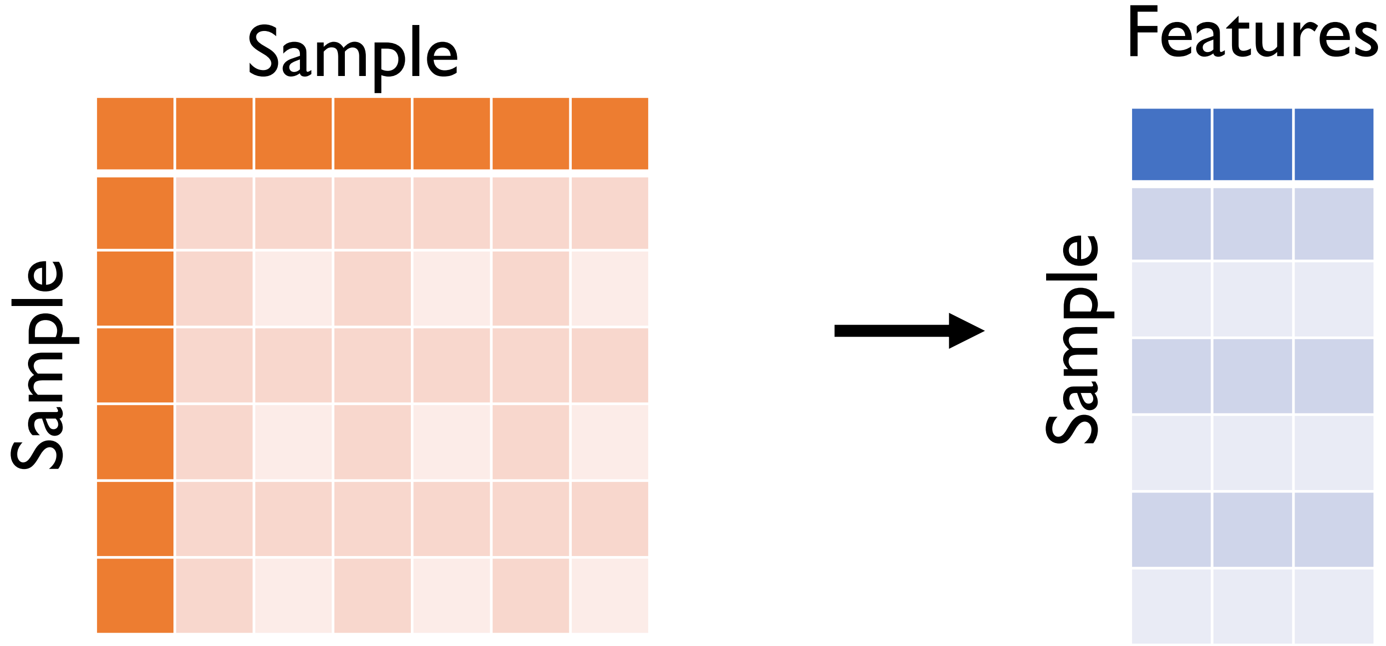
$$\tilde{\boldsymbol{K}} = \boldsymbol{U}\boldsymbol{D}^2\boldsymbol{U}^T = \boldsymbol{Z}\boldsymbol{Z}^T$$

What if we prefer another similarity measure?

Kernel PCA

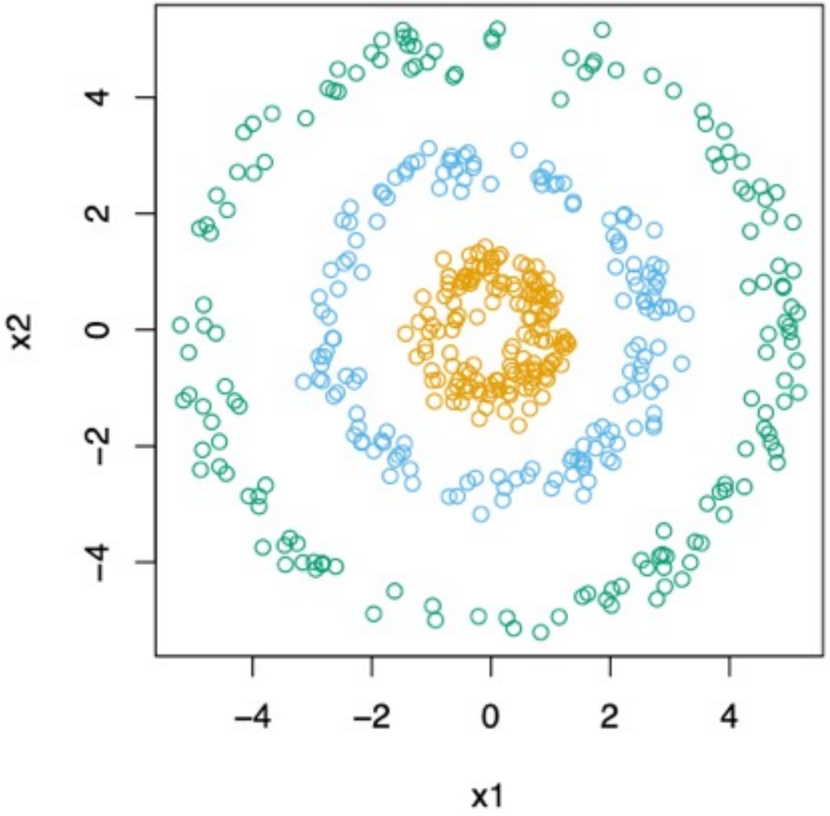
In terms of any similarity $$\tilde{K} = \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right) K \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)$$

computing SVD (Principal Components) $$\tilde{K} = UD^2U^T = ZZ^T$$

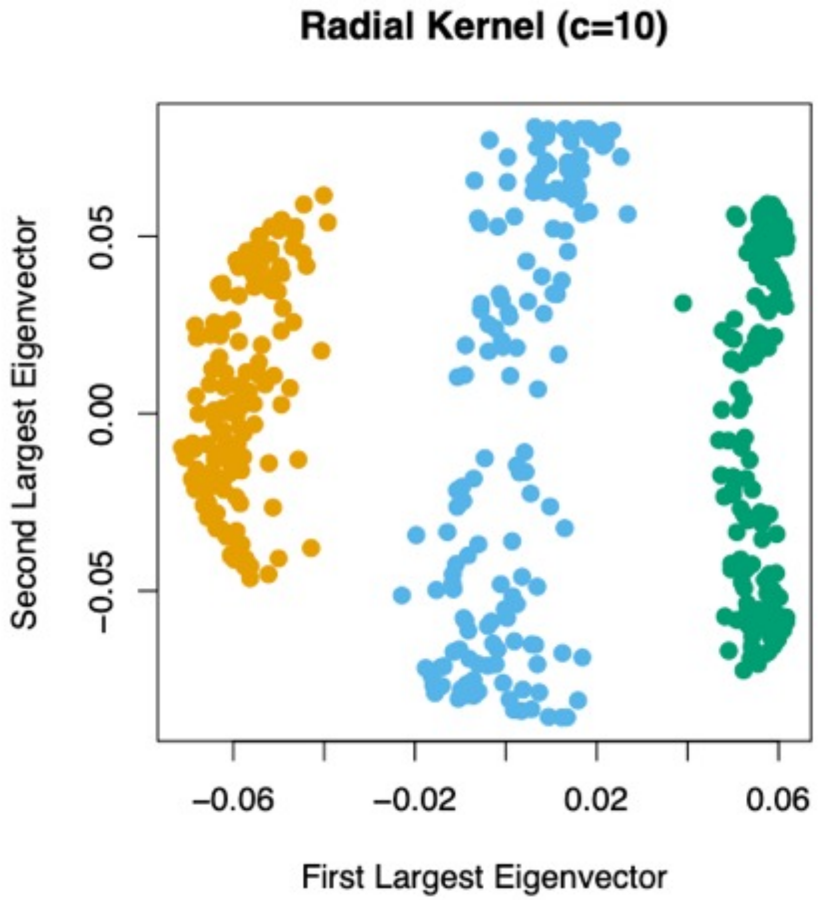


Create features that preserve similarity

Kernel PCA

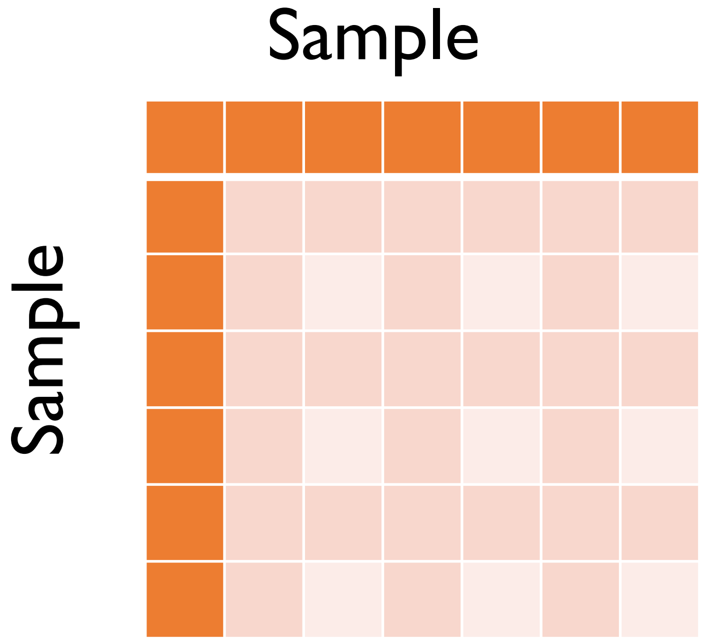


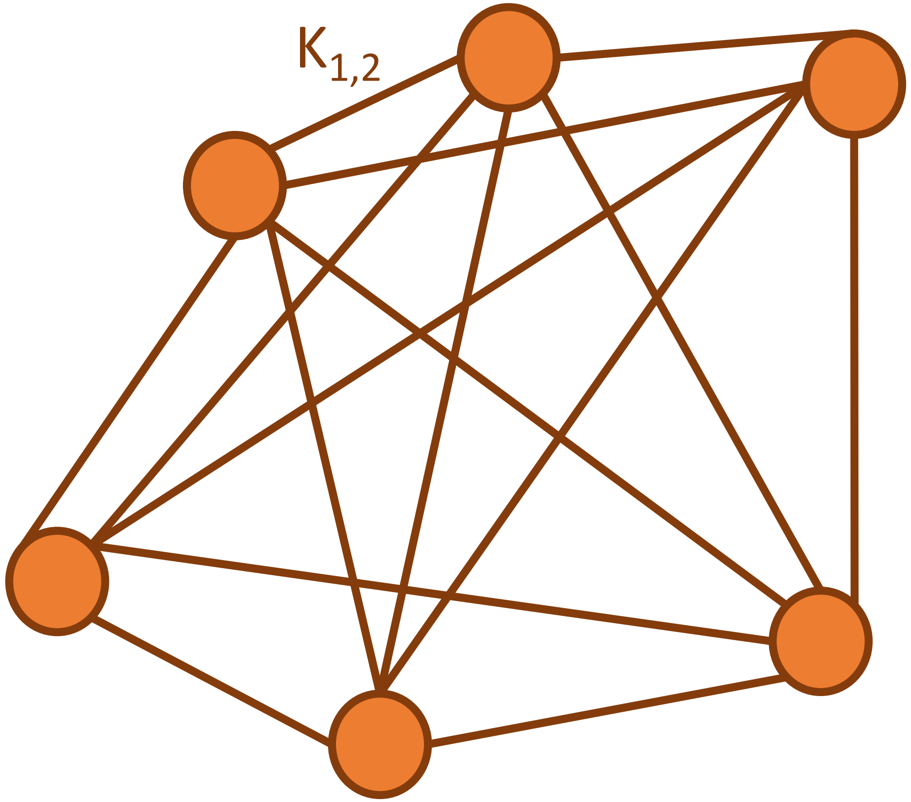
$$K_{i,j} = \exp\left(\frac{\left\|x^{(i)} - x^{(j)}\right\|^2}{c}\right)$$

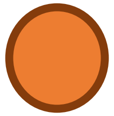


Fragment of Figure 14.30, ESL (2nd edition 2009)

How to interpret similarity matrix?



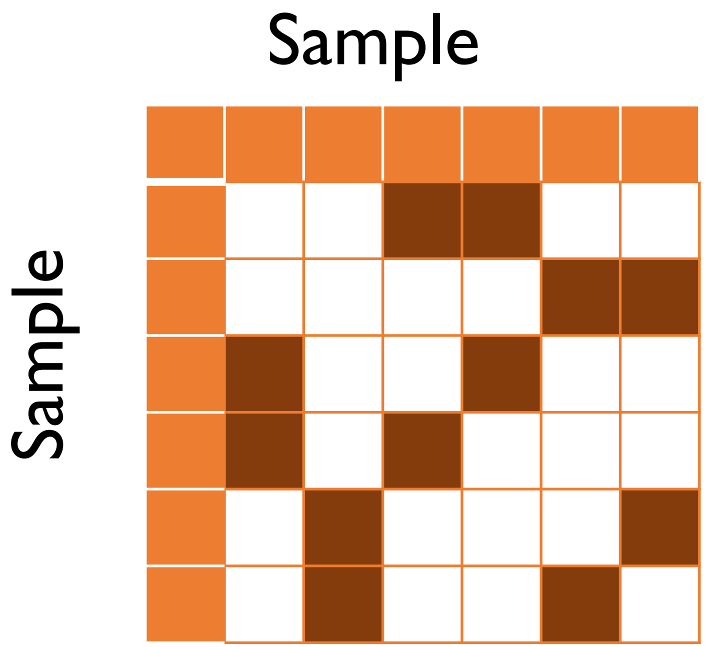




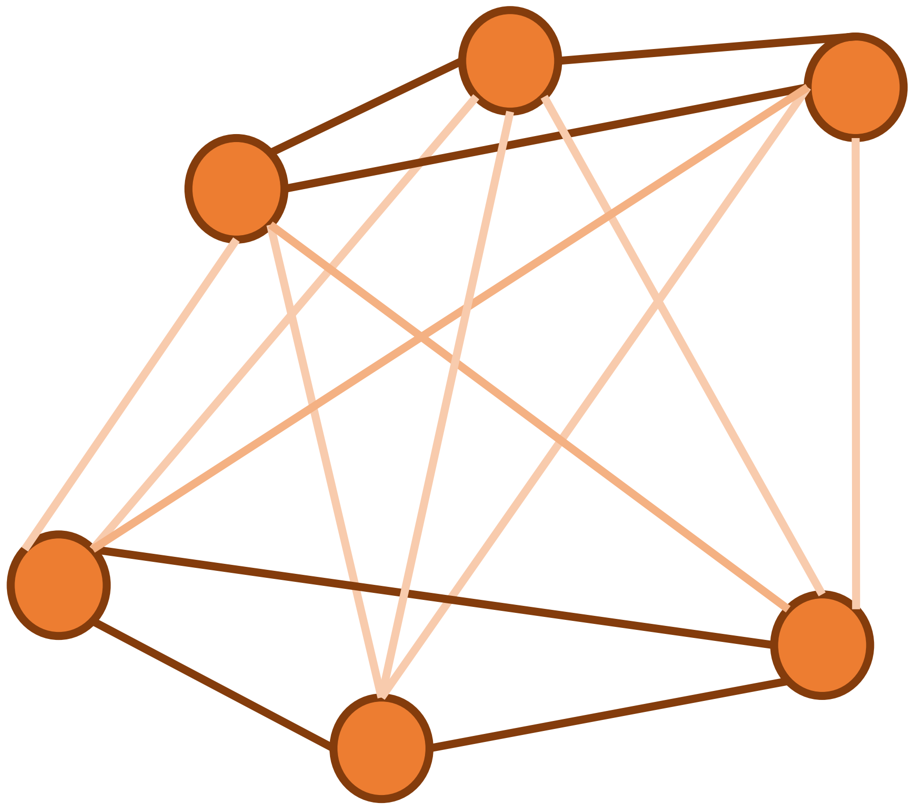
Sample

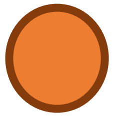
Edge weight = similarity

K-Nearest Neighbors graph



W = adjacency matrix





Sample

Edge weight = similarity

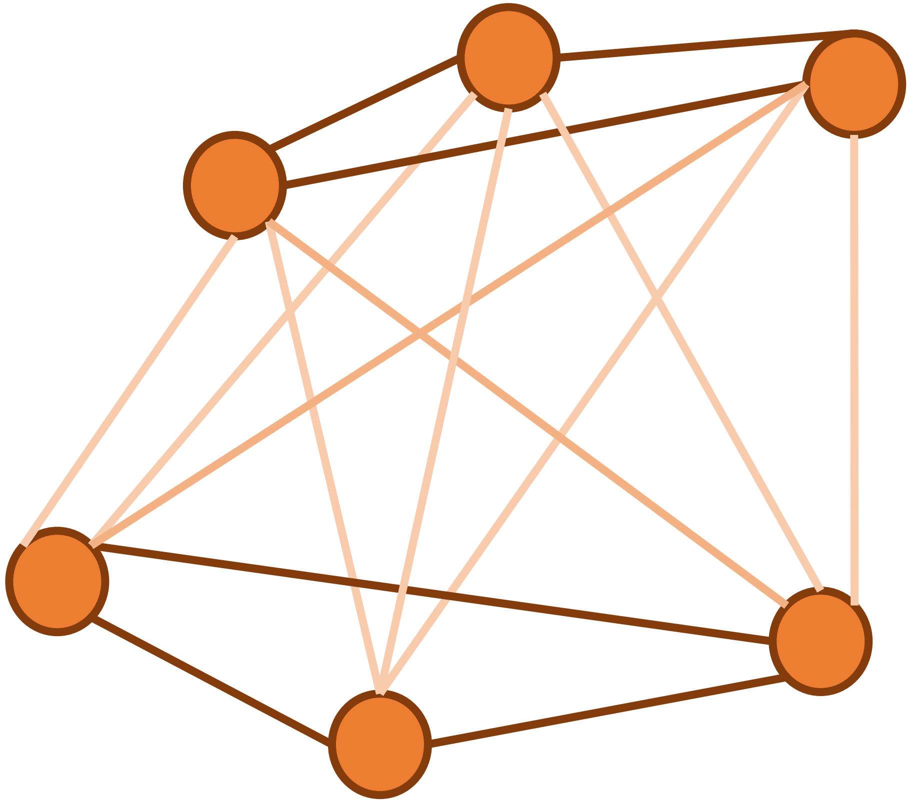
Graph Laplacian

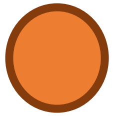
$\boldsymbol{W}$ = Adjacency Matrix

Graph Laplacian

$$\boldsymbol{L} = \boldsymbol{G} - \boldsymbol{W}$$

where $g_{ii} = \sum_{j \neq i} w_{ij}$





Sample

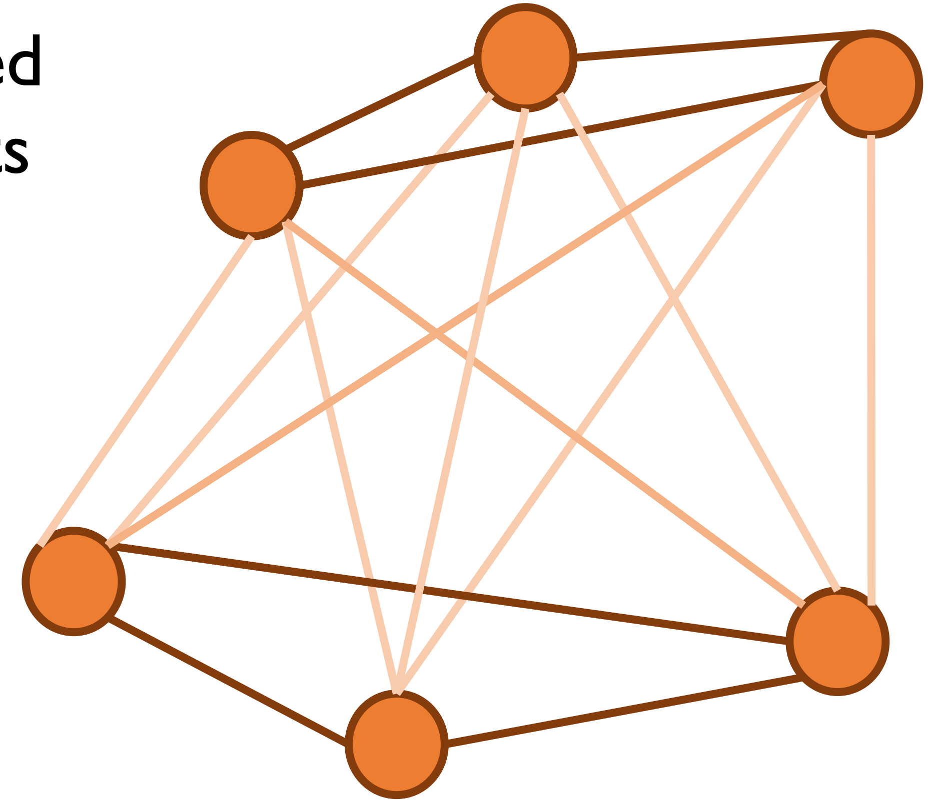
Edge weight = similarity

Spectral Clustering

Eigenvectors of $\boldsymbol{L}$ corresponding to smallest eigenvalue $\approx$ # Connected components

New features

Apply clustering (e.g. K-means)



Sample

Edge weight = similarity

Spectral Clustering

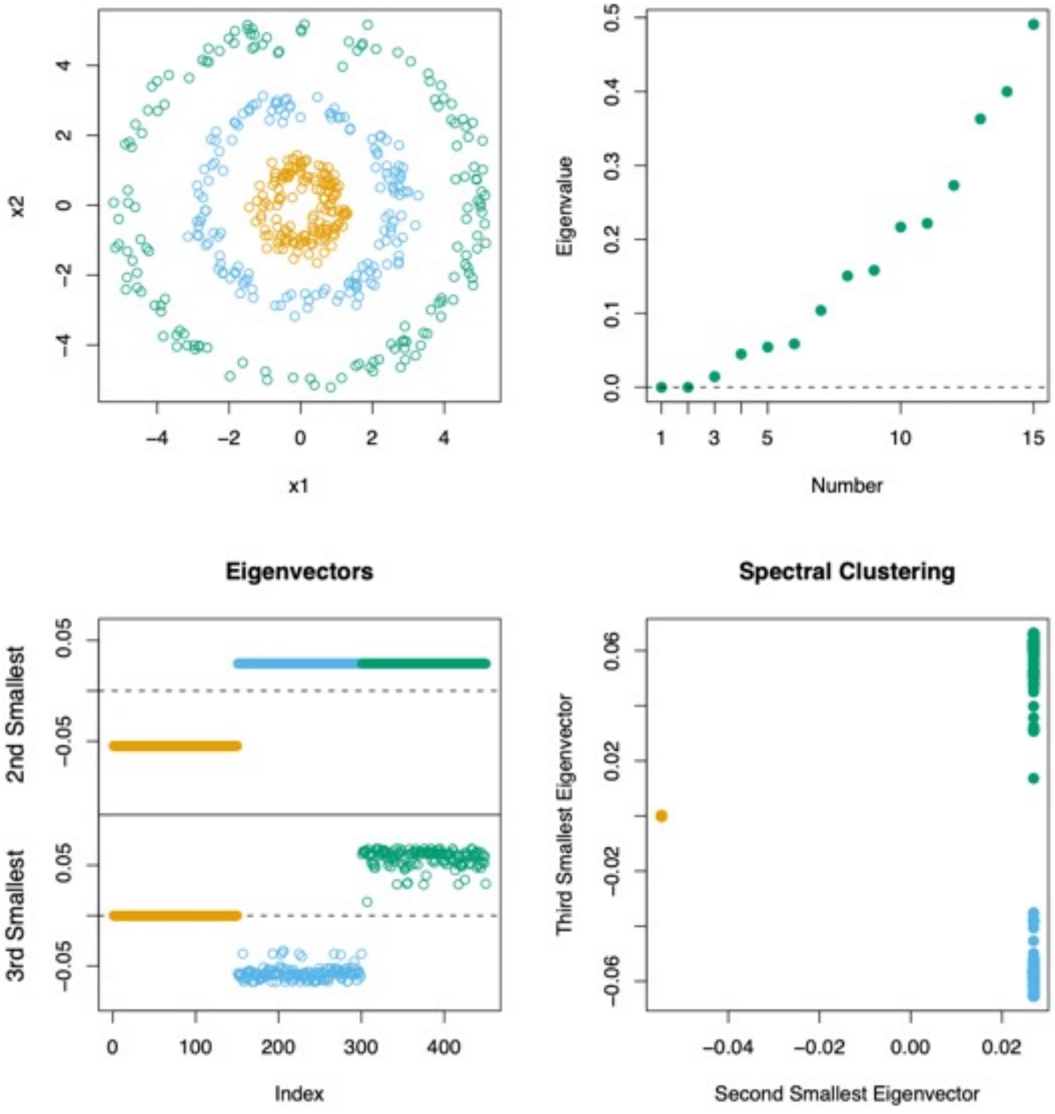
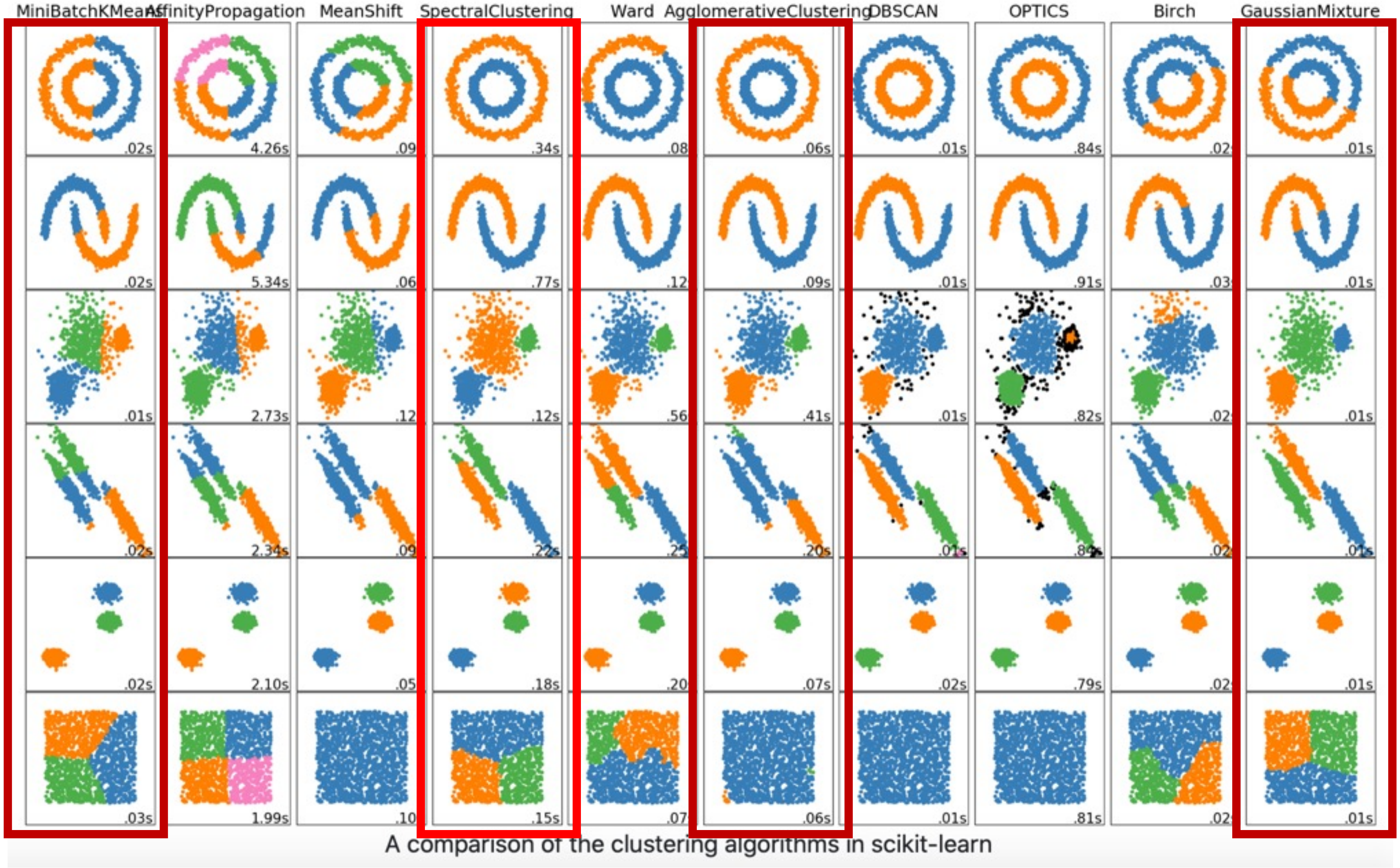
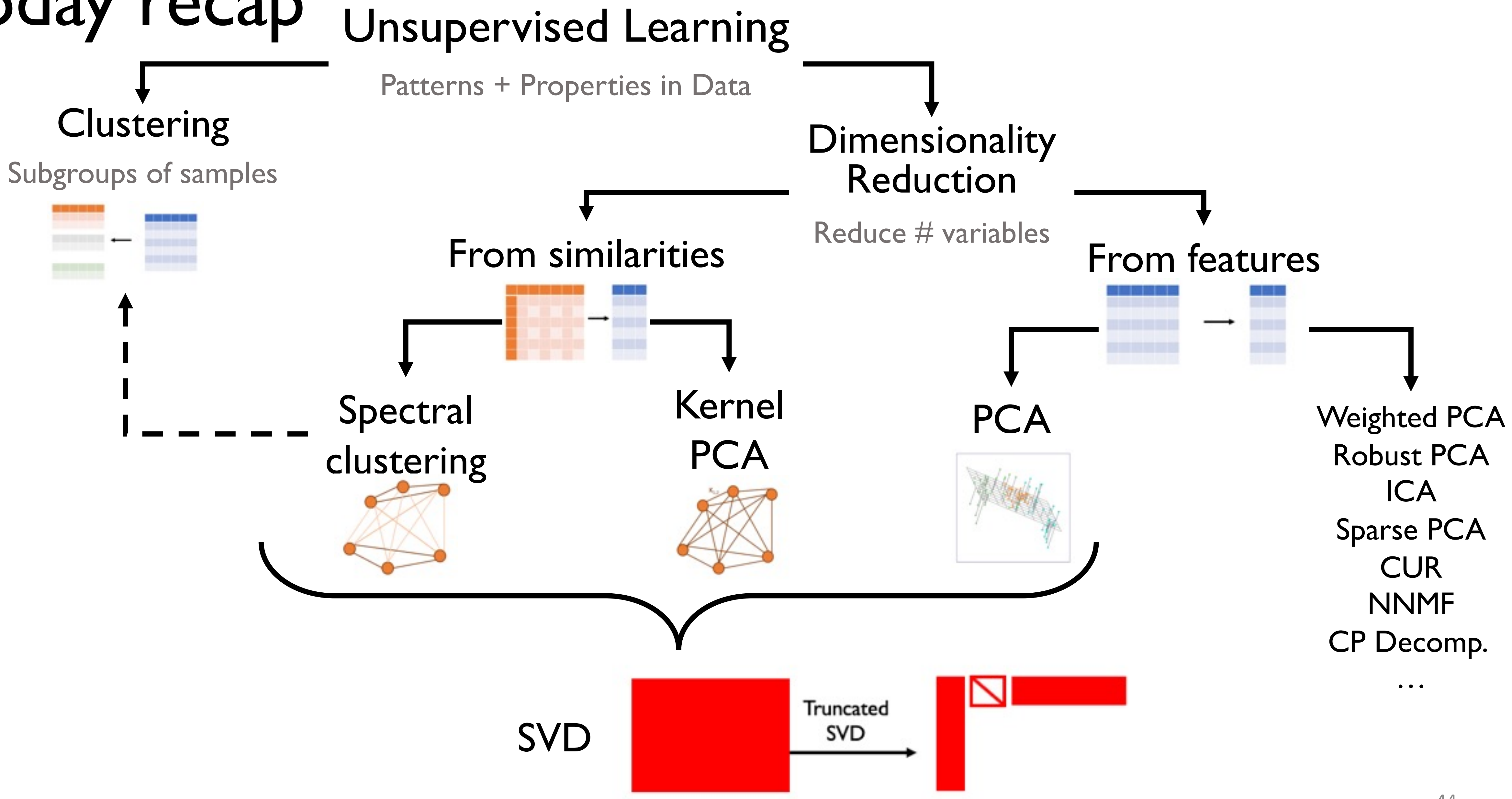


Figure 14.29, ESL (2nd edition 2009)



Code Available https://scikit-learn.org/stable/modules/clustering.html

Today recap


Unsupervised Learning
Patterns + Properties in Data
Clustering
Subgroups of samples
Dimensionality Reduction
Reduce # variables
From similarities
From features
Spectral clustering
Kernel PCA
PCA
Weighted PCA
Robust PCA
ICA
Sparse PCA
CUR
NNMF
CP Decomp.
...
SVD
Truncated SVD

Further reading

- What about categorical variables?
- What about DR for visualization?

Ten quick tips for effective dimensionality reduction
Lan Huong Nguyen, Susan Holmes.(2019)
https://doi.org/10.1371/journal.pcbi.1006907

Logistics

- Join slack channel cme250202.slack.com
- Office hours tomorrow Noon to 1pm (link on Canvas)
- Part I project deadline: April 26.