Lecture 14

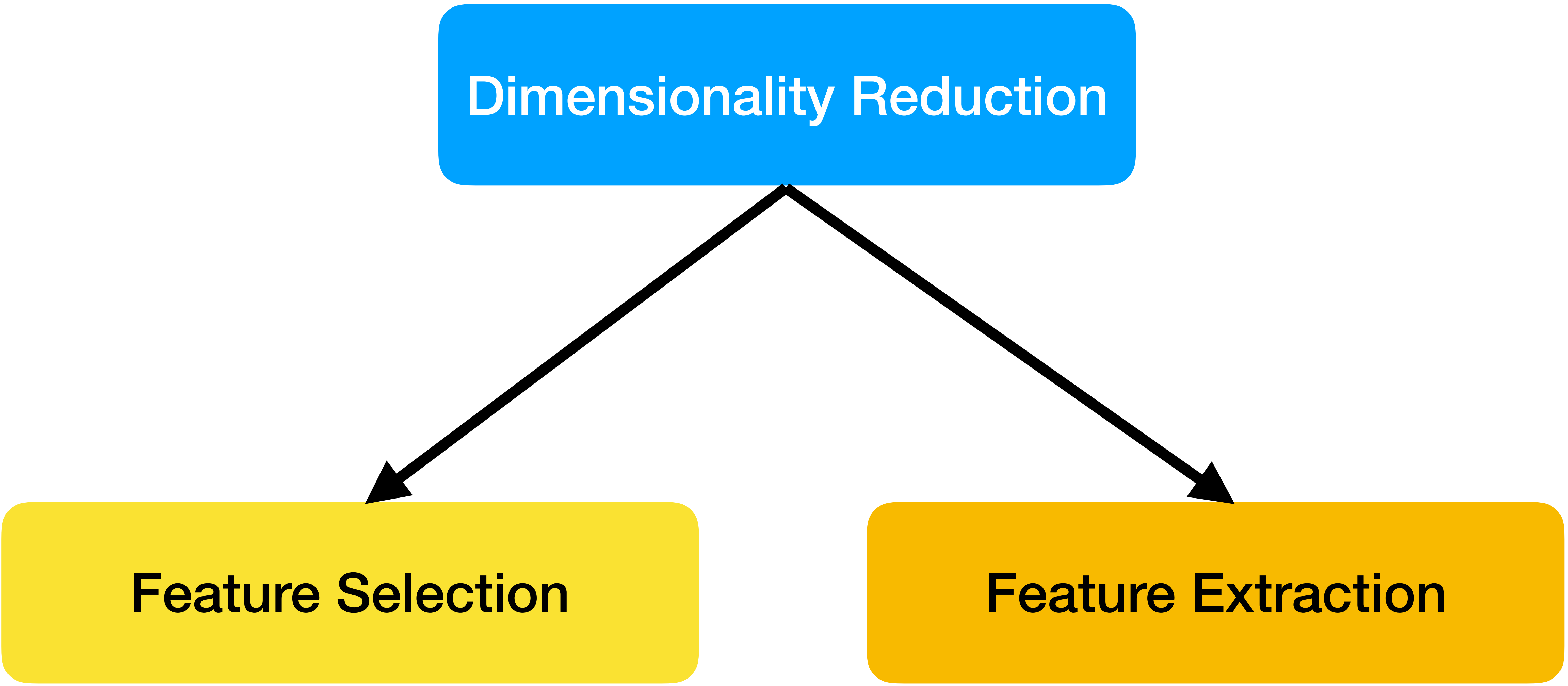
Dimensionality Reduction II: Feature Extraction

[very] short version

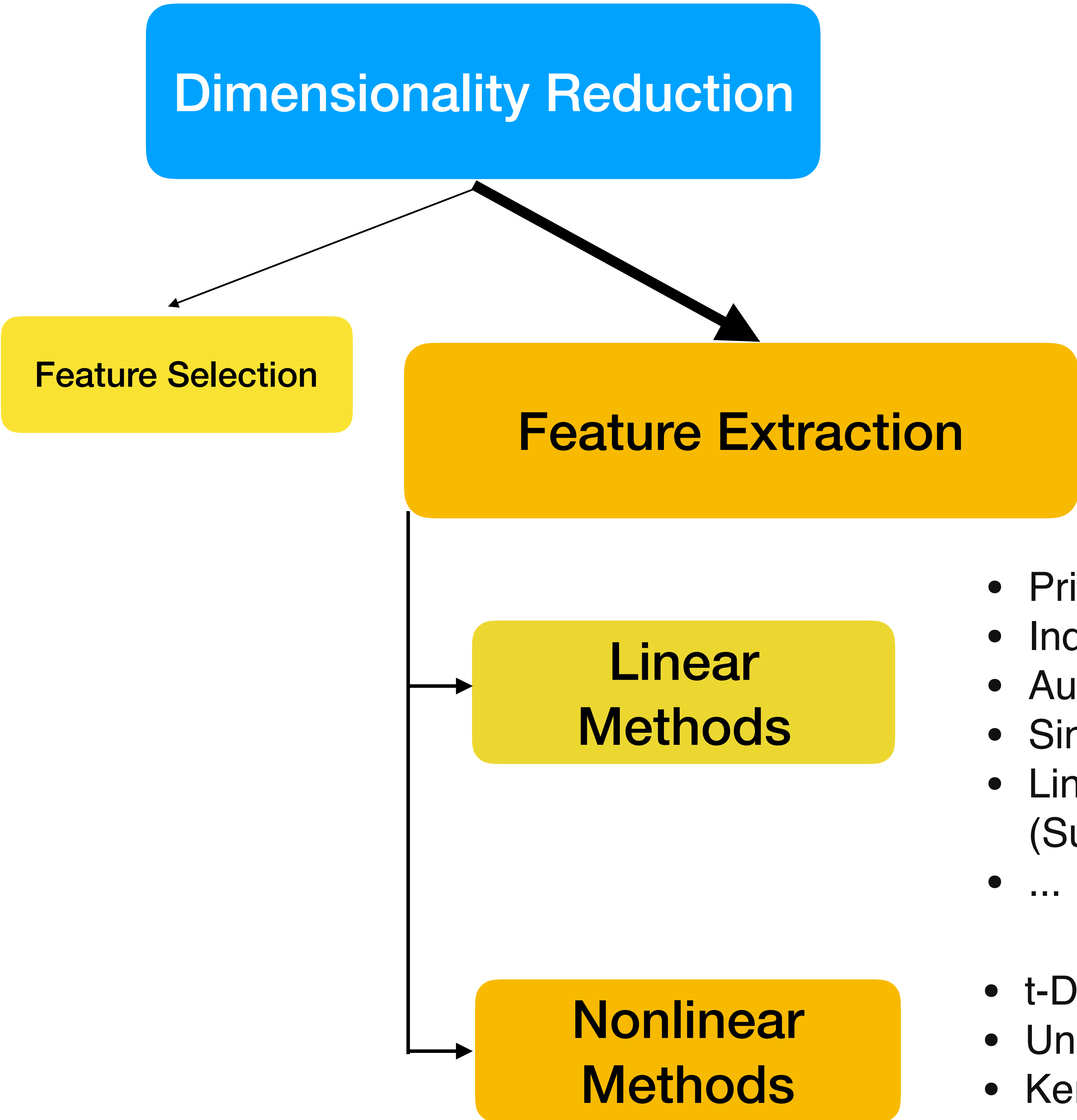
STAT 479: Machine Learning, Fall 2018

Sebastian Raschka

http://stat.wisc.edu/~sraschka/teaching/stat479-fs2018/


Dimensionality Reduction
Feature Selection
Feature Extraction

Today



- Principal Component Analysis (PCA)
- Independent Component Analysis (ICA)
- Autoencoders (linear act. func.)
- Singular Vector Decomposition (SVD)
- Linear Discriminant Analysis (LDA) (Supervised)
- ...

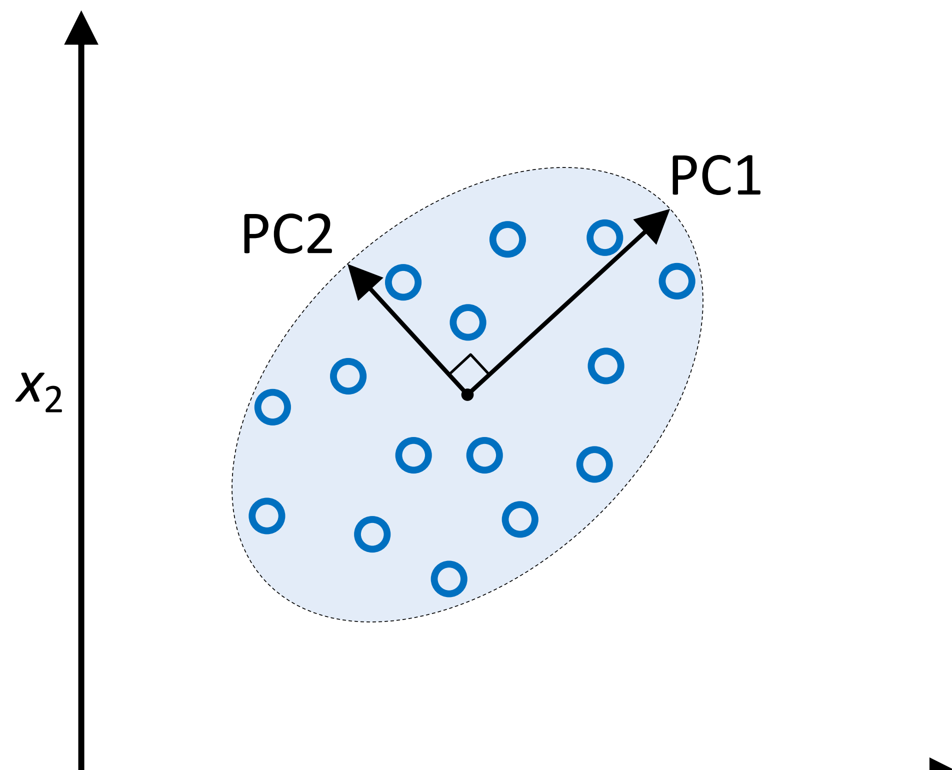
- t-Distr. Stochastic Neigh. Emb. (t-SNE)
- Uniform Manifold Approx. & Proj. (UMAP)
- Kernel PCA
- Spectral Clustering
- Autoencoders (non-linear act. func.)
- ...

Goals of Dimensionality Reduction

- Reduce Curse of Dimensionality problems
- Increase storage and computational efficiency
- Visualize Data in 2D or 3D

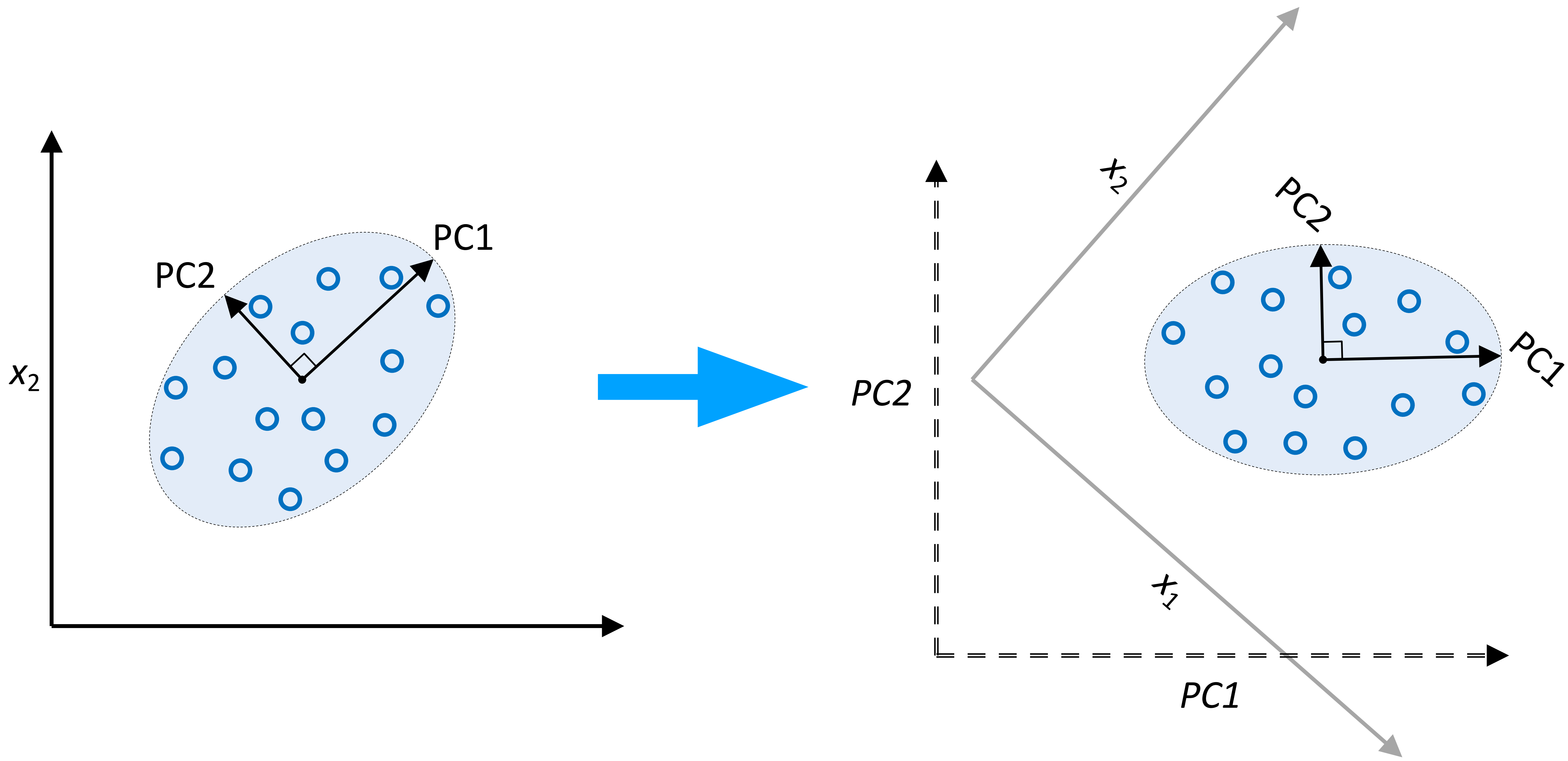
Principal Component Analysis (PCA)

1) Find directions of maximum variance



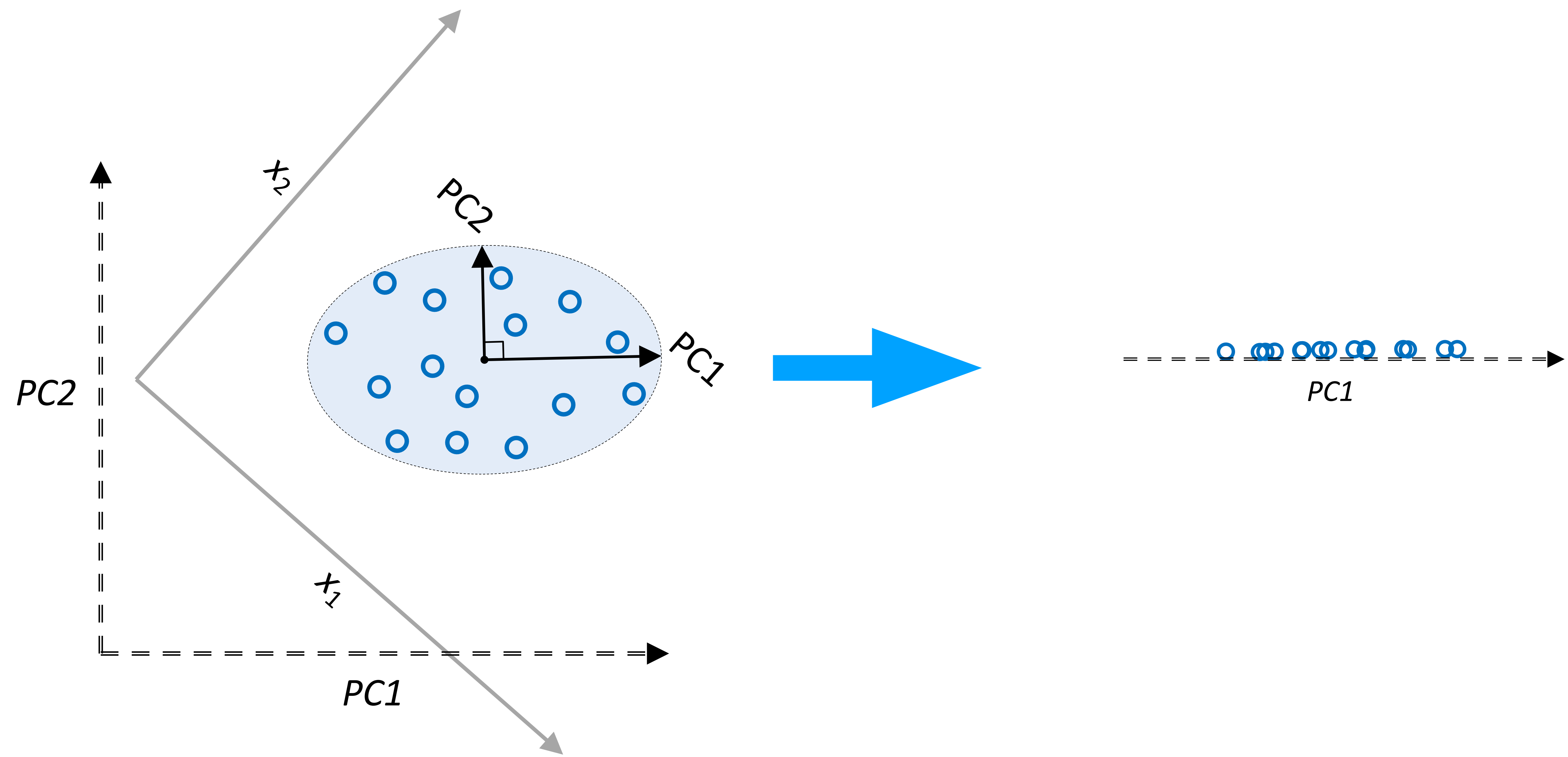
Principal Component Analysis (PCA)

2) Transform features onto directions of maximum variance



Principal Component Analysis (PCA)

3) Usually consider a subset of vectors of most variance (dimensionality reduction)



Principal Component Analysis (PCA) (in a nutshell)

Given design matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$

find vector α_i with maximum variance

repeat: find α_{i+1} with maximum variance uncorrelated with α_i

(repeat *k* times, where *k* is the desired number of dimensions; $k \leq m$)

Principal Component Analysis (PCA) (in a nutshell)

Given design matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$

find vector α_i with maximum variance

repeat: find α_{i+1} with maximum variance uncorrelated with α_i

(repeat *k* times, where *k* is the desired number of dimensions; $k \leq m$)

Principal Component Analysis (PCA) (in a nutshell)

Collect vectors α_i in a projection matrix $\mathbf{A} \in \mathbb{R}^{m \times k}$

(Sorted from highest to lowest associated eigenvalue)

Compute projected data points: $\mathbf{Z} = \mathbf{X}\mathbf{A}$

Principal Component Analysis (PCA)

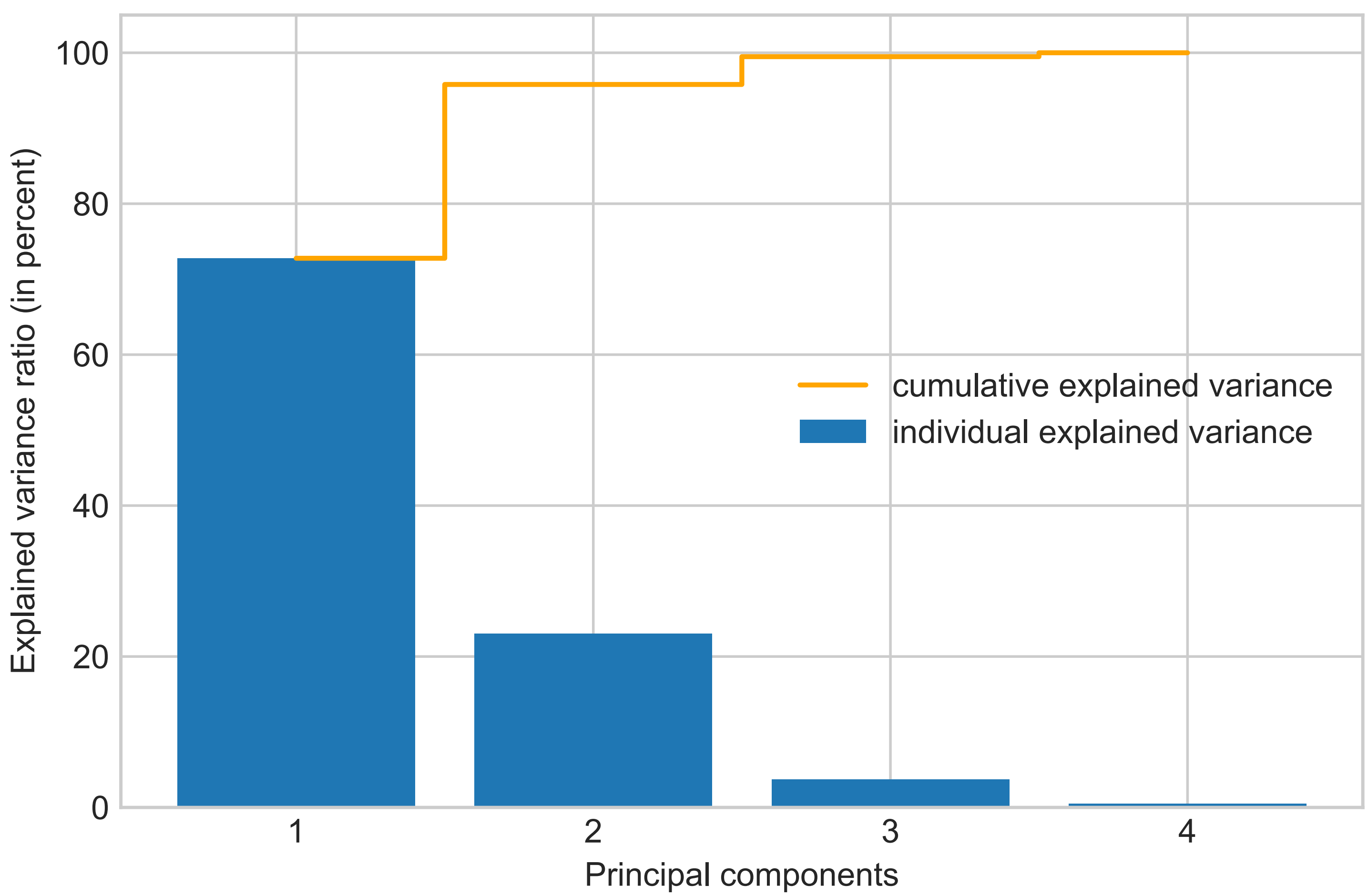
Two approaches to solve PCA (on standardized data):

1. Constrained maximization (e.g., Lagrange multipliers)

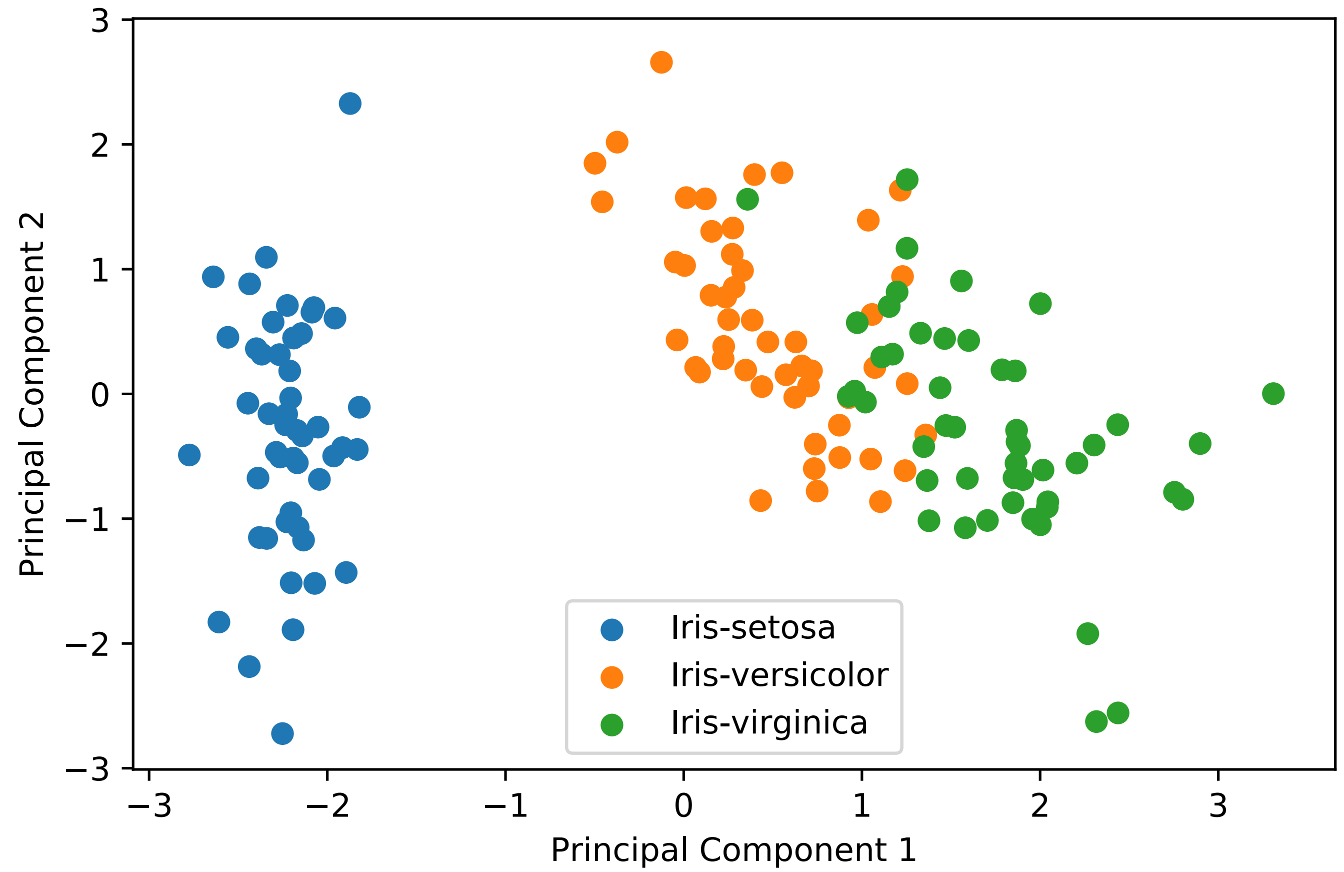
2. Eigen-decomposition of covariance matrix directly

Principal Component Analysis (PCA)

Usually useful to plot the explained variance (normalized eigenvalues)

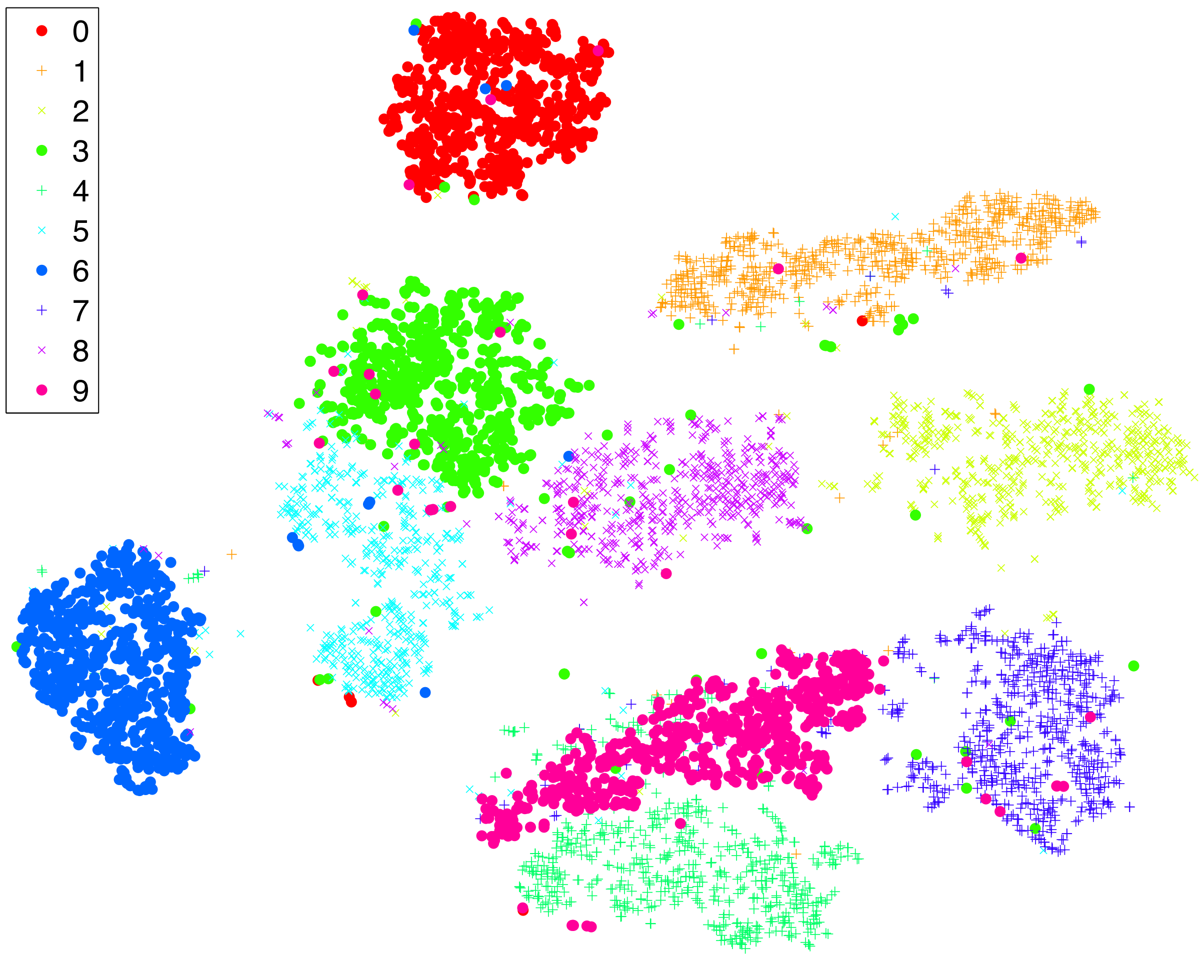


Principal Component Analysis (PCA)



Keep in mind that PCA is unsupervised!

t-Distributed Stochastic Neighbor Embedding (t-SNE)



(a) Visualization by t-SNE.

6000 images from MNIST

Note that MNIST has
28 x 28 = 784 dimensions

Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, *9*(Nov), 2579-2605.

t-Distributed Stochastic Neighbor Embedding (t-SNE)

Idea: Map points near on a manifold to a near position in low-dimensional space

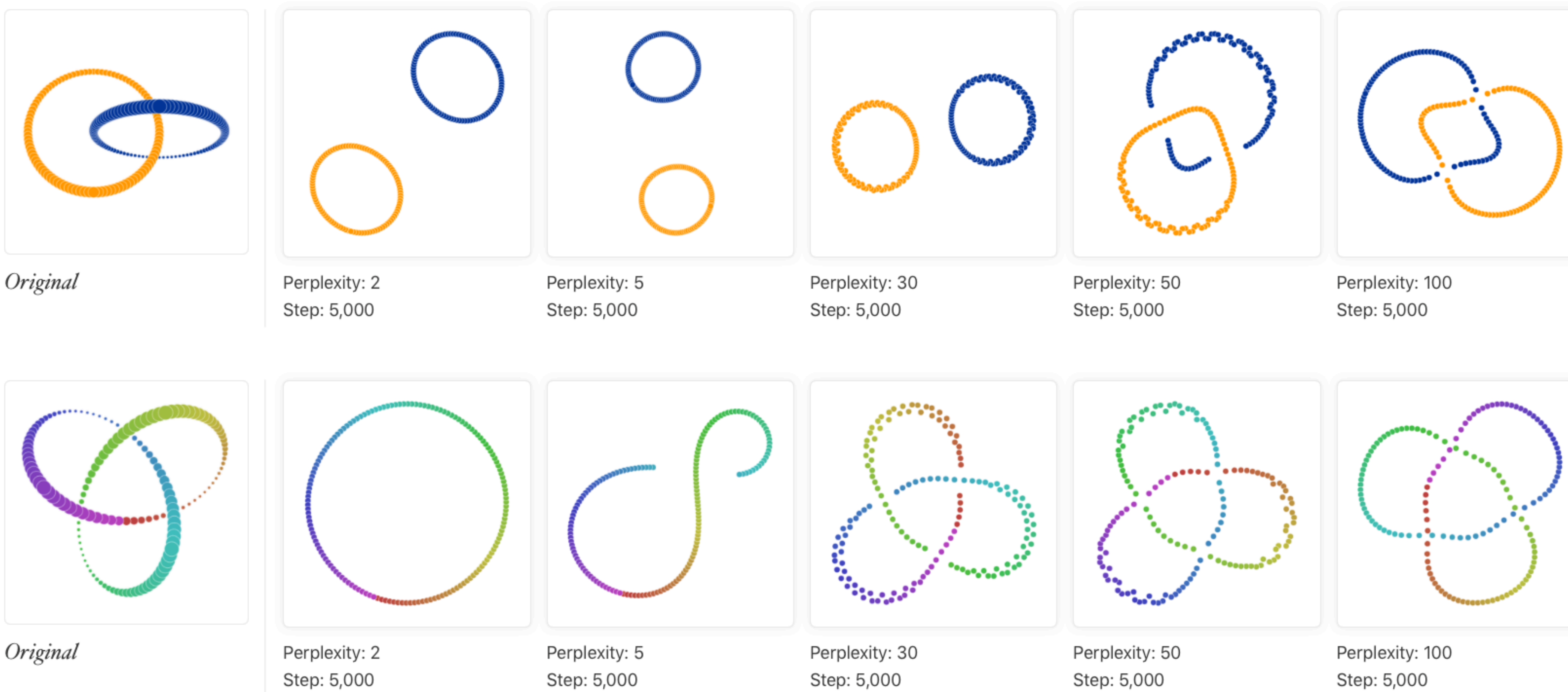
1. Measure euclidean distance in **high** dim & convert to probability of picking a point as a neighbor
   (similarity is proportional to probability); use **Gaussian distribution** for density of each point
2. Same as 1. in **low** dimensionality but with **t distribution** (has heavier tails)
3. Minimize the difference of the conditional probabilities (KL-divergence)

t-Distributed Stochastic Neighbor Embedding (t-SNE)

Idea: Map points near on a manifold to a near position in low-dimensional space

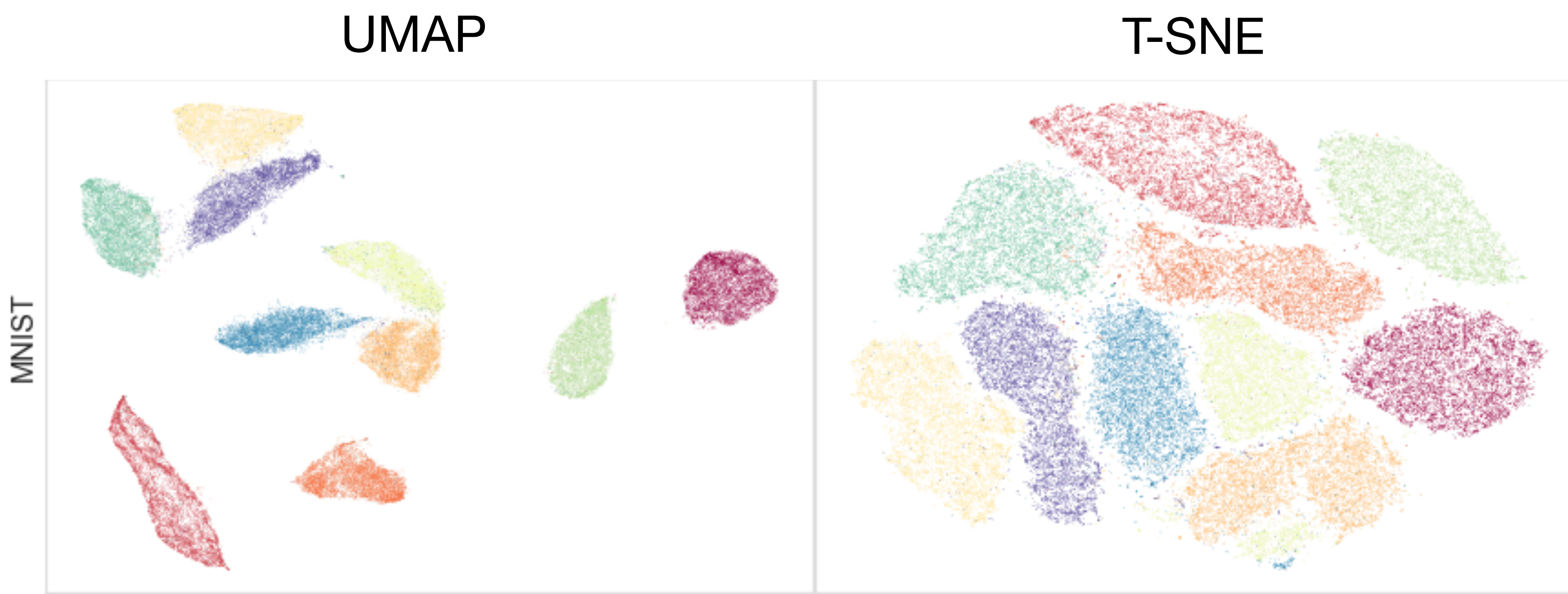
- Great for visualizing datasets in 2D
- Need to analyze multiple perplexity values (tuning parameter related to standard deviation of the Gaussian, to balance local and global attention)
- Not deterministic
- More hyperparameters (learning rate epsilon)

t-Distributed Stochastic Neighbor Embedding (t-SNE)



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

Uniform Manifold Approximation and Projection (UMAP)



McInnes, L., & Healy, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426.

Compared to t-SNE, UMAP seems to be

- faster
- deterministic
- better at preserving clusters

Reading Assignment

- Python Machine Learning, 2nd Edition.
  Chapter 5: Compressing Data via Dimensionality Reduction