Dimensionality Reduction

David Mareček

December 16, 2021

EUROPEAN UNION
European Structural and Investment Fund
Operational Programme Research, Development and Education

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

CC BY SA unless otherwise stated

Dimensionality Reduction

We often have:

- Big and high-dimensional data
- A lot of features
- Many of them may be redundant / correlated / linearly dependent

Dimensionality reduction algorithms map high-dimensional data to a lower dimension while preserving structure.

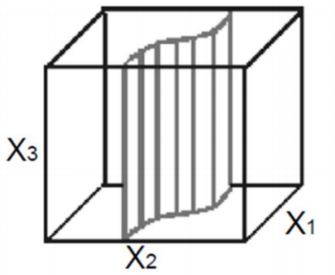
Motivation:

- Visualization
- More efficient use of resources (e.g., time, memory, communication)
- Statistical: fewer dimensions $\rightarrow$ better generalization (curse of dimenzionality)
- Noise removal (improving data quality)

Dimensionality Reduction

Feature selection:

- select a subset of features

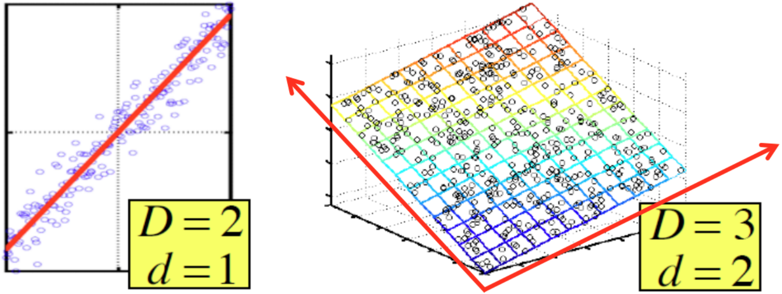


- X_3 is almost irrelevant

Dimensionality Reduction

Feature extraction:

- more general
- not limited to the original features
- Assumption: data (approximately) lies on a lower dimensional space



t-SNE

t-SNE

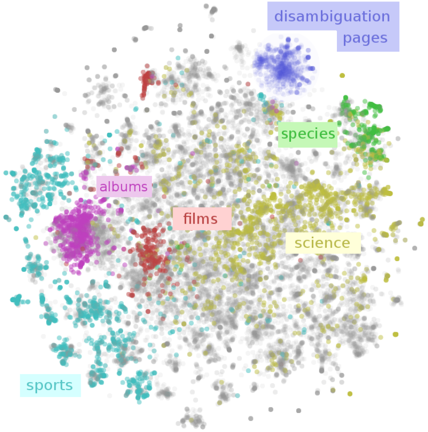
t-distributed Stochastic Neighbor Embedding
developed by Laurens van der Maaten and Geoffrey Hinton in 2008

- a non-linear dimensionality reduction technique
- for visualization of high dimensional data in 2D (3D)
- it keeps the very similar data points close together in lower-dimensional space
- preserves the local structure of the data, not the global structure
- preserves well-separated clusters

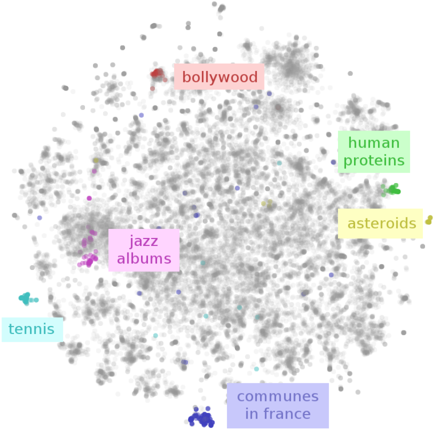
In this part, I am using illustrations by Kemal Erdem.

See `https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a`

Large Clusters



Small Clusters

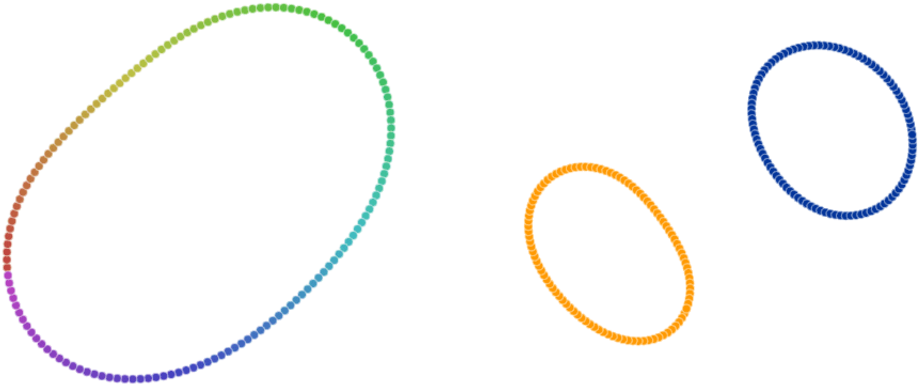


How you would preserve the local structure in 2D?

Original datasets in 3D

How you would preserve the local structure in 2D?

Their t-SNE visualization in 2D



Similarity of two points

Create a probability distribution that represents similarities between neighbors

For each pair of data points (i, j), compute

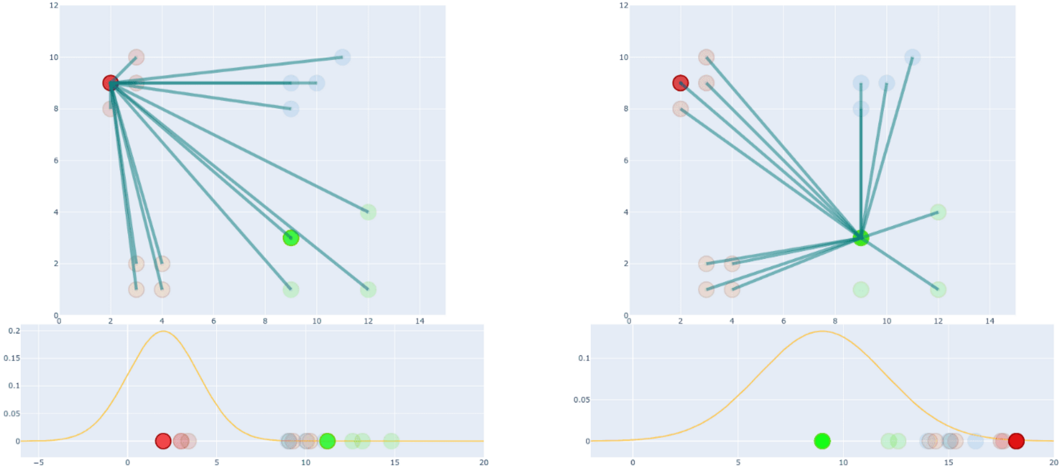
$$p_{i|j} = \frac{exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-||x_i - x_k||^2/2\sigma_i^2)},$$

The similarity of datapoint x_j to datapoint x_i is the conditional probability $p_{j|i}$, that x_i would pick x_j as its neighbor.

The two asymetric distributions are then joined into a symetric one:

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2N}$$

Similarity of two points



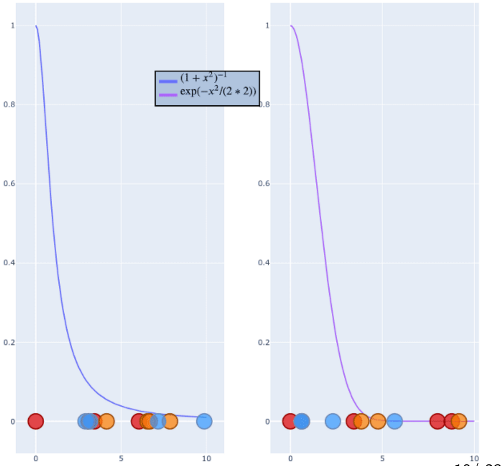
Similarity of two points in the low-dimensional space

As similarity measure in the target low-dimensional space, we will use Student t-distribution instead of the Gaussian

$$q_{i|j} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}}$$

Student t-distribution "falls" more quickly and has longer tail than the Gaussian distribution

Therefore, we will not get similar points squashed into a single point.



Gradient descent

t-SNE starts with all the points y_i randomly distributed in the target 2D (or 3D) space.

It uses Gradient descent optimization using the Kullback-Leibler divergence between p_{ij} and q_{ij} as a cost function.

$$C = D_{KL}(P||Q) = \sum_{x \in X} P(x) log \left(\frac{P(x)}{Qx} \right)$$

In each step, a gradient is calculated for each point and describes how "strongly" it should be pulled and what the direction it should choose.

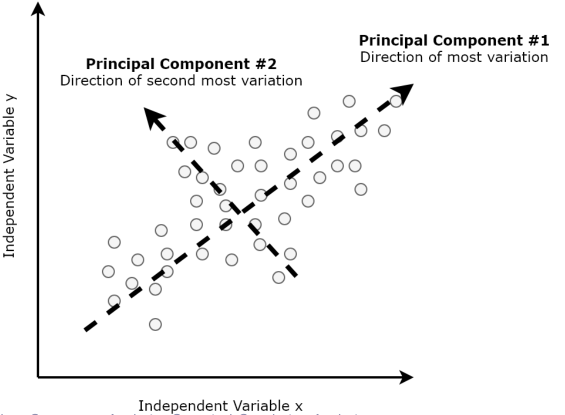
Demo: `projector.tensorflow.org`

Principal Component Analysis

Principal Component Analysis

Principal components (PC) are orthogonal directions that capture most of the variance in the data.

- 1st PC – direction of the greatest variability in data
- 2nd PC – next orthogonal (uncorrelated) direction of greatest variability



Principal Component Analysis

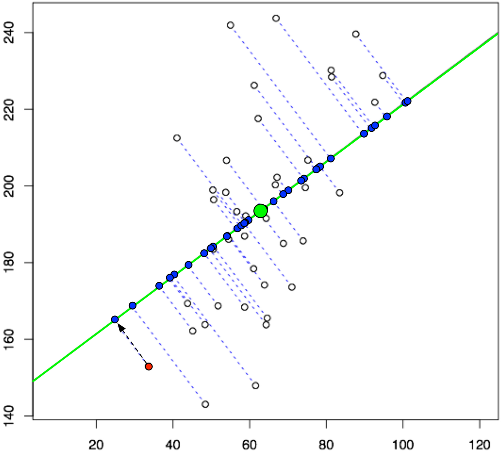
Given the centered data $[x_1, x_2, \dots, x_n]$, the first principal vector is:

$$w_1 = \arg\max_w \frac{1}{m} \sum_{i=1}^{m} (w^T x_i)^2 = \arg\max_w w^T X X^T w, \quad w^T w = 1$$

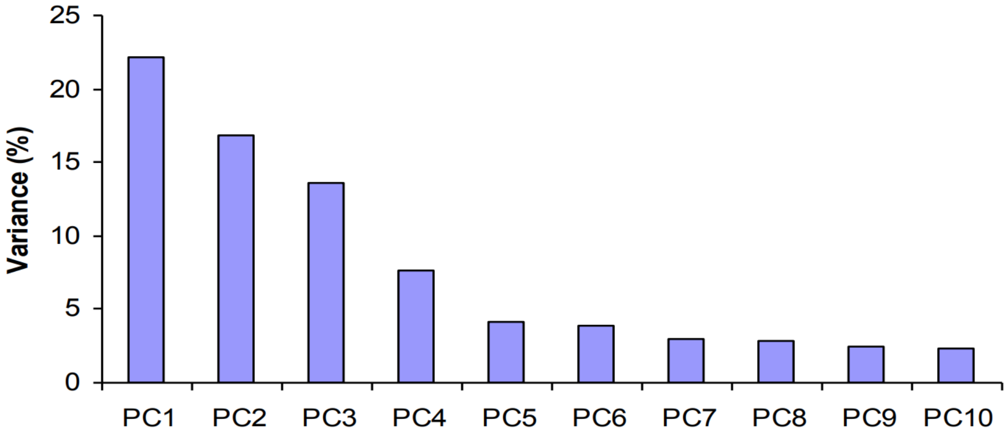
We maximize the variance of projection of x to w.
$\rightarrow$ we maximize the covariance between x and w (the dataset is centered)

For computing the k-th principal vector, we first remove all variability of the previous $k-1$ PC directions and find the next direction of the greatest variability.

Principal Component Analysis



Principal Component Analysis



Principal Component Analysis

1. Standardize the original high-dimensional dataset.
2. Take the standardized data and compute a covariance matrix A that provides a means to measure how all our features relate to each other.

$$A_{xy} = cov(x, y) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})$$

3. Find its eigenvectors w and corresponding eigenvalues λ. Eigenvectors represent the principal components and provide a means to understand the direction of the data. Corresponding eigenvalues represent how much variance there is in the data in that direction.

$$Aw = \lambda w$$

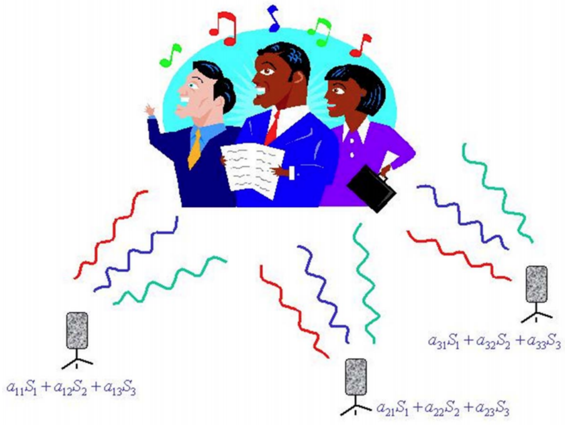
Principal Component Analysis

4. The eigenvectors are then sorted in descending order based on their corresponding eigenvalues, after which the top k eigenvectors are selected representing the most important representations found in the data.
5. A new matrix is then constructed with these k eigenvectors, thereby reducing the original n-dimensional dataset into reduced k dimensions.

Independent Component Analysis

Independent Component Analysis

- The classical "cocktail party" problem
- Separate the mixed signal into sources
- Assumption: different sources are independent



Independent Component Analysis

Let $[v_1, v_2, v_3, \ldots, v_d]$ denote the projection directions of independent components:

ICA: find these directions such that data projected onto these directions have maximum statistical independence

How to actually maximize independence?

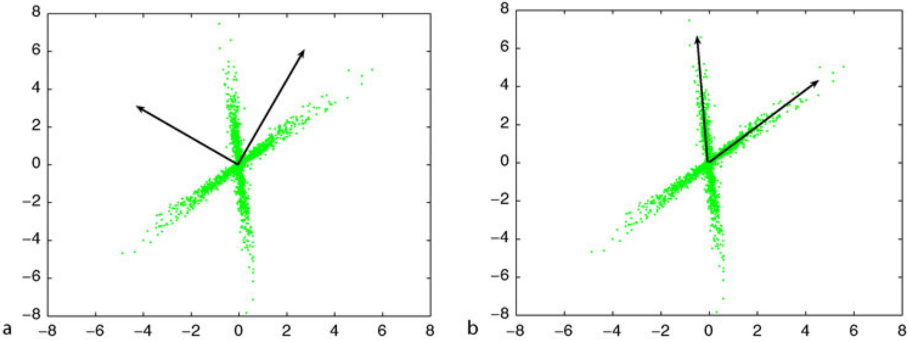
- Minimize the mutual information
- Maximize the non-Gaussianity

PCA versus ICA

Both PCA and ICA reduce dimensions.

Differences:

- PCA with a Gaussian model, ICA with non-Gaussian model
- PCA vectors are orthogonal, ICA vectors are not orthogonal



ICA mathematical approach

$$x_i = a_{i1}s_1 + a_{i2}s_2 + \cdots + a_{in}s_n, \forall i = 1, \ldots, n$$

Giving: observation "x"

Find:

- Original independent components s
- Associated linear combination a_{ij}

Canonical Correlation Analysis

Canonical Correlation Analysis

Now consider two sets of variables x and y

- x is a vector of p variables
- y is a vector of q variables
- Basically, two feature spaces

How to find the connection between two set of variables (or two feature spaces)?

- CCA: find a projection direction u in the space of x, and a projection direction v in the space of y, so that projected data onto u and v has max correlation
- Note: CCA simultaneously finds dimension reduction for two feature spaces

Canonical Correlation Analysis

CCA formulation:

$$\arg\max_{u,v} \frac{u^T X^T Y v}{\sqrt{(u^T X^T X u)(v^T Y^T Y v)}},$$

- X is n by p: n samples in p-dimensional space
- Y is n by q: n samples in q-dimensional space
- The n samples are paired in X and Y

How to solve? ... kind of complicated ...