Data Science

Session 4 - Dimensionality reduction

hadrien.salem@centralelille.fr

introduction-to-data-science

Introduction

What did we do last time?

The CRISP-DM method

Cross-**I**ndustry **S**tandard **P**rocess for **D**ata **M**ining

- ➔ Published in 1999
- ➔ Common in the industry
- ➔ Still relevant today

The CRISP-DM method to carry out data-driven projects

(Image source: Wikipedia)

Course outline

Data science course

Session 1: Understanding data

Session 2: Clean code & Git

Session 3: Preparing data - Cleaning & Missingness

Session 4: Preparing data - Dimensionality reduction

Session 5: Preparing data - Data imbalance

Machine learning course

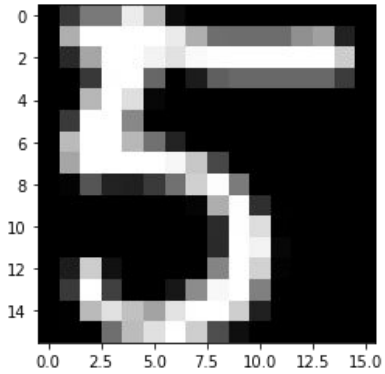
What is dimensionality reduction?

Example: the MNIST dataset

Example: the MNIST dataset

Modified National Institute of Standards and Technology database

- Database of hand-written digits
- Each digit is represented by the greyscale value of each pixel (between 0 and 255)
- Originally, the images are 28x28, but they will be 16x16 in today's practical

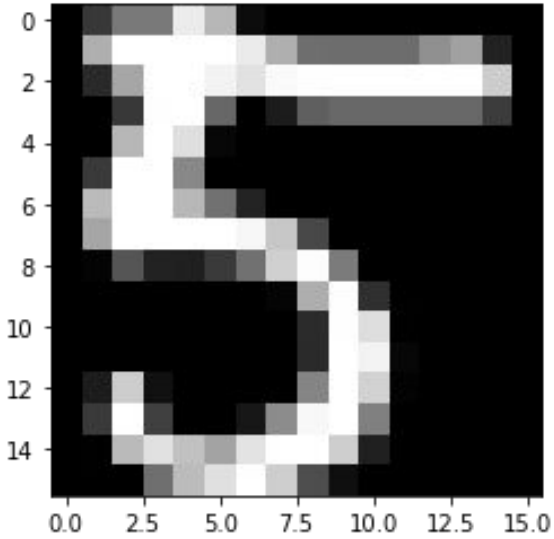


Example: the MNIST dataset

This example shows how the number of features can easily become very large.

Some datasets contain hundreds, or even thousands of features!

Imagine trying to train an algorithm with 4k photographs!



16 x 16 = 256 pixels

⇒ 256 dimensions
⇒ 256 features (columns)

Why would we want to perform dimensionality reduction?

Why would we want to perform dimensionality reduction?

In many cases, having too many features is a disadvantage.

Dimensionality reduction allows for:

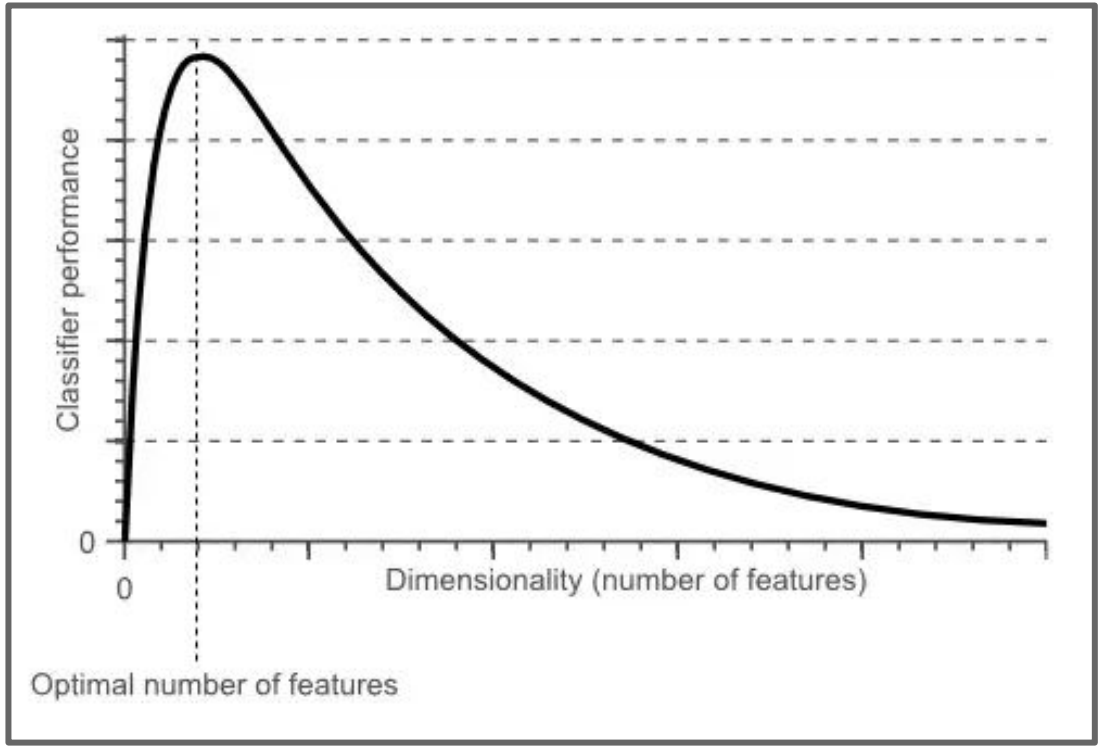
- The reduction of computational charge (i.e. reduction of computing time)
- The reduction of noise in the dataset
- The visualization of data in 2D / 3D
- The alleviation of the curse of dimensionality

⇒ Dimensionality reduction can help improve the performance of machine learning algorithms

Increasing the number of features can help up to a certain point.

However, when features are too numerous, it becomes difficult for algorithms to discernate patterns.

The effects of “high dimensionality” can appear with only 5 features!



The Hughes phenomenon, or curse of dimensionality

Image source

How to perform dimensionality reduction
Method #1 : Feature Selection

How do you select the most important features?

How do you select the most important features?

There are many methods for feature selection

- Selection from expert knowledge (although it can be counter-productive)
- Deletion of low-variance variable
- Determining feature importance with a baseline machine learning model (e.g. Random Forests)
- Iterative choice using a machine learning model
  - Forward method: adding variables
  - Backward method: removing variables
  - Mixt method: doing both
  - Choosing randomly
- Scikit-learn offers many methods for feature selection

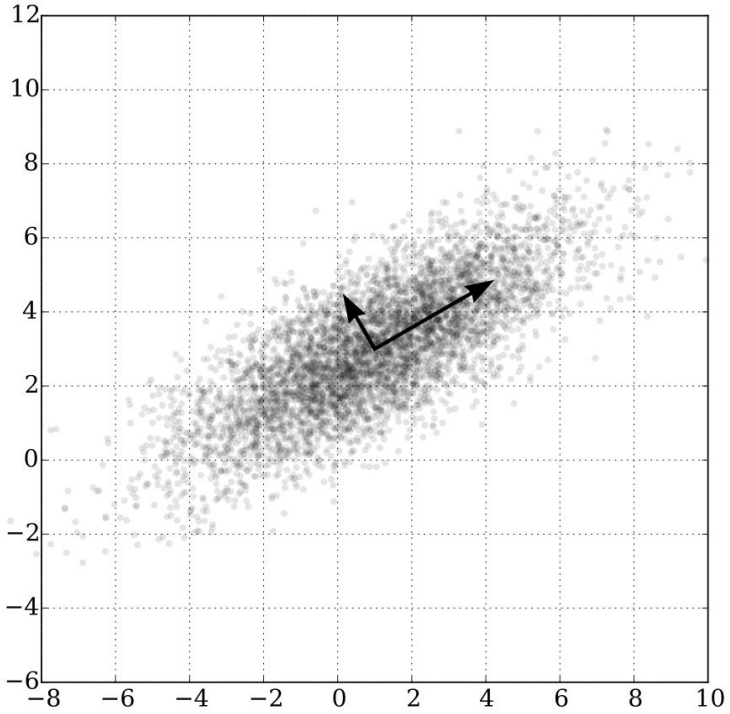
How to perform dimensionality reduction
Method #2 : Feature extraction

Feature extraction

Using existing features to create new ones

The columns created in this way should be more significative than the ones initially in the dataset

It can be simple linear combinations, or more complex functions



Intuitive principle

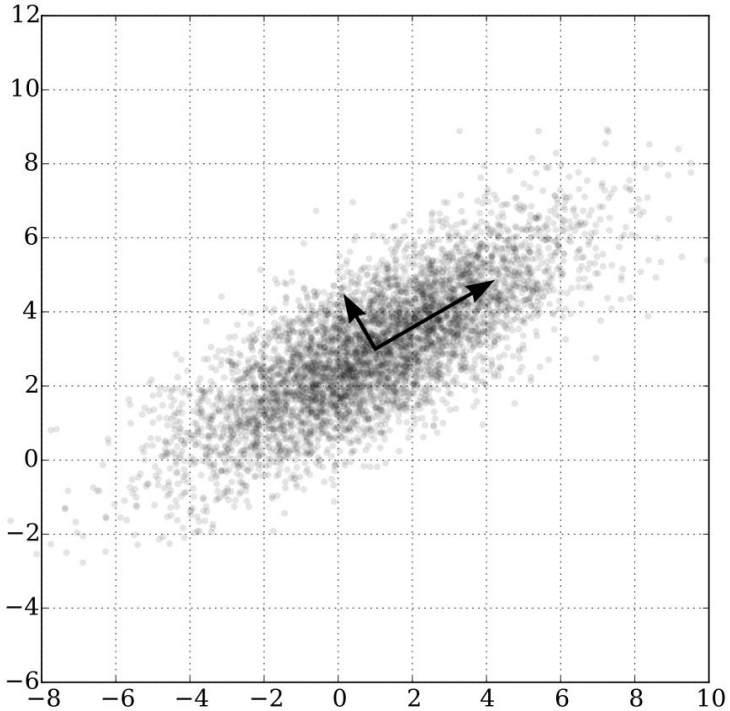
Finding the directions with the highest variance

Principle

We aim to find a **new basis** such that the variance of each projected component is maximized.

In other words, **PCA is not a dimension reduction method per se**. However, in constructing the new basis, we ensure that **the first vectors are the directions of the highest variance**.

To use PCA as a dimension reduction method, it is sufficient to **select the first N vectors** of the new basis.



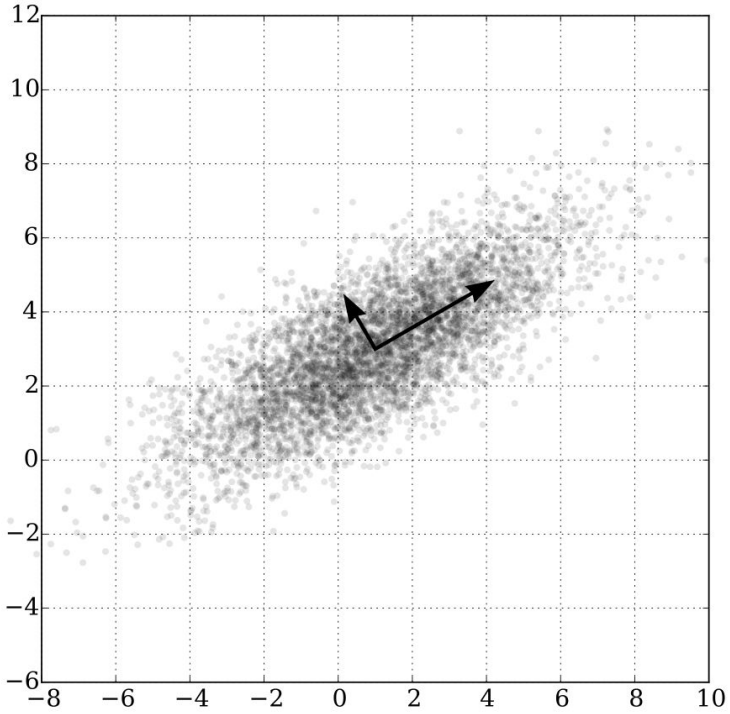
Formally

We are looking for a change of basis matrix that minimizes the approximation error.

$$U = \underset{U^T U = 1}{\arg\min} \sum_{n=1}^{N} ||x_n - \underbrace{U U^T x_n}_{x_{\text{approx}}}||^2$$

It can be shown that this is equivalent to finding the **eigenvectors** of the covariance matrix.

$$\text{Cov}(X, Y) \equiv \text{E}[(X - \text{E}[X])\,(Y - \text{E}[Y])]$$



Using PCA

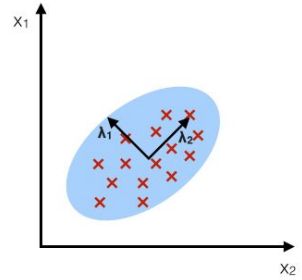
→ **Standardization of data is necessary**

→ Some information is lost despite keeping the most “important” dimensions

→ Features cannot be interpreted anymore

→ In a classification problem, PCA does not take interclass variance into account. Discriminant analysis takes both intra- and interclass variance into account.

PCA:
component axes that maximize the variance



LDA:
maximizing the component axes for class-separation

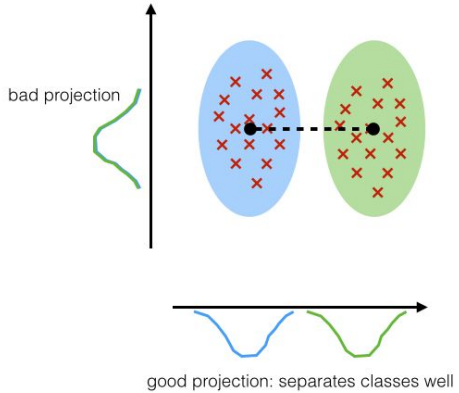


Image source

Practical work

Get the latest version of the notebook from GitHub

Don’t forget to upload your work!

Debrief

Debrief

What did we learn today?

What could we have done better?

What are we doing next time?

Data Science

Session 4 - Dimensionality reduction

hadrien.salem@centralelille.fr

introduction-to-data-science