

# Machine learning in python

## My notes

Maciej jarosz

2019-11-23

# 1 Theoretical bullshit

## 1.1 Introduction

We can divide types of machine learning systems in to few categories because of:

- type of supervision in training process
  - supervised
  - unsupervised
  - semisupervised
  - reinforcement learning
- possibility of real-time learning
  - incremental/online learning
  - batch
- the type of working
  - simple comparition of new data points with already known points
  - detection of patterns in learning data and creating production model

## 1.2 Supervised learning

In supervised learning data transmitted to algorithm contain attached problem solutions, the so called labels.

### 1.2.1 Examples

Typical task of supervised learning is classification. Spam filter is great example. In this case it is trained by a huge amount of sample messages belonging to given class (spam or ham), thank to which it need to be able to classify a new messages.

Another typical task of supervised learning is prediction of the target numerical value. For example like car price using a features named predictors. This type of the task is called regression.

### 1.2.2 Terminology

In the terminology of ML **attribute** is type of data (przebieg samochodu), while **feature** depend on context, can have many of different meaning. Usually when we talk about feature, we think about attribute with it value.

### 1.2.3 Types

This is the list of the most important supervised learning algorithms:

- K-nearest neighbors - k-nn method
- linear method
- logistic regression
- Support Vector Machine - **SVM**
- decision trees and random forests
- neural networks

## 1.3 Unsupervised learning

The main task is to assign data to categories which are containing similar data, based on their features

### 1.3.1 Types

This is the list of the most important unsupervised learning algorithms, because of:

- Data clustering
  - k-means
  - hierarchical cluster analysis - **HCA**
  - expectation-maximization algorithm - **EM**
- Visualization and dimensionality reduction
  - principal component analysis - **PCA**
  - Kernel PCA
  - locally linear embedding - **LLE**
  - t-distributed stochastic neighbor embedding
- learning by using associative rules
  - Apriori algorithm
  - Eclat algorithm

### 1.3.2 Example

Let's assume, You have big amount of data about users visiting your website. You can use clustering to try define groups of similar users. At any moment you can check into which group any user falls into. Every connection between users are made without you interfere. Why? What can give to you? For example you can find out that 40% of your comics are read in the evenings and 20% of users are passionate in science and they are visiting your website only in weekends.

If you will use **Hierarchical cluster analysis** you can divide these groups into smaller subgroups. In this case for example, you can easier decide what kind of posts entries for individual groups.

Another example are **Visualization algorithms**. You can upload huge amount of different data which will be presented in 2D or 3D chart. These algorithms try to save primary structure of data, so you can easier analyze this data and maybe discover some unforeseen patterns.

**Dimensionality reduction** algorithms try to make your data more simply but without loss excessive number of information. We can get this result by merging few attributes in one based on their high correlation level. For example

we can correlate the age of the car with its mileage. This process is called **feature extractions**

Another important task is **anomaly detection**. Good example is to discover some unusual operation on your debit card. Long story short, we can use it to detect some anomaly in our dataset before we end it do our model.

Another one, commonly used algorithm is **association rule learning**. Its task is to analyze very huge amount of data and find some interesting dependencies between attributes. For example by using this method we can discover that people who are when buying ketchup and chips are also often than the other buying steaks.

## 1.4 Semisupervised learning

Some of algorithms are able to process partially marked learning data, mostly composed from unmarked and only a little percent of marked data. Most Semisupervisedlearning algorithm are consisted of supervised and unsupervised learning methods. Example in here can be **deep belief networks** which are arranged in layers unsupervised elements, called restricted Boltzmann machines. These machines are taught sequentially in unsupervised way, a next whole system are tuned by supervised learning techniques.

### 1.4.1 Example

Good example can be google photos. We use unsupervised learning to categorize our photos, then tag some people on our photo and then use supervised learning to give our system ability to recognize every people in the picture.

## 1.5 Reinforcement learning

This topic is very interesting from my perspective. Learning system, named in this context **the agent** may observe environment, make some actions and also receive rewards or panalties(rewards with negative value). This policy forces him to elaborate the best strategy to get the biggest reward. This policy enforce type of action that agent have to do.

### 1.5.1 Example

The great example of this method are selfdriving cars, which have to analyse data whole time and take some action based on collected informations. Also AlphaGo is an example of this method. Program which were learning how to play in Go, developed strategy that gave it a victory over a human master of Go.

## 1.6 Batch learning

Using this method enforce on developer to deliver all data to system, because it have to use all data to learn. This usually mean that application will use a significant amount of time and resources, and this is the reason is always made offline. System is firstly teaching and in the next step implemented to production cycle and never more trained. It is using only resources that it already have. This phenomenon is sometimes called offline learning.

It means that if you want to use some new data, you have to train all your system from the beggining with new and old data. Then turn off old system and turn on the new one. Fortunately, this process can be automated, so this system is able to still learning based on new data. This is very simply but also very efficient. Drawback of this algorithm is time that and resources of the machine that you have to sacrifice to make all the calculations. What does it means for developer or company? Money, money, money, bags of money, trains and airplains of money.



## 1.7 Online learning

In opposition to **batch learning**, system is train periodically by sequential data delivery. They can be single or take form of mini-batches, so small data sets. Every step is quick and don't cost much, so system is able to use new data every time when they appear.

One of the most important feature of online learning is **learning rate** which define the speed of system adjust to changing data. High learning rate means that our system will fast adjust to new data but also fast forgetting the old ones. On the other hand, low learning rate have big inertia factor which mean that system will learning slower but it will be more resistant to interference and some strange, uncorrelated data.

Big drawback of this system is tendency to decrease performance when it get some incorrect data, for example from destroyed sensor. Of course, you can minimalize risk of this situation but it forces monitoring of your system and stop analizing the data then some wrong situation happen. You can use for this some **anomaly detection algorithm**

**Out-of-core learning** is another type of online learning, used when data used to learn system are to big for out physical memory. In this case we are loading only part of data, teaching system, and then loading another part of this data and learning system with all infomation.

### 1.7.1 Examples

This method is great when system have to learning in real-time and take some fast decision, for example in autonomous racing cars in formula student turnament. it is also very useful when we have limited resources: i.e. we can delete old, already used data.

## 1.8 Learning from examples/from model

We can divide also our system in term of generalization: learning from examples and learning from model.

First method is called *instance-based learning*. In this case system is learning by heart and then he compare it with new samples by measure of probability. There is no any complicated math or something. It is just checking data by some features.

Second method is model based learning. It is based on create model from our data and use it to predict the value of new data. We can use many types of models: linear, random trees, decision trees and many, many others.

Let's look at the simplest. Using this equation we can write any linear function:

$$f(\theta) = \theta * \theta * \text{parametric\_value}$$

The only problem is to find this two theta parameters. how can I know what value will give me the best result? We should use to this **learning rate**. U can also **utility function**(also called **matching function**), which tells us how good is our model and **cost function** which have totally opposite meaning. In case of linera regression our task is to minimalize distance between our predicted function and the nearest learning points. **So right now we can see what linear regression method are doing: it is just fitting this theta parameters to our linear function.** This the simplies method of prediction is called **k-nearest neighbours**.

**1.9 Subsection**

Structuring a document is easy!

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

**1.9.1 Subsubsection**

More text.

$$f(x) = x^2$$

$$f(x) = x^2$$

$$g(x) = \frac{1}{x}$$

$$F(x) = \int_b^a \frac{1}{3}x^3$$

$$\frac{1}{\sqrt{x}}$$

**Paragraph** Some more text.

**Subparagraph** Even more text.

**2 Another section**