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 learnning is classification. Spam filter is great example. In this case it is trained by a huge amount of sample messages belonging to given slacc (spam or ham), thank to wchich 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 Therminology

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 imporatnt supervised learning algorithms:

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

1.3 Unsupervised learning

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

1.3.1 Types

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

- Data clustering
  - k-means
  - hierarchical cluster analisys - **HCA**
  - expectation-maximalization 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 webside. You can use clusterring to try define groups of similar users. W any moment you can check into whichgroup any user fall 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 their are visiting your webside onsly in weekends.

If you will use **Hierarchical cluster analisys** you can divide this 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 tries to save primary structure of data, so you can easier analyze this data and maybe discover some unforeseen patterns.

Dimensionality reduction algorithms tries 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 it mileage. This procces 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 annomaly in our dataset before we end it do our model.

Another one, commonly used algorithm is **association rule learning**. It task is to analize very huge abount of data and find some interesing dependenties between atributes. For example by using this method we can discover that people who are when buing ketchup and chips are also often than the other buing 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 panalities(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 oposition 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 /textbfinstance-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 simpliest. Using this equation we can write any linear function:

$$f(\theta) = \theta * \theta * parametric_value$$

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 neghbours.** It will be detailed discussed in the following part of this document.

1.9 Main problems of ML

Yeah, it look too good to be thuth. ML is big oppourtunity but also have so big drawbacks. If we think about it, the first things that are comming to our mind is **wrong algorithm** or **problem with data**. Let's start from the data problem.

1.9.1 Data deficiency

To teach a little child how to speak "mom" or how apple looks like you have to speak a thousands of different words, or show many, many thing included apple. This kid need to see many shapes, colors etc. Need to hears many different sounds included this special one. Only in this case it can learn how to think, how everything looks, sounds and smell. Ok, but this is very, very good situation. Humans are very intelligent.

Our systems are not xD To train our system we often need bilions of data, even in some simple cases and advanced models. BTW. There was a lot of experiments that shows that in the case of very big data, the simpliest and more, more complex algorithms give basicly the same result. Anyway, we have to remember that most data that we are working on are not so big, so we can't belittle the meaning of algorithms.

1.9.2 Unrepresentative data

The next problem of packs of data is what exacly are inside of them. We need to have well prepared data for our purposes. That mean that i.e. when we are doing some prediction based on Ikea's furniture prices we need to have all type of the furnitue available in this shop. In other case, our predictions will be just corrupted and our main goal won't be achieved.

We can also have huge problem because we will have even much more data than we need but that will have just **sampling noise** which mean that we have some data that is burdned with the big error. It can happen i.e. when we are colleting data from some sensor and it will break for some time. Also the big problem is **sampling bias**. For example we have 100 sensors but we are collecting data only from 25. They also can be placed in different environment or something like that. Some of them can also be broken and that is the reason why we didn't collect data from them. The last case is called **nonresponse bias**.

1.10 Poor quality of data

This is another problem but it could sounds similar to previous. In a situation where we have a lot of noise, data burdned with big error or similar, we should clean our dataset. There a lot of different options, only only limit is creativity of developer, he can i.e. delete data or flll missing values.

1.11 Negligible features

Can't make something out of nothing. To many features are bad because system have to analise them and it take time and resources. Process of selecting the right data is called **feature engineering** and it is consist of:

- feature selection: selection of the most useful features among available
- feature extractions: combining alredy existed feature to create some more useful (we can use to do this dimensionality reduction algorithm)
- creating new features from new data

1.12 Overfitting

For example, you bought something on popular auction site and you were cheated. You may want to say that is portal is crap and all sellers are thieves. This behaviour in ML have it own name and this is **overfitting**. This mean taht our model is doing well in case of our learning dataset but not when we try to give him new data. This can happen when our model is overcomplicated and we have to less data or this data is very noised. Imagine, that you want to predict price of a car based on your data. When you will have overfitted data, your system can find some pattern that shows that cars made by manufacturers with letter "b" in name are very expensive because of Bentley na Bugatti. But, is this mean that Mitsubishi is albo worth bilions of dollars? Propably no. In that case you have to optimize your data and this process is called **regularization**. You can do this by collecting more data, make your model simplier or reduce noise. Level of regularization can be controlled by hyperparameters. it is the name of learning algorithm's parameter. They aren't modified by algorithm, we have to specify them before we use it.

1.13 Underfitting

If we have over- then we should have underfitting. I think that now it is quite simple to understand problem and i don't have to write about it. Long story short: We have to simple model and to complicated dataset.

1.14 Subsection

Structuring a document is easy!aaa

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

1.14.1 Subsubsection

More text.

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

$$\begin{aligned} f(x) &= x^2 \\ g(x) &= \frac{1}{x} \\ F(x) &= \int_b^a \frac{1}{3}x^3 \end{aligned}$$

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

Paragraph Some more text.

Subparagraph Even more text.

2 Another section