Lecture 1 - 09-03-2020

1.1 Introduction of the course

In this course we look at the principle behind design of Machine learning. Not just coding but have an idea of algorithm that can work with the data.

We have to fix a mathematical framework: some statistic and mathematics.
Work on ML on a higher level
ML is data inference: make prediction about the future using data about the past

- Clustering → grouping according to similarity
- Planning → (robot to learn to interact in a certain environment)
- Classification → (assign meaning to data) example: Spam filtering
  I want to predict the outcome of this individual or i want to predict whether a person click or not in a certain advertisement.

1.2 Examples

Classify data into categories:

- Medical diagnosis: data are medical records and categories are diseases
- Document analysis: data are texts and categories are topics
- Image analysts: data are digital images and for categories name of objects in the image (but could be different).
- Spam filtering: data are emails, categories are spam vs non spam.
- Advertising prediction: data are features of web site visitors and categories could be click/non click on banners.

Classification : **Different from clustering** since we do not have semantically classification (spam or not spam) → like meaning of the image.
I have a semantic label.

Clustering: i want to group data with similarity function.

Planning: Learning what to do next

Clustering: Learn similarity function

Classification: Learn semantic labels meaning of data

Planning: Learn actions given state

In classification is an easier than planning task since I'm able to make prediction telling what is the semantic label that goes with data points.
If i can do classification i can clustering.
If you do planning you probably classify (since you understanding meaning in your position) and then you can also do clustering probably.
We will focus on classification because many tasks are about classification.

Classify data in categories we can image a set of categories.
For instance the tasks:
'predict income of a person'
'Predict tomorrow price for a stock'
The label is a number and not an abstract thing.

We can distinguish two cases:

- The label set $\rightarrow$ set of possible categories for each data point. For each of this could be finite set of abstract symbols (case of document classification, medical diagnosis). So the task is classification.
- Real number (no bound on how many of them). My prediction will be a real number and is not a category. In this case we talk about a task of regression.

Classification: task we want to give a label predefined point in abstract categories (like YES or NO)
Regression: task we want to give label to data points but this label are numbers.

When we say prediction task: used both for classification and regression tasks.
Supervised learning: Label attached to data (classification, regression)
Unsupervised learning: No labels attached to data (clustering)

In unsupervised the mathematical modelling and way algorithm are score and can learn from mistakes is a little bit harder. Problem of clustering is harder to model mathematically.
You can cast planning as supervised learning: i can show the robot which is the right action to do in that state. But that depends on planning task is formalised.
Planning is higher level of learning since include task of supervised and unsupervised learning.

Why is this important ?
Algorithm has to know how to given the label.
In ML we want to teach the algorithm to perform prediction correctly. Initially algorithm will make mistakes in classifying data. We want to tell algorithm that classification was wrong and just want to perform a score. Like giving a grade to the algorithm to understand if it did bad or really bad. So we have mistakes!

Algorithm predicts and something makes a mistake → we can correct it.
Then algorithm can be more precisely. We have to define this mistake.
Mistakes in case of classification:

- If category is the wrong one (in the simple case). We have a binary signal where we know that category is wrong.

How to communicate it?
We can use the loss function: we can tell the algorithm whether is wrong or not.

Loss function: measure discrepancy between 'true' label and predicted label.
So we may assume that every datapoint has a true label. If we have a set of topic this is the true topic that document is talking about. It is typical in supervised learning.

How good the algorithm did?

$$\ell(y, \hat{y}) \leq 0$$

were y is true label and $\hat{y}$ is predicted label

We want to build a spam filter were 0 is not spam and 1 is spam and that's a Classification task:

$$\ell(y, \hat{y} = \begin{cases} 0, & \text{if } \hat{y} = y \\ 1, & \text{if } \hat{y} \neq y \end{cases}$$

The loss function is the "interface" between algorithm and data.
So algorithm know about the data through the loss function.
If we give a useless loss function the algorithm will not perform good: is important to have a good loss function.

1.2.1 Spam filtering

$Y = \{spam, no\, spam\}$
Binary classification $|Y| = 2$
We have two main mistake:

- False positive: y =non spam, $\hat{y}$ = spam
- False negative: y =spam, $\hat{y}$ = no spam

It is the same mistake? No if i have important email and you classify as spam that's bad and if you show me a spam than it's ok.
So we have to assign a different weight.

$$\ell\left(y, \hat{y}\right) = \begin{cases} 2 & if\ FP \\ 1 & if\ FN \\ 0 & otherwise \end{cases}$$

We have to take more attention on positive mistake
Even in binary classification, mistakes are not equal.