1 Lecture 3 - 07-04-2020

Data point x represented as sequences of measurement and we called this measurements features or attributes.

$x = (x_1, ..., x_d)$ x_1 *feature valuex* $\in X^d$ $X = \mathbb{R}^d$ $X = X_1 \cdot x \cdot ... \cdot X_d \cdot x$

Label space Y
Predictor $f : X \to Y$

Example (x, y) y is the label associated with x
($\to y$ is the correct label, the ground truth)

Learning with example $(x_1, y_1)...(x_m, y_m)$ *training set*

Training set is a set of examples with every algorithm can learn.....

Learning algorithm take training set as input and produces a predictor as output.

.....DISEGNO

With image recognition we use as measurement pixels.
How do we measure the power of a predictor?
A learning algorithm will look at training set, algorithm and generate the predictor. Now the problem is verify the score.
Now we can consider a test set collection of example

$$\textit{Test set} \qquad (x'_1, y'_1)...(x'_n, y'_n)$$

Typically we collect big dataset and then we split in training set and test set randomly.
Training and test are typically disjoint
How we measure the score of a predictor? We compute the average loss.
The error is the average loss in the element in the test set.

$$\textit{Test error} \qquad \frac{1}{n} \cdot \sum_{t=1}^{n} \ell(f(x'_t), y')$$

In order to simulate we collect the test set and take the average loss of the predictor of the test set. This will give us idea of how the..
Proportion of test and train depends in how big the dataset is in general.
Our **Goal**: A learning algorithm 'A' must output f with a small test error.
A does not have access to the test set. (Test set is not part of input of A).
Now we can think in general on how a learning algorithm should be design.
We have a training set so algorithm can say:
'A' may choose f based on performance on training set.

$$\textit{Training error} \qquad \hat{\ell}(f) = \frac{1}{m} \cdot \sum_{t=1}^{m} \ell(f(x_t), y_t)$$

Given the training set $(x_1, ..., x_m)(y_1, ..., y_m)$
If $\hat{\ell}(f)$ for same f, then test of f is also small
Fix F set of predictors output $\hat{f}$

$$\hat{f} = arg\,min\,\hat{\ell}(f) f \in F$$

This algorithm is called Empirical Risk Minimiser (ERM)
When this strategy (ERM) fails?
ERM may fails if for the given training set there are:
Many $f \in F$ with small $\hat{\ell}(f)$, but not all of them have small test error

There could be many predictor with small error but some of them may have big test error. Predictor with the smallest training error doesn't mean we will have the smallest test error.
I would like to pick f^* such that:

$$f^* = arg\,min \frac{1}{n} \cdot \sum_{t=1}^{m} \ell(f(x'_t), y_t) \qquad f \in F$$

where $\ell(f(x'_t), y_t)$ is the test error
ERM works if $f^* such\ that \qquad f^* = arg\,min\,\hat{\ell}(f) \qquad f \in F$
So minimising training and test????? Check videolecture
We can think of f as finite since we are working on a finite computer.
We want to see why this can happen and we want to formalise a model in which we can avoid this to happen by design: We want when we run ERM choosing a good predictor with ..... PD

1.1 Overfitting

We called this as overfitting: specific situation in which 'A' (where A is the learning algorithm) overfits if f output by A tends to have a training error much smaller than the test error.
A is not doing his job (outputting large test error) this happen because test error is misleading.
Minimising training error doesn't mean minimising test error. Overfitting is bad.
Why this happens?
This happen because we have **noise in the data**

1.1.1 Noise in the data

Noise in the data: y_t is not deterministically associated with x_i.

Could be that datapoint appears more times in the same test set. Same datapoint is repeated actually I'm mislead since training and dataset not coincide. Minimising the training error can take me away from the point that minimise the test error.
Why this is the case?

- Some **human in the loop**: label assigned by people.(Like image contains certain object but human are not objective and people may have different opinion)
- **Lack of information**: in weather prediction i want to predict weather error. Weather is determined by a large complicated system. If i have humidity today is difficult to say for sure that tomorrow will rain.

When data are not noise i should be ok.
Labels are not noisy

Fix test set and trainign set.

$$\exists f^* \in F \qquad y'_t = f^*(x'_t) \qquad \forall (x'_t, y'_t) \quad \textit{in test set}$$

$$y_t = f^+(x_t) \qquad \forall (x_t, y_t) \quad \textit{in training set}$$

Think a problem in which we have 5 data points(vectors) :
$\vec{x_1}, ...\vec{x_5}$ *in some space* X
We have a binary classification problem $Y = \{0, 1\}$
$\{\vec{x_1}, ..., \vec{x_5}\} \in X \qquad Y = \{0, 1\}$

F contains all possible calssifier $2^5 = 32 \qquad f : \{x_1, ..., x_5\} \to \{0, 1\}$

Example					
	x_1	x_2	x_3	x_4	x_5
f	0	0	0	0	0
f'	0	0	0	0	1
f"	..	..	..	..	..

$$\textit{Training set} \quad x_1, x_2, x_3 \quad f^+$$

$$\textit{Test set} \quad x_4, x_5 \quad f^*$$

4 *classifier* $f \in F \qquad$ *will have* $\hat{\ell}(f) = 0$

$(x_1, 0) \quad (x_2, 1) \quad (x_3, 0)$
$(x_4, ?) \quad (x_5, ?)$
$f^*(x_4) \quad f^*(x_5)$
If not noise i will have deterministic data but in this example (worst case) we get problem.
I have 32 classifier to choose: i need a larger training set since i can't distinguish predictor with small and larger training(?) error. So overfitting noisy or can happen with no noisy but few point in the dataset to define which predictor is good.

1.2 Underfitting

'A' underfits when f output by A has training error close to test error but they are both large.
Close error test and training error is good but the are both large.

$A \equiv ERM$, *then A undefits if F is too small* $\rightarrow$ *not containing too much predictors*

In general, given a certain training set size:

- Overfitting when $|F|$ is too large (not enough points in training set)
- Underfitting when $|F|$ is too small

Proportion predictors and training set

$$|F|, \ i \ need \ ln|F| \quad bits \ of \ info \ to \ uniquely \ determine \ f^* \in F$$
$$m >> ln|F| \qquad when \quad |F| < \infty where \ m \ is \ the \ size \ of \ traning \ set$$

1.3 Nearest neighbour

This is completely different from ERM and is one of the first learning algorithm. This exploit the geometry of the data. Assume that our data space X is:
$X \equiv \mathbb{R}^d \qquad x = (x_1, ..., x_d) \qquad y - \{-1, 1\}$
S is the traning set $(x_1, y_1)...(x_m, y_m)$
$x_t \in \mathbb{R}^d \qquad y_t \in \{-1, 1\}$

$d = 2 \rightarrow$ *2-dimensional vector*

....– DISEGNO –...
where + and - are labels

Point of test set
If i want to predict this point?
Maybe if point is close to point with label i know then. Maybe they have

the same label.
$\hat{y} = +$ *or* $\hat{y} = -$

.....– DISEGNO – ...
I can came up with some sort of classifier.

Given S training set, i can define $h_N NX \to \{-1, 1\}$
$h_N N(x) =$ label y_t of the point x_t in S closest to X
(the breaking rule for ties)
For the closest we mean euclidian distance
$X = \mathbb{R}^d$

$$\|x - x_t\| = \sqrt{\sum_{e=1}^{d} (x_e - x_t, e)^2}$$

$$\hat{\ell}(h_{NN}) = 0$$
$$h_{NN}(x_t) = y_t$$

training error is 0!