# Lecture 12 - 21-04-2020

## 1.1 Non parametrics algorithms

We talk about **consistency**: as the training size grows unbounded the expected risk of algorithms converge to Bayes Risk.

Now we talk about **non parametric algorithm**: the structure of the model is determined by the data.
Structure of the model is fixed, like the structure of a Neural Network but in non parametric algorithm will change structure of the model as the data grows ($K_{NN}$ and tree predictor).
If I live the tree grow unboundenly then we get a non parametric tree, but if we bound the grows then we get a parametric one.

The converve rate of Bayes Risk (in this case doubled) was small. Converge of 1-$NN$ to $2\,\ell_D(f^*)$ is $m^{-\frac{1}{d+1}}$ so we need an esponential in the dimension. And we need this is under Lips assumption of $\eta$.
It's possible to converge to Bayes Risk and it's called **No free lunch**.

### 1.1.1 Theorem: No free lunch

Let a sequenece of number $a_1, a_2 \ldots \in \mathbb{R}$ such that they converge to 0.
Also $\frac{1}{22222222} \geq a_1 \geq a_2 \geq \ldots \; \forall A$ for binary classification $\exists D$ s. t.
$\ell_D(f^*) = 0$ (zero-one loss) so Bayes risk is zero and $\mathbb{E}[\,\ell_D\,(A(S_M))\,] \geq a_m \quad \forall m \geq 1$
Any Bayes Optimal you should be prepared to do so on long period of time.
This means that:

- For specific data distribution $D$, then $A$ may converge fast to Bayes Risk.
- If $\eta$ is Lipschitz then it is continous. This mean that we perturb the input by the output doesno't change too much.
- If Bayes Risk is 0 ($\ell_D(f^*) = 0$) function will be discontinous

This result typically people think twice for using consistent algorithm because

I have Bayes risk and some non conssitent algorithm that will converge to some value ($\ell_D(\hat{h}^*)$). Maybe i have Bayes risk and the convergence takes a

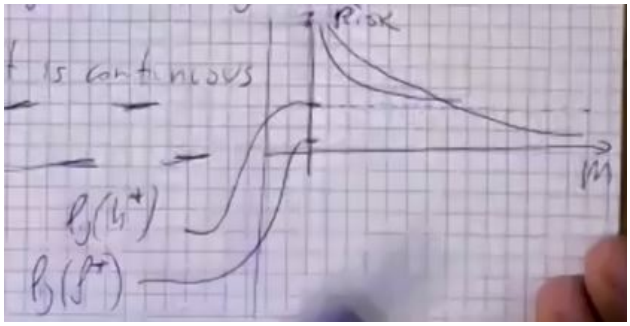


Figure 1.1: Tree building

lot on increasing data points. Before converging was better non parametric (?..)

Picture for binary classification, (similar for other losses)

- Under no assumption on $\eta$, the typicall "parametric" converge rate to risk of best model in $H$ (including ERM) is $m^{-\frac{1}{2}}$. (Bias error may be high)
- Under no assumption on $\eta$ there is no guaranteed convergence to Bayes Risk (in general) and this is **no-free-lunch** that guaranteed me no convergence rate.
- Under Lipshtiz assunption on $\eta$ the typical non parametric convergence to Bayes Risk is $m^{-\frac{1}{d+1}}$. This is exponentially worst than the parametric convergency rate.

The exponential depencendece on $d$ is called **Curse of dimnsionality**.
But if I assume small number of dimension $\longrightarrow$ $K_{NN}$ is ok if $d$ is small (and $\eta$ is "easy")
If you have a non parametric algorithm (no Bayes error but may have exponentially infinity training error). I want them to be balanced and avoid bias and variance. We need to introduce a bit of bias in controlled way.
Inserting bias to reducing variance error. So we sacrify a bit to get a better variance error.

It could be good to inject bias in order to reduce the variance error. In practise instead of having 0 training error i want to have a larger training error and hope to reduce overfitting sacrifing a bit in the training error.
I can increase bias in different technics: one is the unsamble methods.

## 1.2 Highly Parametric Learning Algorithm

### 1.2.1 Linear Predictors

Our domain is Euclidean space (so we have points of numbers).

$$X \; is \; \mathbb{R}^d \qquad x = (x_1, .., x_d)$$

A linear predictor will be a linear function of the data points.

$$h : \mathbb{R}^d \longrightarrow Y \qquad h\,(x) = f(w^T\, x) \quad w \in \mathbb{R}^d$$

$$f : \mathbb{R} \longrightarrow Y$$

And this is the dot product that is

$$w^T\, x = \sum_{t=1}^{d} w_i x_i = \|w\|\, \|x\| \cos\Theta$$

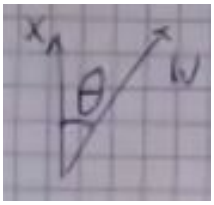


Figure 1.2: Dot product

Suppose we look a regression with square loss.

$$Y = \mathbb{R} \qquad h(x) = w^T\, x \quad w \in \mathbb{R}^d$$

$f^*(x)$ =$\mathbb{E}\,[\,Y|X = x\,]$
Binary classification with zero-one loss $Y = \{-1, 1\}$ We cannot use this since is not a real number but i can do:

$$h(x) = sgn\left(w^T\, x\right) \qquad sgn(x) = \begin{cases} +1 \; if \; z > 0 \\ -1 \; if \; z \leq 0 \end{cases}$$

where sgn is a sign function. Linear classifier.
$\|X\| \cos\Theta$ is the length of the projection of $x$ onto $w$
Now let's look at this set:

$$\{x \in \mathbb{R}^d : w^T x = c\}$$

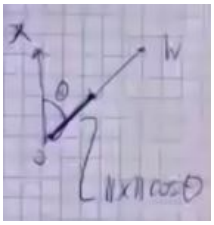


Figure 1.3: Dot product

This is a hyperplane.

$$\|w\|\|x\| \cos \Theta = c \qquad \|x\| \cos \Theta = \frac{c}{\|w\|}$$

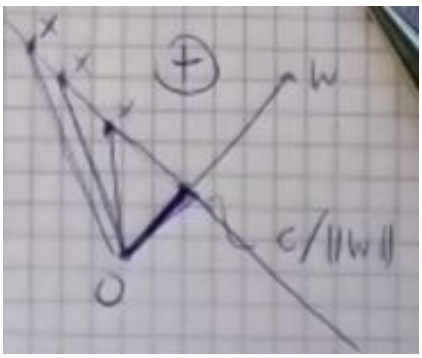


Figure 1.4: Hyperplane

So $(w, c)$ describe an hyperplane.
We can do binary classification using the hyperplane. Any points that lives in the positive half space and the negative. So the hyperplane is splitting in halfs. $H \equiv \{x \in \mathbb{R}^d : w^T x = c\}$

$$H^+ \equiv \{x \in \mathbb{R}^d : w^T x > c\} \qquad \textbf{positive } h_s$$

$$H^- \equiv \{x \in \mathbb{R}^d : w^T x \leq\} \qquad \textbf{negative } h_s$$

$$h(x) = \begin{cases} +1 \; if \; x \in H^+ \\ -1 \; if \; x \notin H^- \end{cases} \qquad h(x) = sgn(w^T - c)$$

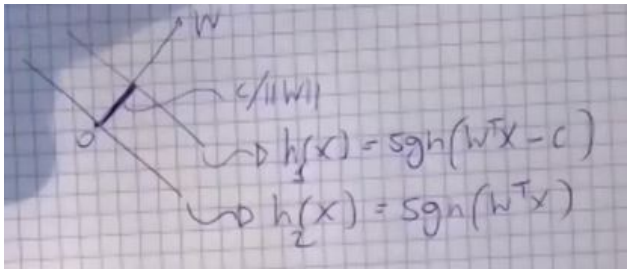


Figure 1.5: Hyperplane

$h_1$ is non-homogenous linear classifier.
$h_2$ is homogenous linear classifier. Any homogenous classifier is equivalent to

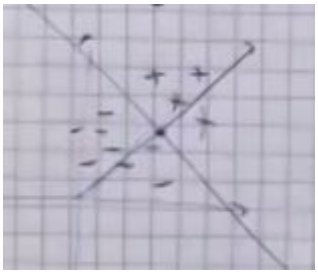

Figure 1.6: Hyperplane

this:

$$\{x \in \mathbb{R}^d : X = c\} \textbf{ is equivalent to } \quad \{x : \mathbb{R}^{d+1} : \nu^T x = 0\}$$

$$\nu = (w_1, .., w_d, -c) \qquad x' = (x_1, ..., x_d, 1)$$

So we added a dimension.

$$w^T x = c \;\Leftrightarrow\; \nu^T x' = 0$$

$$\sum_i w_1 x_1 = c \;\Leftrightarrow\; \sum_i w_1 x_1 - c = 0$$

**Rule**:
**When you learn predictor just add an extra feature to your data points, set it ot 1 and forget about non- homogenous stuff.**

One dimensional example

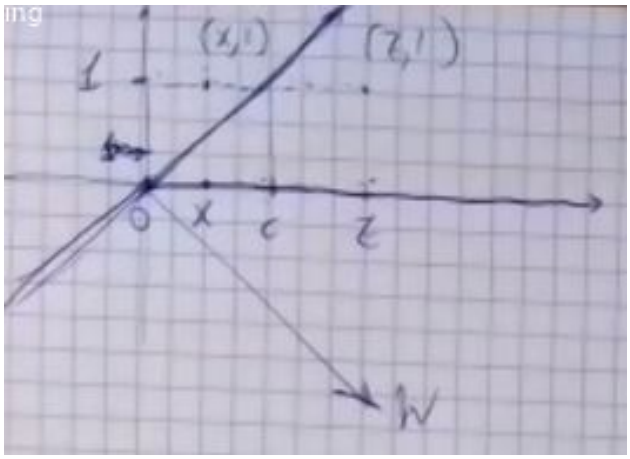


Figure 1.7: Example of one dimensional hyperplane

I have negative (left of $(x, 1)$ and positive point (left of $(z, 1)$ classified

Now i want to learn linear classifier. How can i do it?

$$H_d = \{\ h : \exists w \in \mathbb{R}^d\ h(x) = sgn(w^T x)\ \}$$

Parametric!
We expect high bias a low variance.

$$ERM \qquad \hat{h}_S = arg \min_{h \in H_d} \frac{1}{m} \cdot \sum_{t=1}^{m} I\{h(x_t) \neq y_t\} =$$

$$= \ arg \min_{w \in \mathbb{R}^d} \ \frac{1}{m} \cdot \sum_{t=1}^{m} I\ \{\ y_t\, w^T x_t \leq 0\ \}$$

A bad optimisation problem!

**FACT**:
It is unlikely to find an algorithm that solves ERM for $H_d$ and zero-one loss efficiently.
**NP completeness problems!**
It's very unlikely to solve this problem.
This problem is called **MinDisagreement**

### 1.2.2 MinDisagreement

Instance: $(x_1, y_1)...(x_m, y_m) \in \{0,1\}^d \, x \, \{-1, 1\}, \quad k \in \mathbb{N}$
Question: Is there $w \in x \, D^d$
s.t $y_t \, w^T x_t \leq 0$ for at most $k$ indices $t \in \{1, ...m\}$
This is NP-complete!