Mathematical Tools in Machine Learning

Fadoua Balabdaoui

Seminar für Statistik, ETH

17 octobre 2019

Lecture 4 (Week 5)

Halfspaces (continued)

Linear Regression

Logistic regression

Bias-Complexity Trade-off (Chapter 5)

Lecture 1

Halfspaces : the separable case

- Let S be a **training set** $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ such that
  - $(x_i, y_i), i = 1, \ldots, m$ are i.i.d $\sim \mathcal{D}$ like (x, y)
  - there exists $w^* \in \mathbb{R}^d$ such that

$$\mathrm{sign}\left(\langle w^*, x\rangle\right) = y$$

- This means that we are in the **realizability** case where

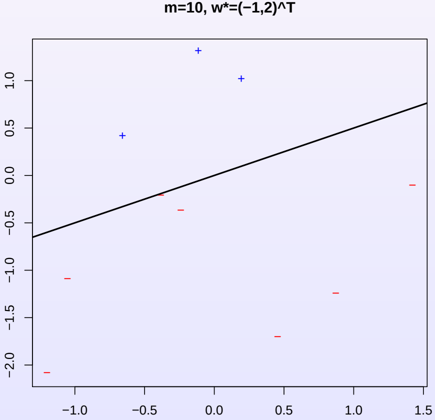
$$\mathrm{sign} \circ h_{w^*}(x) = \mathrm{sign}\left(\langle w^*, x\rangle\right)$$

is a **perfect classifier**

- This case is also called the "**separable case**".

Halfspaces : the separable case

- The following training set S of size $m = 10$ comes from a distribution which is separable with $w^* = (-1, 2)^T$



Halfspaces : the separable case

- In this separable case, we have $\ell_{0-1}(\mathrm{sign} \circ h_{w^*}) = 0$,

$$\mathbb{P}_{(x,y)\sim\mathcal{D}}(\mathrm{sign} \circ h_{w^*}(x) \neq y) = 0$$

- Recall that separability implies that for any ERM rule $\mathrm{sign} \circ h_{w_S}$, that is $\mathrm{sign} \circ h_{w_S} \in \mathrm{argmin}_{f\in\mathcal{H}} L_S(f)$ (with $\mathcal{H} = \mathsf{HS}_d$) we have :

$$L_S(\mathrm{sign} \circ h_{w_S}) = 0, \quad \text{with probability 1}$$

because $\mathrm{sign} \circ h_{w^*}(x_i) = y_i$ with probability 1 and hence

$$L_S(\mathrm{sign} \circ h_{w_S}) \leq L_S(\mathrm{sign} \circ h_{w^*}) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{\mathrm{sign}\circ h_{w^*}(x_i)\neq y_i} = 0.$$

- This means that if we want to find an ERM rule, we need to find $w_S \in \mathbb{R}^d$ such that w_S **perfectly classifies** all the examples in S.

Halfspaces : Solution 1 (LP)

- A **Linear Program** (LP) aims at finding the **solution** of the optimization problem

$$\max_{w \in \mathbb{R}^d} \langle u, w \rangle \quad \text{subject to} \quad Aw \geq v$$

where $u \in \mathbb{R}^d, v \in \mathbb{R}^m, A \in \mathbb{R}^m \times \mathbb{R}^d$ are given.

- **Example** : Solve $\max_{(w_1,w_2)^T \in \mathbb{R}^2} \{250 w_1 + 75 w_2\}$ subject to

$$5w_1 + w_2 \leq 100, \quad w_1 + w_2 \leq 60, \quad w_1 \geq 0 \;\; \text{and} \;\; w_2 \geq 0$$

Here : $u = (250, 75)^T, v = (-100, -60, 0, 0)^T$ and

$$A = \begin{pmatrix} -5 & -1 \\ -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

- The problem admits the unique solution $w = (10, 50)^T$.

Halfspaces : Solution 1 (LP)

- We know by the separability that there exists $w^* \in \mathbb{R}^d$ such that

$$\mathrm{sign}(\langle w^*, x_i \rangle) = y_i$$

for $i = 1, \ldots, m$.

- Let $\gamma = \min_{i \in \{1,\ldots,m\}} y_i \langle w^*, x_i \rangle$

$$\bar{w} = \frac{w^*}{\gamma}.$$

Then, for all $i \in \{1, \ldots, m\}$

$$y_i \langle \bar{w}, x_i \rangle = \frac{1}{\gamma} y_i \langle w^*, x_i \rangle \geq 1.$$

Halfspaces : Solution 1 (LP)

- Hence, we have found a vector $w \in \mathbb{R}^d \in \mathrm{argmin}_{w' \in \mathbb{R}^d} L_S(\mathsf{sign} \circ h_{w'})$ such that

$$y_i \langle w, x_i \rangle \geq 1, \quad \forall\, i \in \{1, \ldots, m\} \tag{1}$$

- Consider $v = (1, \ldots, 1)^T \in \mathbb{R}^m$ and the matrix $A \in \mathbb{R}^m \times \mathbb{R}^d$:

$$A = \begin{pmatrix} y_1 x_{1,1} & \ldots & y_1 x_{1,d} \\ \vdots & \vdots & \vdots \\ y_i x_{i,1} & \ldots & y_i x_{i,d} \\ \vdots & \vdots & \vdots \\ y_m x_{m,1} & \ldots & y_m x_{m,d} \end{pmatrix}.$$

- Then, (1) is **equivalent** to $Aw \geq v$.

Halfspaces : Solution 2 (The Perceptron)

- The Perceptron is an **iterative algorithm** due to Rosenblatt (1957) :
  - it produces a **sequence** of vectors $w^{(1)}, w^{(2)}, \dots, w^{(T)}$
  - when it stops at iteration T, the output $w^{(T)}$ yields a **perfect classifier** (and hence an ERM rule)
- The algorithm runs as follows
  - **Start** with $w^{(1)} = (0, \dots, 0)^T \in \mathbb{R}^d$.
  - At **iteration** t, if $w^{(t)}$ is **not** a perfect classifier, then **find** an $i \in \{1, \dots, m\}$ such that (x_i, y_i) is **mislabeled** : $y_i \langle x_i, w^{(t)} \rangle \leq 0$.
  - At **iteration** $(t+1)$, **update** as follows :

$$w^{(t)} \to w^{(t+1)} = w^{(t)} + y_i x_i.$$

- The update **guides** the sequence toward a more **correct** labeling :

$$y_i \langle w^{(t+1)}, x_i \rangle = y_i \langle w^{(t)} + y_i x_i \rangle x_i \quad = \quad y_i \langle x_i, w^{(t)} \rangle + \|x_i\|^2.$$

Halfspaces : Solution 2 (The Perceptron)

- **Theorem.** Assume separability and let

$$B = \min\{\|w\| : y_i\langle w, x_i\rangle \geq 1,\ \forall i = 1, \ldots, m\}, \quad \text{and} \quad R = \max_{1 \leq i \leq m} \|x_i\|.$$

Then, the algorithm **stops** after at most $\lfloor (RB)^2 \rfloor$ iterations,

$$y_i\langle w^{(T)}, x_i\rangle > 0,\ \forall\ i = 1, \ldots, m.$$

- **Proof.** For $t \geq 1$ an integer, we will show that if at iteration t, $w^{(t)}$ mislabels some example (x_i, y_i) then we must have

$$\frac{\langle w^*, w^{(t+1)}\rangle}{\|w^*\|\|w^{(t+1)}\|} \geq \frac{\sqrt{t}}{RB}. \tag{2}$$

Halfspaces : Solution 2 (The Perceptron)

- Assume for now that (2) is true.

$$\|w^*\|\|w^{(t+1)}\| \geq |\langle w^*, w^{(t+1)}\rangle| \geq \langle w^*, w^{(t+1)}\rangle$$

and hence it follows from (2) that

$$1 \geq \frac{\langle w^*, w^{(t+1)}\rangle}{\|w^*\|\|w^{(t+1)}\|} \geq \frac{\sqrt{t}}{RB} \implies t \leq (RB)^2.$$

- **Recall** that $w^{(1)} = (0, \ldots, 0)^T$.

$$w^{(2)} = w^{(1)} + y_i x_i$$

- **Suppose** that for $t \geq 3$ **we have that** $\langle w^*, w^{(t)}\rangle \geq t - 1$.

Halfspaces : Solution 2 (The Perceptron)

- Let $i \in \{1, \ldots, m\}$ such that $y_i \langle w^{(t)}, x_i \rangle \leq 0$.

$$\langle w^*, w^{(t+1)} \rangle = \langle w^*, w^{(t)} + y_i x_i \rangle \quad \geq \quad t - 1 + 1 = t.$$

We conclude by induction that if the Perceptron output $w^{(t)}$ is not a perfect classifier, then

$$\langle w^*, w^{(t+1)} \rangle \geq t. \tag{3}$$

$$\begin{aligned} \|w^{(j+1)}\|^2 = \|w^{(j)} + y_i x_i\|^2 \quad &= \quad \|w^{(j)}\|^2 + 2 \underbrace{y_i \langle w^{(j)}, x_i \rangle}_{\leq 0} + \|x_i\|^2 \\ &\leq \quad \|w^{(j)}\|^2 + R^2. \end{aligned}$$

Halfspaces : Solution 2 (The Perceptron)

and hence,

$$\sum_{j=1}^{t}\left(\|w^{(j+1)}\|^2 - \|w^{(j)}\|^2\right) \leq R^2 t$$

Therefore,

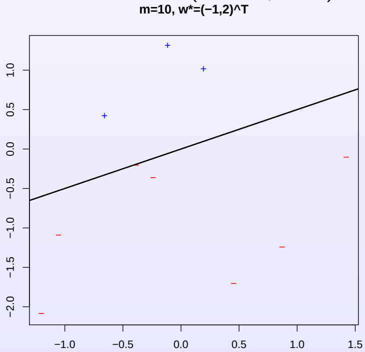
$$\frac{\langle w^*, w^{(t+1)}\rangle}{\|w^*\|\|w^{(t+1)}\|} \geq \frac{t}{BR\sqrt{t}} = \frac{\sqrt{t}}{BR}$$

- The Perceptron stops indeed in at most $\lfloor (RB)^2 \rfloor$ iterations :
- If $RB >> 1$, the Perceptron can be slow in finding the solution.

The Perceptron applied to the previous example with $m = 10$

- The Perceptron terminates after 2 iterations with

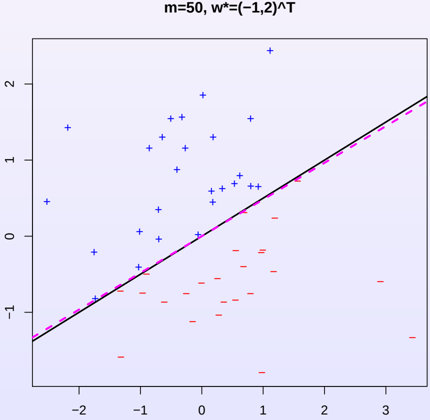
$$w_S = w^{(2)} = (-0.22, 2.20)^T.$$



The Perceptron applied to $m = 50$

- The Perceptron terminates at iteration $T = 11$ with

$$w_S = w^{(11)} = (-0.93, 1.94)^T.$$



Lecture 1

Predicting a real outcome : linear regression

- In linear regression, we assume that $(x, y) \sim \mathcal{D}$ such that

$$\mathbb{E}[y|x] = \langle w^*, x\rangle + b^*, \quad \text{for some } w^* \in \mathbb{R}^d, b^* \in \mathbb{R}.$$

- Hence, we want learn about the relationship between x and y by considering the hypothesis class

$$\mathcal{H}_{\mathrm{reg}} = L_d = \left\{ h_{w,b} : \quad w \in \mathbb{R}^d, b \in \mathbb{R} \right\}$$

with $h_{w,b}(x) = \langle w, x\rangle + b$.

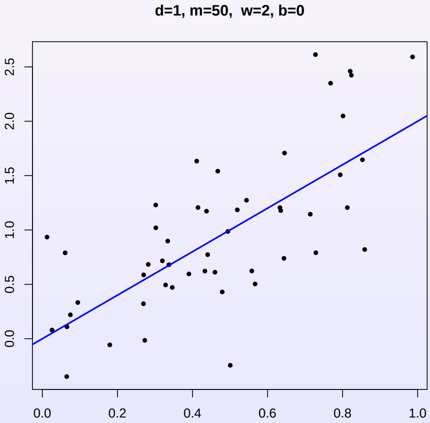
- The loss function is given by

$$\ell_{\mathrm{sq}}(h, (x, y)) = (h(x) - y)^2$$

so that the true risk is $L_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\left(h(x) - y\right)^2\right]$

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m}(h(x_i) - y_i)^2.$$

Linear regression : example



Here, $x \sim \mathcal{U}[0,1]$ and $y|x \sim \mathcal{N}(2x, 0.5^2)$ with $\mathbb{E}[y|x] = 2x$.

Linear regression : Least Squares Estimator

- In the following, we assume the homogeneous representation $h_w(x) = \langle w, x \rangle$.

$$\min_{w \in \mathbb{R}^d} L_S(h_w) = \min_{w \in \mathbb{R}^d} \left\{ \frac{1}{m} \sum_{i=1}^{m} \left(\langle w, x_i \rangle - y_i \right)^2 \right\}.$$

- A solution has to be a critical vector of $w \mapsto L_S(h_w)$:

$$\nabla L_S(h_w) = (0, \ldots, 0)^T,$$

that is we look for $w_S \in \mathbb{R}^d$ such that

$$\begin{pmatrix} \frac{\partial L_S(h_w)}{\partial w_1}\big|_{w_S} \\ \vdots \\ \frac{\partial L_S(h_w)}{\partial w_d}\big|_{w_S} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^d.$$

Linear regression : Least Squares Estimator

- for $j \in \{1, \ldots, d\}$

$$\frac{\partial L_S(h_w)}{\partial w_j}|_w = \frac{2}{m} \sum_{i=1}^{m} x_{i,j}(\langle w, x_i \rangle - y_i)$$

with $x_i = (x_{i,1}, \ldots, x_{i,d})^T, \; i = 1, \ldots, m.$

$$\begin{aligned} \nabla L_S(h_w) &= \frac{2}{m} \sum_{i=1}^{m} x_i(\langle w, x_i \rangle - y_i) \\ &= \frac{2}{m} \sum_{i=1}^{m} x_i(x_i^T w - y_i) = \mathbf{0} \end{aligned}$$

if and only if

$$\sum_{i=1}^{m} x_i x_i^T w = \sum_{i=1}^{m} y_i x_i.$$

Linear regression : Least Squares Estimator

- Put $A = \sum_{i=1}^{m} x_i x_i^T \in \mathbb{R}^d$.

$$w = w_S = A^{-1}\left(\sum_{i=1}^{m} y_i x_i\right).$$

The solution is called also the **Least Squares Estimator**.

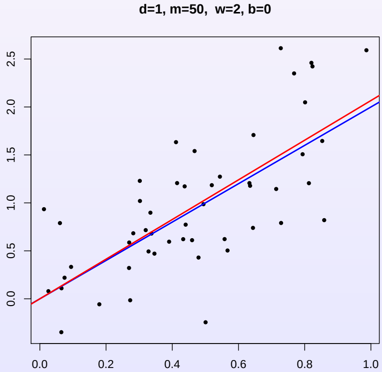
- **Remark 1 :** Note that if we write

$$X = \begin{pmatrix} x_{1,1} & \dots & x_{1,d} \\ \vdots & \vdots & \vdots \\ x_{m,1} & \dots & x_{m,d} \end{pmatrix} \in \mathbb{R}^m \times \mathbb{R}^d$$

(X is called the **design matrix**),

Linear regression : Least Squares Estimator

- **Remark 2 :** if $d = 1$, the LSE is given by $w_S = \frac{\sum_{i=1}^{m} y_i x_i}{\sum_{i=1} x_i^2}$



with $w = 2.0672$, the obtained LSE (slope of the red line).

Regression with polynomial predictors

- Consider $\mathcal{X} = \mathcal{Y} = \mathbb{R}$ and $(x, y) \sim \mathcal{D}$ such that

$$\mathbb{E}[y|x] = a_0^* + a_1^* x + \ldots + a_n^* x^n$$

for some integer $n \geq 1$, and $a_k^* \in \mathbb{R}$ for $k = 0, \ldots, n$.

- The hypothesis class is then given by

$$\mathcal{H}_{\text{poly}} = \left\{ x \mapsto \sum_{k=0}^{n} a_k x^k, a_k \in \mathbb{R} \right\}.$$

- If $\psi(x) = (1, x, \ldots, x^n)^T \in \mathbb{R}^{n+1}$,

$$\begin{aligned} \mathcal{H}_{\text{poly}} \quad &= \quad \left\{ h_a : a = (a_0, \ldots, a_n)^T \in \mathbb{R}^{n+1} \right\} \\ \text{with} \quad & \qquad h_a(x) = \langle a, \psi(x) \rangle, \quad x \in \mathcal{X}. \end{aligned}$$

Regression with polynomial predictors

- Let $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ be a training set.

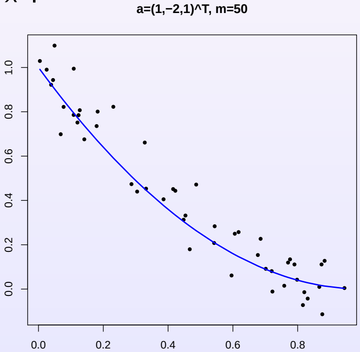
$$a_S \in \operatorname{argmin}_{a \in \mathbb{R}^{n+1}} \left\{ \frac{1}{m} \sum_{i=1}^{m} \Big(\langle a, \psi(x_i) \rangle - y_i \Big)^2 \right\}.$$

- Put $A = \sum_{i=1}^{m} \psi(x_i)\psi(x_i)^T$.

$$a_S = A^{-1} \left(\sum_{i=1} y_i \psi(x_i) \right) \in \mathbb{R}^{n+1}.$$

Regression with polynomial predictors : Example

- $\mathbb{E}[y|x] = 1 - 2x + x^2$.



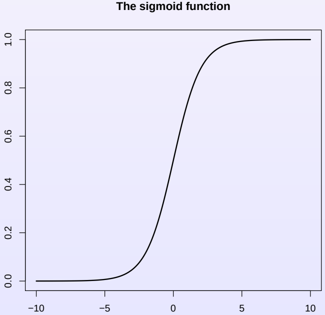
with $a_S = (1.05, -2.26, 1.24)^T$.

Lecture 1

Back to binary classification

- Consider again binary classification with $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{-1, 1\}$.

$$\phi_{\text{sig}}(z) = \frac{1}{1 + \exp(-z)}$$



$\lim_{z \to -\infty} \phi_{\text{sig}}(z) = 0$ and $\lim_{z \to -\infty} \phi_{\text{sig}}(z) = 1$.

Binary classification with logistic regression

- In **logistic regression**, the goal is learn the hypothesis class

$$\begin{aligned}\mathcal{H}_{\mathrm{sig}} &= \phi_{\mathrm{sig}} \circ L_d = \left\{\phi_{\mathrm{sig}} \circ h_w, \quad w \in \mathbb{R}^d\right\} \\ &= \left\{x \mapsto \frac{1}{1+\exp(-\langle w, x\rangle)}, \quad w \in \mathbb{R}^d\right\}.\end{aligned}$$

- Question : what **loss function** should we choose ?
- If we interpret $\phi_{\mathrm{sig}} \circ h_w(x)$ as $\mathbb{P}[y=1|x]$,

$$\begin{aligned}\langle w, x\rangle \text{ very large} &\implies \mathbb{P}[y=1|x] \approx 1 \\ \langle w, x\rangle \text{ very small} &\implies \mathbb{P}[y=1|x] \approx 0 \quad (\mathbb{P}[y=-1|x] \approx 1)\end{aligned}$$

- We want $\phi_{\mathrm{sig}} \circ h_w$ to take values close to 1 when $y=1$, and values close to 0 if $y=-1$.

Binary classification with logistic regression

- Answer : any reasonable loss function should give a **greater penalty** to an element $\phi_{\text{sig}} \circ h_w$ in case $-y\langle w, x\rangle$ takes **greater values** :
  - if $y = 1$, this means $\langle w, x\rangle$ small and we know that $\phi_{\text{sig}} \circ h_w(x) \approx 0$,
  - if $y = -1$, this means $\langle w, x\rangle$ large and we know that $\phi_{\text{sig}} \circ h_w(x) \approx 1 \iff 1 - \phi_{\text{sig}} \circ h_w(x) \approx 0$.
- Thus, the loss function should be increasing in $-y\langle w, x\rangle$,
- A possible choice is $\ell(\phi_{\text{sig}} \circ h_w, (x, y)) = \log\Big(1 + \exp(-y\langle w, x\rangle)\Big)$.
- Given a training set $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$,

$$w_S \in \mathrm{argmin}_{w \in \mathbb{R}^d} \left\{ \frac{1}{m} \sum_{i=1}^{m} \log\Big(1 + \exp(-y_i\langle w, x_i\rangle)\Big) \right\}.$$

Binary classification with logistic regression : some remarks

- Let us write $p_w = \phi_{\text{sig}} \circ h_w \in \mathcal{H}_{\text{sig}}$. Note that for $w \in \mathbb{R}^d$

$$\exp(-\langle w, x \rangle) = \frac{1}{p_w(x)} - 1 = \frac{1 - p_w(x)}{p_w(x)}$$

or equivalently

$$\langle w, x \rangle = \log\left(\frac{p_w(x)}{1 - p_w(x)}\right) = \text{logit}(p_w(x)).$$

- Also, we can make the dependence of the loss on p_w clear :

$$\begin{aligned} \ell(p_w, (x, y)) &= \log(1 + \exp(-y\langle w, x \rangle)) \\ &= \log\left(1 + \left(\frac{1 - p_w(x)}{p_w(x)}\right)^y\right) \end{aligned}$$

Binary classification with logistic regression : some remarks

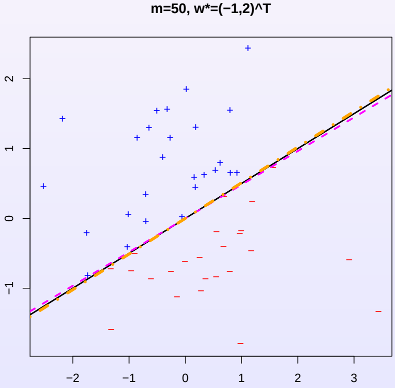
- Hence,

$$\ell(p_w,(x,y)) = \begin{cases} \log(1/p_w(x)), & \text{if } y = 1 \\ \log(1/(1-p_w(x))), & \text{if } y = -1. \end{cases}$$

- Minimizing $\sum_{i=1}^{m} \ell(p_w,(x_i,y_i))$ "pushes"
  - $p_w(x_i)$ to be large / close to 1 when $y_i = 1$
  - $p_w(x_i)$ to be small / close to 0 when $y_i = -1$.

Logistic regression : The example with $m = 50$

- The function $w \mapsto \sum_{i=1}^{m} \log\left(1 + \exp(-y_i \langle w, x_i \rangle)\right)$ is **convex**.



Summary

- The hypothesis class

$$L_d = \{x \mapsto h_{w,b}(x) = \langle w, x \rangle + b : w \in \mathbb{R}^d, b \in \mathbb{R}\}$$

  and related classes $\phi \circ L_d$ for some ϕ form classes of linear predictors.
- Linear predictor classes are widely used in binary classification, linear and polynomial regression.
- Finding ERM rules in such classes can be done using well-known and efficient algorithms (LP, Perceptron, gradient descent).

Lecture 1

Introduction into the trade-off problem

- Training data can be **misleading** and result in **overfitting**.
- That is one motivation to consider **finite hypothesis classes**.
- When we choose a finite hypothesis class $\mathcal{H}$, we reflect some **prior knowledge** we have about the learning problem.
- **Some questions.**
  - Is such a prior knowledge necessary to be a successful learner ?
  - Is it possible that some "super" learner can be successful at **any** learning task ?

Introduction into the trade-off problem

- Recall that in a learning task
  - we have a training set S of i.i.d. examples coming from an **unknown** distribution $\mathcal{D}$ over some domain $\mathcal{Z}$ $(= \mathcal{X} \times \mathcal{Y})$
  - our goal is to find a **prediction rule** $h : \mathcal{X} \to \mathcal{Y}$ with a **small true risk** $L_{\mathcal{D}}(h) = \mathbb{E}_{z \sim \mathcal{D}}[\ell(h, z)]$.
- The question about the super learner can be reformulated as follows :

$$S \mapsto A(S)$$

and m

The No-Free-Lunch Theorem

- The answer is **no**!
- **Theorem.** Let $m < |\mathcal{X}|/2$ representing a training set size.

  Then, there **exists** a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ such that

  1. There exists a labeling function $f : \mathcal{X} \mapsto \{0,1\}$ with $L_{\mathcal{D}}(f) = \mathbb{P}_{(x,y)\sim\mathcal{D}}(f(x) \neq y) = 0$.
  2. With probability of at least $1/7$ over the choice of $S \sim \mathcal{D}^m$ we have that $L_{\mathcal{D}}(A(S)) \geq 1/8$; i.e.,

$$\mathcal{D}^m(L_{\mathcal{D}}(A(S)) \geq 1/8) \geq 1/7.$$

Immediate consequence of the No-Free Lunch Theorem

- **Corollary.**

Proof.

- for any $(\epsilon, \delta) \in (0,1)^2$
- for any distribution $\mathcal{D}$

for which the realizability assumption holds ($L_{\mathcal{D}}(f) = 0$),

$$\mathcal{D}^m(S : L_{\mathcal{D}}(A(S)) \leq \epsilon) \geq 1 - \delta.$$

Immediate consequence of the No-Free-Lunch Theorem

Now, choose $\epsilon_0 = 1/9 < 1/8$, $\delta_0 = 1/8 < 1/7$.

Since $|\mathcal{X}| > 2m$, it follows from the No-Free-Lunch Theorem that there exists a distribution $\mathcal{D}_0$ and a labeling function f_0 such that $L_{\mathcal{D}_0}(f_0) = 0$ and

$$\mathcal{D}_0^m(S : L_{\mathcal{D}_0}(A(S)) > \epsilon_0)$$

or equivalently

$$\mathcal{D}_0^m(S : L_{\mathcal{D}_0}(A(S)) \leq \epsilon_0) < 1 - \delta_0. \qquad \square$$

Proof of the No-Free Lunch Theorem

On the blackboard.