

# Empirical Risk Minimization

Fabrice Rossi

SAMM  
Université Paris 1 Panthéon Sorbonne

2018

Introduction

PAC learning

ERM in practice

# General setting

## Data

- ▶  $\mathcal{X}$  the “input” space and  $\mathcal{Y}$  the “output” space
- ▶  $D$  a fixed and unknown distribution on  $\mathcal{X} \times \mathcal{Y}$

## Loss function

A loss function  $l$  is

- ▶ a function from  $\mathcal{Y} \times \mathcal{Y}$  to  $\mathbb{R}^+$
- ▶ such that  $\forall \mathbf{Y} \in \mathcal{Y}, \quad l(\mathbf{Y}, \mathbf{Y}) = 0$

## Model, loss and risk

- ▶ a model  $g$  is a function from  $\mathcal{X}$  to  $\mathcal{Y}$
- ▶ given a loss function  $l$  the risk of  $g$  is  $R_l(g) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim D}(l(g(\mathbf{x}), \mathbf{y}))$
- ▶ optimal risk  $R_l^* = \inf_g R_l(g)$

## Data set

- ▶  $\mathcal{D} = ((\mathbf{X}_i, \mathbf{Y}_i))_{1 \leq i \leq N}$
- ▶  $(\mathbf{X}_i, \mathbf{Y}_i) \sim D$  (i.i.d.)
- ▶  $\mathcal{D} \sim D^N$  (product distribution)

## General problem

- ▶ a learning algorithm creates from  $\mathcal{D}$  a model  $g_{\mathcal{D}}$
- ▶ does  $R_l(g_{\mathcal{D}})$  reaches  $R_l^*$  when  $|\mathcal{D}|$  goes to infinity?
- ▶ if so, how quickly?

## Empirical risk

$$\widehat{R}_l(g, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^N l(g(\mathbf{X}_i), \mathbf{Y}_i) = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} l(g(\mathbf{x}), \mathbf{y})$$

## ERM algorithm

- ▶ choose a class of functions  $\mathcal{G}$  from  $\mathcal{X}$  to  $\mathcal{Y}$
- ▶ define

$$g_{ERM, l, \mathcal{G}, \mathcal{D}} = \arg \min_{g \in \mathcal{G}} \widehat{R}_l(g, \mathcal{D})$$

- ▶ is ERM a “good” machine learning algorithm?

# Three distinct problems

## 1. an optimization problem

- ▶ given  $l$  and  $\mathcal{G}$  how difficult is finding  $\arg \min_{g \in \mathcal{G}} \widehat{R}_l(g, \mathcal{D})$ ?
- ▶ given limited computational resources, how close can we get to  $\arg \min_{g \in \mathcal{G}} \widehat{R}_l(g, \mathcal{D})$ ?

## 2. an estimation problem

- ▶ given  $\mathcal{G}$  a class of function, define  $R_{l, \mathcal{G}}^* = \inf_{g \in \mathcal{G}} R_l(g)$
- ▶ can we bound  $R_l(g_{\mathcal{D}}) - R_{l, \mathcal{G}}^*$ ?

## 3. an approximation problem

- ▶ can be bound  $R_{l, \mathcal{G}}^* - R_l^*$ ?
- ▶ in a way that is compatible with estimation?

# Three distinct problems

## 1. an optimization problem

- ▶ given  $l$  and  $\mathcal{G}$  how difficult is finding  $\arg \min_{g \in \mathcal{G}} \widehat{R}_l(g, \mathcal{D})$ ?
- ▶ given limited computational resources, how close can we get to  $\arg \min_{g \in \mathcal{G}} \widehat{R}_l(g, \mathcal{D})$ ?

## 2. an estimation problem

- ▶ given  $\mathcal{G}$  a class of function, define  $R_{l, \mathcal{G}}^* = \inf_{g \in \mathcal{G}} R_l(g)$
- ▶ can we bound  $R_l(g_{\mathcal{D}}) - R_{l, \mathcal{G}}^*$ ?

## 3. an approximation problem

- ▶ can be bound  $R_{l, \mathcal{G}}^* - R_l^*$ ?
- ▶ in a way that is compatible with estimation?

## Focus of this course

- ▶ the estimation problem
- ▶ and then the approximation problem
- ▶ with a few words about the optimization problem

Introduction

**PAC learning**

ERM in practice



## Learning concepts

- ▶ a concept  $c$  is a mapping from  $\mathcal{X}$  to  $\mathcal{Y} = \{0, 1\}$
- ▶ in concept learning, the loss function  $l_b$  with  $l_b(p, t) = \mathbf{1}_{p \neq t}$
- ▶ we consider only a distribution  $D_{\mathcal{X}}$  over  $\mathcal{X}$
- ▶ risk and empirical risk definitions are adapted to this setting:
  - ▶ risk:  $R(g) = \mathbb{E}_{\mathbf{x} \sim D_{\mathcal{X}}}(\mathbf{1}_{g(\mathbf{x}) \neq c(\mathbf{x})})$
  - ▶ empirical risk:  $\widehat{R}(g, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{g(\mathbf{x}_i) \neq c(\mathbf{x}_i)}$
- ▶ in essence the pair  $(D_{\mathcal{X}}, c)$  replaces  $D$ : this corresponds to a noise free situation
- ▶ as a consequence a data set is  $\mathcal{D} = \{\mathbf{X}_1, \dots, \mathbf{X}_N\}$  and has to be complemented by a concept to learn

## Notations

If  $\mathcal{A}$  is a learning algorithm, then  $\mathcal{A}(\mathcal{D})$  is the model produced by running  $\mathcal{A}$  on the data set  $\mathcal{D}$

## Definition

A concept class  $\mathcal{C}$  (i.e. a set of concepts) is PAC-learnable if there is an algorithm  $\mathcal{A}$  and a function  $\mathcal{N}_{\mathcal{C}}$  from  $[0, 1]^2$  to  $\mathbb{N}$  such that: for any  $1 > \epsilon > 0$  and any  $1 > \delta > 0$ , for any distribution  $D_{\mathcal{X}}$  and any concept  $c \in \mathcal{C}$ , if  $N \geq \mathcal{N}_{\mathcal{C}}(\epsilon, \delta)$  then

$$\mathbb{P}_{\mathcal{D} \sim D_{\mathcal{X}}^N} \{R(\mathcal{A}(\mathcal{D})) \leq \epsilon\} \geq 1 - \delta$$

- ▶ probably  $\geq 1 - \delta$
- ▶ approximately correct  $\leq \epsilon$

# Concept learning and ERM

## Remark

- ▶ the concept to learn  $c$  is in  $C$
- ▶ thus  $R_G^* = 0$
- ▶ in addition, for any  $\mathcal{D}$ ,  $\widehat{R}(g_{ERM,g,\mathcal{D}}, \mathcal{D}) = 0$
- ▶ then ERM provides PAC-learnability if for any  $g \in C$  such that  $\widehat{R}(g, \mathcal{D}) = 0$ ,  $\mathbb{P}_{\mathcal{D} \sim D_X^N} \{R(g) \leq \epsilon\} \geq 1 - \delta$

## Theorem

Let  $C$  be a *finite* concept class and let  $\mathcal{A}$  be an algorithm that outputs  $\mathcal{A}(\mathcal{D})$  such that  $\widehat{R}(\mathcal{A}(\mathcal{D}), \mathcal{D}) = 0$ . Then when  $N \geq \left\lceil \frac{1}{\epsilon} \log \frac{|C|}{\delta} \right\rceil$ ,

$$\mathbb{P}_{\mathcal{D} \sim D_X^N} \{R(\mathcal{A}(\mathcal{D})) \leq \epsilon\} \geq 1 - \delta$$

1. we consider ways to break the AC part, i.e. having both  $\widehat{R}(g, \mathcal{D}) = 0$  and  $R(g) > \epsilon$ . We have

$$Q = \mathbb{P}(\exists g \in \mathcal{C}, \widehat{R}(g, \mathcal{D}) = 0 \text{ and } R(g) > \epsilon) = \mathbb{P}\left(\bigcup_{g \in \mathcal{C}} (\widehat{R}(g, \mathcal{D}) = 0 \text{ and } R(g) > \epsilon)\right)$$

2. union bound  $Q \leq \sum_{g \in \mathcal{C}} \mathbb{P}(\widehat{R}(g, \mathcal{D}) = 0 \text{ and } R(g) > \epsilon)$
3. then we have

$$\begin{aligned} \mathbb{P}(\widehat{R}(g, \mathcal{D}) = 0 \text{ and } R(g) > \epsilon) &= \mathbb{P}(\widehat{R}(g, \mathcal{D}) = 0 | R(g) > \epsilon) \mathbb{P}(R(g) > \epsilon) \\ &\leq \mathbb{P}(\widehat{R}(g, \mathcal{D}) = 0 | R(g) > \epsilon) \end{aligned}$$

- ▶ notice that  $R(g) = \mathbb{P}_{\mathbf{X} \sim D_{\mathcal{X}}}(g(\mathbf{X}) \neq c(\mathbf{X}))$
- ▶ thus  $\mathbb{P}_{\mathbf{X} \sim D_{\mathcal{X}}}(g(\mathbf{X}) = c(\mathbf{X}) | R(g) > \epsilon) \leq 1 - \epsilon$
- ▶ as the observations are i.i.d,  
 $\mathbb{P}(\widehat{R}(g, \mathcal{D}) = 0 | R(g) > \epsilon) \leq (1 - \epsilon)^N \leq e^{-N\epsilon}$
- ▶ finally

$$\mathbb{P}(\exists g \in \mathcal{C}, \widehat{R}(g, \mathcal{D}) = 0 \text{ and } R(g) > \epsilon) \leq |\mathcal{C}| e^{-N\epsilon}$$

- ▶ then if  $\widehat{R}(\mathcal{A}(\mathcal{D}), \mathcal{D}) = 0$ ,  $\mathbb{P}_{\mathcal{D} \sim D_{\mathcal{X}}^N} \{R(\mathcal{A}(\mathcal{D})) \leq \epsilon\} \geq 1 - |\mathcal{C}| e^{-N\epsilon}$
- ▶ we want  $|\mathcal{C}| e^{-N\epsilon} \leq \delta$ , which happens when  $N \geq \frac{1}{\epsilon} \log \frac{|\mathcal{C}|}{\delta}$

## ERM

- ▶ ERM provides PAC-learnability for finite concept classes
- ▶ optimization computational cost in  $\Theta(N|C|)$

## Data consumption

- ▶ the data needed to reach some PAC level grows with the logarithm of the concept class
- ▶ a finite set  $C$  can be encoded with  $\log_2 |C|$  bits (by numbering the elements)
- ▶ each observation  $\mathbf{X}$  fixes one bit of the solution

# Generalization

## Concept learning is too limited

- ▶ no noise
- ▶ fixed loss function

## Agnostic PAC learnability

A class of models  $\mathcal{G}$  (functions from  $\mathcal{X}$  to  $\mathcal{Y}$ ) is PAC-learnable with respect to a loss function  $l$  if there is an algorithm  $\mathcal{A}$  and a function  $\mathcal{N}_{\mathcal{G}}$  from  $[0, 1]^2$  to  $\mathbb{N}$  such that: for any  $1 > \epsilon > 0$  and any  $1 > \delta > 0$ , for any distribution  $D$  on  $\mathcal{X} \times \mathcal{Y}$  if  $N \geq \mathcal{N}_{\mathcal{G}}(\epsilon, \delta)$  then

$$\mathbb{P}_{\mathcal{D} \sim D^N} \{R_l(\mathcal{A}(\mathcal{D})) \leq R_{l, \mathcal{G}}^* + \epsilon\} \geq 1 - \delta$$

## Main questions

- ▶ does ERM provide agnostic PAC learnability?
- ▶ does that apply to infinite classes of models?

# Uniform approximation

## Lemma

Controlling the ERM can be done by ensuring the empirical risk is *uniformly* a good approximation of the true risk:

$$R_I(g_{ERM,I,\mathcal{G},\mathcal{D}}) - R_{I,\mathcal{G}}^* \leq 2 \sup_{g \in \mathcal{G}} |R_I(g) - \hat{R}_I(g, \mathcal{D})|$$



# Uniform approximation

## Lemma

Controlling the ERM can be done by ensuring the empirical risk is *uniformly* a good approximation of the true risk:

$$R_I(g_{ERM,I,\mathcal{G},\mathcal{D}}) - R_{I,\mathcal{G}}^* \leq 2 \sup_{g \in \mathcal{G}} \left| R_I(g) - \widehat{R}_I(g, \mathcal{D}) \right|$$

## Proof.

for any  $g \in \mathcal{G}$ , we have

$$\begin{aligned} R_I(g_{ERM,I,\mathcal{G},\mathcal{D}}) - R_I(g) &= R_I(g_{ERM,I,\mathcal{G},\mathcal{D}}) - \widehat{R}_I(g_{ERM,I,\mathcal{G},\mathcal{D}}, \mathcal{D}) + \widehat{R}_I(g_{ERM,I,\mathcal{G},\mathcal{D}}, \mathcal{D}) - R_I(g), \\ &\leq R_I(g_{ERM,I,\mathcal{G},\mathcal{D}}) - \widehat{R}_I(g_{ERM,I,\mathcal{G},\mathcal{D}}, \mathcal{D}) + \widehat{R}_I(g, \mathcal{D}) - R_I(g), \\ &\leq \left| R_I(g_{ERM,I,\mathcal{G},\mathcal{D}}) - \widehat{R}_I(g_{ERM,I,\mathcal{G},\mathcal{D}}, \mathcal{D}) \right| + \left| \widehat{R}_I(g, \mathcal{D}) - R_I(g) \right|, \\ &\leq 2 \sup_{g \in \mathcal{G}} \left| R_I(g) - \widehat{R}_I(g, \mathcal{D}) \right|, \end{aligned}$$

which leads to the conclusion. □

## Theorem

If  $|\mathcal{G}| < \infty$  and if  $l \in [a, b]$  then when  $N \geq \left\lceil \frac{\log\left(\frac{2|\mathcal{G}|}{\delta}\right)(b-a)^2}{2\epsilon^2} \right\rceil$

$$\mathbb{P}_{\mathcal{D} \sim \mathcal{D}_{\mathcal{X}}^N} \left\{ \sup_{g \in \mathcal{G}} \left| R_l(g) - \widehat{R}_l(g, \mathcal{D}) \right| \geq \epsilon \right\} \leq 1 - \delta$$

## Proof.

very rough sketch

1. we use the union technique to focus on a single model  $g$
2. then we use the Hoeffding inequality to bound the difference between an empirical average and an expectation. In our context it says

$$\mathbb{P}_{\mathcal{D} \sim \mathcal{D}_{\mathcal{X}}^N} \left\{ \left| R_l(g) - \widehat{R}_l(g, \mathcal{D}) \right| \geq \epsilon \right\} \leq 2 \exp \left( -2N \frac{\epsilon^2}{(b-a)^2} \right)$$

3. the conclusion is obtained as in the simple case of concept learning



## Theorem

If  $|\mathcal{G}| < \infty$  and if  $I \in [0, 1]$  the ERM provides agnostic PAC-learnability

with  $\mathcal{N}_{\mathcal{G}}(\epsilon, \delta) = \left\lceil \frac{2 \log\left(\frac{2|\mathcal{G}|}{\delta}\right)(b-a)^2}{\epsilon^2} \right\rceil$

## Discussion

- ▶ obvious consequence of the uniform approximation result
- ▶ the limitation  $I \in [a, b]$  can be lifted but only asymptotically
- ▶ the dependency of the data size to the quality (i.e. to  $\epsilon$ ) is far less satisfactory than in the simple case: this is a consequence of allowing noise

## Restriction

- ▶ we keep the noise but move back to a simple case
- ▶  $\mathcal{Y} = \{0, 1\}$  and  $l = l_b$

## Growth function

- ▶ if  $\{v_1, \dots, v_m\}$  is a finite subset of  $\mathcal{X}$

$$\mathcal{G}_{\{v_1, \dots, v_m\}} = \{(g(v_1), \dots, g(v_m)) \mid g \in \mathcal{G}\} \subset \{0, 1\}^m$$

- ▶ the growth function of  $\mathcal{G}$  is

$$S_{\mathcal{G}}(m) = \sup_{\{v_1, \dots, v_m\} \subset \mathcal{X}} |\mathcal{G}_{\{v_1, \dots, v_m\}}|$$

## Going back to finite things

- ▶  $|\mathcal{G}_{\{v_1, \dots, v_m\}}|$  gives the number of models as seen by the inputs  $\{v_1, \dots, v_m\}$
- ▶ it corresponds to the number of possible classification decisions (a.k.a. binary labelling) of those inputs
- ▶ the growth function corresponds to the worst case analysis: the set of inputs that can be labelled in the largest number of different ways

## Vocabulary

- ▶ if  $|\mathcal{G}_{\{v_1, \dots, v_m\}}| = 2^m$  then  $\{v_1, \dots, v_m\}$  is said to be *shattered* by  $\mathcal{G}$
- ▶  $\mathcal{S}_{\mathcal{G}}(m)$  is the  $m$ -th shatter coefficient of  $\mathcal{G}$

## Theorem

For any  $1 > \epsilon > 0$  and any  $1 > \delta > 0$  and for any distribution  $D$

$$\mathbb{P}_{\mathcal{D} \sim D_x^N} \left\{ \sup_{g \in \mathcal{G}} \left| R_{l_b}(g) - \widehat{R}_{l_b}(g, \mathcal{D}) \right| \geq \frac{4 + \sqrt{\log(\mathcal{S}_{\mathcal{G}}(2N))}}{\delta \sqrt{2N}} \right\} \leq 1 - \delta$$

## Consequences

- ▶ strong link between the growth function and uniform approximation
- ▶ useful only if  $\frac{\log(\mathcal{S}_{\mathcal{G}}(2m))}{m}$  goes to zero when  $m \rightarrow \infty$
- ▶ if  $\mathcal{G}$  shatters sets of arbitrary sizes  $\log(\mathcal{S}_{\mathcal{G}}(2m)) = 2m \log 2$

# Vapnik Chervonenkis dimension

## VC-dimension

$$VCdim(\mathcal{G}) = \sup \{m \in \mathbb{N} \mid \mathcal{S}_{\mathcal{G}}(m) = 2^m\}$$

## Characterization

$VCdim(\mathcal{G}) = m$  if and only if

1. there is **a** set of  $m$  points  $\{v_1, \dots, v_m\}$  that is shattered by  $\mathcal{G}$
2. **no** set of  $m + 1$  points  $\{v_1, \dots, v_{m+1}\}$  is shattered by  $\mathcal{G}$

## Lemma (Sauer)

If  $VCdim(\mathcal{G}) < \infty$ , for all  $m$   $\mathcal{S}_{\mathcal{G}}(m) \leq \sum_{k=0}^{VCdim(\mathcal{G})} \binom{m}{k}$ . In particular when  $m \geq VCdim(\mathcal{G})$

$$\mathcal{S}_{\mathcal{G}}(m) \leq \left( \frac{em}{VCdim(\mathcal{G})} \right)^{VCdim(\mathcal{G})}$$

## Consequences

If  $VCdim(\mathcal{G}) = d < \infty$ , for any  $1 > \epsilon > 0$  and any  $1 > \delta > 0$  and for any distribution  $D$ , if  $N \geq d$  then

$$\mathbb{P}_{\mathcal{D} \sim D_x^N} \left\{ \sup_{g \in \mathcal{G}} \left| R_{l_b}(g) - \widehat{R}_{l_b}(g, \mathcal{D}) \right| \geq \frac{4 + \sqrt{\log\left(\frac{2eN}{d}\right)}}{\delta\sqrt{2N}} \right\} \leq 1 - \delta$$

## Learnability

- ▶ a finite VC-dimension ensures agnostic PAC-learnability of the ERM
- ▶ it can be shown that  $\mathcal{N}_{\mathcal{G}}(\epsilon, \delta) = \Theta\left(\frac{VCdim(\mathcal{G}) + \log \frac{1}{\delta}}{\epsilon^2}\right)$



VC-dimension calculation is very difficult! A useful result:

## Theorem

Let  $\mathcal{F}$  be a vector space of functions from  $\mathcal{X}$  to  $\mathbb{R}$  of dimension  $p$ . Let  $\mathcal{G}$  be the class of models given by

$$\mathcal{G} = \{g : \mathcal{X} \rightarrow \{0, 1\} \mid \exists f \in \mathcal{F}, \forall \mathbf{X} \in \mathcal{X} g(\mathbf{X}) = \mathbf{1}_{f(\mathbf{X}) \geq 0}\}.$$

Then  $VCdim(\mathcal{G}) \leq p$ .

# Is a finite VC-dimension needed?

## Theorem

Let  $\mathcal{G}$  be a class of models from  $\mathcal{X}$  to  $\mathcal{Y} = \{0, 1\}$ . Then the following properties are equivalent:

1.  $\mathcal{G}$  is agnostic PAC-learnable with the binary loss  $l_b$
2. ERM provides agnostic PAC-learnable with the binary loss  $l_b$  for  $\mathcal{G}$
3.  $VCdim(\mathcal{G}) < \infty$

## Interpretation

- ▶ learnability in the PAC sense is therefore uniquely characterized by the VC-dimension of the class of models
- ▶ no algorithmic tricks can be used to circumvent this fact!
- ▶ but this applies only to a fix class!

# Beyond binary classification

- ▶ numerous extensions are available
  - ▶ to the regression setting (with quadratic or absolute loss)
  - ▶ to classification with more than two classes
- ▶ refined complexity measures are available
  - ▶ Rademacher complexity
  - ▶ Covering numbers
- ▶ better bounds are also available
  - ▶ in general
  - ▶ in the noise free situation

But the overall message remains the same: learnability is only possible in classes of bounded complexity.

Introduction

PAC learning

**ERM in practice**

## Empirical risk minimization

$$g_{ERM,I,\mathcal{G},\mathcal{D}} = \arg \min_{g \in \mathcal{G}} \widehat{R}_I(g, \mathcal{D})$$

## Implementation?

- ▶ what class  $\mathcal{G}$  should we use?
  - ▶ potential candidates?
  - ▶ how to chose among them?
- ▶ how to implement the minimization part
  - ▶ complexity?
  - ▶ approximate solutions?

## Some examples

- ▶ fixed “basis” models, e.g. for  $\mathcal{Y} = \{-1, 1\}$

$$\mathcal{G} = \left\{ \mathbf{X} \mapsto \text{sign} \left( \sum_{k=1}^K \alpha_k f_k(\mathbf{X}) \right) \right\},$$

where the  $f_k$  are fixed functions from  $\mathcal{X}$  to  $\mathbb{R}$

- ▶ parametric “basis” models

$$\mathcal{G} = \left\{ \mathbf{X} \mapsto \text{sign} \left( \sum_{k=1}^K \alpha_k f_k(\mathbf{w}_k, \mathbf{X}) \right) \right\},$$

where the  $f_k(\mathbf{w}_k, \cdot)$  are fixed functions from  $\mathcal{X}$  to  $\mathbb{R}$  and the  $\mathbf{w}_k$  are parameters that enable tuning the  $f_k$

- ▶ useful also for  $\mathcal{Y} = \mathbb{R}$  (remove the indicator function)

## Linear models

- ▶  $\mathcal{X} = \mathbb{R}^P$
- ▶ the linearity is with respect to  $\alpha$
- ▶ basic model
  - ▶  $f_k(\mathbf{X}) = X_k$
  - ▶  $\sum_{k=1}^P \alpha_k f_k(\mathbf{X}) = \alpha^T \mathbf{X}$
- ▶ general models
  - ▶  $f_k(\mathbf{X})$  can be any polynomial function on  $\mathbb{R}^P$  or more generally a function from  $\mathbb{R}^P$  to  $\mathbb{R}$
  - ▶ e.g.  $f_k(\mathbf{X}) = X_1 X_2^2$ ,  $f_k(\mathbf{X}) = \log X_3$ , etc.

## Nonlinear models

- ▶ Radial Basis Function (RBF) neural networks:

- ▶  $\mathcal{X} = \mathbb{R}^P$

- ▶  $f_k((\beta, \mathbf{w}_k), \mathbf{X}) = \exp(-\beta \|\mathbf{X} - \mathbf{w}_k\|^2)$

- ▶ one hidden layer perceptron:

- ▶  $\mathcal{X} = \mathbb{R}^P$

- ▶  $f_k((\beta, \mathbf{w}_k), \mathbf{X}) = \frac{1}{1 + \exp(-\beta - \mathbf{w}_k^T \mathbf{X})}$

## More complex outputs

- ▶ if  $|\mathcal{Y}| < \infty$ , write  $\mathcal{Y} = \{y_1, \dots, y_L\}$
- ▶ possible class

$$\mathcal{G} = \left\{ \mathbf{X} \mapsto y_{t(\mathbf{X})}, \text{ with } t(\mathbf{X}) = \arg \max_l \left( \exp \left( - \sum_{k=1}^K \alpha_{kl} f_{kl}(\mathbf{w}_{kl}, \mathbf{X}) \right) \right) \right\}$$



## Parametric view

- ▶ previous classes are described by parameters
- ▶ ERM is defined at the model level but can equivalently be considered at the parameter level
- ▶ if e.g.  $\mathcal{G} = \left\{ \mathbf{X} \mapsto \text{sign}\left(\sum_{k=1}^K \alpha_k f_k(\mathbf{X})\right) \right\}$  solving  $\min_{g \in \mathcal{G}} \widehat{R}_b(g, \mathcal{D})$  is equivalent to solving  $\min_{\alpha \in \mathbb{R}^K} \widehat{R}_b(g_\alpha, \mathcal{D})$ , where  $g_\alpha$  is the model associated to  $\alpha$

# (Meta)Parameters

## Parametric view

- ▶ previous classes are described by parameters
- ▶ ERM is defined at the model level but can equivalently be considered at the parameter level
- ▶ if e.g.  $\mathcal{G} = \left\{ \mathbf{X} \mapsto \text{sign}\left(\sum_{k=1}^K \alpha_k f_k(\mathbf{X})\right) \right\}$  solving  $\min_{g \in \mathcal{G}} \widehat{R}_b(g, \mathcal{D})$  is equivalent to solving  $\min_{\alpha \in \mathbb{R}^K} \widehat{R}_b(g_\alpha, \mathcal{D})$ , where  $g_\alpha$  is the model associated to  $\alpha$

## Meta-parameters

- ▶ to avoid confusion, we use the term "meta-parameters" to refer to parameters of the machine learning algorithm
- ▶ in ERM, those are class level parameters ( $\mathcal{G}$  itself):
  - ▶  $K$
  - ▶ the  $f_k$  functions
  - ▶ the parametric form of  $f_k(w_k, \cdot)$

## Standard optimization problem

- ▶ computing  $\arg \min_{g \in \mathcal{G}} \hat{R}_l(g, \mathcal{D})$  is a classical optimization problem
- ▶ no closed-form solution in general
- ▶ ERM relies on standard algorithms: gradient based algorithms if possible, combinatorial optimization tools if needed

## Very different complexities

- ▶ from easy cases: linear models with quadratic loss
- ▶ to NP-hard ones: binary loss even with super simple models

## ERM version of linear regression

- ▶ class of models

$$\mathcal{G} = \left\{ g : \mathbb{R}^P \rightarrow \mathbb{R} \mid \exists(\beta_0, \boldsymbol{\beta}), \forall \mathbf{X} \in \mathbb{R}^P \ g_{\beta_0, \boldsymbol{\beta}}(\mathbf{x}) = \beta_0 + \boldsymbol{\beta}^T \mathbf{x} \right\}$$

- ▶ loss:  $l(p, t) = (p - t)^2$

- ▶ empirical risk:  $\widehat{R}_l(g_{\beta_0, \boldsymbol{\beta}}, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^N \left( Y_i - \beta_0 - \boldsymbol{\beta}^T \mathbf{X}_i \right)^2$

- ▶ standard solution  $(\beta_0^*, \boldsymbol{\beta}^*)^T = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \mathbb{Y}$  with

$$\mathbb{X} = \begin{pmatrix} 1 & \mathbf{X}_1^T \\ \dots & \dots \\ 1 & \mathbf{X}_N^T \end{pmatrix} \quad \mathbb{Y} = \begin{pmatrix} Y_1^T \\ \dots \\ Y_N^T \end{pmatrix}$$

- ▶ computational cost in  $\Theta(NP^2)$

## Linear model with binary loss

- ▶ class of models

$$\mathcal{G} = \left\{ g : \mathbb{R}^P \rightarrow \{0, 1\} \mid \exists(\beta_0, \beta), \forall \mathbf{x} \in \mathbb{R}^P \ g_{\beta_0, \beta}(\mathbf{x}) = \text{sign}(\beta_0 + \beta^T \mathbf{x}) \right\}$$

- ▶ loss:  $l_b(p, t) = \mathbf{1}_{p \neq t}$
- ▶ empirical risk: misclassification rate
- ▶ in this context ERM is NP-hard and tight approximations are also NP-hard
- ▶ notice that the input dimension is the source of complexity

## Noise

- ▶ if the optimal model makes zero error, then ERM is polynomial!
- ▶ complexity comes from both noise and the binary loss

## Smooth functions

- ▶  $\mathcal{Y} = \mathbb{R}$
- ▶ parametric case  $\mathcal{G} = \{\mathbf{X} \mapsto F(\mathbf{w}, \mathbf{X})\}$
- ▶ assume the loss function  $l$  and the models in the class  $\mathcal{G}$  are differentiable (can be extended to subgradients)
- ▶ gradient of the empirical loss

$$\nabla_{\mathbf{w}} \hat{R}_l(\mathbf{w}, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^N \frac{\partial l}{\partial \rho}(F(\mathbf{w}, \mathbf{X}_i), Y_i) \nabla_{\mathbf{w}} F(\mathbf{w}, \mathbf{X}_i)$$

- ▶ ERM through standard gradient based algorithms, such as gradient descent

$$\mathbf{w}^t = \mathbf{w}^{t-1} - \gamma^t \nabla_{\mathbf{w}} \hat{R}_l(\mathbf{w}^{t-1}, \mathcal{D})$$

## Finite sum

- ▶ leverage the structure of  $\widehat{R}_l(\mathbf{w}, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^N l(F(\mathbf{w}, \mathbf{X}_i), Y_i)$
- ▶ what about updating according to only one example?
- ▶ stochastic gradient descent
  1. start with a random  $\mathbf{w}^0$
  2. iterate
    - 2.1 select  $i^t$  randomly uniformly in  $\{1, \dots, N\}$
    - 2.2  $\mathbf{w}^t = \mathbf{w}^{t-1} - \gamma^t \frac{\partial l}{\partial \mathbf{p}}(F(\mathbf{w}^{t-1}, \mathbf{X}_{i^t}), Y_{i^t}) \nabla_{\mathbf{w}} F(\mathbf{w}^{t-1}, \mathbf{X}_{i^t})$
- ▶ practical tips:
  - ▶ use the Polyak-Ruppert averaging:  $\bar{\mathbf{w}}^m = \frac{1}{m} \sum_{t=0}^{m-1} \mathbf{w}^t$
  - ▶  $\gamma^t = (\gamma^0 + t)^{-\kappa}$ ,  $\kappa \in ]0.5, 1]$ ,  $\gamma^0 \geq 0$
  - ▶ numerous acceleration techniques such as momentum ("averaged" gradients)

## Heuristics

- ▶ numerous heuristics have been proposed for ERM and related problems
- ▶ one of the main tools is alternate/separate optimization: optimize with respect to some parameters while holding the other constants

## Radial basis function

- ▶  $\mathcal{G} = \left\{ \mathbf{X} \mapsto \sum_{k=1}^K \alpha_k \exp(-\beta \|\mathbf{X} - \mathbf{w}_k\|^2) \right\}$
- ▶ set  $\beta$  heuristically, e.g. to the inverse of the smallest squared distance between two  $\mathbf{X}$  in the data set
- ▶ set the  $\mathbf{w}_k$  via a unsupervised method applied to the  $(\mathbf{X}_i)_{1 \leq i \leq N}$  only (for instance the K-means algorithm)
- ▶ consider  $\beta$  and the  $\mathbf{w}_k$  fixed and apply standard ERM to the  $\alpha_k$ , e.g. linear regression if the loss function is quadratic and  $\mathcal{Y} = \mathbb{R}$



## Maximum likelihood

- ▶ the classical way of estimating parameters in statistics consists in maximizing the likelihood function
- ▶ this is empirical risk minimization in disguise
- ▶ learnability results apply!

## Linear regression

- ▶ in linear regression one assumes that the following conditional distribution:  $Y_i | \mathbf{X}_i = \mathbf{x} \sim \mathcal{N}(\beta_0 + \beta^T \mathbf{x}, \sigma^2)$
- ▶ the MLE estimate of  $\beta_0$  and  $\beta$  is obtained as

$$(\widehat{\beta_0}, \widehat{\beta})_{MLE} = \arg \min_{\beta_0, \beta} \sum_{i=1}^N (Y_i - \beta_0 - \beta^T \mathbf{x}_i)^2$$

- ▶ MLE=ERM here

## MLE

- ▶ in logistic regression one assumes that (with  $\mathcal{Y} = \{0, 1\}$ )

$$\mathbb{P}(Y_i = 1 | \mathbf{X}_i = \mathbf{x}) = \frac{1}{1 + \exp(-\beta_0 - \boldsymbol{\beta}^T \mathbf{x})} = h_{\beta_0, \boldsymbol{\beta}}(\mathbf{x})$$

- ▶ the MLE estimate is obtained by maximizing over  $(\beta_0, \boldsymbol{\beta})$  the following function

$$\sum_{i=1}^N (Y_i \log h_{\beta_0, \boldsymbol{\beta}}(\mathbf{X}_i) + (1 - Y_i) \log(1 - h_{\beta_0, \boldsymbol{\beta}}(\mathbf{X}_i)))$$

## Machine learning view

- ▶ assume  $\mathcal{Y} = \mathbb{R}$
- ▶ use again the class of linear models
- ▶ the loss is given by

$$l(p, t) = t \log(1 + \exp(-p)) + (1 - t) \log(1 + \exp(p))$$

## Extended ML framework

- ▶ the standard ML approach consists in looking for  $g$  in the set of functions from  $\mathcal{X}$  to  $\mathcal{Y}$
- ▶ the logistic regression does not model directly the link between  $\mathbf{X}$  and  $\mathbf{Y}$  but rather a probabilistic link
- ▶ the ML version is based on a new ML paradigm where the loss function is defined on  $\mathcal{Y}' \times \mathcal{Y}$  and  $g$  is a function from  $\mathcal{X}$  to  $\mathcal{Y}'$

## Simplifying the ERM

- ▶ ERM with binary loss is complex: non convex loss with no meaningful gradient
- ▶ goal: keep the binary decision but remove the binary loss
- ▶ solution:
  - ▶ ask to the model a score rather than a binary decision
  - ▶ build a loss function that compares a score to the binary decision
  - ▶ use a decision technique consistent with the score
  - ▶ generally simpler to formulate with  $\mathcal{Y} = \{-1, 1\}$  and the sign function
- ▶ this a relaxation as we do not look anymore for a crisp 0/1 solution but for a continuous one

## Numerous possible solutions

$\mathcal{Y} = \{-1, 1\}$  with decision based on  $\text{sign}(p)$

- ▶ logistic loss:  $l_{logi}(p, t) = \log(1 + \exp(-pt))$
- ▶ perceptron loss:  $l_{per}(p, t) = \max(0, -pt)$
- ▶ hinge loss (Support Vector Machine):  $l_{hinge}(p, t) = \max(0, 1 - pt)$
- ▶ exponential loss (Ada boost):  $l_{exp}(p, t) = \exp(-pt)$
- ▶ quadratic loss:  $l_2(p, t) = (p - t)^2$

## This is not ERM anymore!

- ▶  $\text{sign} \circ g$  is a model in the original sense (a function from  $\mathcal{X}$  to  $\mathcal{Y}$ )
- ▶ but  $I_{\text{relax}}(g(\mathbf{X}), Y) \neq I_b(\text{sign}(g(\mathbf{X})), \mathbf{Y})$
- ▶ **surrogate loss minimization**
  - ▶ some theoretical results are available
  - ▶ but in general no guarantee to find the best model with a surrogate loss

## Are there other reasons to avoid ERM?

# Negative results (binary classification)

$$VCdim(\mathcal{G}) = \infty$$

- ▶ consider a fixed ML algorithm that picks up a classifier in  $\mathcal{G}$  with infinite VC dimension (using whatever criterion)
- ▶ for all  $\epsilon > 0$  and all  $N$ , there is  $D$  such that  $R_{\mathcal{G}}^* = 0$  and

$$\mathbb{E}_{\mathcal{D} \sim D^N}(R(g_D)) \geq \frac{1}{2e} - \epsilon$$

$$VCdim(\mathcal{G}) < \infty$$

- ▶ for all  $\epsilon > 0$ , there is  $D$  such that

$$R_{\mathcal{G}}^* - R^* > \frac{1}{2} - \epsilon$$

## Summary

The empirical risk minimization framework seems appealing at first but it has several limitations

- ▶ the binary loss is associated to practical difficulties:
  - ▶ implementation is difficult (because of the lack of smoothness)
  - ▶ complexity can be high in the case of noisy data
- ▶ learnability is guaranteed but
  - ▶ only for model classes with finite VC dimension
  - ▶ which are strictly limited!

## Beyond ERM

- ▶ surrogate loss function
- ▶ data adaptive model class





This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

<http://creativecommons.org/licenses/by-sa/4.0/>

Last modification: 2018-02-05

By: Fabrice Rossi (Fabrice.Rossi@apiacoa.org)

Git hash: 8100ae41a33a8993be5363018b56bfce56d63e01