Empirical Risk Minimization

Fabrice Rossi

CEREMADE
Université Paris Dauphine

2021

Outline

Introduction

Model classes

Optimization

ERM and statistics

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 $R_l^* = \inf_g R_l(g)$

Supervised learning

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 minimization

Empirical risk

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

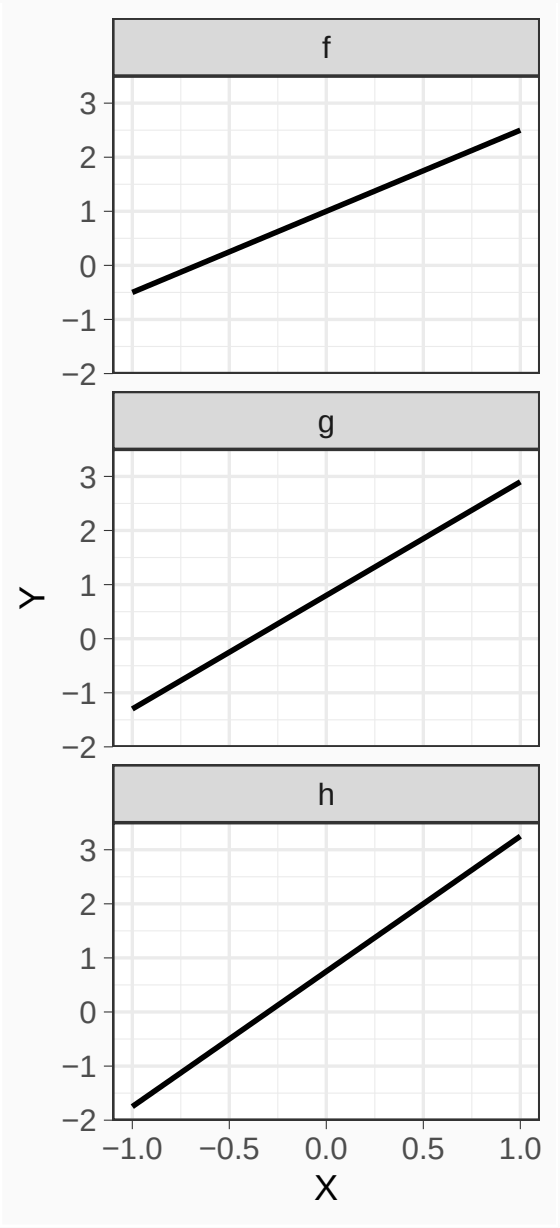
ERM algorithm

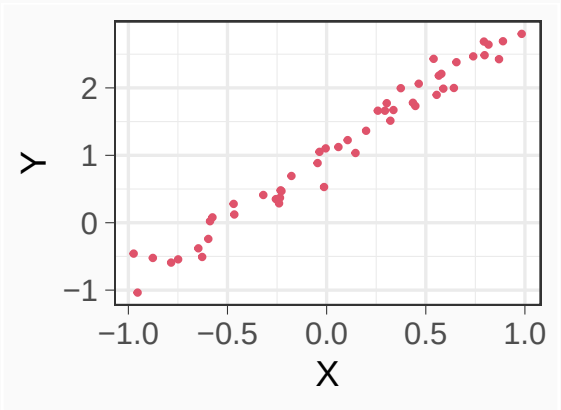
- choose a class of functions $\mathcal{G}$ from $\mathcal{X}$ to $\mathcal{Y}$
- define

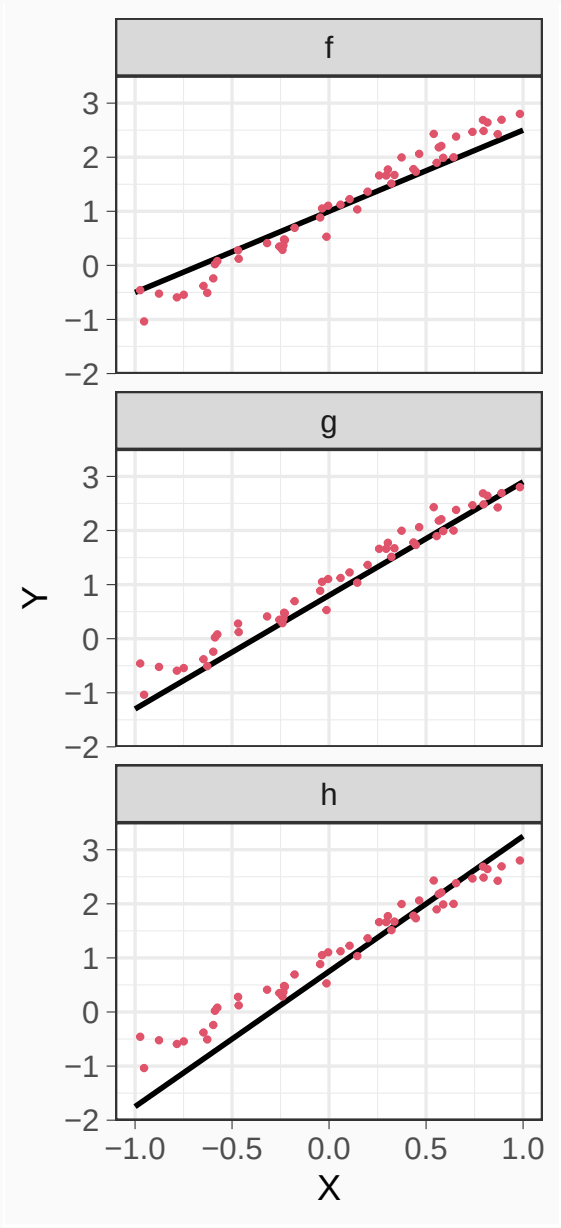
$$g_{ERM,l,\mathcal{G},\mathcal{D}} = \arg\min_{g \in \mathcal{G}} \widehat{R}_l(g, \mathcal{D})$$

- is ERM a “good” machine learning algorithm?

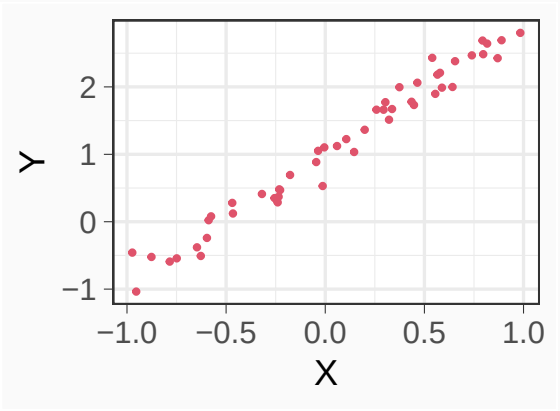
Empirical risk minimization

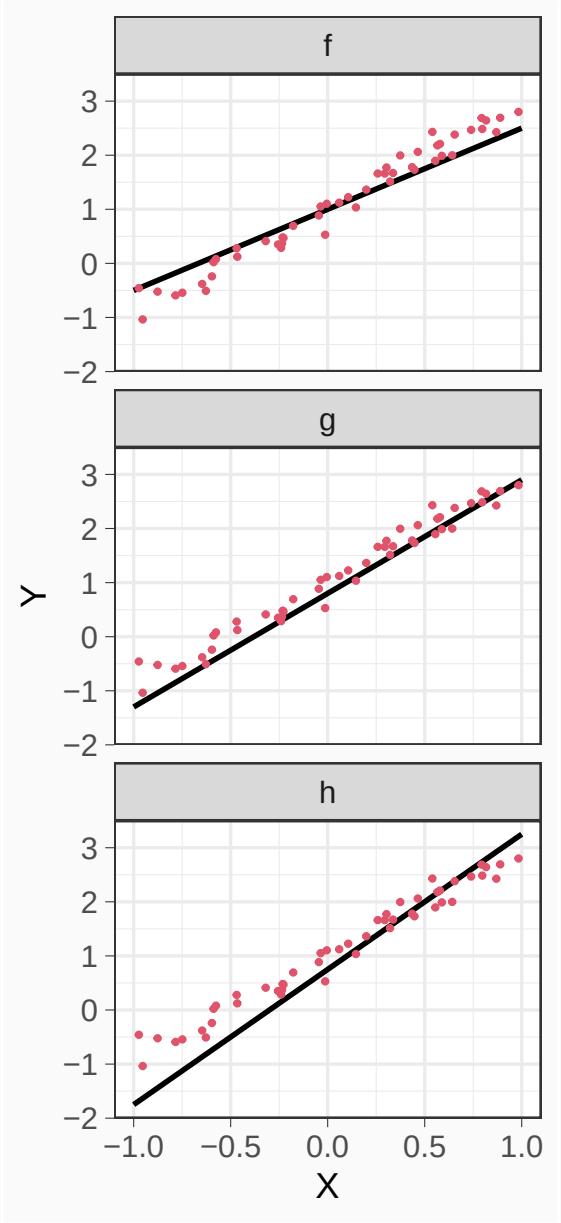






Empirical risk minimization



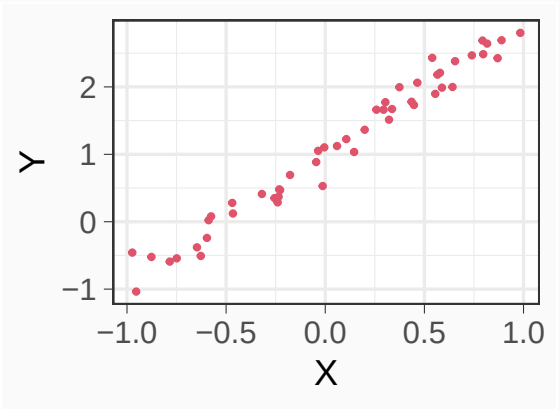


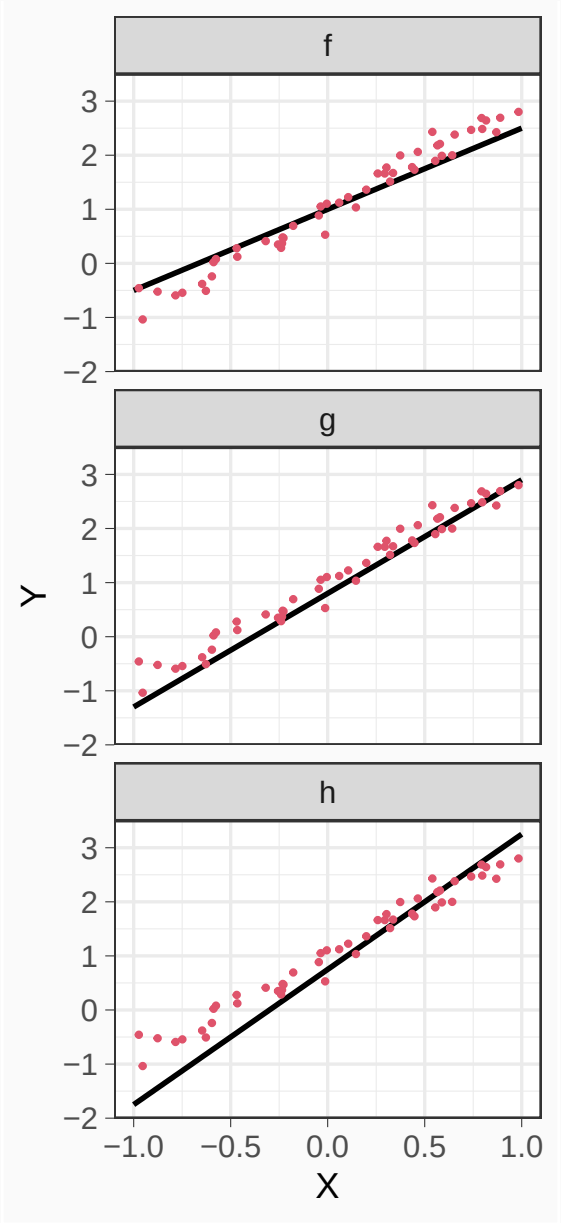
$$\widehat{R}_{l_2}(f, \mathcal{D}) = 0.086$$

$$\widehat{R}_{l_2}(g, \mathcal{D}) = 0.071$$

$$\widehat{R}_{l_2}(h, \mathcal{D}) = 0.167$$

Empirical risk minimization



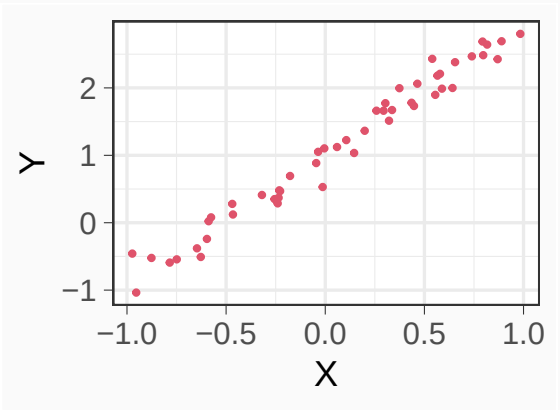


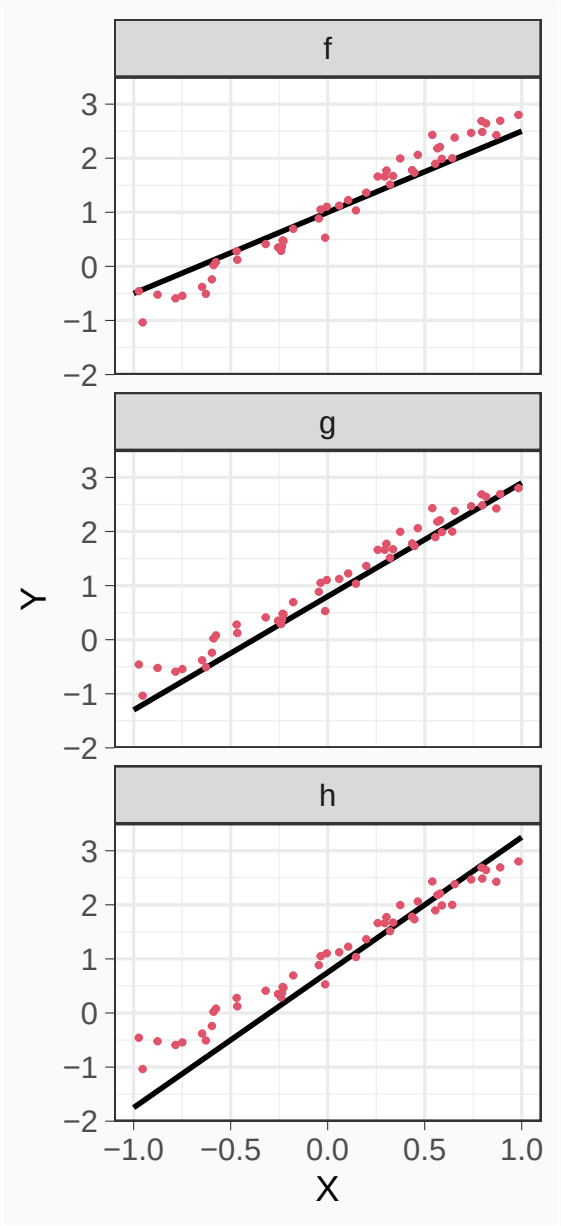
$$\widehat{R}_{l_2}(f, \mathcal{D}) = 0.086$$

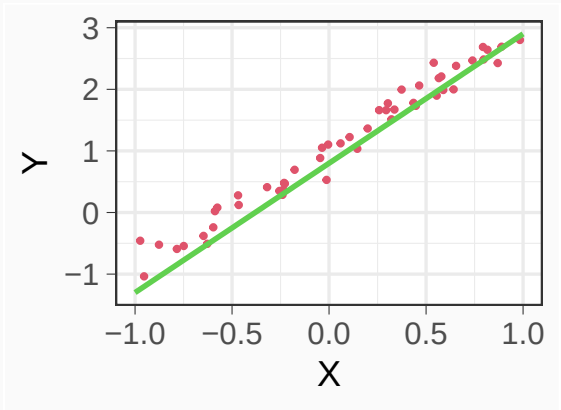
$$\widehat{R}_{l_2}(g, \mathcal{D}) = 0.071$$

$$\widehat{R}_{l_2}(h, \mathcal{D}) = 0.167$$

Empirical risk minimization





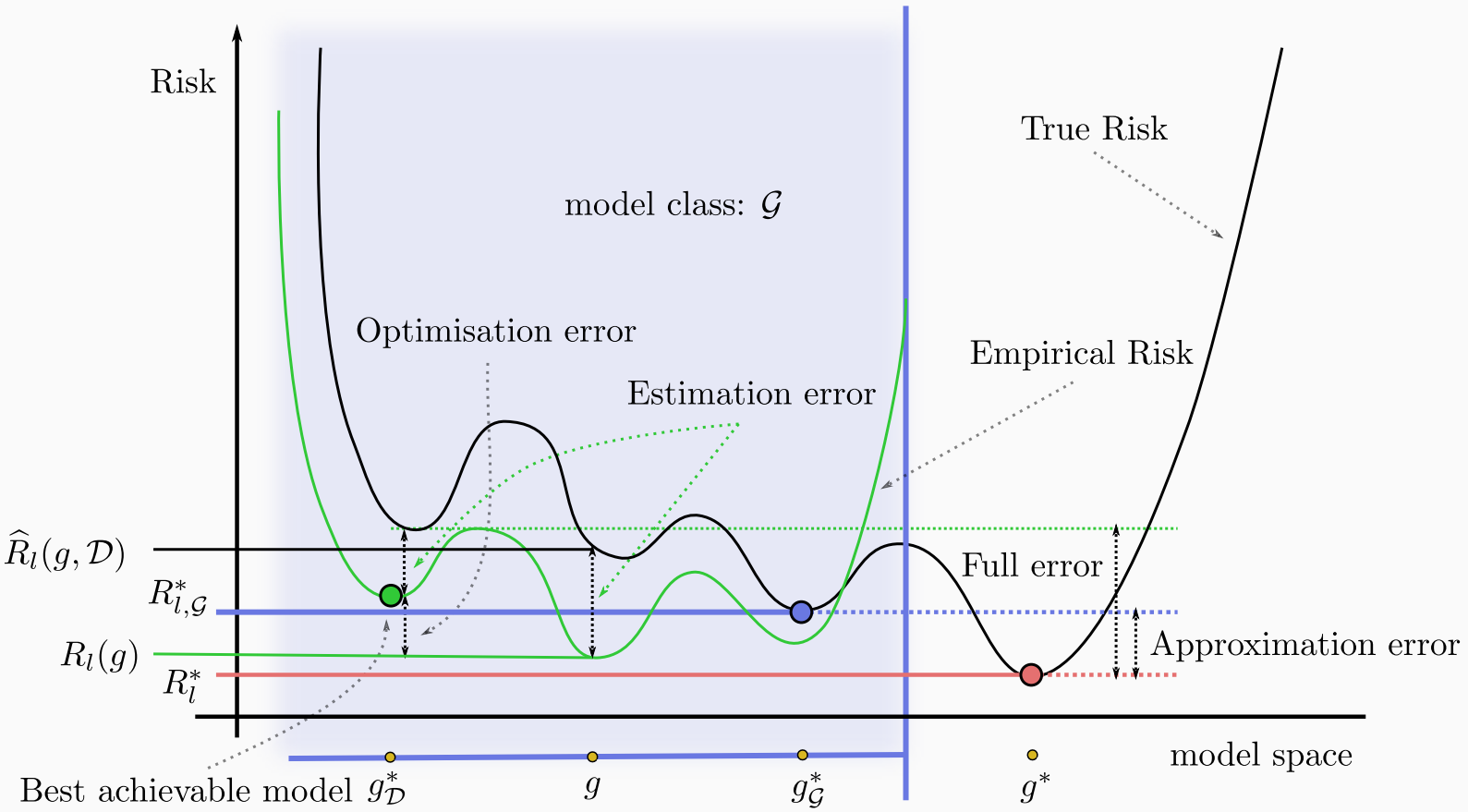


Theoretical questions

Learning theory

1. Optimization quality
   - 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. Estimation quality
   - 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. Approximation quality
   - can we bound $R^*_{l,\mathcal{G}} - R^*_l$?
   - in a way that is compatible with estimation?

Learning theory



ERM in practice

Empirical risk minimization

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

Implementation?

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

Outline

Introduction

Model classes

Optimization

ERM and statistics

A simple example

Linear models

- for $\mathcal{Y} = \mathbb{R}$ and $\mathcal{X} = \mathbb{R}^P$
- chose some fixed functions ϕ_j from $\mathbb{R}^P$ to $\mathbb{R}$, e.g.
  - $\phi_j(\mathbf{X}) = \mathbf{x}_j$
  - $\phi_{j,l}(\mathbf{X}) = \mathbf{x}_j \mathbf{x}_l$
  - $\phi_{\sin,j}(\mathbf{X}) = \sin(\mathbf{x}_j)$
- write $g(\mathbf{X}) = \sum_{j=1}^{K} \beta_j \phi_j(\mathbf{X})$ for well chosen $\beta^T = (\beta_j)_{1 \leq j \leq K}$ (data dependant)
- model classes

$$\mathcal{G}_{(\phi_j)_{1 \leq j \leq K}} = \left\{ g : \mathbb{R}^P \to \mathbb{R} \middle| \exists \beta^T = (\beta_j)_{1 \leq j \leq K},\ g(\mathbf{X}) = \sum_{j=1}^{K} \beta_j \phi_j(\mathbf{X}) \right\}$$

Model versus class

Possible confusion

- $g : \mathbf{X} \mapsto \sum_{j=1}^{K} \beta_j \phi_j(\mathbf{X})$ is a model, described by $\beta^T = (\beta_j)_{1 \leq j \leq K}$ and by $(\phi_j)_{1 \leq j \leq K}$
- β^T characterises the model in the class, among other models of the class
- $(\phi_j)_{1 \leq j \leq K}$ characterise the class among all classes of the same form
- parameters (model) versus meta-parameters (class)

A simple example

Least-Squares

- quadratic loss $l_2(p,t) = (p-t)^2$
- empirical risk (parameter oriented point of view)

$$\widehat{R}_l((\beta_j)_{1\leq j\leq K}, (\phi_j)_{1\leq j\leq K}, \mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{Y}_i - \sum_{j=1}^{K}\beta_j\phi_j(\mathbf{X}_i)\right)^2$$

- ERM $\Leftrightarrow$ least squares linear regression

A simple example

Well known (partial) solution

- $\widehat{R}_l((\beta_j)_{1\leq j\leq K}, (\phi_j)_{1\leq j\leq K}, \mathcal{D})$ is easy to optimize over $\beta^T = (\beta_j)_{1\leq j\leq K}$

$$\beta^* = \left(\mathbb{X}^T\mathbb{X}\right)^{-1}\mathbb{X}^T\mathbb{Y}$$

with

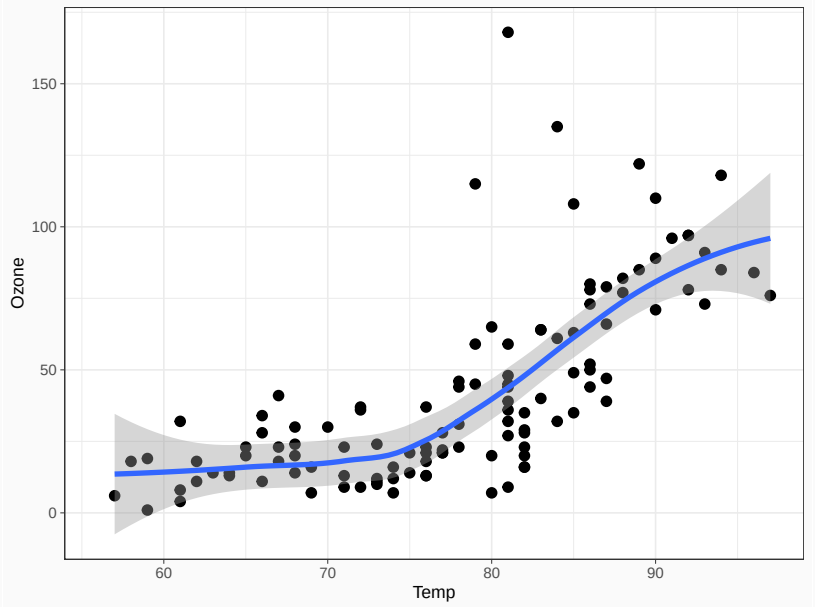
$$\mathbb{X} = \begin{pmatrix} \phi_1(\mathbf{X}_1) & \dots & \phi_K(\mathbf{X}_1) \\ \dots & \dots & \\ \phi_1(\mathbf{X}_N) & \dots & \phi_K(\mathbf{X}_N) \end{pmatrix} \quad \mathbb{Y} = \begin{pmatrix} Y_1 \\ \dots \\ Y_N \end{pmatrix}$$

- but no closed form solution over $(\phi_j)_{1\leq j\leq K}$

Example

Air quality

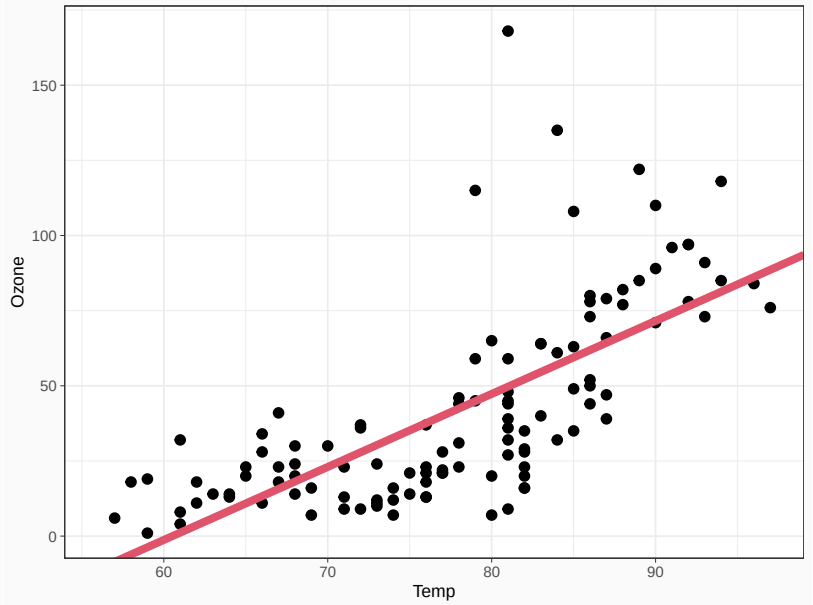
- Mean ozone concentration from May to September 1973 (New York)
- Meteorologic measures (temperature, solar radiation, etc.)



Example

Air quality

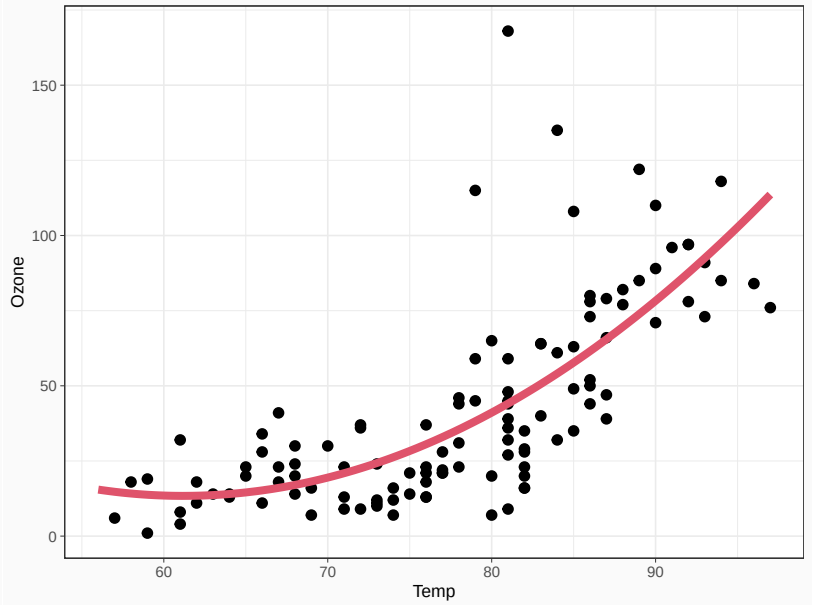
- Ozone as a function of temperature
- linear model



Example

Air quality

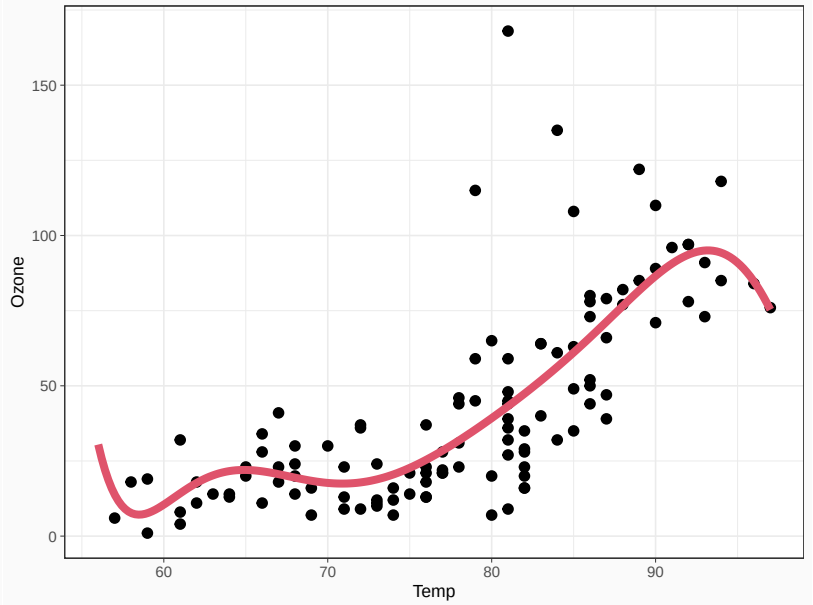
- Ozone as a function of temperature
- linear model
- quadratic model



Example

Air quality

- Ozone as a function of temperature
- linear model
- quadratic model
- order 10 model



Empirical risk minimisation

- two quite different optimisation problems
- best model in the class
  - standard optimisation problem (e.g., finite number of parameters)
  - sometimes with simple solutions
- best class
  - complex optimisation problem (e.g., finding functions in a functional space)
  - somewhat ill posed problem, e.g.
    - with $\mathcal{X} = \mathbb{R}^P$, one can chose N independent functions $(\phi_j)_{1 \leq j \leq N}$ (for instance polynomial functions)
    - then in general $\min \widehat{R}_l((\beta_j)_{1 \leq j \leq N}, (\phi_j)_{1 \leq j \leq N}, \mathcal{D}) = 0$

Model classes

Nonlinear models

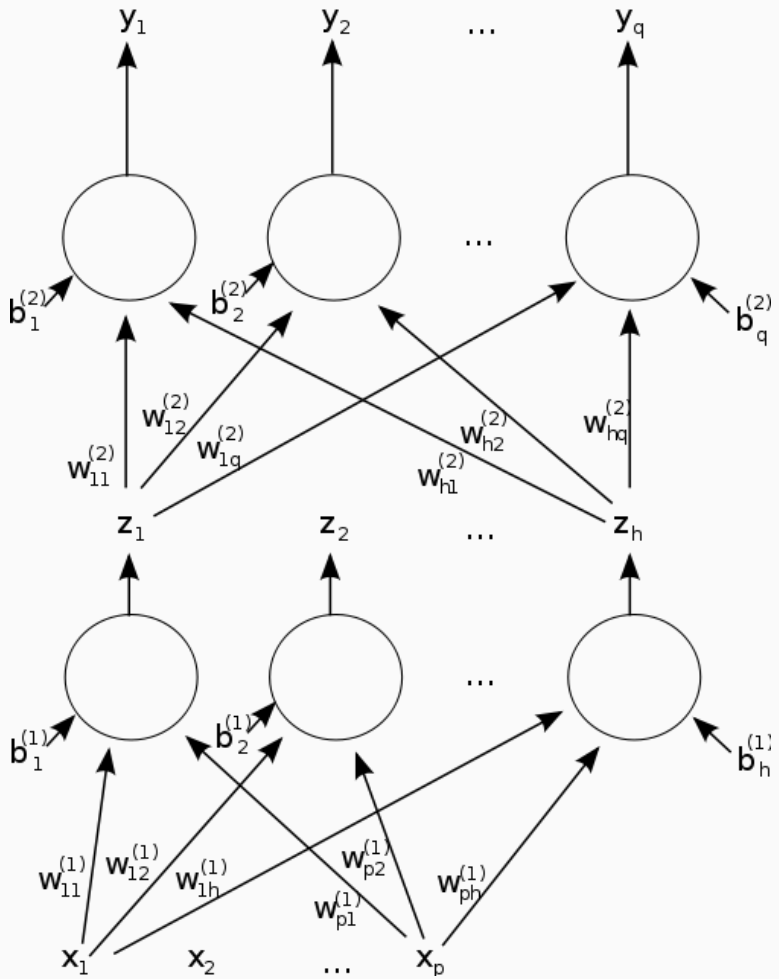
- for $\mathcal{Y} = \mathbb{R}$
- chose some fixed functions ψ_j from $\mathbb{R}^P \times W_j$ to $\mathbb{R}$
- model classes

$$\mathcal{G}_{(\psi_j)_{1 \leq j \leq K}} = \left\{ g \middle| \exists \beta, (w_1, \ldots, w_K),\ g(\mathbf{X}) = \sum_{j=1}^{K} \beta_j \psi_j(\mathbf{X}, w_j) \right\}$$

- examples
  - $\psi_j(\mathbf{X}, w_j) = \exp\left(-\beta \|\mathbf{X} - w_j\|^2\right)$ (Radial basis functions)
  - e.g. one hidden layer perceptron, given a nonlinear function T such as $T(x) = \max(0, x)$

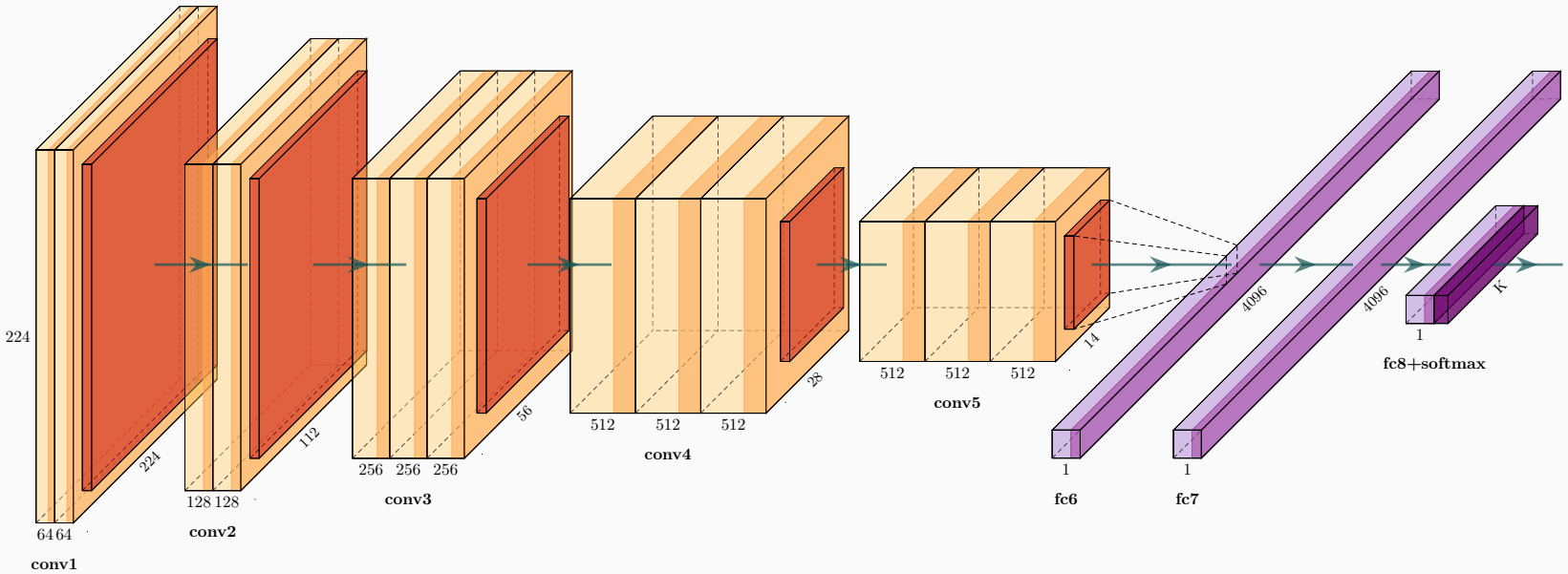
$$\psi_j(\mathbf{X}, w_j) = T(w_j^T \mathbf{X})$$

Artifical neural networks



https://commons.wikimedia.org/wiki/File:Two_layer_ann.svg

Artifical neural networks



https://github.com/HarisIqbal88/PlotNeuralNet

More complex inputs

Nominal data

- $|\mathcal{X}| < \infty$ and no structure
- one hot encoding
  - $\mathcal{X} = \{x_1, \ldots, x_P\}$
  - mapped to $\{0, 1\}^P$
  - x_k translated to $(0, \ldots, 0, 1, 0, \ldots, 0)$ where the 1 is in position k

Dissimilarity data

- dissimilarity space $(\mathcal{X}, d)$
  1. d is a function from $\mathcal{X} \times \mathcal{X}$ to $\mathbb{R}^+$
  2. $\forall \mathbf{X}, \mathbf{X}'$, $d(\mathbf{X}, \mathbf{X}') = d(\mathbf{X}', \mathbf{X})$
  3. $\forall \mathbf{X}, \mathbf{X}'$, $\mathbf{X} \neq \mathbf{X}' \Leftrightarrow d(\mathbf{X}, \mathbf{X}') > 0$
- typical solution: $\psi_j(\mathbf{X}, w_j) = \exp\left(-d(\mathbf{X}, w_j)^k\right)$ (with $k = 1$ or $k = 2$)

More complex outputs

$|\mathcal{Y}| = 2$

- write $\mathcal{Y} = \{-1, 1\}$
- assume given a class $\mathcal{F}$ of functions from $\mathcal{X}$ to $\mathbb{R}$
- use (score based approach)

$$\mathcal{G} = \left\{ g \middle| \exists f \in \mathcal{F}, g(\mathbf{X}) = \text{sign}(f(\mathbf{X})) \right\}$$

More complex outputs

$2 < |\mathcal{Y}| < \infty$

- write $\mathcal{Y} = \{1, \ldots, L\}$
- assume given a class $\mathcal{F}$ of functions from $\mathcal{X}$ to $\mathbb{R}^L$
- define a decision function from $\mathbb{R}^L$ to $\mathcal{Y}$ for instance
  - $h_{\max}(s_1, \ldots, s_L) = \arg\max_l s_l$
  - $h_{\exp \max}(s_1, \ldots, s_L) = \arg\max_l \frac{\exp s_l}{\sum_{k=1}^{L} \exp s_k}$
- possible class

$$\mathcal{G} = \left\{ g \middle| \exists f \in \mathcal{F}, g(\mathbf{X}) = h(f(\mathbf{X})) \right\}$$

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. $\mathcal{G} = \left\{ \mathbf{X} \mapsto \text{sign}(\sum_{j=1}^{K} \beta_j \phi_j(\mathbf{X})) \right\}$ solving $\min_{g \in \mathcal{G}} \widehat{R}_{l_b}(g, \mathcal{D})$ is equivalent to solving $\min_{\boldsymbol{\beta} \in \mathbb{R}^K} \widehat{R}_{l_b}(g_{\boldsymbol{\beta}}, \mathcal{D})$, where $g_{\boldsymbol{\beta}}$ is the model associated to $\boldsymbol{\beta}$

Meta-Parameters

Class level parameters

- specify the class of models
- interpolation regime: well chosen meta-parameters allow the class to interpolate any (reasonable) data set of a given size
- cannot be chosen by ERM (overfitting!)
- examples:
  - K
  - the ϕ_j functions
  - the parametric form of $\psi_j(., w_j)$
  - etc.

In summary

A possible generic ML algorithm

- chose a set of model classes
  - models in a class should have roughly the same interpolation capabilities
  - classes are indexed by meta-parameters: $\mathcal{G}_\theta$ for $\theta \in \Theta$
- chose a way to select $g \in \mathcal{G}_\theta$ for any θ using a data set $\mathcal{D}$, e.g.

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

- chose the best class using another independent data set $\mathcal{D}'$, e.g.

$$\theta^* = \arg\min_{\theta \in \Theta} \widehat{R}_I(g_{ERM,\mathcal{D},\theta}, \mathcal{D}')$$

- compute the best model, e.g.

$$g^* = \arg\min_{g \in \mathcal{G}_{\theta^*}} \widehat{R}_I(g, \mathcal{D} \cup \mathcal{D}')$$

Outline

Introduction

Model classes

Optimization

ERM and statistics

ERM and optimization

Standard optimization problem

- computing $\arg\min_{g \in \mathcal{G}} \widehat{R}_l(g, \mathcal{D})$ 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

Linear classification

Linear model with binary loss

- class of models

$$\mathcal{G} = \left\{ g : \mathbb{R}^P \to \{-1, 1\} \mid \exists(\beta_0, \boldsymbol{\beta}), \forall \mathbf{X} \in \mathbb{R}^P \; g_{\beta_0, \boldsymbol{\beta}}(\mathbf{x}) = \mathsf{sign}(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}) \right\}$$

- 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

Gradient descent

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

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

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

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

Stochastic gradient descent

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 $\{1, \ldots, N\}$
     - 2.2 $\mathbf{w}^t = \mathbf{w}^{t-1} - \gamma^t \frac{\partial l}{\partial p}(F(\mathbf{w}^{t-1}, \mathbf{X}_{i^t}), Y_{i^t}) \nabla_{\mathbf{w}} F(\mathbf{w}^{t-1}, \mathbf{X}_{i^t})$
- practical tips:
  - use the Polyak-Ruppert averaging: $\overline{\mathbf{w}}^m = \frac{1}{m}\sum_{t=0}^{m-1} \mathbf{w}^t$
  - $\gamma^t = (\gamma^0 + t)^{-\kappa}$, $\kappa \in]0.5, 1]$, $\gamma^0 \geq 0$
  - numerous acceleration techniques such as momentum (“averaged" gradients)

Ad hoc methods

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

- $\mathcal{G} = \left\{ \mathbf{X} \mapsto \sum_{k=1}^{K} \beta_k \exp\left(-\rho \|\mathbf{X} - \mathbf{w}_k\|^2\right) \right\}$
- set ρ 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 ρ and the $\mathbf{w}_k$ fixed and apply standard ERM to the β_k, e.g. linear regression if the loss function is quadratic and $\mathcal{Y} = \mathbb{R}$

Outline

Introduction

Model classes

Optimization

ERM and statistics

ERM and statistics

Maximum likelihood

- the classical way of estimating parameters in statistics consists in maximizing the likelihood function
- this can be seen as empirical risk minimization

Linear regression

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

$$\widehat{\beta}_{MLE} = \arg\min_{\beta} \sum_{i=1}^{N} \left(Y_i - \beta^T \phi(\mathbf{X}_i) \right)^2$$

- MLE=ERM here

Derivation

- likelihood function

$$\mathcal{L}\left(\beta, \sigma | (Y_i)_{1 \leq i \leq N}\right) = \prod_{i=1}^{N} \left(\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(Y_i - \beta^T \phi(\mathbf{X}_i)\right)^2}{2\sigma^2} \right) \right)$$

- log likelihood

$$\log \mathcal{L}\left(\beta, \sigma | (Y_i)_{1 \leq i \leq N}\right) = -\frac{N}{2} \log 2\pi - N \log \sigma - \sum_{i=1}^{N} \frac{\left(Y_i - \beta^T \phi(\mathbf{X}_i)\right)^2}{2\sigma^2}$$

- maximizing can be done by
  - minimizing $\sum_{i=1}^{N} \left(Y_i - \beta^T \phi(\mathbf{X}_i)\right)^2 \rightarrow \widehat{\beta}_{MLE}$
  - minimizing (over σ) $N \log \sigma + \sum_{i=1}^{N} \frac{\left(Y_i - \widehat{\beta}_{MLE}^T \mathbf{x}_i\right)^2}{2\sigma^2} \rightarrow$ $\widehat{\sigma}^2_{MLE} = \frac{1}{N} \sum_{i=1}^{N} \left(Y_i - \widehat{\beta}_{MLE}^T \phi(\mathbf{X}_i) \right)^2$

General regression

Noise and loss functions

- assume $Y_i = g_\beta(\mathbf{X}_i) + \varepsilon_i$ where ε_i is a i.i.d noise with density q
- likelihood function

$$\mathcal{L}\left(\beta|(Y_i)_{1\leq i\leq N}\right) = \prod_{i=1}^{N} q\left(Y_i - g_\beta(\mathbf{X}_i)\right)$$

- maximum likelihood $\Leftrightarrow$ ERM with the loss

$$l(p, v) = -\log q(p - v)$$

- example: ε Laplace $\Leftrightarrow$ $l(p, v) = |p - v|$
- can be extended to non additive noise

Logistic Regression

MLE

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

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

- the MLE estimate is obtained by maximizing over β the following function

$$-\sum_{i=1}^{N} \log(1 + \exp(-Y_i \beta^T \phi(\mathbf{X}_i)))$$

Derivation

$$\mathcal{L}(\beta|(Y_i)_{1\leq i\leq N}) = \prod_{Y_i=1} \mathbb{P}(Y_i = 1|\mathbf{X}_i = \mathbf{x}) \prod_{Y_i=-1} \mathbb{P}(Y_i = -1|\mathbf{X}_i = \mathbf{x})$$

$$= \prod_{i=1}^{N} (h_\beta(\mathbf{X}_i))^{\frac{Y_i+1}{2}} (1 - h_\beta(\mathbf{X}_i))^{\frac{1-Y_i}{2}}$$

$$\log \mathcal{L}(\beta|(Y_i)_{1\leq i\leq N}) = \frac{1}{2} \sum_{i=1}^{N} ((Y_i + 1) \log h_\beta(\mathbf{X}_i)) + (1 - Y_i) \log(1 - h_\beta(\mathbf{X}_i)))$$

$$\log(h_\beta(\mathbf{X}_i)) = -\log(1 + \exp(-\beta^T \phi(\mathbf{X}_i)))$$

$$\log(1 - h_\beta(\mathbf{X}_i)) = -\log(1 + \exp(\beta^T \phi(\mathbf{X}_i)))$$

$$\log \mathcal{L}(\beta|(Y_i)_{1\leq i\leq N}) = -\sum_{i=1}^{N} \log(1 + \exp(-Y_i \beta^T \phi(\mathbf{X}_i)))$$

ERM version

Machine learning view

- assume $\mathcal{Y} = \mathbb{R}$
- use again the class of linear models $\mathbf{X} \mapsto \beta^T \phi(\mathbf{X})$
- the loss is given by

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

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 $\mathcal{Y}'$

Relaxation

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 -1/1 solution but for a continuous one

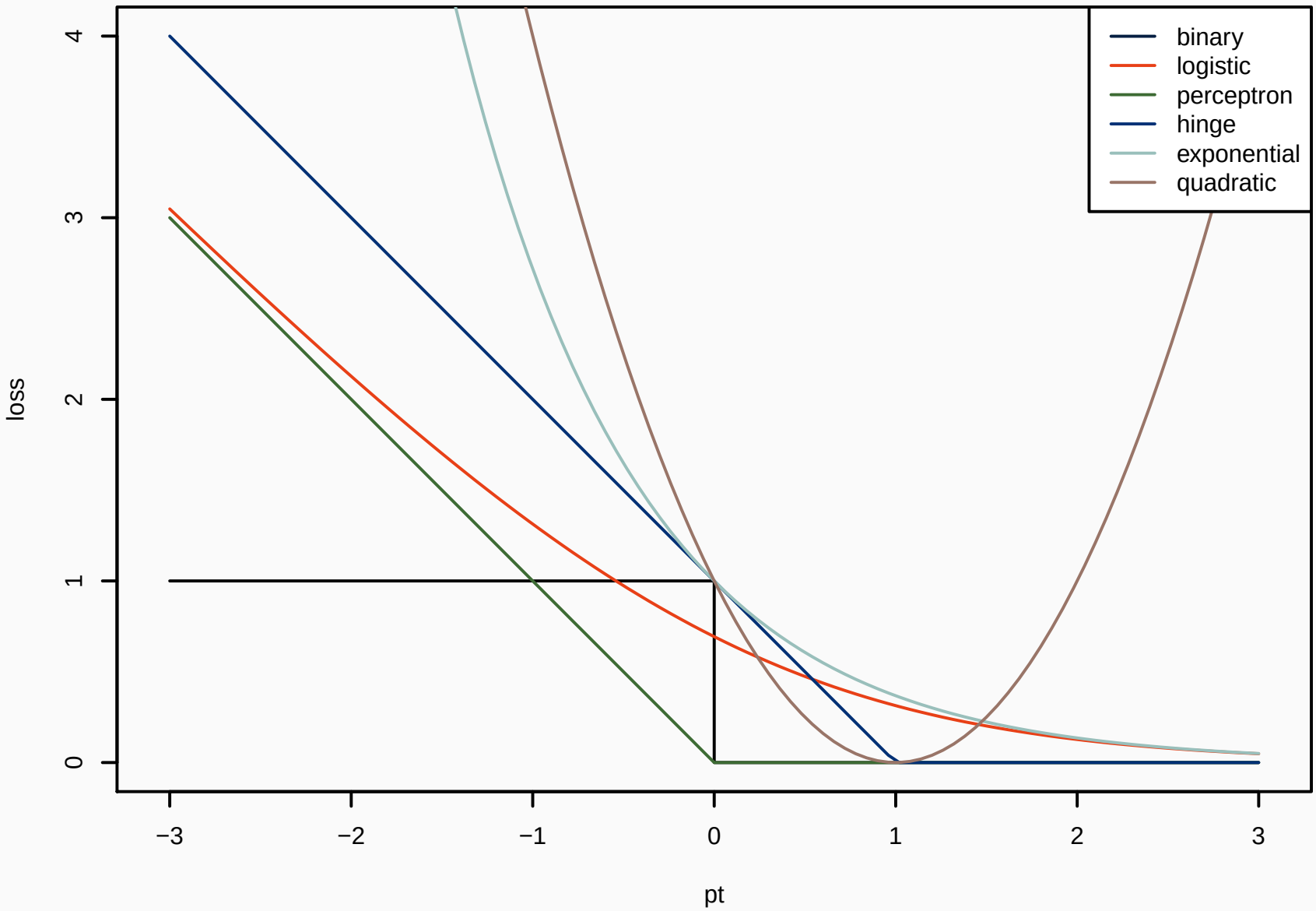
Examples

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: $l_2(p, t) = (p - t)^2 = 1 + (pt)^2 - 2pt$

Losses


binary
logistic
perceptron
hinge
exponential
quadratic
loss
pt
0
1
2
3
4
−3
−2
−1
0
1
2
3

Beyond ERM

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 $l_{relax}(g(\mathbf{X}), Y) \neq l_b(\text{sign}(g(\mathbf{X})), \mathbf{Y})$
- surrogate loss minimization
  - some theoretical results are available
  - but in general no guarantee to find the best model with a surrogate loss

Conclusion

Summary

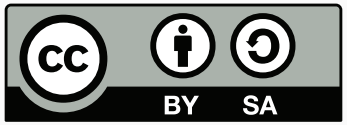
The empirical risk minimization framework seems appealing at first but it has several limitations

- 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 (see the theoretical course)
  - which are strictly limited!

Beyond ERM

- surrogate loss function
- data adaptive model class

Licence



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

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

Version

Last git commit: 2021-01-19
By: Fabrice Rossi (Fabrice.Rossi@apiacoa.org)
Git hash: ef0e3848ef7038aa293211c155495bad460d32b1

Changelog

- January 2021:
  - added an illustration of error types
  - added more examples
- January 2020: clarified some points
- December 2019: initial version