



école —  
normale —  
supérieure —  
paris-saclay —

université  
PARIS-SACLAY

# Introduction to Statistical Learning

Nicolas Vayatis

Lecture 3 - Regularization and stability

## Course overview

- Chapter 1 : Optimality in statistical learning  
Probabilistic view / Performance criteria / Optimal elements
- **Chapter 2 : Mathematical foundations of statistical learning**  
Concentration inequality / Complexity measures / **Regularization and stability**
- Chapter 3 : Consistency of mainstream machine learning methods  
Boosting, SVM, Neural networks / Bagging, Random forests

# Chapter 2 - Mathematical tools

- A. Probability inequalities
- B. Complexity measures
- C. Regularization and stability  $\rightarrow$  today's lecture

# Reminder : The key concepts of Machine Learning

Bias-variance dilemma

Discretization of prediction objective through ERM

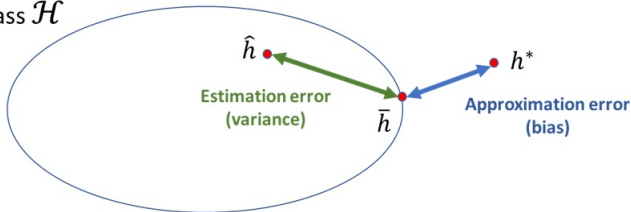
Complexity measures

## How to modify ERM to achieve the key trade-off in Machine Learning?

- Denote by  $L(h)$  the error measure for any decision function  $h$
- We have :  $L(\bar{h}) = \inf_{\mathcal{H}} L$  , and  $L(h^*) = \inf L$
- Bias-Variance type decomposition of error for any output  $\hat{h}$  :

$$L(\hat{h}) - L(h^*) = \underbrace{L(\hat{h}) - L(\bar{h})}_{\text{estimation (stochastic)}} + \underbrace{L(\bar{h}) - L(h^*)}_{\text{approximation (deterministic)}}$$

Hypothesis class  $\mathcal{H}$



# Regularization in linear models

# Regression model

## Basic theory

- Random pair :  $(X, Y)$  over  $\mathbb{R}^d \times \mathbb{R}$
- Decision rules :  $f : \mathbb{R}^d \rightarrow \mathbb{R}$
- Least-square prediction error :  $R(f) = \mathbb{E}(Y - f(X))^2$
- Optimal predictor :  $f^*(x) = \mathbb{E}(Y|X = x)$
- ERM over a class  $\mathcal{F}$  of decision rules given a sample  $\{(X_i, Y_i) : i = 1, \dots, n\}$  :

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^n (Y_i - f(X_i))^2$$

## Linear regression model

- Vector notations :

Response vector  $\mathbf{Y} \in \mathbb{R}^n$ , input data matrix  $\mathbf{X}$  (size  $n \times d$ )

- Linear model with vector notations : parameter  $\beta^* \in \mathbb{R}^d$

$$\mathbf{Y} = \mathbf{X}\beta^* + \varepsilon$$

where  $\varepsilon$  random noise vector (centered, independent of  $\mathbf{X}$ )

- When  $\text{rank}(\mathbf{X}) = d$ , then the ERM is given by :

$$\hat{f}(x) = \hat{\beta}^T x \quad \text{where} \quad \hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} = \hat{\Pi}_d \mathbf{Y}$$

where  $\hat{\Pi}_d$  projection matrix over the columns of  $X$  in  $\mathbb{R}^n$

# Linear regression model

## Limitations

- When  $\text{rank}(\mathbf{X}) < d$  (e.g.  $d > n$ ), then for solution  $\hat{\beta}$  and  $b \in \ker(\mathbf{X})$ , we have that  $\hat{\beta} + b$  is also a solution
- As a consequence :
  - (a) coefficients cannot benefit of interpretation
  - (b) out-of-sample predictions are not unique (while in-sample predictions are unique)
- Solution : assume that effective dimension of the linear model is smaller than  $d$  (and  $n$ !)

# The sparse linear regression model

- Intuition : what if there are uninformative variables in the model but we do not know which they are?
- Sparsity assumption : Let  $\beta^*$  the true parameter which only a subset of variables (called *support*)

$$m^* = \{j : \beta_j^* \neq 0\} \subset \{1, \dots, d\}$$

- $\ell_0$  norm of any  $\beta$  :  $\|\beta\|_0 = \sum_{j=1}^d \mathbb{I}\{\beta_j \neq 0\}$

## Two possible formulations Constrained vs. Penalized optimization

- 1 Ivanov formulation : take  $k$  between 0 and  $\min\{n, d\}$

$$\min_{\beta \in \mathbb{R}^d} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 \quad \text{subject to } \|\beta\|_0 \leq k$$

- 2 Tikhonov formulation : take  $\lambda > 0$

$$\min_{\beta \in \mathbb{R}^d} \{ \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_0 \}$$

## Comments

- Tikhonov looks as a Lagrange formulation of Ivanov
- But here the two formulations are NOT equivalent due to the lack of smoothness of the  $\ell_0$  norm
- Ivanov with  $\ell_0$  constraint is known as the Best Subset Selection problem for which there are algorithms based on heuristics (e.g. Forward Stagewise Regression) which work ok up to  $k \simeq 35$ . Recent advances : check Mixed Integer Optimization (MIO) formulation by Bertsimas et al. (2016).
- Focus on Tikhonov regularization from now on

# Connecting the dots

## Tikhonov penalty and variance

Recall :

- Tikhonov formulation with  $\ell_0$  penalty : take  $\lambda > 0$

$$\min_{\beta \in \mathbb{R}^d} \{ \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_0 \} \quad (1)$$

- Bias-variance decomposition of the error for the LSE  $\hat{\beta}$  :

$$\frac{1}{n} \mathbb{E}(\|\mathbf{X}\beta^* - \mathbf{X}\hat{\beta}_n\|^2) = \sigma^2 \frac{d}{n} \quad (2)$$

where  $d$  is the dimension of the data and  $\sigma^2$  is the variance of the Gaussian noise

Questions for now : does the bias-variance decomposition (2) explains (1) ? Is the penalty correct ?

## Model selection in linear models

- Model :  $\mathbf{Y} = \mathbf{X}\beta^* + \varepsilon$
- Consider a model for  $\beta^*$  that is a subset  $m$  of indices of  $\{1, \dots, d\}$
- Example : In dimension  $d = 3$ , we have :
  - 1 model of size  $|m| = 0$  : constant model
  - 3 models of size  $|m| = 1$  :  $\{1\}, \{2\}, \{3\}$
  - 3 models of size  $|m| = 2$  :  $\{1, 2\}, \{2, 3\}, \{1, 3\}$
  - 1 model of size  $|m| = 3$  :  $\{1, 2, 3\}$

We potentially have 8 versions of Least Square Estimator (LSE), we call constrained LSE (except for the case  $|m| = 3$  which is unconstrained).

## Model selection in linear models

- Model :  $\mathbf{Y} = \mathbf{X}\beta^* + \varepsilon$
- Consider the set  $\mathcal{M}$  of subsets  $m$  of the variables among indices  $\{1, \dots, d\}$ . There are  $2^d$  such sets  $m$ .
- For every  $m \in \mathcal{M}$ , there is a standard linear regression model with dimension  $k_m = |m|$ . In other words, for those  $j \notin m$ , we have  $\beta_j^* = 0$ .
- For each model  $m \in \mathcal{M}$ , compute the constrained Least Square Estimator  $\hat{\beta}_m$ .
- The final estimator is the "best" among  $\hat{\beta}_m$  over all  $m \in \mathcal{M}$

# What "Best" actually means

## The oracle

- In-sample error given by :  $r_m = \frac{1}{n} \mathbb{E}(\|\mathbf{X}\beta^* - \mathbf{X}\hat{\beta}_m\|^2)$
- Best theoretical estimator (called *oracle*) :

$$\hat{\beta}_{\bar{m}} \quad \text{where} \quad \bar{m} = \arg \min_{m \in \mathcal{M}} r_m$$

- Example of an empirical estimator : Akaike Information Criterion (AIC penalty of Least Squares)

$$\hat{m} = \arg \min_{m \in \mathcal{M}} \left\{ \|\mathbf{Y} - \mathbf{X}\hat{\beta}_m\|^2 + 2|m|\sigma^2 \right\}$$

(can be computed from data as long as  $\sigma^2$  is assumed to be known)

# Optional material

## Derivation of Akaike Information Criterion

# Akaike Information Criterion (1/2)

## Derivation

- Recall from least square bias-variance decomposition in linear models : error of estimator

$$r_m = \frac{1}{n} \mathbb{E}(\|\mathbf{X}\beta^* - \mathbf{X}\hat{\beta}_m\|^2) = \frac{1}{n} \mathbb{E}(\|(I_n - \hat{\Pi}_m)\mathbf{X}\beta^*\|^2) + \sigma^2 \frac{|m|}{n}$$

with  $\mathbf{X}\hat{\beta}_m = \hat{\Pi}_m \mathbf{Y}$  where  $\hat{\Pi}_m$  is the orthogonal projection on the subspace  $S_m$  generated with the subset  $m$  of variables

- Similarly, we can derive :

$$\frac{1}{n} \mathbb{E}(\|\mathbf{Y} - \mathbf{X}\hat{\beta}_m\|^2) = \frac{1}{n} \mathbb{E}(\|(I_n - \hat{\Pi}_m)\mathbf{X}\beta^*\|^2) + \sigma^2 \frac{(n - |m|)}{n}$$

- Then, we observe :

$$\frac{1}{n} \mathbb{E}(\|\mathbf{Y} - \mathbf{X}\hat{\beta}_m\|^2) = r_m + \sigma^2 \frac{(n - 2|m|)}{n}$$

## Akaike Information Criterion (2/2)

### Empirical estimator of the error

- We have obtained that :

$$r_m = \frac{1}{n} \mathbb{E}(\|\mathbf{Y} - \mathbf{X}\hat{\beta}_m\|^2) + \sigma^2 \frac{(2|m| - n)}{n}$$

- Unbiased estimator of the error (assuming known variance) :

$$\hat{r}_m = \frac{1}{n} \|\mathbf{Y} - \mathbf{X}\hat{\beta}_m\|^2 + \sigma^2 \frac{(2|m| - n)}{n}$$

- Akaike Information Criterion

$$\hat{m} = \arg \min_{m \in \mathcal{M}} \left\{ \|\mathbf{Y} - \mathbf{X}\hat{\beta}_m\|^2 + 2|m|\sigma^2 \right\}$$

End of optional material

## Bottom line on AIC

### Is AIC an optimal penalty for model selection in linear models?

- Tikhonov regularization for  $\ell_0$  norm is equivalent to AIC with  $\lambda = 2\sigma^2$  in this case ( $\lambda$  also depends on  $n$  if we minimize the average square error on the data)
- In practice, AIC does not pick the right dimension : in high dimensions,  $\hat{r}_m$  fluctuates around  $r_m$  due to a large amount of models with same cardinality  $|m|$
- The correct penalty should be of the order  $2\sigma^2|m| \log(d)$

NB : The number of linear models of given size  $|m|$  in dimension  $d$  is :

$$\binom{d}{|m|} \leq \exp(|m|(1 + \log(d/|m|)))$$

## AIC in large dimensions

- When  $d$  is large, is this practical?
- There are about  $e^{d/2}$  models to scan in the worst case where  $|m| \simeq d/2 \dots$

# Solving the computation burden

## The power of convexity

- Practical methods for model selection are essentially greedy heuristics consisting in adding and/or retrieving one variable at the time to explore part of the whole model space which is exponential in the dimension. Examples are : Forward Stagewise Regression, Forward-Backward algorithm...
- Question : would it be possible to solve the optimization wrt the unknown parameter  $\beta$  AND wrt to its support subset of indices jointly ?
- Answer is yes at the cost of the so-called relaxation of the non-convex formulation with the  $\ell_0$  penalty to a convexified problem with an  $\ell_1$  penalty.

# The LASSO for linear models

## From $\ell_0$ to $\ell_1$

- Consider the relaxation of the previous problem replacing the  $\ell_0$ -norm by the  $\ell_1$ -norm :

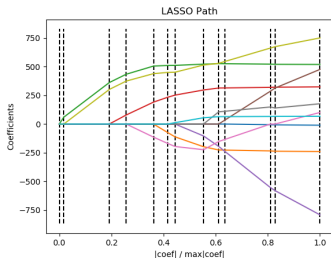
$$\|\beta\|_1 = \sum_{j=1}^d |\beta_j|$$

- The new estimator is called the LASSO : for any  $\lambda > 0$ ,

$$\hat{\beta}_\lambda \in \arg \min_{\beta \in \mathbb{R}^d} \{ \|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda \|\beta\|_1 \}$$

## Blessings of the LASSO

- Approximate solutions via efficient algorithms building the so-called regularization paths  $\lambda \rightarrow \hat{\beta}_\lambda$  :



- Theoretical soundness : it can be shown that : as  $n, d \rightarrow \infty$ , in-sample error resists to curse of dimensionality

$$\frac{1}{n} \mathbb{E}(\|\mathbf{X}\beta^* - \mathbf{X}\hat{\beta}\|^2) \leq C \|\beta^*\|_1 \sqrt{\frac{\log d}{n}}$$

(holds for the constrained formulation)

## Penalized least-squares in linear regression

- LASSO

$$\hat{\beta}_\lambda \in \arg \min_{\beta \in \mathbb{R}^d} \left\{ \frac{1}{n} \|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda \|\beta\|_1 \right\}$$

- Ridge regression

$$\hat{\beta}_\lambda \in \arg \min_{\beta \in \mathbb{R}^d} \left\{ \frac{1}{n} \|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda \|\beta\|_2^2 \right\}$$

- Structured sparsity with  $\|\beta\|_S$  being a sparsity inducing norm (group LASSO, graph LASSO, ...)

$$\hat{\beta}_\lambda \in \arg \min_{\beta \in \mathbb{R}^d} \left\{ \frac{1}{n} \|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda \|\beta\|_S \right\}$$

# The "mother" of shallow ML algorithms

From classical statistics to Machine Learning

# "Shallow Learning"

- **Shallow Learning** are algorithms which will only depend on very few hyperparameters beyond the  $\lambda$ .
- **Deep Learning** relies on many architectural hyperparameters (e.g., number of layers, nodes, etc.) whose calibration is a very complex optimization problem.
- The theory of supervised Machine Learning *should* apply to both shallow and deep learning.

## Penalized optimization

- Learning process as the optimization of a data-dependent criterion :

$$\text{Criterion}(h) = \text{Training error}(h) + \lambda \text{Penalty}(h)$$

- Training error : data-fitting term related to a loss function
- Penalty : complexity of the decision function or function norms (e.g. LASSO)
- Constant  $\lambda$  : smoothing parameter tuned through cross-validation procedure

## How to create shallow ML algorithms ?

- Standard function classes (e.g. linear functions) and risk (e.g. least squares) and variations on the penalties

$$\text{Criterion}(h) = \text{Training error}(h) + \lambda \text{Penalty}(h)$$

(e.g. in least square minimization : LASSO, Group LASSO, Elastic Net, Fused LASSO, structures penalties...)

- Playing with losses changing the training error

$$\text{Criterion}(h) = \text{Training error}(h) + \lambda \text{Penalty}(h)$$

# Changing loss functions

A few examples:

**Ridge regression:**

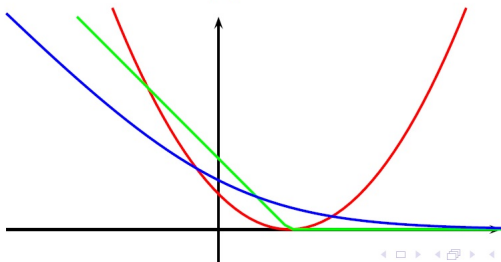
$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n \frac{1}{2} (y_i - \beta^\top \mathbf{x}_i)^2 + \lambda \|\beta\|_2^2.$$

**Linear SVM:**

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n \max(0, 1 - y_i \beta^\top \mathbf{x}_i) + \lambda \|\beta\|_2^2.$$

**Logistic regression:**

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n \log(1 + e^{-y_i \beta^\top \mathbf{x}_i}) + \lambda \|\beta\|_2^2.$$



## The principle of Structural risk minimization (SRM)

- Given a training set of size  $n$  and the corresponding empirical error  $\hat{L}_n$ , consider the ERM principle over an increasing sequence of hypothesis classes  $\mathcal{H}_1 \subset \dots \mathcal{H}_j \subset \dots$  of increasing complexity (e.g. dimension in linear models, VC dimension in nonlinear models)

## SRM leads to penalized ERM

- In order to achieve the estimation-approximation ("bias-variance") trade-off, the idea is to penalize the empirical risk with a complexity term :

$$\hat{h}_n^{\text{SRM}} = \hat{h}_{\hat{j},n}^{\text{ERM}}$$

where :

$$\hat{h}_{j,n}^{\text{ERM}} = \arg \min_{h \in \mathcal{H}_j} \hat{L}_n(h)$$

and

$$\hat{j} = \arg \min_{j \geq 1} \left\{ \hat{L}_n(\hat{h}_{j,n}^{\text{ERM}}) + \lambda \text{pen}(n, \text{complexity}(\mathcal{H}_j)) \right\}$$

where  $\lambda = \lambda_n$  is called the regularization parameter, or smoothing parameter.

# SRM example

## Regularization in decision trees

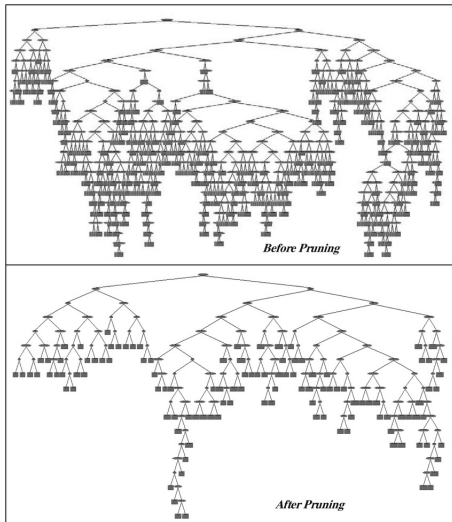
Building a complexity calibrated decision tree involves two steps :

- 1 Growing a decision tree - output : a tree classifier  $\hat{h}_\pi$  with data-dependent partition  $\hat{\pi}$  (which may overfit the data !)
- 2 Pruning the tree - optimization over all subpartitions (subtrees) a penalized criterion of the form

$$\arg \min_{\pi \subset \hat{\pi}} \hat{L}_n(\hat{h}_\pi) + \lambda |\pi|$$

where  $\hat{h}_\pi$  is a tree classifier obtained with the training data based on a partition  $\pi$  which is a subpartition of  $\hat{\pi}$

# Example of original and pruned decision tree



## Other examples of regularized formulations

- Linear (soft) SVM (hinge loss, L2 penalty)

$$\hat{\beta}_\lambda \in \arg \min_{\beta \in \mathbb{R}^d} \left\{ \frac{1}{n} \sum_{i=1}^n (1 - Y_i \cdot \beta^T X_i)_+ + \lambda \|\beta\|_2^2 \right\}$$

- Kernel ridge regression :  $K$  kernel and its parameter

$$\hat{\alpha}_\lambda = \min_{\alpha \in \mathbb{R}^n} \{ \alpha^T K \alpha - 2\alpha^T Y + \lambda \alpha^T \alpha \}$$

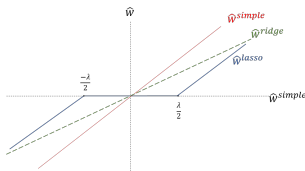
where  $K = (K(X_i, X_j))_{i,j}$

Example of kernel  $K$  with one parameter  $\mu$  :

$$K(x, x') = \exp \left( -\frac{\|x - x'\|_2^2}{\mu} \right), \quad \mu > 0$$

# The four faces of regularization (in linear models)

- Penalized optimization : Tikhonov regularization
- Bayesian priors :
  - LS with gaussian prior  $\leftrightarrow$  Ridge regression
  - LS with Laplace prior  $\leftrightarrow$  LASSO
- Soft order constraint : minimize least squares subject to  $\|\beta\|^2 \leq C$  (budget  $C$  is the twin sister of smoothing parameter  $\lambda$ )
- Weight decay (also known as 'shrinkage' in mathematical statistics)



## ...and other faces of regularization in Machine Learning

- Local rules :
  - regularization through local smoothing
- Ensemble learning :
  - Regularization through aggregation
  - Dimension subsampling in random forests
  - Early stopping
- Deep learning : Implicit regularization, e.g.
  - Dropout
  - Batch normalization
  - Data augmentation
  - ...

# Stability of ML algorithms

## The principle of stability

- Builds on sensitivity analysis approach applied to machine learning algorithms with respect to changes in the training set
- Stability is a property of the algorithm (e.g. ERM, KRR...) and depends on the loss function
- It builds upon the good old concept of robustness in statistics, revisited with modern tools from probability (concentration inequalities) and applied to the analysis of learning algorithms
- Key references : Bousquet and Elisseeff (2002) and Mukherjee, Niyogi, Poggio, and Rifkin (2006)

## Definition of (uniform) stability

- Consider an algorithm which provides an estimator  $\tilde{h}_n$  on a sample of size  $n$  and we denote  $\tilde{h}'_n$  the estimator resulting from the same sample where one observation was changed.
- For a given loss function  $\ell$ , We say that the algorithm is (uniformly)  $\gamma$ -stable if there exists a constant  $\gamma$  for which we have : for any training sample, and for any pair  $(x, y)$ ,

$$|\ell(y, \tilde{h}_n(x)) - \ell(y, \tilde{h}'_n(x))| \leq \gamma$$

## Error bound based on stability

- Consider a loss function  $\ell$  which is uniformly bounded by  $M > 0$  and  $\tilde{h}_n$  is the output of a  $\gamma$ -uniformly stable learning algorithm.
- We have, with probability at least  $1 - \delta$  :

$$L(\tilde{h}_n) \leq \hat{L}_n(\tilde{h}_n) + \gamma + (2n\gamma + M) \sqrt{\frac{\log(1/\delta)}{2n}}$$

- Proof based on McDiarmid's concentration inequality

Hint : under  $\gamma$ -uniform stability assumption, the function  $L(\tilde{h}_n) - \hat{L}_n(\tilde{h}_n)$  of the sample satisfies the bounded difference assumption with  $c/n = 2\gamma + M/n$

Consequence : the upper bound converges to zero when  $\gamma = \gamma_n \rightarrow 0$  and  $\gamma_n \sqrt{n} \rightarrow 0$

## Stability of regularized methods

- Consider the case of linear models estimated by minimizing an error with  $\rho$ -Lipschitz loss with an  $\ell_2$ -penalty : for any  $\lambda > 0$

$$\widehat{\beta}_n^R(\lambda) = \arg \min_{\beta \in \mathbb{R}^d} \left\{ \frac{1}{n} \sum_{i=1}^n \ell(Y_i, \beta^T X_i) + \lambda \|\beta\|_2^2 \right\}$$

- It can be shown that this algorithm is stable with parameter  $\gamma$  such that :

$$\gamma \leq \frac{2\rho^2}{n\lambda}$$

- A similar result holds with self-bounded loss such that : at any  $(x, y)$

$$\|\nabla \ell(y, \beta^T x)\|^2 \leq 2\beta \ell(y, \beta^T x)$$

# End of Chapter 2

Coming next : analysis of mainstream ML algorithms