M2 CHPS

[Data & Apprentissage]

Introduction à la science des données et à l'apprentissage

Nicolas Vayatis

Modèles linéaires et parcimonie

Supervised Machine Learning

The bias-variance decomposition in Machine Learning

General setup
Notations

- Goal of learning: an optimal decision function $h^* : \mathcal{X} \to \mathcal{Y}$
  $\mathcal{X}$: domain set, $\mathcal{Y}$: label set
- Input of learning:
  - **Training data:** a set of labeled data

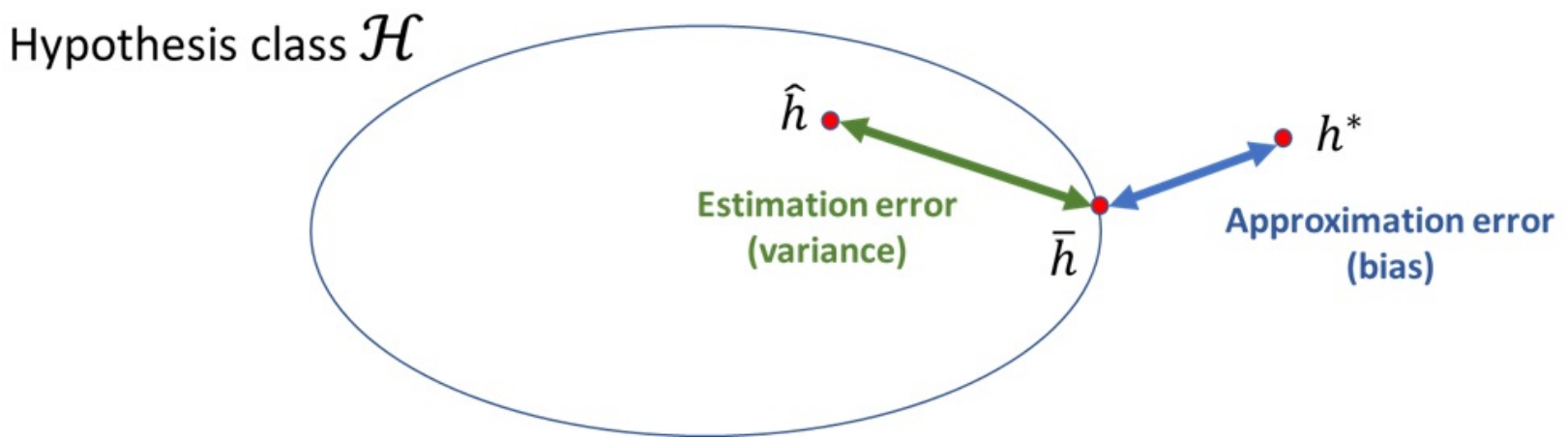
    $$D_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$$

    of size n, where the (X, Y)'s are in $\mathcal{X} \times \mathcal{Y}$
  - **Hypothesis space:** a collection $\mathcal{H}$ of candidate decision functions $h : \mathcal{X} \to \mathcal{Y}$
- Output of learning: an empirical decision function $\widehat{h}$ in the hypothesis space $\mathcal{H}$ estimated from training data D_n
- Reference in $\mathcal{H}$: the best decision function $\bar{h}$ in the class (the more data, the closer $\widehat{h}$ to $\bar{h}$)

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 $\widehat{h}$:

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



About approximation error

- Cybenko (1989) - Denseness result in the spirit of Stone-Weierstrass showing that any linear combination of compositions of sigmoid with linear functions is dense wrt the supremum norm in the space of continuous functions over the d-dimensional unit cube.
- Barron (1994) - Approximation error bound involves a parameter quantifying the smoothness of the target function.
- Status of this question in the regression setup:
  - For kernel machines: a full theory is available thanks to Smale (2003), Steinwart (2008).
  - For deep learning: recent work by Grohs, Perekrestenko, Elbrächter, and Bölcskei (2019) .
  - In the classification setup, tough problem, still open issue...

Reminder on linear models in statistics

The regression case

The regression model

- Goal of learning: $h^* : \mathbb{R}^d \to \mathbb{R}$
- Observations: IID random pairs $(X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}$:

$$Y_i = h^*(X_i) + \varepsilon_i \ , \quad i = 1, \ldots, n$$

where ε_i is a random noise variable independent of X

- We shall use a vector notation as follows:

$$\mathbf{Y} = \mathbf{h}^* + \varepsilon$$

where the three terms are all in $\mathbb{R}^n$

Vector notations

- Elements in $\mathbb{R}^n$:

$$\mathbf{Y} = (Y_1, \ldots, Y_n)^T,$$
$$\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^T$$

- Image vectors: for any $h \in \mathcal{H}$, we use the bold characters as

$$\mathbf{h} = (h(X_1), \ldots, h(X_n))^T$$

- The image of data points through elements of $\mathcal{H}$ by

$$\mathcal{H}(X) = \{\mathbf{h} = (h(X_1), \ldots, h(X_n))^T, : h \in \mathcal{H}\}$$

- Norm in $\mathbb{R}^n$:

$$\forall u = (u_1, \ldots, u_n)^T \in \mathbb{R}^n \ , \quad \|u\|^2 = \sum_{i=1}^{n} u_i^2$$

Least square estimator (LSE)

- Definition of the LSE :

$$\widehat{\mathbf{h}}_{\mathbf{n}} = \underset{\mathbf{h} \in \mathcal{H}(X)}{\arg\min} \frac{1}{n} \|\mathbf{Y} - \mathbf{h}\|^2$$

  where $\mathcal{H}(X) = \{\mathbf{h} = (h(X_1), \ldots, h(X_n))^T, : h \in \mathcal{H}\}$

- Assuming that the linear model is gaussian (i.e. the noise variables are IID and follow a centered gaussian distribution with fixed and known variance), then the LSE also corresponds to the Maximum Likelihood estimator.

Gaussian linear model in $\mathbb{R}^d$
Two additional assumptions

- The hypothesis space $\mathcal{H}$ is the class of *linear* functions of rank d
- The noise vector $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^T$ is a *gaussian* random vector in $\mathbb{R}^n$ with distribution $\mathcal{N}_n(0, \sigma^2 I_n)$

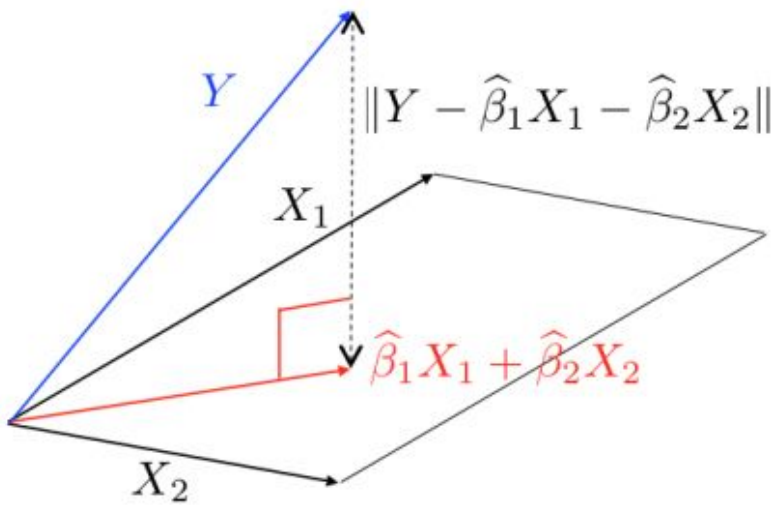
Linear models in $\mathbb{R}^d$
Examples

Notations: $x = (x^{(1)}, \ldots, x^{(d)})^T \in \mathbb{R}^d$

- Linear regression: $h(x) = \sum_{k=1}^{d} \beta_k x^{(k)}$
- Basis/frame expansion (Fourier, splines, wavelets, etc.)
- Additive models: $h(x) = \sum_{k=1}^{d} f_k(x^{(k)})$
- Piecewise constant regression (taking into account breakpoints)

LSE in linear regression

- Denote by $\mathbf{X} \in \mathbb{R}^{n \times d}$ the data matrix and $\beta \in \mathbb{R}^d$ the parameter to estimate
- Assumption: $\mathbf{X}$ is of full rank equal to d and assume $d \leq n$
- Definition of LSE: $\widehat{\beta}_{\mathbf{n}} = \arg\min_{\beta \in \mathbb{R}^d} \frac{1}{n} \|\mathbf{Y} - \mathbf{X}\beta\|^2$
- Least square estimate: $\widehat{\mathbf{h}}_{\mathbf{n}} = \mathbf{X}\widehat{\beta}_{\mathbf{n}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} = \widehat{\Pi}\mathbf{Y}$



Proof of the LSE computation in linear models

- $\widehat{\beta}_{\mathbf{n}} = \underset{\beta \in \mathbb{R}^d}{\arg\min}\, R(\beta)$ where $R(\beta) = \frac{1}{n}\|\mathbf{Y} - \mathbf{X}\beta\|^2$
- Gradient computation:

$$\frac{d}{d\beta}\left((\mathbf{Y} - \mathbf{X}\beta)^T(\mathbf{Y} - \mathbf{X}\beta)\right) = -2\mathbf{Y}^T\mathbf{X} + 2\beta^T\mathbf{X}^T\mathbf{X} \in \mathbb{R}^{1\times d}$$

- First-order condition for the minimizer of a convex function:

$$-\mathbf{Y}^T + \beta^T\mathbf{X}^T\mathbf{X} = 0$$

- Final result: $\widehat{\beta}_{\mathbf{n}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$

Alternate proof based on chain rule

- Let: $R(\beta) = \ell(e(\beta))$ where $\ell(e) = \|e\|^2$ and $e(\beta) = \mathbf{Y} - \mathbf{X}\beta$
- Chain rule: $\frac{dR}{d\beta} = \frac{\partial \ell}{\partial e}\frac{\partial e}{\partial \beta}$ where the j-th element is given by:

$$\frac{dR}{d\beta}[j] = \sum_{k=1}^{n} \frac{\partial \ell}{\partial e}[k]\frac{\partial e}{\partial \beta}[k,j]$$

- Note that: $\frac{\partial \ell}{\partial e} = 2e^T \in \mathbb{R}^{1\times n}$ and $\frac{\partial e}{\partial \beta} = -\mathbf{X} \in \mathbb{R}^{n\times d}$
- Finally: $\frac{dR}{d\beta} = -2e^T\mathbf{X} = -2(\mathbf{Y} - \mathbf{X}\beta)^T\mathbf{X}$

What all student (should) know

The bias-variance trade-off in regression

We have seen so far

- Model: $\mathbf{Y} = \mathbf{h}^* + \varepsilon \in \mathbb{R}^n$
- Computation of LSE: $\widehat{\mathbf{h}}_\mathbf{n} = \widehat{\Pi}\mathbf{Y}$ where $\widehat{\Pi} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$
- A notion of risk:

$$L(\widehat{\mathbf{h}}_\mathbf{n}) = \frac{1}{n}\mathbb{E}\left(\|\mathbf{h}^* - \widehat{\mathbf{h}}_\mathbf{n}\|^2\right)$$

Bias-variance decomposition (1/2)
Derivation

- First note that: $\widehat{\mathbf{h}}_\mathbf{n} = \widehat{\Pi}\mathbf{Y} = \widehat{\Pi}(\mathbf{h}^* + \varepsilon)$ and then

$$\mathbf{h}^* - \widehat{\mathbf{h}}_\mathbf{n} = (I_n - \widehat{\Pi})\mathbf{h}^* - \widehat{\Pi}\varepsilon$$

- Note that $\widehat{\Pi} : \mathbb{R}^n \to \mathbb{R}^n$ the orthogonal projection onto $\mathcal{H}(X)$:

$$\widehat{\Pi} \circ \widehat{\Pi} = \widehat{\Pi}$$

- By orthogonality of the images of $I_n - \widehat{\Pi}$ and $\widehat{\Pi}$:

$$\begin{aligned} L(\widehat{h}_n) &= \frac{1}{n}\mathbb{E}\left(\|\mathbf{h}^* - \widehat{\mathbf{h}}_\mathbf{n}\|^2\right) \\ &= \frac{1}{n}\mathbb{E}(\|(I_n - \widehat{\Pi})\mathbf{h}^*\|^2 + \|\widehat{\Pi}\varepsilon\|^2) \end{aligned}$$

Bias-variance decomposition (2/2)
Result

- Using an additional technical result (next slide):

$$
\begin{aligned}
L(\widehat{h}_n) &= \frac{1}{n}\mathbb{E}\big(\|\mathbf{h}^* - \widehat{\mathbf{h}}_\mathbf{n}\|^2\big) \\
&= \frac{1}{n}\mathbb{E}(\|(I_n - \widehat{\Pi})\mathbf{h}^*\|^2 + \|\widehat{\Pi}\varepsilon\|^2) \\
&= \underbrace{\frac{1}{n}\mathbb{E}(\|(I_n - \widehat{\Pi})\mathbf{h}^*\|^2)}_{\text{bias}} + \underbrace{\sigma^2\frac{d}{n}}_{\text{variance}}
\end{aligned}
$$

- Used in model selection (e.g. AIC = Akaike Information Criterion)

Explanation of the d/n term

Property on the norm of projections of gaussian random vectors:

- Assume $\mathbf{Z}$ is a gaussian random vector $\mathcal{N}_n(0, I_n)$ in $\mathbb{R}^n$, $\mathcal{H}$ is a linear subspace of $\mathbb{R}^n$ and $\Pi : \mathbb{R}^n \to \mathbb{R}^n$ a linear projection onto $\mathcal{H}$
- Then: the random vector $\Pi_{\mathcal{H}}\mathbf{Z}$ has gaussian distribution $\mathcal{N}_n(0, \Pi)$ on $\mathbb{R}^n$ (linear transformation of a gaussian is a gaussian)
- Furthermore: $\|\Pi\mathbf{Z}\|^2$ follows a chi-square distribution with

$$\mathbb{E}(\|\Pi\mathbf{Z}\|^2) = \dim(\mathcal{H})$$

How Machine Learning takes over linear regression

1. What if non-additive noise? Other tasks than regression?
2. From linear to nonlinear models
3. What replaces the dimension d as a measure of complexity in nonlinear models?
4. Is the d/n rate also typical for larger hypothesis classes? What if d larger than n ?

From classical statistics to Machine Learning

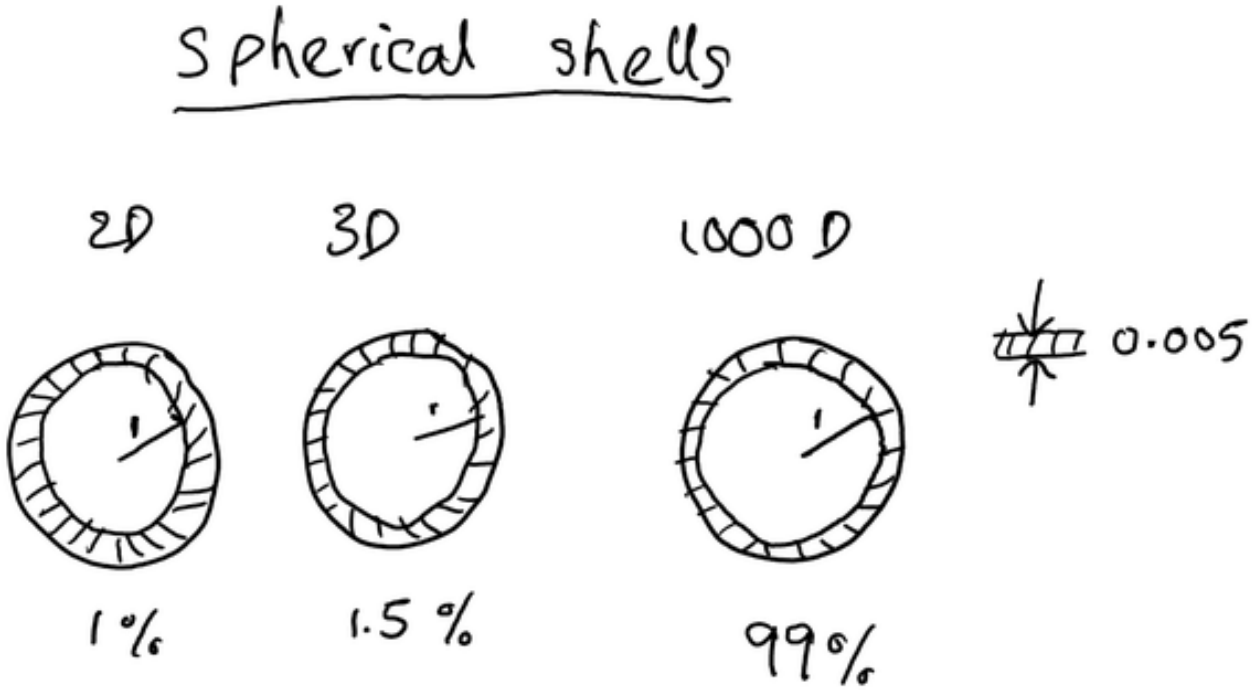
- Handling models in high dimensions:

  $d \gg 1$, $d \gg n$

- Questioning dimensionality:

  number of parameters vs. complexity of set of functions

High Dimensional Interlude
Some surprising fact



- Ratio shell/volume:

$$\frac{vol(B_d(0,1) - B_d(0,1-\varepsilon))}{vol(B_d(0,1))} = 1-(1-\varepsilon)^d \to 1 \text{ when } d \to \infty$$

High Dimensional Interlude Readings

- High level position paper:

  "High Dimensional Data Analysis : The Curses and Blessings of Dimensionality" by D. Donoho (2000)

- Maths book:

  "High-Dimensional Probability" by Roman Vershynin (2018)

From classical statistics to Machine Learning: Handling high dimensions

A. Sparsity and linear models

B. Estimating nonlinear functions

C. Generalizations

What 'high' means for linear regression

- So far, we assumed: $n \geq d$ and $(\mathbf{X}) = d$
- If d gets large, two things may/will happen:
  - Instability in computing the inverse $(\mathbf{X}^T\mathbf{X})^{-1}$
  - Rank-deficiency of data matrix $\mathbf{X}$
- Rank-deficient case (but still $n \geq d$):
  - There are many solutions to the LS problem...
  - Get one LSE by using Moore-Penrose pseudo-inverse instead of $(\mathbf{X}^T\mathbf{X})^{-1}$ in the projection matrix
  - The LSE is the solution with minimal ℓ_2-norm
- Cure to instability: (ridge) regularization...

A. Sparsity and linear models

Tuning the dimension of the model

Linear regression model
Notations

- Vector notations:

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

- Linear model with vector notations:

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

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

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 β^* the true parameter which only a subset of variables (called *support*)

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

- ℓ_0 norm of any β: $\|\beta\|_0 = \sum_{j=1}^{d} \mathbb{I}\{\beta_i \neq 0\}$

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 $\lambda > 0$

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

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 ℓ_0 norm
- Ivanov with ℓ_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

Sparsity and linear models

Model selection

Connecting the dots
Tikhonov penalty and variance

Recall:

- Tikhonov formulation with ℓ_0 *penalty*: take $\lambda > 0$

$$\min_{\beta \in \mathbb{R}^d} \left\{ \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_0 \right\} \tag{1}$$

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

$$\frac{1}{n}\mathbb{E}\left(\|\mathbf{X}\beta^* - \mathbf{X}\widehat{\beta}_{\mathbf{n}}\|^2\right) \simeq \text{Bias} + \sigma^2 \frac{d}{n} \tag{2}$$

  where d is the dimension of the data and σ^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 β^* that is a subset m of indices of $\{1, \ldots, d\}$
- 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 call constrained LSE (except for the case $|m| = 3$ which is unconstrained).

Model selection in linear models

- 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 $|m|$. In other words, for those $j \notin m$, we have $\theta_j^* = 0$.
- Denote by $\mathbf{X}_m$ the submatrix of $\mathbf{X}$ of size $n \times |m|$ which contains only the columns whose index belongs to m
- For each model $m \in \mathcal{M}$, compute the constrained Least Square Estimator $\widehat{\theta}_n^{(m)} = (\mathbf{X}_m^T \mathbf{X}_m)^{-1} \mathbf{X}_m^T \mathbf{Y} \in \mathbb{R}^{|m|}$.
- The final estimator is the "best" among $\widehat{\theta}_n^{(m)}$ over all $m \in \mathcal{M}$

What "Best" actually means

- Denote by $\mathbf{X}_m$ the data matrix of size $n \times |m|$
- Risk of the predictor: $r_m = \frac{1}{n}\mathbb{E}\left(\|\mathbf{X}\theta^* - \mathbf{X}_m\widehat{\theta}_n^{(m)}\|^2\right)$
- Best theoretical estimator (called *oracle*):

$$\widehat{\theta}_n^{(\overline{m})} \quad \text{where} \quad \overline{m} = \arg\min_{m \in \mathcal{M}} r_m$$

- Penalized LS with Akaike Information Criterion (AIC)

$$\widehat{m} = \arg\min_{m \in \mathcal{M}} \left\{ \|\mathbf{Y} - \mathbf{X}_m\widehat{\theta}_n^{(m)}\|^2 + 2|m|\sigma^2 \right\}$$

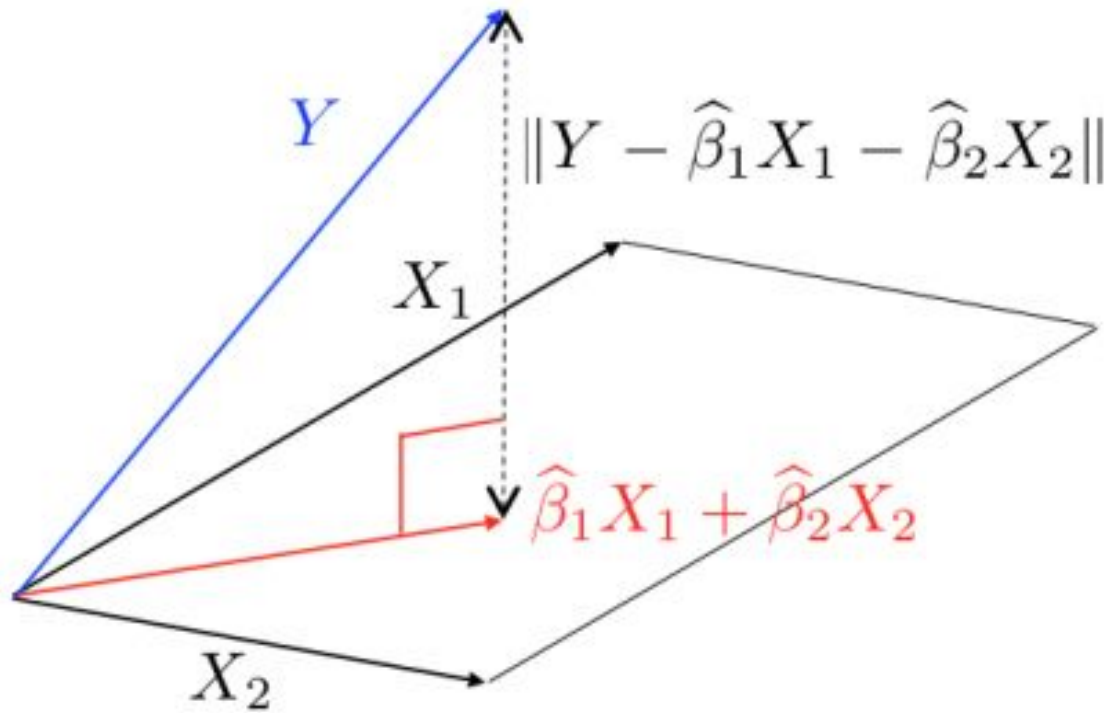
(can be computed from data assuming σ^2 is known)

Optional material

Derivation of Akaike Information Criterion

LSE in linear regression

- Denote by $\mathbf{X}_m$ the data matrix $(n \times |m|)$ and $\widehat{\theta}_n^{(m)}$ the LSE
- Prediction vector: $\mathbf{X}_m \widehat{\theta}_n^{(m)} = \mathbf{X}_m (\mathbf{X}_m^T \mathbf{X}_m)^{-1} \mathbf{X}_m^T \mathbf{Y} = \widehat{\Pi}_m \mathbf{Y}$



Bias-variance decomposition (1/2)
Derivation

- Note that $\widehat{\Pi}_m : \mathbb{R}^n \to \mathbb{R}^n$ is the orthogonal projection onto the space generated by the directions in m:

$$\widehat{\Pi}_m \circ \widehat{\Pi}_m = \widehat{\Pi}_m$$

- We have: $\mathbf{X}_m \widehat{\theta}_n^{(m)} = \widehat{\Pi}_m \mathbf{Y} = \widehat{\Pi}_m(\mathbf{X}\theta^* + \varepsilon)$ and then

$$\mathbf{X}\theta^* - \mathbf{X}_m \widehat{\theta}_n^{(m)} = (I_n - \widehat{\Pi}_m)\mathbf{X}\theta^* - \widehat{\Pi}_m \varepsilon$$

Bias-variance decomposition (2/2)
Result

- Property of the projector: images of $I_n - \widehat{\Pi}$ and $\widehat{\Pi}$ are orthogonal
- Therefore:

$$\begin{aligned} r_m &= \frac{1}{n}\mathbb{E}\left(\|(I_n - \widehat{\Pi}_m)\mathbf{X}\theta^*\|^2 + \|\widehat{\Pi}_m\varepsilon\|^2\right) \\ &= \frac{1}{n}\mathbb{E}\left(\|(I_n - \widehat{\Pi}_m)\mathbf{X}\theta^*\|^2\right) + \sigma^2\frac{|m|}{n} \end{aligned}$$

since $\|\widehat{\Pi}_m\varepsilon\|^2$ follows a chi-square distribution with $|m|$ degrees of freedom (property of projections of multivariate gaussian vectors).

Akaike Information Criterion (1/2)
Derivation

- Similarly, we can derive:

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

  Indeed: $\mathbf{Y}-\mathbf{X}_m\widehat{\theta}_n^{(m)}=(I_n-\widehat{\Pi}_m)(\mathbf{X}\theta^*+\varepsilon)$ and $\|I_n-\widehat{\Pi}_m\varepsilon\|^2$ follows a chi-square distribution with $n-|m|$ degrees of freedom

- Combining the two identities, prediction error can be related to risk:

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

Akaike Information Criterion (2/2)
Empirical estimator of the error

- We have obtained that:

$$r_m = \frac{1}{n}\mathbb{E}\left(\|\mathbf{Y} - \mathbf{X}_m\widehat{\theta}_n^{(m)}\|^2\right) + \sigma^2\frac{(2|m| - n)}{n}$$

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

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

- Akaike Information Criterion

$$\widehat{m} = \arg\min_{m\in\mathcal{M}}\left\{\|\mathbf{Y} - \mathbf{X}_m\widehat{\theta}_n^{(m)}\|^2 + 2|m|\sigma^2\right\}$$

End of optional material

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 $|m| \simeq d/2$...

A. Sparsity and linear models

From (mathematical) statistics to optimization

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 β 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 ℓ_0 penalty to a convexified problem with an ℓ_1 penalty.

The LASSO for linear models
From ℓ_0 to ℓ_1

- Consider the relaxation of the previous problem replacing the ℓ_0-norm by the ℓ_1-norm:

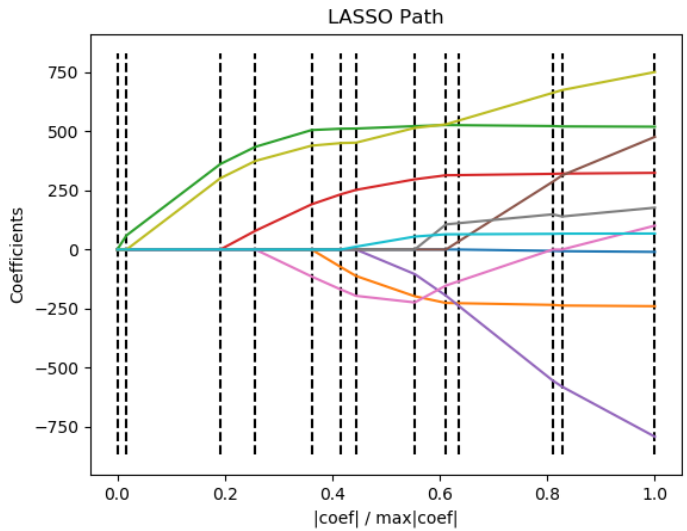
$$\|\beta\|_1 = \sum_{j=1}^{d} |\beta_j|$$

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

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

Blessings of the LASSO

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



- Theoretical soundness: it can be shown that: as $n, d \to \infty$

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

The "mother" of ML algorithms
Penalized optimization

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

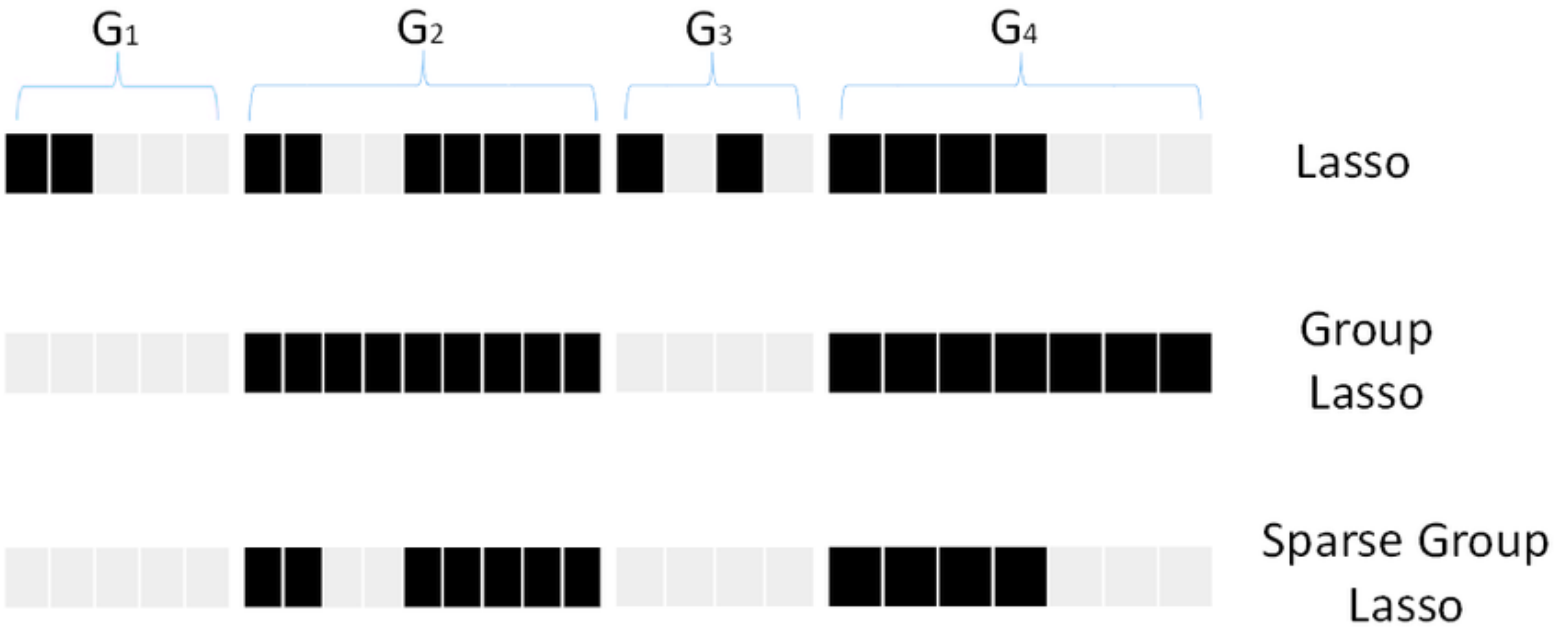
$$\text{Criterion}(h) = \text{Training error}(h) + \lambda \, \text{Penalty}(h)$$

- Training error: data-fitting term related to a loss function
- Penalty: complexity of the decision function
- Constant λ: smoothing parameter tuned through cross-validation procedure

A. Sparsity and linear models

Structured sparsity

Putting human priors in penalties
Sparsity patterns

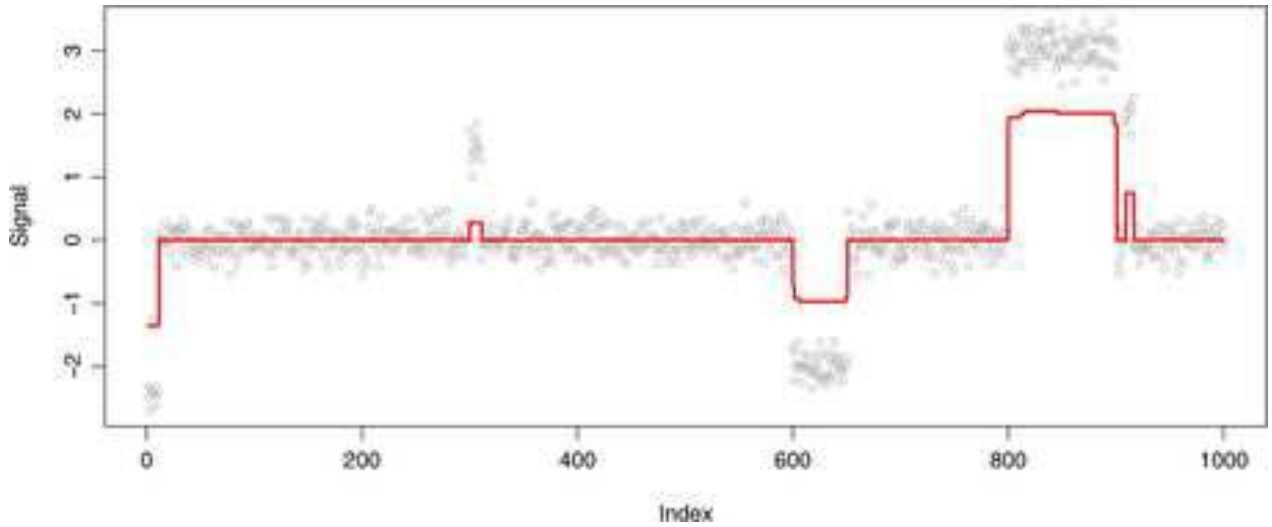


The simplest structured penalty
Group LASSO

- Group structure on the parameter β^*: let G the number of groups of subsets of indices in $\{1, \ldots, d\}$ and, for $g = 1, \ldots, G$, we denote by $\mathbf{X}^{(g)}$ the submatrix of $\mathbf{X}$ with variables in group g and by $\beta^{(g)}$ the coefficient vector applied to variables in group g and d_g is the size of group g.
- Group LASSO formulation:

$$\widehat{\beta}_\lambda \in \underset{\beta \in \mathbb{R}^d}{\arg\min} \left\{ \|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda \sum_{g=1}^{G} \sqrt{d_g} \|\beta^{(g)}\| \right\}$$

Case of temporal patterns
Fused LASSO



- Enforcing temporal coherence leads to adding a penalty term:

$$\widehat{\beta}_\lambda \in \underset{\beta \in \mathbb{R}^d}{\arg\min} \left\{ \|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda \|\beta\|_1 + \mu \sum_{j=2}^{d} |\beta_j - \beta_{j-1}| \right\}$$

A. Sparsity and linear models

Ridge regression

Penalized optimization
Other penalties?

- Until now: hypothesis class with linear functions $h \in \mathcal{H}$ and variations on sparsity-inducing penalties

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

- This idea goes back to the 60s (Ivanov, John, Lavrent'ev, Tikhonov) where the penalty operated as a regularizer of solutions for ill-posed problems.

Ill-posed problem in statistics
High dimensional least square regression

- Assume d larger than n
- Then when solving the least square optimization problem, we observe that we have less equations than variables: this is the case of an *underdetermined* linear system.
- Another way to put this is to observe that $\mathbf{X}^T\mathbf{X}$ is not full rank, hence it is not invertible and there is an infinity of solutions.

The oldest regularizer in statistics
Ridge regression

- The Ridge estimator is the solution of the following penalized optimization problem: for any $\lambda > 0$,

$$\widehat{\beta}_\lambda \in \underset{\beta \in \mathbb{R}^d}{\arg\min} \left\{ \|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda \|\beta\|_2^2 \right\}$$

Derivation of ridge regression estimator

- We denote the objective function:

$$F(\beta) = (\mathbf{Y} - \mathbf{X}\beta)^T (\mathbf{Y} - \mathbf{X}\beta) + \lambda \beta^T \beta$$

- Thanks to convexity and differentiability of F, we obtain the solution by solving

$$\nabla F(\beta) = 2\mathbf{X}^T(\mathbf{X}\beta - \mathbf{Y}) + 2\lambda\beta = 0$$

- Solution:

$$\widehat{\beta}_\lambda = \left(\mathbf{X}^T\mathbf{X} + \lambda I_d\right)^{-1} \mathbf{X}^T\mathbf{Y}$$

  because $\mathbf{X}^T\mathbf{X} + \lambda I_d$ always invertible.

- Computation still painful for d very large...

Dual optimization problem
Formulation and KKT conditions

- Equivalent formulation of ridge regression optimization:

$$\min_{\beta \in \mathbb{R}^d, r \in \mathbb{R}^n} \left\{ \frac{1}{2}\|r\|^2 + \frac{\lambda}{2}\|\beta\|^2 \right\} \quad \text{subject to } r = \mathbf{X}\beta - \mathbf{Y}$$

- Lagrange formulation with multiplier vector α

$$\mathcal{L}(\beta, r, \alpha) = \frac{1}{2}\|r\|^2 + \frac{\lambda}{2}\|\beta\|^2 + \alpha^T(r - \mathbf{X}\beta + \mathbf{Y})$$

- Karush-Kuhn-Tucker conditions: zeroing gradient wrt primal variables β, r, leads to:

$$\beta(\alpha) = \frac{1}{\lambda}\mathbf{X}^T\alpha \quad \text{and} \quad r(\alpha) = -\alpha$$

Dual optimization problem
Resolution

- Then, an equivalent formulation of ridge regression optimization is given by:

$$\mathcal{L}(\beta(\alpha), r(\alpha), \alpha) = \frac{1}{2}\|\alpha\|^2 + \frac{1}{2\lambda}\|\mathbf{X}^T\alpha\| + \alpha^T\left(-\alpha - \frac{1}{\lambda}\mathbf{X}\mathbf{X}^T\alpha + \mathbf{Y}\right)$$

- Solution:

$$\widehat{\alpha} = \lambda\left(\mathbf{X}\mathbf{X}^T + \lambda I_n\right)^{-1}\mathbf{Y} \quad \text{and} \quad \widehat{\beta} = \frac{1}{\lambda}\mathbf{X}^T\widehat{\alpha}$$

Dual optimization problem
Interpretation of the result

- The prediction on $x \in \mathbb{R}^d$ can be expressed in terms of α

$$x^T \widehat{\beta} = \frac{1}{\lambda} x^T \mathbf{X}^T \widehat{\alpha} = \frac{1}{\lambda} \sum_{i=1}^{n} \widehat{\alpha}_i x^T X_i$$

- We can use the identity:
  $\mathbf{X}^T \left(\mathbf{X}\mathbf{X}^T + \lambda I_n\right)^{-1} = \left(\mathbf{X}^T\mathbf{X} + \lambda I_d\right)^{-1} \mathbf{X}^T$ to check the solutions are the same.

- Important observation! Optimization and function evaluation only require the pairwise scalar product between x's and data points X_i's

Elastic Net
The best of LASSO and Ridge?

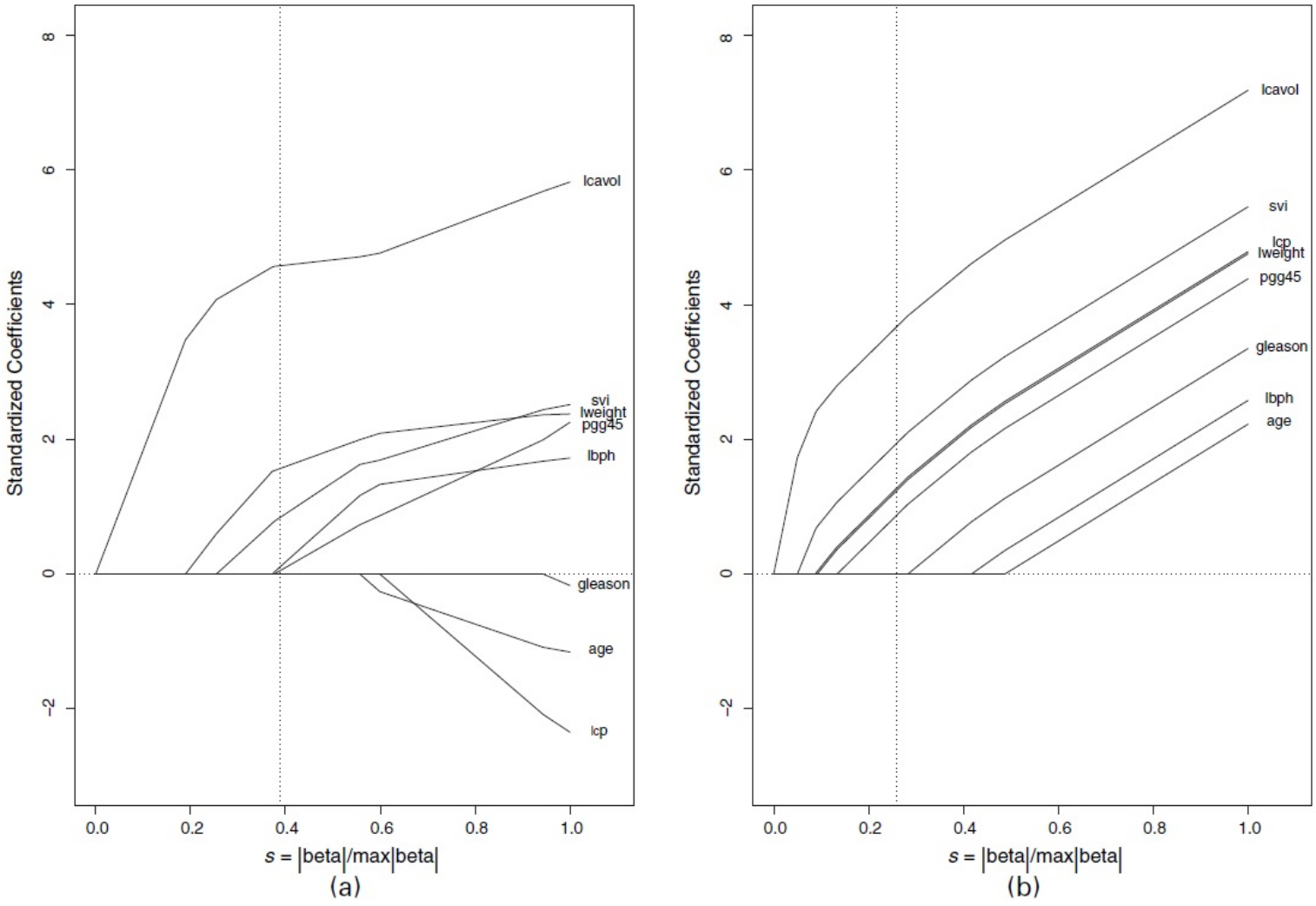
- Rationale (from [Zou and Hastie, 2005])

  (a) In the $p > n$ case, the lasso selects at most n variables before it saturates, because of the nature of the convex optimization problem. This seems to be a limiting feature for a variable selection method. Moreover, the lasso is not well defined unless the bound on the L_1-norm of the coefficients is smaller than a certain value.
  (b) If there is a group of variables among which the pairwise correlations are very high, then the lasso tends to select only one variable from the group and does not care which one is selected. See Section 2.3.
  (c) For usual $n > p$ situations, if there are high correlations between predictors, it has been empirically observed that the prediction performance of the lasso is dominated by ridge regression (Tibshirani, 1996).

- Combination of ℓ_1 and ℓ_2 penalties

$$\widehat{\beta}_\lambda \in \underset{\beta \in \mathbb{R}^d}{\arg\min} \left\{ \|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda \|\beta\|_1 + \mu \|\beta\|_2^2 \right\}$$

LASSO vs. Elastic Net
Comparison of regularization paths



Tuning the hyperparameters
Cross-validation

- How do we select the parameters λ and μ? These are called hyperparameters or smoothing parameters or regularization parameters.
- This is a universal problem in monitoring the overfitting effect of Machine Learning methods.
- The procedure of cross-validation will be developed later in the course.

Exercise
Comparison of the three penalties

- Consider the following toy problem: $Y \sim \mathcal{N}_1(\beta^*, 1)$ where β is a real-valued parameter ($d = 1$).
- Find the three estimators when minimizing the following three functions:

$$\text{(i)}\frac{1}{2}(Y-\beta)^2+\lambda,\ \text{(ii)}\frac{1}{2}(Y-\beta)^2+\lambda|\beta|,\ \text{(iii)}\frac{1}{2}(Y-\beta)^2+\lambda\beta^2$$

- Show a plot of the estimators as functions of the unconstrained LSE and explain the use of the following terminology for the penalized procedures: hard thresholding, soft thresholding, shrinkage.

B. Estimating nonlinear functions

From nonlinear to linear
Polynomial regression example

- Consider a polynomial regression in dimension $d = 2$: this corresponds to a linear model of dimension $d' = 7$ with feature vector:

$$\Phi(x_1, x_2) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2)^T$$

- Note that:

$$\Phi(x)^T\Phi(x') = (x^Tx' + 1)^2$$

- We call $K(x, x') = (x^Tx' + 1)^2$ a polynomial kernel. A kernel has the property to be represented as a scalar product in a high dimensional feature space. The feature space is the image of the original input space of dimension d through Φ. The feature space can be of huge dimension.

The magic of kernels
Kernel ridge regression

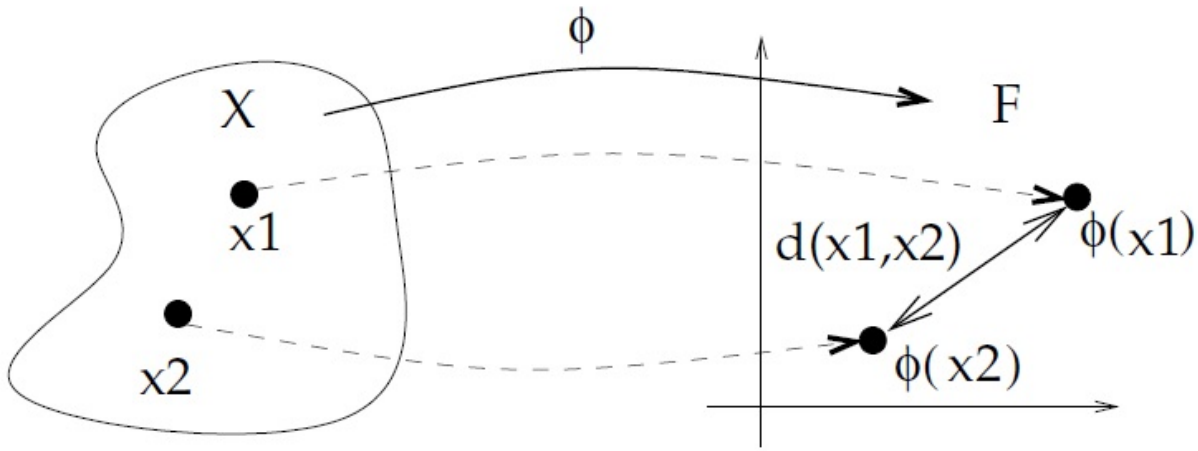
- In the linear case of ridge regression, we have seen that the only data-dependent quantities that matter in both problem formulation and evaluation of predictions are the pairwise scalar products of $X_i^T X_j$ and $x^T X_i$.
- We can basically replace any scalar product by the kernel evaluation of the considered pair without changing at all the algorithmic complexity of resolution. We are then able to estimate the parameters α_i of nonlinear functions of the form:

$$f(x) = \sum_{i=1}^{n} \alpha_i K(x, X_i)$$

Examples of basic kernels

- Linear: $\mathrm{K}(\mathrm{x}, \mathrm{z}) = \mathrm{x} \cdot z$
- Polynomial: $\mathrm{K}(\mathrm{x}, z) = (\mathrm{x} \cdot z)^{\mathrm{d}}$ or $\mathrm{K}(\mathrm{x}, z) = (1 + \mathrm{x} \cdot z)^{\mathrm{d}}$
- Gaussian: $\mathrm{K}(\mathrm{x}, z) = \exp\left[-\frac{\|x-z\|^2}{2\,\sigma^2}\right]$
- Laplace Kernel: $\mathrm{K}(\mathrm{x}, z) = \exp\left[-\frac{\|x-z\|}{2\,\sigma^2}\right]$

How a kernel defines a metric
Definition



$$
\begin{aligned}
d_K(\mathbf{x}_1, \mathbf{x}_2)^2 &= \| \Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2) \|_{\mathcal{H}}^2 \\
&= \langle \Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2), \Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2) \rangle_{\mathcal{H}} \\
&= \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_1) \rangle_{\mathcal{H}} + \langle \Phi(\mathbf{x}_2), \Phi(\mathbf{x}_2) \rangle_{\mathcal{H}} - 2 \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2) \rangle_{\mathcal{H}} \\
d_K(\mathbf{x}_1, \mathbf{x}_2)^2 &= K(\mathbf{x}_1, \mathbf{x}_1) + K(\mathbf{x}_2, \mathbf{x}_2) - 2K(\mathbf{x}_1, \mathbf{x}_2)
\end{aligned}
$$

Kernel engineering

- Specific kernels have been design to process structured data such as strings (text, DNA sequence...)
- Example of spectrum kernel used for DNA sequences:

Kernel definition

- The 3-spectrum of

$$\mathbf{x} = \texttt{CGGSLIAMMWFGV}$$

is:

$$(\texttt{CGG},\texttt{GGS},\texttt{GSL},\texttt{SLI},\texttt{LIA},\texttt{IAM},\texttt{AMM},\texttt{MMW},\texttt{MWF},\texttt{WFG},\texttt{FGV}) \ .$$

- Let $\Phi_u(\mathbf{x})$ denote the number of occurrences of u in $\mathbf{x}$. The k-spectrum kernel is:

$$K(\mathbf{x},\mathbf{x}') := \sum_{u \in \mathcal{A}^k} \Phi_u(\mathbf{x}) \Phi_u(\mathbf{x}') \ .$$

Kernel machine learning estimation
Is it doable?

- Nice modeling properties of kernel functions
- The question is whether the penalized optimization in the sense of least squares is feasible?

C. Generalizations

Application to other estimation problems

Penalized optimization
What about other variations?

- Until now: hypothesis class with linear functions $h \in \mathcal{H}$ and variations on the penalties

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

- From now on: play with other losses affects the Training error

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

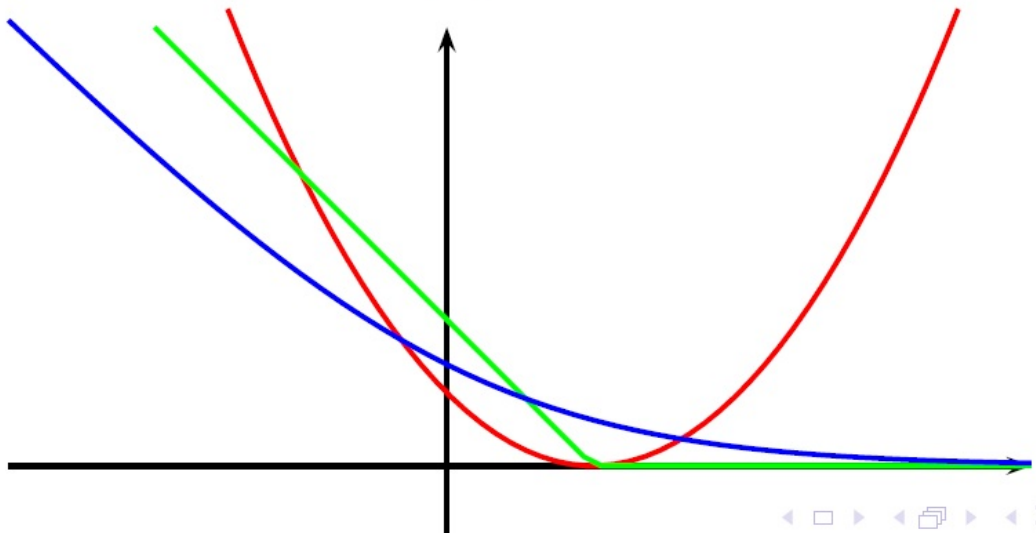
Using other loss functions

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



Next sessions

- Other tasks: linear models for classification
- The issue of representation: feature engineering, variable selection , representation learning
- From linear to nonlinear models: what can be saved?