Linear models, validation and selection

Formation ENSTA-ParisTech
Conférence IA

Florent Chatelain * Olivier Michel *

* Univ. Grenoble Alpes, GIPSA-lab

10-13 February 2020

Model based approaches

Reminder on Supervised Learning

- input data $X \in \mathbb{R}^p$
- reponse Y to be predicted
- training set $(X_1, Y_1), \ldots, (X_n, Y_n)$

Discriminative models

Direct learning of $P(Y|X)$, e.g. generalized linear models s.t.

- Linear regression
- Logistic regression ($\leftarrow$ classification tasks)
- ...

Outline

- Linear Models
  - Linear regression
  - Linear model for classification: Logistic regression
- Model Validation
  - Cross-Validation
  - Information Criterion
- Model Selection
  - Subset selection
  - Regularization and shrinkage methods
    - Ridge regression
    - Lasso estimator
- Applications
  - prostate data
  - Heart diseases data

Linear Regression Problem

- $X_i = (X_{i,1}, \ldots, X_{i,p})^T \in \mathbb{R}^p$,
- $Y_i \in \mathbb{R}$,

for $i = 1, \ldots, n$ (sized n training set)

Linear Regression Model

$$Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_{i,j} + \sigma \epsilon_i, \quad \text{for } i = 1, \ldots, n,$$

- ϵ_i is a centered with unit variance ($E[\epsilon_i] = 0$, var$(\epsilon_i) = 1$) white noise
- β_0 is the "intercept" (reduces to the ordinate at the origin when $p = 1$)
- $\beta \equiv (\beta_0, \ldots, \beta_p) \in \mathbb{R}^{p+1}$ is the coefficient vector

Objective: estimation of $\beta \leftarrow$ supervised learning problem

Linear Regression Problem (Cont'd)

Linear Regression Model

$$Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_{i,j} + \sigma\epsilon_i, \qquad \text{for } i = 1, \ldots, n,$$

Remark: model linear w.r.t. $\beta \equiv (\beta_0, \ldots, \beta_p) \in \mathbb{R}^{p+1}$, but not necessarily linear w.r.t.

- the inputs X_i: we can add non linear predictors $h(X_1, \ldots, X_p)$ in the model, e.g. X_i^2, $X_i X_j$...
- the outputs Y_i: we can introduce a non linear link function $\leftarrow$ generalized linear model, e.g. logistic regression

Linear model: Keep it simple!

Simple linear approach may seem overly simplistic

- true regression functions are never linear

+ extremly useful, both conceptually and practically

Practically

Gorge Box, 60': "Essentially, all models are wrong, but some are very useful"

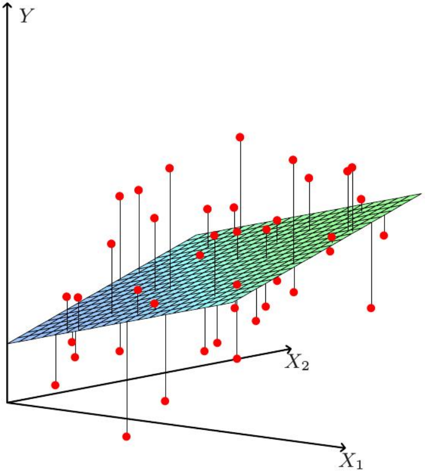
☞ *Simple is actually very good*: works very well in a lot of situations by capturing the main effects (which are generally the most interesting)

Conceptually

Many concepts developped for the linear problem are important for a lot of the supervised learning techniques

☞ Although it is nerver correct, a linear model serves as a good and interpretable approximation of the unknown true function $f(X)$

Least Squares (LS) Estimator



Linear least squares fitting with $X \in \mathbb{R}^2$

LS estimate defined by minimizing the Residual Sum of Squares (RSS)

$$\widehat{\beta} = \arg\min_{\beta} \underbrace{\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2}_{\mathrm{RSS}(\beta)}$$

- RSS(β) $\propto$ training error rate for quadratic loss

Least Squares Estimator (Cont'd)

$$\widehat{\beta} = \arg\min_{\beta} \mathrm{RSS}(\beta), \quad \text{where } \mathrm{RSS}(\beta) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2$$

Matrix expression of RSS

$$\mathrm{RSS}(\beta) = ||Y - X\beta||_2^2,$$

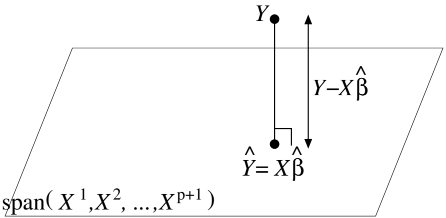
$$\text{where } Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n, \qquad X = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n,1} & \dots & x_{n,p} \end{pmatrix} \in \mathbb{R}^{n \times (p+1)}$$

LS Estimator derivation

$\widehat{Y} = X\widehat{\beta}$ is the prediction in the space spanned by the column vectors of X such that the euclidean error norm $||Y - X\widehat{\beta}||_2$ is minimized

Orthogonality principle

Let X^j be the jth column of X



for $j = 1, \ldots, p+1$

$$\langle X^j, Y - X\widehat{\beta}\rangle = (X^j)^T \left(Y - X\widehat{\beta}\right) = 0,$$

$$\Leftrightarrow \quad X^T \left(Y - X\widehat{\beta}\right) = 0,$$

$$\Leftrightarrow \quad \left(X^T X\right)\widehat{\beta} = X^T Y$$

LS Estimator computation

Assumption: $\operatorname{rank} X = p + 1$, hence $X^T X$ is invertible

Analytical expression

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

Numerical computation in high dimension

When $p > 10^3$ or $p > 10^4$, too expansive to compute $\left(X^T X\right)^{-1}$ ... More efficient to use a numerical procedure to minimize the criterion $J(\beta) \equiv \frac{1}{2}\mathrm{RSS}(\beta)$, e.g. steepest descent

$$\beta_{k+1} = \beta_k - \alpha_k \nabla_\beta \ J(\beta_k),$$

where step size $\alpha_k \in \mathbb{R}$ is the *learning rate*, and descent direction is computed as

- batch gradient $\nabla_\beta \ J(\beta) = X^T X \beta - X^T Y$
- stochastic gradient $\nabla_\beta \ J(\beta) \approx X_i^T X_i \beta - X_i^T Y_i$ for $i = 1, \ldots, n$ (scan of the training set) $\leftarrow$ cheaper than batch one for a single iteration
- mini-batch gradient: tradeoff between batch and stochastic gradients

Linear model for classification: Logistic regression

Classification problem $Y \in \mathcal{Y} \leftarrow$ discrete set

Binary classification problem: $\mathcal{Y} = \{1, 2\}$

Consider the following model

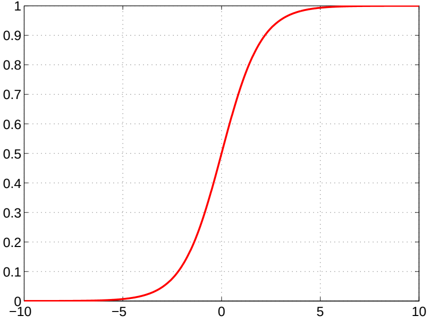
$$\Pr(Y_i = 1 | X_i = x_i) = \phi(x_i^T \beta) = \frac{\exp(x_i^T \beta)}{1 + \exp(x_i^T \beta)},$$

where

- $x_i = (1, x_{i,1}, \dots, x_{i,p})^T \in \mathbb{R}^{p+1} \leftarrow$ intercept term included by default,
- $\phi: \; u \in \mathbb{R} \mapsto \frac{\exp(u)}{1+\exp(u)} \in (0, 1)$ is the logistic function: maps a real value to a probability

Logistic function

$$
\begin{aligned}
\phi(x) \; &: \mathbb{R} \to]0,1[\\
& u \mapsto \frac{\exp u}{1+\exp u} = \frac{1}{1+\exp(-u)}.
\end{aligned}
$$



Logit link function

Consider

- $p_i \equiv \Pr(Y_i = 1|X_i = x_i) = \phi(x_i^T \beta)$
- ϕ^{-1} : $p \in (0,1) \mapsto \log \frac{p}{1-p} \in \mathbb{R}$ is the logit function

Generalized linear model

- Linear equation w.r.t. β,

$$\text{logit}\,(p_i) = x_i^T \beta,$$

- additional nonlinear constraint:

$$\Pr(Y_i = 2|X_i = x_i) = 1 - \underbrace{\Pr(Y_i = 1|X_i = x_i)}_{p_i} = \frac{1}{1 + \exp\,(x_i^T \beta)}$$

- Maximum Likelihood Estimates

$$\widehat{\beta} = \arg\min_{\beta} -\ell(\beta)$$

where $\ell(\beta)$ is the log-likelihood (here logistic, but normal model yields LSE for linear regression)

Outline

Linear Models
- Linear regression
- Linear model for classification: Logistic regression

Model Validation
- Cross-Validation
- Information Criterion

Model Selection
- Subset selection
- Regularization and shrinkage methods
  - Ridge regression
  - Lasso estimator

Applications
- prostate data
- Heart diseases data

Outline

Linear Models
- Linear regression
- Linear model for classification: Logistic regression

Model Validation
- Cross-Validation
- Information Criterion

Model Selection
- Subset selection
- Regularization and shrinkage methods
  - Ridge regression
  - Lasso estimator

Applications
- prostate data
- Heart diseases data

Limitations of LS estimator

Problem

when rank $X < p+1$, or when X has singular values close to zero, then $X^T X$ is no more invertible, or ill conditioned (eigenvalues close to zero)...

Causes

- redundant or nearly-collinear predictors, e.g. $X^k \approx aX^l + b$, where X^j is the jth column of X
- high dimensional problem where $p \approx n$ (or $p > n$)

Effects

no single, or stable, solution for $\widehat{\beta}$

- high variance of $\widehat{\beta}$ as an eigenvalue λ_i of $X^T X$ is close to zero ($||\widehat{\beta}|| \to +\infty$ as $\lambda_i \to 0$),
- true error rate explodes since a small perturbation in the training set yields a substantially different estimate $\widehat{\beta}$ and prediction rule $\widehat{y} = x^T \widehat{\beta}$

☞ over-fitting problem

Instability of LSE: Deconvolution illustration

- $y \in \mathbb{R}^n$ with $n = 256^2$, $\beta \in \mathbb{R}^p$ with $p = 256^2$,
- $X \in \mathbb{R}^{n \times p} \leftarrow$ sized $(256^2) \times (256^2)$ matrix...

$\beta \leftarrow$ original image

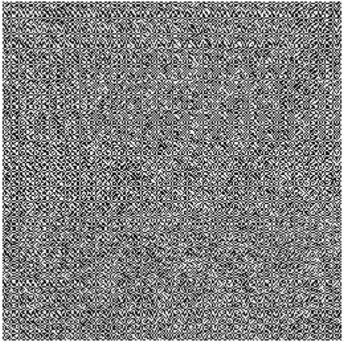
$y = X\beta \leftarrow$ blurred image

Instability of LSE: Deconvolution illustration

- $y \in \mathbb{R}^n$ with $n = 256^2$, $\beta \in \mathbb{R}^p$ with $p = 256^2$,
- $X \in \mathbb{R}^{n \times p} \leftarrow$ sized $(256^2) \times (256^2)$ matrix...

$\beta \leftarrow$ original image

$y = X\beta \leftarrow$ blurred image



$\widehat{\beta} = (X^T X)^{-1} X^T y \leftarrow$ LS estimate

Reminder on Train and Test Errors

- Loss-function
  - Classification: $L(y, \hat{y}) = 0$ if $y = \hat{y}$ else $1 \leftarrow$ 0-1 loss
  - Regression: $L(y, \hat{y}) = (y - \hat{y})^2 \leftarrow$ quadratic loss
- **Train error**: average loss over the training sample

$$\text{Err}_{\text{train}} = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{y}_i)$$

- **Test/Prediction error**: average loss over a new test sample $\rightarrow$ Generalization error
- General picture:

$$\text{Err}_{\text{test}} \approx \text{Err}_{\text{train}} + O$$

O would be the average *optimism* (overfitting problem!)

Model Selection vs Model Validation

Model selection

- Estimate the best set of hyperparameters
- Estimate the performance of differents models

Model Validation

Estimate the generalization error on unseen/test sample

Model Selection vs Model Validation

Model selection

- Estimate the best set of hyperparameters
- Estimate the performance of differents models

Model Validation

Estimate the generalization error on unseen/test sample

K-fold Cross-Validation (CV): Principle

- Method to estimate prediction error using the training sample
- Based on splitting the data in K-folds, here $K = 5$:

$\mathrm{Err}_1(\hat{f}_1, \lambda)$	Validation	Train	Train	Train	Train
$\mathrm{Err}_2(\hat{f}_2, \lambda)$	Train	Validation	Train	Train	Train
$\mathrm{Err}_3(\hat{f}_3, \lambda)$	Train	Train	Validation	Train	Train
$\mathrm{Err}_4(\hat{f}_4, \lambda)$	Train	Train	Train	Validation	Train
$\mathrm{Err}_5(\hat{f}_5, \lambda)$	Train	Train	Train	Train	Validation

  where λ are some hyperparameters of the model/method

- Estimate of Test error:

$$\mathrm{CV}(\hat{f}, \lambda) = \sum_{k=1}^{K} \mathrm{Err}_k(\hat{f}_k, \lambda)$$

K-fold Cross-Validation (CV): Algorithm

`Input`: input variables X (dimension $n \times p$), responses y (dim. n), number of folds k

`Divide` randomly the set $\{1, 2, \ldots, n\}$ in k subsets (i.e., folds) of roughly equal sizes (e.g., size equals to the integer part of n/k with a little smaller last part if n is not a multiple of k) denoted as $F_1, \ldots, F_k$

`for i = 1 to k:`

- `Form` the validation set (X_{val}, y_{val}) where the indexes of the X and y variables belongs to the ith fold F_i
- `Form` the training set (X_{train}, y_{train}) where the indexes of the X and y variables belongs to all the folds except F_i
- `Train` the algorithm/model on the training set (X_{train}, y_{train})
- `Apply` the resulting prediction rule on the input X_{val} of the validation set
- `Compute` the error rate on the validation set based on the predictions and the true responses y_{val}

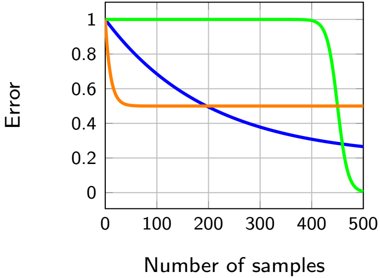
`Output`: the average error rate computed over all the k folds

Pratical advices

- K? Usually K=5 or 10 is a good trade-off (K=n is called leave-one-out)

	Bias	Variance
K low	High	Low
K high	Low	High
K = n	Low	Very High

- Be careful to the learning curve



- Model should be trained completely for each fold (i.e., data normalization, optimization, etc ...)
- Notebooks: `N1_validation_and_model_selection.ipynb`, `N2_nested_cross_validation_iris.ipynb`

Information Criterion

Penalized log-likelihood criterion

$$C(K) = -\hat{\ell}(x;\lambda) + \mathrm{pen}(\lambda, n)$$

- $\hat{\ell}(x;\lambda) \equiv \ell(x;\hat{\beta}_\lambda)$ with $\hat{\beta}_\lambda$ the MLE of the model parameters
- $\hat{\ell}(x;\lambda) \propto \mathrm{RSS}(\hat{\beta}_\lambda)$ for linear regression (normal model)
- λ are the hyperparameters of the model that drive its complexity (e.g. number k of variables that enter in the model)

Trade-off between two terms (to minimize)

- $-\hat{\ell}(x;\lambda)$: fidelity to the data (likelihood)
- $\mathrm{pen}(\lambda, n)$: low complexity of the model

Bayesian Information Criterion (BIC)

Asymptotic ($n \gg k_\lambda$) criterion for Bayesian models (i.e. with a prior on the model parameters)

$$\mathrm{pen}(k_\lambda, n) = \frac{1}{2} k_\lambda \log(n)$$

- n is the size of the dataset
- k_λ is the *effective* number of parameters for the λ hyperparameter

$$\mathrm{BIC}(\lambda) = -2\hat{\ell}(x;\lambda) + k_\lambda \log(n)$$

☞ Model comparison: select model $\hat{\ell}_j(x;\lambda_j)$ with minimal BIC

☞ Hyperparmeter tuning: choose value $\widehat{\lambda}$ that minimizes BIC

Akaïke Information Criterion (AIC)

Other popular asymptotic ($n \gg k_\lambda$) criterion

$$\mathrm{AIC}(\lambda) = -2\hat{\ell}(x; \lambda) + 2k_\lambda$$

- k_λ is the *effective* number of parameters for the λ hyperparameter
- Comparison with BIC: $\log(n)\, k_\lambda$ replaced by $2\, k_\lambda$

☞ Variant less aggressive (as long as $\log(n) > 2$, where n is the sample size)

Outline

Linear Models
- Linear regression
- Linear model for classification: Logistic regression

Model Validation
- Cross-Validation
- Information Criterion

Model Selection
- Subset selection
- Regularization and shrinkage methods
  - Ridge regression
  - Lasso estimator

Applications
- prostate data
- Heart diseases data

Outline

Linear Models
- Linear regression
- Linear model for classification: Logistic regression

Model Validation
- Cross-Validation
- Information Criterion

Model Selection
- Subset selection
- Regularization and shrinkage methods
  - Ridge regression
  - Lasso estimator

Applications
- prostate data
- Heart diseases data

Subset selection

Motivation

- interpretation: with a large number of predictors, we often would like to determine a smaller subset that exhibit the main (strongest) effects.
- prediction accuracy: can always be improved by shrinking or setting some coefficients to zero, thus preventing from overfitting

Principle

- Retain only a subset of the variables, and eliminate the rest from the model
- ☞ different strategies for choosing the subset

Forward-stepwise selection

Greedy algorithm producing an increasing nested sequence of models:

- starts with the intercept
- then sequentially adds into the model the predictor that most improves the fit

Produces a sequence of models indexed by k, the subset size, which must be determined.

☞ choosing k? cross-validation, AIC/BIC, 'significance' criterion, ...

Remarks

- suboptimal procedure, but we can always compute the forward stepwise sequence (even when $p \gg n$)
- similar to *orthogonal matching pursuit* in signal processing

Backward-stepwise selection

Greedy algorithm producing a decreasing nested sequence of models:

- starts with the full model,
- deletes the predictor that has the least impact on the fit.

How to choose the candidate for dropping: variable X_j with the smallest absolute Z-score

$$Z_j \equiv \frac{\hat{\beta}_j}{\widehat{\mathrm{sd}}(\hat{\beta}_j)},$$

where $\widehat{\mathrm{sd}}(\hat{\beta}_j)$ is the approx. standard error for $\hat{\beta}_j$

Remarks

- Backward selection can only be used when $n > p$, while forward stepwise can always be used.

Random Forests

Variable importance measure

- For bagged/RF regression trees, we record the total amount that the RSS is decreased due to splits over a given predictor, averaged over all B trees. A large value indicates an important predictor
- Similarly, for bagged/RF classification trees, we add up the total amount that the Gini index is decreased by splits overa given predictor, averaged over all B trees.

☞ Greedy selection procedures, e.g. *Backward-stepwise selection*

Regularization: shrinkage

Idea: introducing a little bias in the estimation of β may lead to a substantial decrease in variance and, hence, in the true error rate

Penalized regression

Regularize the estimation problem by introducing a penalization term for β

$$\widetilde{\beta} = \arg\min_{\beta} \left[\mathrm{RSS}(\beta) + \lambda\,\mathrm{Pen}(\beta)\right]$$

- $\mathrm{RSS}(\beta)$ is the *fidelity term* to the training set (replace with the opposite log-likehihood $-\ell(\beta)$ for generalized linear model, e.g. logistic regression)
- $\mathrm{Pen}(\beta)$ is the *a priori* to regularize the solution,
- $\lambda > 0$ is the penalization coefficient

Choosing λ: tradeoff between overfitting (small λ) and underfitting (large λ)

☞ standard practice is to use cross-validation to estimate an optimal λ for the test error rate (but AIC/BIC can also be used)

Ridge regression

Penalization in the (squared) ℓ_2 sense:

$$\mathrm{Pen}(\beta) \equiv \beta^T \beta = ||\beta||_2^2, \quad \leftarrow \text{Tychonov regularization}$$

$\widetilde{\beta}$ is thus obtained by minimizing

$$\begin{aligned}\mathrm{RSS}(\beta)+\lambda\,\mathrm{Pen}(\beta) &= (Y-X\beta)^T(Y-X\beta)+\lambda\beta^T\beta,\\ &= \left(\beta-(X^TX+\lambda I)^{-1}X^TY\right)^T\left(X^TX+\lambda I\right)\left(\beta-(X^TX+\lambda I)^{-1}X^TY\right)+\mathrm{Cst},\end{aligned}$$

Ridge estimator: $\widetilde{\beta} = (X^TX + \lambda I)^{-1}X^TY$

Remark

similar to LS estimator, with an additional 'ridge' on the diagonal of X^TX

- $X^TX + \lambda I$ has all its eigenvalues greater than $\lambda > 0$, $\leftarrow$ ensures that $\widetilde{\beta}$ is always defined, and stable for large enough λ
- ☞ when $\lambda \to 0$, then $\widetilde{\beta} \to \widehat{\beta}$,
- ☞ when $\lambda \to +\infty$, then $\widetilde{\beta} \to 0$

Ridge Regression: deconvolution illustration

- $y \in \mathbb{R}^n$ with $n = 256^2$, $\beta \in \mathbb{R}^p$ with $p = 256^2$,
- $X \in \mathbb{R}^{n \times p} \leftarrow$ sized $(256^2) \times (256^2)$ matrix...

$\beta \leftarrow$ original image

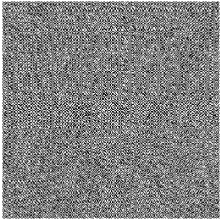
$y = X\beta \leftarrow$ blurred image

Ridge Regression: deconvolution illustration

- $y \in \mathbb{R}^n$ with $n = 256^2$, $\beta \in \mathbb{R}^p$ with $p = 256^2$,
- $X \in \mathbb{R}^{n \times p} \leftarrow$ sized $(256^2) \times (256^2)$ matrix...

$\beta \leftarrow$ original image

$y = X\beta \leftarrow$ blurred image



$\widehat{\beta} = (X^T X)^{-1} X^T y \leftarrow$ LS estimate

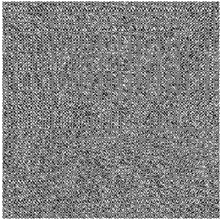
Ridge Regression: deconvolution illustration

- $y \in \mathbb{R}^n$ with $n = 256^2$, $\beta \in \mathbb{R}^p$ with $p = 256^2$,
- $X \in \mathbb{R}^{n \times p} \leftarrow$ sized $(256^2) \times (256^2)$ matrix...

$\beta \leftarrow$ original image

$y = X\beta \leftarrow$ blurred image

$\widetilde{\beta} = (X^T X + \lambda I)^{-1} X^T y \leftarrow$ ridge estimate



$\widehat{\beta} = (X^T X)^{-1} X^T y \leftarrow$ LS estimate

Regularization by promoting sparsity

Sparse representations/approximations

A representation, or an approximation, is said to be sparse when most of the coefficients (in a given basis) are zero

'Bet on Sparsity' principle

Sparsity is a good option in high dimension!

- if the sparsity assumption does not hold, no method will be able to recover the underlying model in high dimension where $p \approx n$ or $p > n$
- but if the sparsity assumption holds true, then the parameters can be efficiently estimated by a method that promotes sparsity

☞ Occam's razor or KISS (keep it simple, stupid) principles: same idea that simpler models are preferable than more complex ones

Application to the regression problem

choosing a penalization function $\mathrm{Pen}(\beta)$ that promotes the sparsity of β (i.e. with many components $\beta_j = 0$ for $j = 1, \ldots, p+1$) $\leftarrow$ Lasso estimator

Lasso ('least absolute shrinkage and selection operator') estimator

Definition

$$\widetilde{\beta}_{\mathrm{lasso}} = \arg\min_{\beta} \left[\mathrm{RSS}(\beta) + \lambda \, ||\beta||_1 \right],$$

where $||\beta||_1 = \sum_{j=1}^{p+1} |\beta_j|$ is the ℓ_1 norm

- no analytical expression of $\widetilde{\beta}_{\mathrm{lasso}}$
- but convex optimization problem where very efficient numerical procedures are available to compute $\widetilde{\beta}_{\mathrm{lasso}}$

Lasso advantages

Converges to a generally sparse solution, i.e. such that $\beta_k = 0$ for a subset of index k

- ☞ the less significant variables are explicitly discarded
- ☞ similar stability than ridge estimator + variable selection

- Tensorflow Playground: `https://playground.tensorflow.org/`
- Notebook: `N3_curse_dimensionality.ipynb`

Penalization with ℓ_1 and ℓ_2 norms: geometrical interpretation

- Least Squares estimator: $\widehat{\beta} = \text{arg min } \mathrm{RSS}(\beta)$,
- Regularized estimator: $\widetilde{\beta} = \text{arg min}\,(\mathrm{RSS}(\beta) + \lambda \mathrm{Pen}(\beta))$
  $\Leftrightarrow \widetilde{\beta} = \text{arg min } \mathrm{RSS}(\beta)$ under the constraint $\mathrm{Pen}\left(\widetilde{\beta}\right) \leq s(\lambda)$.

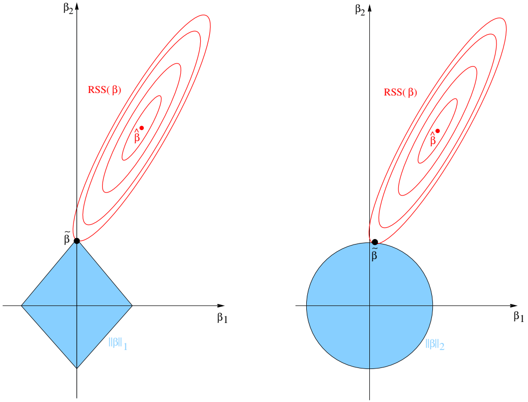


Illustration in dimension $p = 2$: $\beta = (\beta_1, \beta_2)^T$

- Lasso (ℓ_1 norm) : $\mathrm{Pen}\,(\beta) = ||\beta||_1 = \sum_{k=1}^{p} |\beta_k|$
- Ridge regression (ℓ_2 squared norm) :
  $\mathrm{Pen}\,(\beta) = ||\beta||_2^2 = \beta^T \beta$

ℓ_1 norm promotes the sparsity of the estimator: the less significant predictors are explicitly discarded (coeffs β_k are zero) $\leftarrow$ model selection

Outline

Linear Models
- Linear regression
- Linear model for classification: Logistic regression

Model Validation
- Cross-Validation
- Information Criterion

Model Selection
- Subset selection
- Regularization and shrinkage methods
  - Ridge regression
  - Lasso estimator

Applications
- prostate data
- Heart diseases data

Application: prostate data

Stamey et al. (1989) study to examine the association between prostate specific antigen (PSA) and several clinical measures that are potentially associated with PSA in men. Objective is to predict the Log PSA from eight variables

- lcavol: Log cancer volume
- lweight: Log prostate weight
- age: The man's age
- lbph: Log of the amount of benign hyperplasia
- svi: Seminal vesicle invasion; 1=Yes, 0=No
- lcp: Log of capsular penetration
- gleason: Gleason score
- pgg45: Percent of Gleason scores 4 or 5

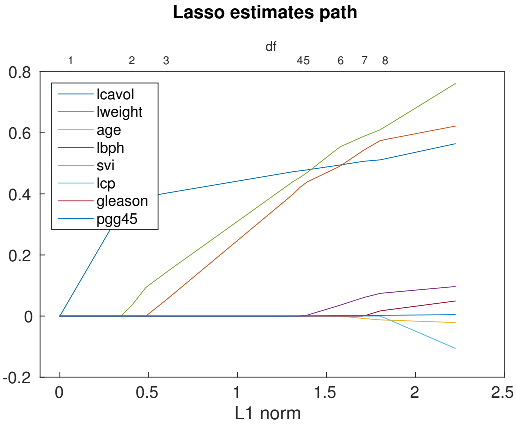
Application : prostate data

Model selection: ℓ_1 penalization (Lasso)

$$\widetilde{\beta}(\lambda) = \arg\min_{\beta} \mathrm{RSS}(\beta) + \lambda||\beta||_1,$$

$\rightarrow$ function of λ where less significant variables are explicitly discarded

Path of the ℓ_1-penalized coefficients vs $||\widetilde{\beta}(\lambda)||_1$



Choosing λ

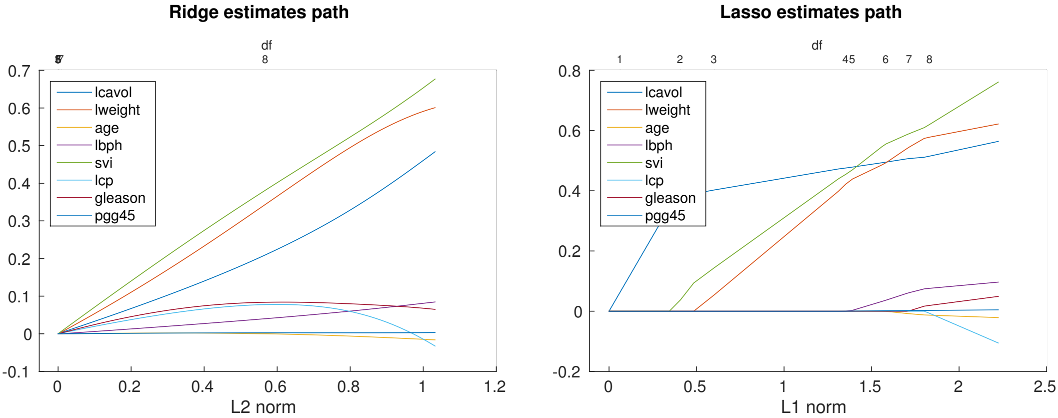
- large $||\widetilde{\beta}(\lambda)||_1$ (small λ) $\rightarrow$ overfitting
- small $||\widetilde{\beta}(\lambda)||_1$ (large λ) $\rightarrow$ underfitting
- $0.48 \leq ||\widetilde{\beta}(\lambda)||_1 \leq 1.43 \rightarrow$ 3 predictors (lcavol,svi,lweight)

$||\widetilde{\beta}(\lambda)||_1 = 1.06$ ($\lambda = 0.21$) estimated by cross validation

Notebook: `N4_lasso_model_selection.ipynb`

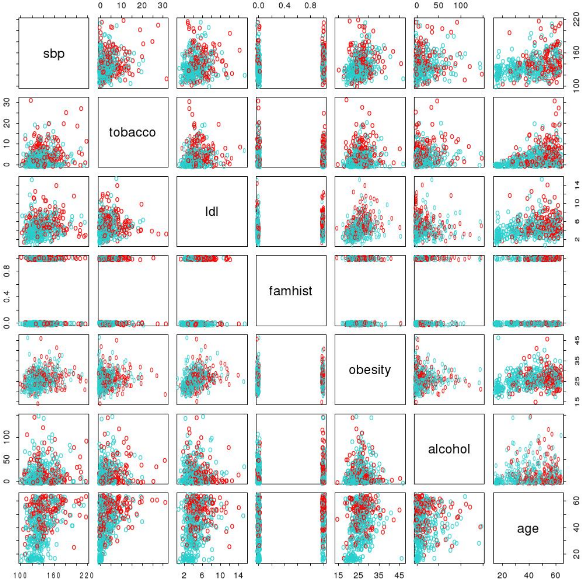
Application : prostate data

Comparison of ridge and lasso estimators



Path of the penalized coefficients as a function of $||\widetilde{\beta}(\lambda)||$

Application: South African coronary heart disease (CHD)



Matrix of the predictor scatterplots

- each plot $\equiv$ pair of risk factors
- 160 cases / 302 controls
- *ldl*: $\sim$ cholesterol, *sbp*: systolic blood pressure

Application: South African CHD (Cont'd)

Logistic regression fit

	Coefficient	Std. Error	Z score
(Intercept)	−4.130	0.964	−4.285
sbp	0.006	0.006	1.023
tobacco	0.080	0.026	3.034
ldl	0.185	0.057	3.219
famhist	0.939	0.225	4.178
obesity	−0.035	0.029	-1.187
alcohol	0.001	0.004	0.136
age	0.043	0.010	4.184

- A Z score ($\equiv$ Coeff / Std. Error) > 2 in absolute value is significant at the 5% level.

Must be interpreted with caution!

- systolic blood pressure (sbp) is not significant!
- nor is obesity (conversely, < 0 coefficient)!

→ result of the strong correlations between the predictors

Application: South African CHD (Cont'd)

Model selection: greedy backward procedure

Find the variables that are sufficient for explaining the CHD outputs

- drop the least significant predictor, and refit the model
- repeat until no further terms can be dropped ← backward selection

Logistic regression fit with backward model selection procedure

	Coefficient	Std. Error	Z score
`(Intercept)`	−4.204	0.498	−8.45
`tobacco`	0.081	0.026	3.16
`ldl`	0.168	0.054	3.09
`famhist`	0.924	0.223	4.14
`age`	0.044	0.010	4.52

Interpretations

- Tobacco is measured in total lifetime usage in kilograms, with a median of 1kg for the controls and 4.1kg for the cases
- An increase of 1kg $\Rightarrow$ increase of the CHD proba of $\exp(0.081) = 1.084$ or 8.4% (confidence interval at 95% $[1.03, 1.14]$)

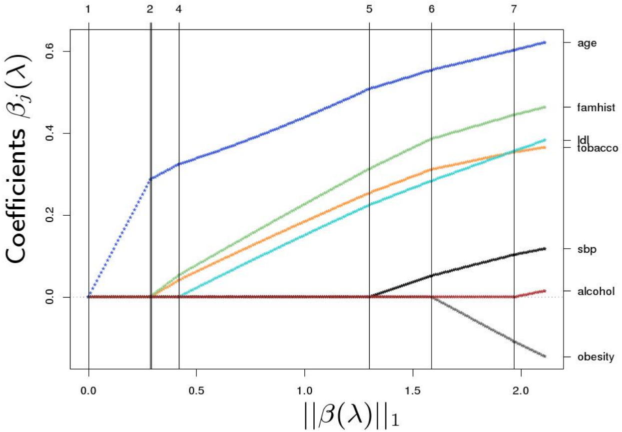
Application: South African CHD (Cont'd)

Model selection: ℓ_1 penalization (Lasso type method)

$$\widetilde{\beta}(\lambda) = \arg\min_{\beta} -\ell(\beta) + \lambda||\beta||_1,$$

$\rightarrow$ function of λ where less significant variables are explicitly discarded

Path of the des coefficients ℓ_1-penalized coefficients as a function of $||\hat{\beta}(\lambda)||_1$



Choosing λ

- large $||\widetilde{\beta}(\lambda)||_1$ (small λ) $\rightarrow$ overfitting
- small $||\widetilde{\beta}(\lambda)||_1$ (large λ) $\rightarrow$ underfitting
- $0.43 \leq ||\widetilde{\beta}(\lambda)||_1 \leq 1.3 \rightarrow$ 4 same predictors than backward selection procedure

Notebook:
`N5_LR_heart_diseases_SA.ipynb`

Conclusions

Generalized Linear Models

Learning of the prediction rule based on a model of Y given X

☞ Linear regression, Logistic regression

Properties

- Simplicity: useful to capture the main effects
- Interpretability
- Shrinkage and Selection procedures

☞ extends to non-linear methods

Notebook: `N6_select_features_boston.ipynb`

Conclusions (Cont'd)

Model Selection

Model selection methods are an essential tool for data analysis, especially for big datasets involving many predictors

- Information criterion (AIC, BIC) make an adjustment to the training error to account for the overfitting bias and can be used to select among models with different numbers of variables,
- AIC/BIC are simple, but are known to be inacurate in high dimension (e.g. when the number of predictors is comparable or greater than the sample size)
- Cross-validation has an advantage relative to AIC, BIC in that it provides a direct estimate of the test error. Can be used for model selection and/or hyperparameter tuning