Lecture 7 : Methods for Regression

Biased regression

K. Meziani

Dauphine | PSL
UNIVERSITÉ PARIS

Section 1

1. Introduction

Statistical Learning

Given a training sample $(x_i, Y_i)_{i=1}^n$, we want to predict futur value of Y.

1. Set a family of model $\mathcal{F}$.

$$\text{Here,} \quad \mathcal{F} = \{x^\top\beta, \beta \in \mathbb{R}^p\}$$

2. Pick $\widehat{f} \in \mathcal{F}$ based on the training set

$$\text{Here,} \quad \widehat{f}(x) = \widehat{y} = x^\top\widehat{\beta}, \quad \text{where} \quad \widehat{\beta} = (X^\top X)^{-1}X^\top Y$$

3. Asses the accuracy of the model $\widehat{f}$ on the new observations (test sample)

Generalization property

Generalization property

A good model should predict futur value correctly

$$R(\widehat{f}) = \mathbb{E}\|Y - \widehat{f}(X)\|^2 = \sigma^2 + bias^2(\widehat{f}) + \mathrm{Tr}(\mathbb{V}\mathrm{ar}(\widehat{f}))$$

The risk is **not directly computable** since the law (X, Y) is unknown.

$$\text{Here,} \quad R(\widehat{f}) = \mathbb{E}\|Y - \widehat{f}(X)\|^2 = \sigma^2 + bias^2(\widehat{f}) + \sigma^2 p$$

Generalization property

Empirical risk

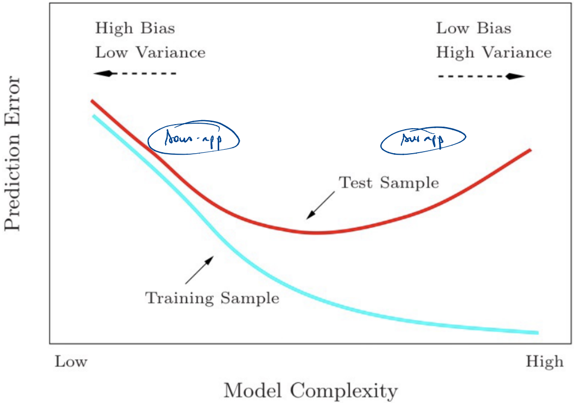
For $\widehat{f}$, we can estimate $R(\widehat{f})$ from a test sample $(X_t, Y_t)_{t=1,\cdots,T}$

$$\text{Here,} \quad \widehat{R}(\widehat{f}) = \frac{1}{T}\sum_{t=1}^{T}\left(Y_t - \widehat{f}(X_t)\right)^2$$

Bias/Variance compromise

- Define the error (risk) of an estimator:

$$R(\widehat{f}) = \sigma^2 + \text{bias}(\widehat{f}) + \text{Var}(\widehat{f})$$



Comments

Consider the regression linear model is correct. Then, Gauss-Markov theorem guarantee that the OLSE is the best linear unbiased estimator.

Question : When does it make sense to use a low variance biased estimator?

Consider the Gram matrix $G = \frac{1}{n} X^\top X$, then

$$\mathbb{V}\mathrm{ar}(\widehat{\beta}) = \sigma^2 (X^\top X)^{-1} = \sigma^2 \frac{G^{-1}}{n}$$

Comments

Estimating β^* is an inverse problem whose difficulty is measured by the condition number depending on the eigenvalues of G, $\lambda(G)$,

$$\kappa(G) = \frac{\lambda_{\max}(G)}{\lambda_{\min}(G)} > 0$$

- G is well-conditioned if $\kappa(G)$ is small. This guarantees a good performance of the OLSE.
- Otherwise, G is said to be ill-conditioned and the OLSE performs poorly.
- Strongly correlated covariates implies that G is ill-conditioned and we may need to use other estimators than OLSE.
- Large p increases complexity of the model but may cause numerical instability and/or overfitting

First strategy : model selection

Model selection :

- step by step methods look for a small number of active predictors $s.t.$ $Y \approx X_s\beta_s$.

Advantages :

- better interpretability,
- improve prediction performance.

Disadvantages

- Greedy algorithm,
- may provide a suboptimal model for correlated predictors,
- Backward selection does not work if $n < p$.

Second strategy : Regularization

For correlated predictors, the LSE of β_j may have large variance (and thus take large values). To control the variance of estimators, we can impose constraints on the size of estimators. This may produce an estimator with improved prediction accuracy.

$$\widehat{\beta}_\lambda \in \operatorname{argmin}_{\beta\in\mathbb{R}^p} \|\boldsymbol{Y} - X\beta\|_n^2 + \lambda\,\mathrm{pen}(\beta),$$

↳ ideal penalty: $\|\beta\|_{\ell^0}$

→ impossible computationnellement parlant

→ what about $\|\cdot\|_{\ell^2}$

→ what about $\|\cdot\|_{\ell^1}$

where $\lambda > 0$ is a regularization parameter.

Objectives:

- Numerical stability (bad conditionning of $X^\top X$ for correlated predictors $(X_1, \ldots, X_p)$).
- Improve prediction accuracy: Increase bias to decrease variance, Bound/control the impact of nonactive predictors, Improve interpretability, Control model complexity
- Variable selection

Section 2

2. Penalized least squares estimator

Ridge regression

Proposition.

Let $\lambda > 0$, then the Ridge estimator[a] is such that

$$\widehat{\beta}^R(\lambda) = (X^T X + \lambda \mathbb{I}_p)^{-1} X^T Y = \arg\min_{\beta \in \mathbb{R}^p} \|Y - X\beta\|_n^2 + \lambda \|\beta\|^2.$$

The parameter λ is called the ***tuning/regularization parameter***.

[a] Hoerl and Kennard, 1970

Comments:

☛ If $\lambda > 0$, then $(X^T X + \lambda \mathbb{I}_p)^{-1}$ is well defined.

☛ Let consider the conditions of existance of $\widehat{\beta}$ the ordinary least squares estimator (OLSE), then

$$\boxed{\widehat{\beta}^R(\lambda) = (X^T X + \lambda \mathbb{I}_p)^{-1} X^T X \widehat{\beta}.}$$

Remarks

- We do not penalise the intercept parameter β_0.
- Strongly convex functional ⇒ unique solution
- Closed form solution (fast computation)
- Improved numerical stability
- Better prediction accuracy

Ridge regression

Let denote $\Lambda_j \geq 0$ eigenvalues of $X^\top X$. As $\lambda > 0$, the eigenvalues of $(X^\top X + \lambda \mathbb{I}_p)^{-1}$ are $(\Lambda_j + \lambda)^{-1} > 0$

Proposition Let $\lambda > 0$, then

1. $\mathbb{E}[\widehat{\beta}^R(\lambda)] = \beta - \lambda(X^TX + \lambda\mathbb{I}_p)^{-1}\beta$
2. $\mathbb{V}\mathrm{ar}[\widehat{\beta}^R(\lambda)] = \sigma^2(X^TX + \lambda\mathbb{I}_p)^{-1}X^TX(X^TX + \lambda\mathbb{I}_p)^{-1}$
3. Let $X^TX = P\mathrm{Diag}(\tilde{\ })\mathrm{P}^{\mathrm{T}}$, where P is a matrix such that $P^T = P^{-1}$ and $||P^T\beta||^2 = ||\beta||^2$, then the mean square error (MSE) is equal to

$$\mathbb{E}\|\widehat{\beta}^R(\lambda) - \beta\|_2^2 = \sum_{j=1}^{p} \frac{\sigma^2\Lambda_j + \lambda^2[P^T\beta]_j^2}{(\lambda + \Lambda_j)^2}$$

Proof of 1.

Let $\widehat{\beta}^R(\lambda) = (X^T X + \lambda \mathbb{I}_p)^{-1} X^T X \widehat{\beta}$, where $\widehat{\beta}$ is such that

$$\mathbb{E}[\widehat{\beta}] = \beta \quad \text{and} \quad \mathbb{V}\text{ar}[\widehat{\beta}] = \sigma^2 (X^T X)^{-1}$$

Then,

$$\begin{aligned}
\mathbb{E}[\widehat{\beta}^R(\lambda)] &= (X^T X + \lambda \mathbb{I}_p)^{-1} X^T X \mathbb{E}[\widehat{\beta}] = (X^T X + \lambda \mathbb{I}_p)^{-1} X^T X \beta \\
&= (X^T X + \lambda \mathbb{I}_p)^{-1} (X^T X + \lambda \mathbb{I}_p - \lambda \mathbb{I}_p) \beta \\
&= (X^T X + \lambda \mathbb{I}_p)^{-1} (X^T X + \lambda \mathbb{I}_p) \beta - \lambda (X^T X + \lambda \mathbb{I}_p)^{-1} \beta \\
&= \beta - \lambda (X^T X + \lambda \mathbb{I}_p)^{-1} \beta
\end{aligned}$$

Proof of 2.

As $\mathbb{V}\text{ar}[\widehat{\beta}] = \sigma^2(X^TX)^{-1}$, it comes

$$\begin{aligned}
\mathbb{V}\text{ar}[\widehat{\beta}^R(\lambda)] &= (X^TX+\lambda\mathbb{I}_p)^{-1}X^TX\mathbb{V}\text{ar}[\widehat{\beta}]X^TX(X^TX+\lambda\mathbb{I}_p)^{-1}\\
&= \sigma^2(X^TX+\lambda\mathbb{I}_p)^{-1}X^TX(X^TX)^{-1}X^TX(X^TX+\lambda\mathbb{I}_p)^{-1}\\
&= \sigma^2(X^TX+\lambda\mathbb{I}_p)^{-1}X^TX(X^TX+\lambda\mathbb{I}_p)^{-1}
\end{aligned}$$

Proof of 3.

EXAM

First recall that for any estimator $\widehat{\theta}$ of an parameter $\theta \in \mathbb{R}^p$

$$H(\widehat{\theta}) = \mathbb{E}\left[(\theta - \widehat{\theta})(\theta - \widehat{\theta})^T\right] = \left(\theta - \mathbb{E}[\widehat{\theta}]\right)\left(\theta - \mathbb{E}[\widehat{\theta}]\right)^T + \mathbb{V}\text{ar}[\widehat{\theta}]. \tag{1}$$

Applying (1) to $\widehat{\beta}^R(\lambda)$, we get

$$H(\widehat{\beta}^R) = \left(\beta - \mathbb{E}[\widehat{\beta}^R(\lambda)]\right)\left(\beta - \mathbb{E}[\widehat{\beta}^R(\lambda)]\right)^T + \mathbb{V}\text{ar}[\widehat{\beta}^R(\lambda)].$$

Moreover,

$$\begin{aligned} H(\widehat{\beta}^R) &= \left(\lambda(X^TX + \lambda\mathbb{I}_p)^{-1}\beta\right)\left(\lambda(X^TX + \lambda\mathbb{I}_p)^{-1}\beta\right)^T + \mathbb{V}\text{ar}[\widehat{\beta}^R(\lambda)] \\ &= \lambda^2(X^TX + \lambda\mathbb{I}_p)^{-1}\beta\beta^T(X^TX + \lambda\mathbb{I}_p)^{-1} \qquad (2) \\ &\quad + \sigma^2(X^TX + \lambda\mathbb{I}_p)^{-1}X^TX(X^TX + \lambda\mathbb{I}_p)^{-1} \\ &= (X^TX + \lambda\mathbb{I}_p)^{-1}\left(\lambda^2\beta\beta^T + \sigma^2X^TX\right)(X^TX + \lambda\mathbb{I}_p)^{-1} \qquad (3) \end{aligned}$$

Proof of 3.

From now, we denote by $\text{diag}(\Lambda)$ and $\text{diag}\left(1/(\Lambda+\lambda)\right)$ the $p \times p$ diagonal matrices whose diagonal elements are the respectively $1/(\Lambda_i+\lambda)$ and $\Lambda_i+\lambda$, with Λ_i the eigenvalues of the matrix X^TX. Then, let us decompose $X^TX = P\text{diag}(\Lambda)P^T$, it comes

$$X^TX = P\text{diag}(\Lambda)P^T \quad \text{and} \quad (X^TX+\lambda\mathbb{I}_p)^{-1} = P\text{diag}\left(1/(\Lambda+\lambda)\right)P^T$$

Proof of 3.

Then, as $P^TP = PP^T = \mathbb{I}_p$ and by using equation (2) becomes

$$\begin{aligned} H(\widehat{\beta}^R) &= (X^TX + \lambda\mathbb{I}_p)^{-1}\left(\lambda^2\beta\beta^T + \sigma^2X^TX\right)(X^TX + \lambda\mathbb{I}_p)^{-1} \\ &= P\mathrm{diag}\left(1/(\Lambda+\lambda)\right)P^T\left(\lambda^2\beta\beta^T + \sigma^2P\mathrm{diag}(\Lambda)P^T\right)P\mathrm{diag}\left(1/(\Lambda+\lambda)\right)P^T \\ &= P\mathrm{diag}\left(1/(\Lambda+\lambda)\right)\left(\lambda^2P^T\beta\beta^TP + \sigma^2P^TP\mathrm{diag}(\Lambda)P^{T}P^{\cancel{T}}\right)\mathrm{diag}\left(1/(\Lambda+\lambda)\right)P^T \\ &= P\mathrm{diag}\left(1/(\Lambda+\lambda)\right)\left(\lambda^2P^T\beta\beta^TP + \sigma^2\mathrm{diag}(\Lambda)\right)\mathrm{diag}\left(1/(\Lambda+\lambda)\right)P^T \qquad (4) \end{aligned}$$

Proof of 3.

We are now ready to prove 3. of the proposition

$$
\begin{aligned}
\mathbb{E}\left[\|\widehat{\beta}^R(\lambda)-\beta\|_2^2\right] &= \mathbb{E}\left[\mathrm{Tr}\left(\|\widehat{\beta}^R(\lambda)-\beta\|_2^2\right)\right] = \mathbb{E}\left[\mathrm{Tr}\left((\beta-\widehat{\beta}^R(\lambda))^T(\beta-\widehat{\beta}^R(\lambda))\right)\right] \\
&= \mathbb{E}\left[\mathrm{Tr}\left((\beta-\widehat{\beta}^R(\lambda))(\beta-\widehat{\beta}^R(\lambda))\right)^T\right] \\
&= \mathrm{Tr}\left[\mathbb{E}\left((\beta-\widehat{\beta}^R(\lambda))(\beta-\widehat{\beta}^R(\lambda))\right)^T\right] = \mathrm{Tr}(H(\widehat{\beta}^R))
\end{aligned}
$$

Proof of 3.

By using (4), we have as $P^T P = \mathbb{I}_p$

$$\begin{aligned}
\mathbb{E}\left[\|\widehat{\beta}^R(\lambda) - \beta\|_2^2\right] &= \mathrm{Tr}\left[\mathbb{E}\left((\beta - \widehat{\beta}^R(\lambda))(\beta - \widehat{\beta}^R(\lambda))\right)^T\right] \\
&= \mathrm{Tr}\left[P\mathrm{diag}\left(1/(\Lambda+\lambda)\right)\left(\lambda^2 P^T\beta\beta^T P + \sigma^2\mathrm{diag}(\Lambda)\right)\mathrm{diag}\left(1/(\Lambda+\lambda)\right)P^T\right] \\
&= \mathrm{Tr}\left[P^T P\mathrm{diag}\left(1/(\Lambda+\lambda)\right)\left(\lambda^2 P^T\beta\beta^T P + \sigma^2\mathrm{diag}(\Lambda)\right)\mathrm{diag}\left(1/(\Lambda+\lambda)\right)\right] \\
&= \mathrm{Tr}\left[\mathrm{diag}\left(1/(\Lambda+\lambda)\right)\left(\lambda^2 P^T\beta\beta^T P + \sigma^2\mathrm{diag}(\Lambda)\right)\mathrm{diag}\left(1/(\Lambda+\lambda)\right)\right] \\
&= \mathrm{Tr}\left[\mathrm{diag}\left(1/(\Lambda+\lambda)^2\right)\left(\lambda^2 P^T\beta\beta^T P + \sigma^2\mathrm{diag}(\Lambda)\right)\right] \\
&= \mathrm{Tr}\left[\mathrm{diag}\left(1/(\Lambda+\lambda)^2\right)\left(\lambda^2 P^T\beta\beta^T P\right)\right] \\
&\quad +\mathrm{Tr}\left[\mathrm{diag}\left(1/(\Lambda+\lambda)^2\right)\left(\sigma^2\mathrm{diag}(\Lambda)\right)\right] \\
&= \mathrm{Tr}\left[\mathrm{diag}\left(\lambda^2/(\Lambda+\lambda)^2\right)\left(P^T\beta\beta^T P\right)\right] + \mathrm{Tr}\left[\mathrm{diag}\left(\sigma^2\Lambda/(\Lambda+\lambda)^2\right)\right] \\
&= \sum_{j=1}^{p}\frac{\lambda^2[P^T\beta]_j^2}{(\Lambda_j+\lambda)^2} + \sum_{j=1}^{p}\frac{\sigma^2\Lambda_j}{(\Lambda_j+\lambda)^2} = \sum_{j=1}^{p}\frac{\lambda^2[P^T\beta]_j^2 + \sigma^2\Lambda_j}{(\Lambda_j+\lambda)^2} \qquad \square
\end{aligned}$$

Trace linéaire

Comments

☛ $\widehat{\beta}^R(\lambda)$ is biased.

☛ It may be better from the MSE point of view. Indeed if λ increases the variance of the Ridge estimator decreases.

☛ If $\lambda \to +\infty$ (large biais and null variance) then $\widehat{\beta}^R(\lambda) \to 0 \Rightarrow$ no regressor is selected and

$$\mathbb{E}\|\widehat{\beta}^R(\lambda) - \beta\|_2^2 \to \|\beta\|^2.$$

☛ If $\lambda \to 0$ (small bias, large variance) then $\widehat{\beta}^R(\lambda) \to \widehat{\beta} \Rightarrow$ all regressors are selected and

$$\mathbb{E}\|\widehat{\beta}^R(\lambda) - \beta\|_2^2 \to \sigma^2 \mathrm{Tr}((\mathrm{X}^{\mathrm{T}}\mathrm{X})^{-1}).$$

Comments

☛ If $X^T X = \mathbb{I}_p$ (orthogonal matrix X) then

$$\widehat{\beta}^R(\lambda) = \frac{X^T Y}{1+\lambda} = \frac{\widehat{\beta}}{1+\lambda}.$$

The values of $\widehat{\beta}_k$ the OLSE are decreasing with λ. Ridge estimator belongs to the "*shrinkage estimators*", it shrinks the coefficients but keeps all variables.

☛ Ridge regression is interesting when eigenvalues Λ_j are small.

☛ Finding a good compromise requires the proper tuning of λ. It has to be calibrated !!!

Toy example

$$Y = X_1\beta_1 + X_2\beta_2 + \xi.$$

with X_1 and X_2 strongly correlated.

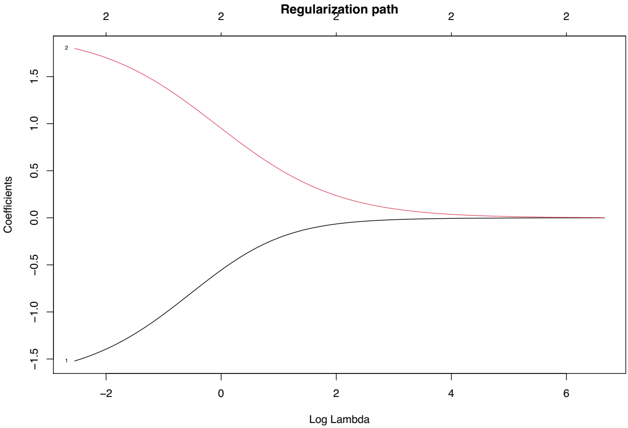
- If $X_1 \approx X_2$, then for all $\gamma \geq 0$,

  $$Y \approx X_1(\beta_1 + \gamma) + X_2(\beta_2 - \gamma) + \xi.$$

  We have several almost equivalent representations of Y.
- OLSE provides the same large variance prediction for all of these possible representations β indexed by γ.
- Constraint on the norm of β allows to select one representation with small l_2-norm: $(\beta_1 + \gamma)^2 + (\beta_2 - \gamma)^2$ that is minimized for $\gamma = (\beta_2 - \beta_1)/2$, and in that case $\beta_j = (\beta_1 + \beta_2)/2$.

$\rightsquigarrow$ Ridge estimator tends to average the coefficients corresponding to correlated prédictors.

Toy Example


Regularization path
2
2
2
2
2
Coefficients
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
2
1
−2
0
2
4
6
Log Lambda

Differents library/functions

- Function `lm.ridge` of the library `MASS`.
- Function `train( ,method='glmnet',`$\alpha = 0$`)` of the library `caret`. In library `caret` by default, the regressors and the response variable are re-centered and standardizeds. The output coefficient are in the original scale!
- Function `glmnet()` of the library `glmnet` with $\alpha = 0$. glmnet(x.train,y.train,standardize=TRUE,alpha=0,lambda=lambda_try). The regressors and/or the response variable are re-centered and/or standardizeds with the option `standardize=TRUE`.

Second Strategy: Regularization

- Ridge performs shrinkage but does not provide sparse solution. Unfit to select predictors.
- Is it possible to design a penalty function that can enforce sparsity on $\widehat{\beta}$?

↳ mettre des coefficients à 0.

$\Rightarrow$ LASSO (Least Absolute Shrinkage and Selection Operator)

Lasso

Definition **LASSO**[a]

$$\widehat{\beta}^{\mathrm{L}}(\lambda) = \operatorname{argmin}_{\beta \in \mathbb{R}^{p+1}} \|\boldsymbol{Y} - X\beta\|_n^2 + \lambda \|\beta\|_1,$$

where the regularization parameter $\lambda > 0$.

[a] Tibshirani, 1996

Comments

☛ The intercept is not penalized.

☛ In general, there is no closed form for the LASSO estimator.

☛ Convex optimization problem and there are efficient approximation algorithms (LARS,. . .).

☛ In the setting of orthogonal matrix X, we can derive a closed form.

☛ Lasso regularizes and selects a subset of predictors.

Orthogonal setting

Proposition Let $\lambda > 0$, if $X^\top X = \mathbb{I}_p$ then the LASSO estimator is equal to

$$\widehat{\beta}^L_j(\lambda) = \mathrm{sign}(X_j^T Y)\left[|X_j^T Y| - \lambda/2\right]_+ ,$$

where $[x]_+ = x$ if $x \geq 0$ and 0 otherwise.

Proof : Let as exercice.

$X_j^T Y$ quantifie la corrélation entre X_j et Y.

Comments

☛ $\forall j = 1, \cdots, p$, $|X_j^\top Y|$ can be seen as an indicator of the correlation between the regressor X_j and the response Y.

☛ If $\lambda/2 > |X_j^\top Y|$ then $\widehat{\beta}_j^L(\lambda) = 0$, so the regressor X_j is removed from the model. Therefore, if

$$\lambda/2 > \|X^\top Y\|_\infty = \max_j |X_j^\top Y| \Rightarrow \widehat{\beta}^L(\lambda) = 0.$$

So there is an infinity of λ values such that $\widehat{\beta}^L(\lambda) = 0$.

☛ Let $k \in \{1, \cdots, p\}$

$$\|X^\top Y\|_\infty = \max_j |X_j^\top Y| = |X_k^\top Y|,$$

i.e. the regressor X_k is the most correlated to Y. So as soon as $\lambda/2$ goes below the threshold $\|X^\top Y\|_\infty$, a first explanatory variable is retained, the regressor X_k.

Differents library/functions

- Function `l1ce` with the library `lasso2`.
- Function `cv.lars` with the library `lars`.
- Function `train(  ,method='glmnet',`$\alpha = 1$`)` of the library `caret`.
- Function `glmnet()` of the library `glmnet` with $\alpha = 1$.

Comments

☛ Choice of λ:

- $\lambda \to 0$ implies small bias, large variance (LSE)
- $\lambda \to \infty$ implies large biais and null variance.

$\Rightarrow$ Finding a good compromise requires proper tuning of λ.

☛ The Ridge estimator prevents overfitting. But it does not make the variable selection, it only shrinkes the β coefficients but keeps all regressors.

☛ The LASSO estimator also shrinks coefficients but puts some coefficients to zeros. It makes variables selection. Sparse solution (interpretability)

Elastic-Net

definition

• Let $\lambda_1 > 0$ and $\lambda_2 > 0$, the Elastic-Net[a] estimator denoted by $\widehat{\beta}^{EN}(\lambda_1, \lambda_2)$ is defined as follows

$$\widehat{\beta}^{EN}(\lambda_1, \lambda_2) = \arg\min_{\beta \in \mathbb{R}^p} ||Y - X\beta||^2 + \lambda_1 ||\beta||_1 + \lambda_2 ||\beta||_2^2.$$

• Let $\lambda > 0$, the Elastic-Net estimator can also be defined as follows

$$\widehat{\beta}^{EN}(\lambda, \alpha) = \arg\min_{\beta \in \mathbb{R}^p} ||Y - X\beta||^2 + \lambda \left(\alpha ||\beta||_1 + (1 - \alpha) ||\beta||_2^2 \right),$$

where $\alpha \in [0, 1]$.

[a] Zou and Hastie, 2005

Comments

☛ As for the LASSO, there is no closed form for the Elastic-net, but the optimisation problem is convex and there exist efficient approximation algorithms.

☛ The Elastic-net regression is a mix of the Ridge regression and the LASSO regression.

☛ As for Ridge and LASSO regressions, there is no theory on tests and confidence intervals for the Elastic-net regression.

☛ The Ridge estimator prevents overfitting. But it does not make the variable selection, it only shrinkes the β coefficients but keeps all regressors.

☛ The LASSO estimator also shrinks coefficients with some shrunk to zeros. It helps with features selection.

☛ Here again, hyperprameters λ and α have to be calibrated.

Differents library/functions

- Function `enet` with the library `elascticnet`.
- Function `train( ,method='glmnet',`$\alpha =$`,` $\lambda =$`)` of the library `caret`.
- Function `glmnet()` of the library `glmnet` with $\alpha = \cdot, \lambda = \cdot$.

PLS regression (Partial least square)

The principle of PSL regression is “close” to principal component regression (PCA). The purpose of this method is to introduce new regressors $t^{(1)}, \cdots, t^{(k)}$ such that :

- They are linear combinations of the departure regressors (which will have previously centered and renormalized). We assume $\overline{Y} = \overline{X}_j = 0$ and $s_Y^2 = s_{X_j}^2 = 1$.
- They are 2 by 2 orthogonal

$$\forall i \neq i' \quad \text{we have} \quad < t^{(i)}, t^{(i')} >= 0.$$

- They are ranked in order of importance by taking as their criterion their link with the variable Y.

PLS Algorithm : Step 1

- We set $X^{(1)} = X$ and $Y^{(1)} = Y$.
- Calculation of the first PLS component $t^{(1)}$: for $w \in \mathbb{R}^p$

$$t^{(1)} = \underset{t = X^{(1)} w,\ \|w\|_2 = 1}{\arg\max} < t, Y^{(1)} >$$

- Then, regress $Y^{(1)}$ on $t^{(1)}$ such that $Y^{(1)} = r_1 t^{(1)} + \widehat{\varepsilon}_1$,
  - with $r_1 \in \mathbb{R}$, the coefficient of the orthogonal projection on $t^{(1)}$:

$$r_1 t^{(1)} = P_{[t^{(1)}]} Y^{(1)} \quad \text{and} \quad r_1 = (t^{(1)\,T} t^{(1)})^{-1} t^{(1)\,T} Y^{(1)}$$

  - and the residual $\widehat{\varepsilon}_1$ which corresponds to the part not explained by $t^{(1)}$

$$\widehat{\varepsilon}_1 = P_{[t^{(1)}]^\perp} Y^{(1)}$$

PLS Algorithm : Step k

- We set $X^{(k)} = P_{[t^{(k-1)}]^{\perp}} X^{(k-1)}$ *i.e.* the part of $X^{(k-1)}$ that was not used in the first step to explain and $Y^{(k)} = P_{[t^{(k-1)}]^{\perp}} Y^{(k-1)} = \widehat{\varepsilon}_{k-1}$.
- Calculation of the $k-th$ PLS component $t^{(k)}$: for $w \in \mathbb{R}^p$

$$t^{(k)} = \underset{t=X^{(k)}w,\ \|w\|_2=1}{\arg\max} < t, Y^{(k)} >$$

- Then, regress $Y^{(k)}$ on $t^{(k)}$ such that $Y^{(k)} = r_k t^{(k)} + \widehat{\varepsilon}_k$,
  - with $r_k \in \mathbb{R}$, the coefficient of the orthogonal projection on $t^{(k)}$:

$$r_k t^{(k)} = P_{[t^{(k)}]} Y^{(k)}$$

  - and the residual $\widehat{\varepsilon}_k$ which corresponds to the part not explained by $t^{(k)}$

$$\widehat{\varepsilon}_k = P_{[t^{(k)}]^{\perp}} Y^{(k)}$$

Comments

☛ Here again, the hyperprameter k has to be calibrated.

☛ The PLS components are orthogonal to each other by construction, indeed $t^{(j)}$ is a linear combination of the columns of $X^{(j)}$

$$X^{(j)} = P_{[t^{(j-1)}]^{\perp}} X^{(j-1)}$$

Therefore, $X^{(j)} \in \mathrm{span}(t^{(1)}, \cdot, t^{(j-1)})$ and $t^{(j)} \perp \{t^{(1)}, \cdot, t^{(j-1)}\}$.

☛ Note that the $t^{(j)}$ are chosen according to the maximum empirical correlation with Y, the variables X and Y being centered.

Theorem

Theorem The PLS model is written as follows :

$$\begin{aligned} Y &= P_{[t^{(1)}]} Y^{(1)} + \cdots + P_{[t^{(k)}]} Y^{(k)} + \widehat{\varepsilon}_k \\ &= r_1 t^{(1)} + \cdots + r_k t^{(k)} + \widehat{\varepsilon}_k, \end{aligned}$$

where $\widehat{\varepsilon}_k = P_{[t^{(k}]^{\perp}} Y^{(k)} = P_{\mathrm{span}(t^{(1)},\cdot,t^{(j-1)})} Y$.

Proof : Admited or let in exercice.

Differents library/functions

- Function `plsr` with the library `pls`.
- Function `train(  ,method='pls')` of the library `caret`.

Here the regressors and/or the response variable are re-centered and/or standardizeds by adding the following option in the command `preProc = c("center", "scale")`.

An exemple : Prostate cancer dataset

Consider a dataset of size $n = 97$ split into 2 subset (train/test=67/30). we want to predict **lcavol** with respect to the other variables.

```
##  [1] "lcavol"  "lweight" "age"     "lbph"    "svi"     "lcp"     "gleason"
##  [8] "pgg45"   "lpsa"    "train"
```

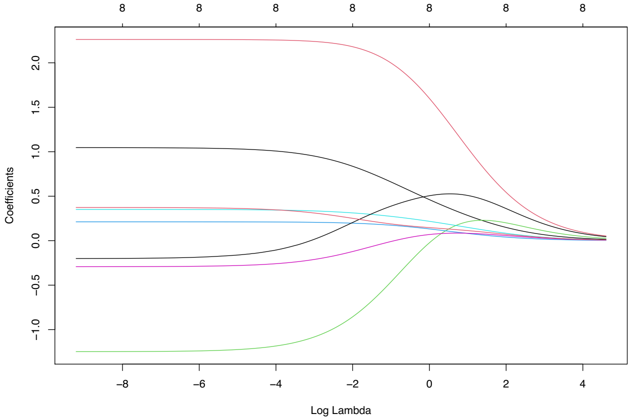
Consider all the variables and set

```
model.full <- lm(lpsa~.,prostate.train)
```

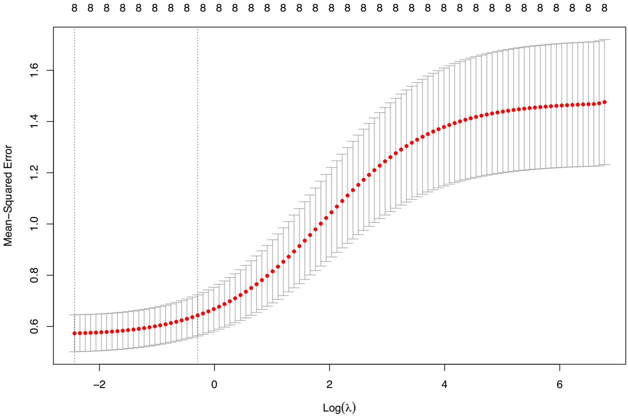
The RMSE on the test for the OLSE is 0.7225678.

Ridge : Prostate cancer

Solution path (for all $\lambda \in (0, \infty)$)



Ridge : Prostate cancer



The optimal parameter $\lambda^\star$ that minimizes the RMSE over the whole solution path is 0.087888

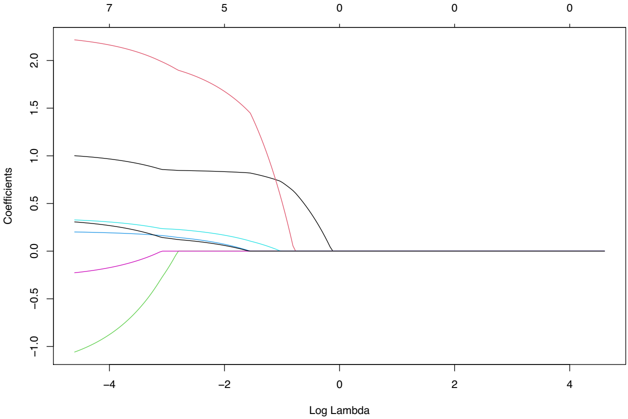
Ridge : Prostate cancer

The RMSE on the test for the Ridge estimator is 0.6975781

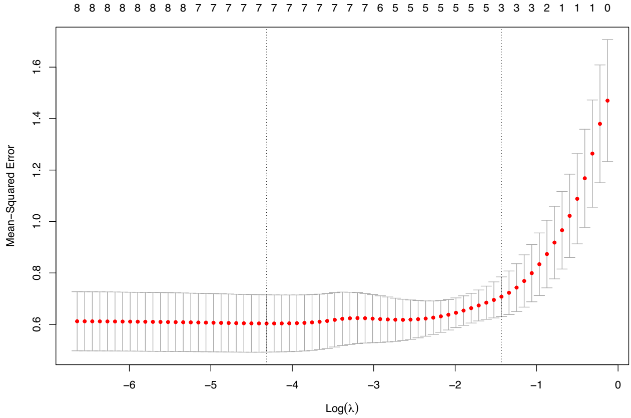
While the RMSE on the test for OLSE is 0.7225678.

Lasso : Prostate cancer data

Solution path (for all $\lambda \in (0, \infty)$)



Lasso : Prostate cancer data



The optimal parameter $\lambda^{\star}$ that minimizes the RMSE over the whole solution path is 0.0133582

Lasso : Prostate cancer data

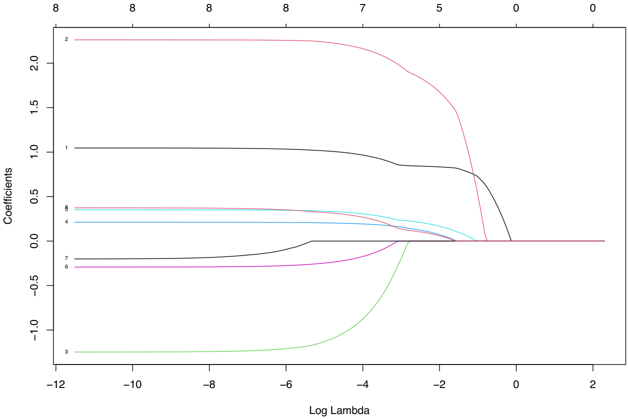
The RMSE on the test for the Lasso estimator is 0.6668793

While the RMSE on the test for the Ridge estimator is 0.6975781

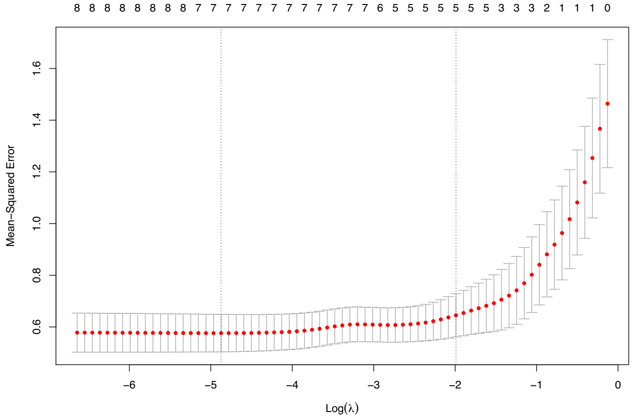
While the RMSE on the test for OLSE is 0.7225678.

Elastic-Net : Prostate cancer data

Solution path (for all $\lambda \in (0, \infty)$)



Elastic-Net : Prostate cancer data



The optimal parameter $\lambda^\star$ that minimizes the RMSE over the whole solution path is 0.0076441

Elastic-Net : Prostate cancer data

The RMSE on the test for the Elastic.net estimator is 0.6669006

The RMSE on the test for the Lasso estimator is 0.6668793

The RMSE on the test for the Ridge estimator is 0.6975781

The RMSE on the test for OLSE is 0.7225678.

Section 3

3. Cross-Validation

Discussion

- Use of the full dataset to fit $\widehat{\beta} \Rightarrow$ no guarantee as to the quality of prediction on a new dataset. **Need a proper dataset !**
- If fitting $\widehat{\beta}$ requires to choose hyperparameters (*e.g.* tuning λ) $\Rightarrow$ **need a proper dataset !**

Ideally, we would have 3 datasets

- A *train*-dataset for ℓearning/training to compute $\{\widehat{\beta}_{\ell}(\lambda)\}_{\lambda}$.
- A *validation*-dataset to validate/fix the "optimal" hyperparameters λ_{opt} according to a **criterion** and find the optimal procedure $\widehat{\beta}_{\ell}(\lambda_{opt})$.
- A *test*-dataset to evaluate the accuracy by calculating a **performance score** of the model $\widehat{\beta}_{train}(\lambda_{opt})$ on the test sample.

Splitting the dataset into 3 sub-samples could be the solution. Unfortunately, datas are rare and expensive. The *validation***-sample can be avoided by doing** *cross-validation* **on the** *train***-sample.**

Holdout methout

In practice, we divide the original sample $(x_i^T, y_i)_{i=1,\cdots,n}$ into two sub-samples : the *train* and the *test*.

- A *training*-sample of size n_ℓ for learning/training $(x_\ell^T, y_\ell)_\ell$: to build the model and to estimate the (hyper)parameters, say $\widehat{\beta_\ell}$ (or $\widehat{\beta_\ell}(\lambda_{opt})$).
- An independent *test*-sample of size n_t to validate that the estimated model responds to validation as well as during learning.

Exemple : of score performance on the *test*-sample: the predicted residual error sum of squares PRESS:

$$\text{PRESS} = \|\widehat{Y_t} - Y_t\|^2 = \|X_t\widehat{\beta_\ell} - Y_t\|^2 = \sum_{i=1}^{n_t}(\widehat{Y_i} - Y_i)^2,$$

where $\widehat{\beta_\ell}$, the estimation of β has been calculated with the *train*-sample.

Comments

- When dividing the sample by 2, the proportion is (70%,30%).
- When dividing the sample by 3, the proportion is (60%,20%,20%).
- Careful attention should be paid to the choice of sub-samples, choosing them in an intelligent way so that each sample represents at best the entire sample.

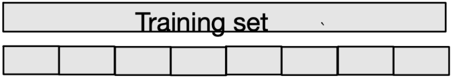
Cross-validation

If we need to select the "optimal" hyperparamter(s), we can do it on the *train*-data by cross-validation. And, for a upper bound $\overline{\lambda}$ of λ, the **optimal** λ_{opt} is the one that will minimize the criterion

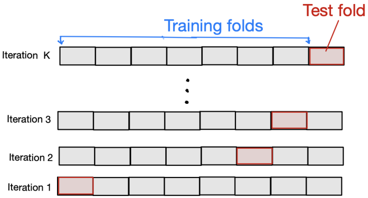
$$\lambda_{opt} = \arg\min_{\lambda\in[0,\overline{\lambda}]} \mathrm{Crit}(\lambda).$$

K-fold cross validation

We divide the *train*-sample $(x_i^T, y_i)_{i=1}^{n_{train}}$ into k sub-samples $(x_i^T, y_i)_{i=1}^{n_{\ell_k}}$ of size n_{ℓ_k}.



K-fold cross validation



The performance score is calculated by the PRESS at each iteration k on the sub-sample k

$$\mathrm{PRESS}^k(\lambda) = \|\widehat{Y}_{\ell_k}(\lambda) - Y_{\ell_k}\|^2 = \|X_{\ell_k}\widehat{\beta}_{(-\ell_k)}(\lambda) - Y_{t_v}\|^2$$

where $\widehat{Y}_{\ell_k} = X_{\ell_k}\widehat{\beta}_{\ell}(\lambda)$ and $\widehat{\beta}_{(-\ell_k)}(\lambda)$ have been calculated with the $(K-1)$ remaining folds.

K-fold cross validation

The average of the K-PRESS is finally calculated to estimate the prediction error

$$\mathrm{Crit}(\lambda) := \frac{1}{K}\sum_{k=1}^{K} \mathrm{PRESS}^{k}(\lambda) = \frac{1}{K}\sum_{k=1}^{K} \|X_{\ell_k}\widehat{\beta}_{(-\ell_k)}(\lambda) - Y_{\ell_k}\|^2$$

Leave-one-out cross validation (LOOCV)

Here $K = n_{train}$, *i.e.* we learn with $(n_{train} - 1)$ observations then we validate the model on the n_{train}-*th* observation and we repeat the full operation n_{train} times and

$$\mathrm{Crit}(\lambda) := \mathrm{PRESS}_{CV}(\lambda) = \frac{1}{n_{train}} \sum_{i=1}^{n_{train}} (\widehat{Y}_i(\lambda) - Y_i)^2 = \frac{1}{n_{train}} \sum_{i=1}^{n_{train}} (x_i^T \widehat{\beta}_{(-i)}(\lambda) - Y_i)^2,$$

where $\widehat{\beta}_{(-i)}(\lambda)$ have been calculated with the *train*-sub-sample where the observation (x_i^T, Y_i) has been removed.

Comments

☛ If we denote by $H(\lambda) := X(X^TX + \lambda\mathbb{I}_p)^{-1}X^T$ the projector in Ridge setting, it comes that

$$x_i^T\widehat{\beta}_{(-i)}(\lambda) - Y_i = \frac{x_i^T\widehat{\beta}(\lambda) - Y_i}{1 - H_{ii}} = \frac{\widehat{\varepsilon}_i(\lambda)}{1 - H_{ii}},$$

where $\widehat{\beta}(\lambda)$ have been compute with all the *train*-sample and H_{ii} is the *i-th* diagonal element of H. Then, we can write

$$\mathrm{PRESS}_{CV}(\lambda) = \frac{1}{n_{train}}\sum_{i=1}^{n_{train}}\left(\frac{\widehat{\varepsilon}_i(\lambda)}{1 - H_{ii}(\lambda)}\right)^2.$$

☛ The standard (*LOOCV*) gives a very precise estimate of the optimal parameters. Unfortunately, its implementation is limited in practice by a high numerical cost. We prefer the *generalized cross validation* (Carefull, nothing to do with *(LOOCV)*) where $1 - H_{ii}(\lambda)$ is replaced by $(1 - \mathrm{Tr}(H(\lambda))/n)$.

Generalized cross-validation (GCV)

- In **our Ridge setting**, we calculate the estimators $\left(\widehat{\beta}_t^R(\lambda)\right)_\lambda$ with this *train*-sample (X, Y) of size n_{train} such that

$$\widehat{\beta}_\ell^R(\lambda) = (X^TX + \lambda\mathbb{I}_p)^{-1}X^TY \quad \text{and} \quad \widehat{Y}^R(\lambda) = H(\lambda)Y$$

with $H(\lambda) = X(X^TX + \lambda\mathbb{I}_p)^{-1}X^T$ the projector in Ridge setting for the *train*-sample.

- We define the Generalized cross-validation criterion as follows

$$\mathrm{GCV}(\lambda) := \frac{\|Y - \widehat{Y}^R(\lambda)\|^2/n_{train}}{(\mathrm{Tr}(\mathbb{I}_n - H(\lambda))/n_{train})^2} = \frac{1}{n_{train}}\sum_{i=1}^{n_{train}}\left(\frac{Y_i - \widehat{Y}_i^R(\lambda)}{1 - \mathrm{Tr}(H(\lambda))/n_{train}}\right)^2$$