# Un esempio di LASSO e regressione ridge in R

Un esempio di regolarizzazione

N. Torelli (DEAMS, University of Trieste)

17 novembre 2020

## Contents



## LASSO e regressione ridge

L'uso del LASSO (e in generale delle tecniche di regolarizzazione) consentono di risolvere eventuali problemi di multicollinearità, di ridurre la complessità del modello e di selezionare quindi un modello che bilanci capacità di adattamento e semplicità includendo solo quelle variabili più influenti.

L'idea del (**LASSO**) è quella di penalizzare la funzione verosimigliinza con un termine che dipende dalla complesssità del modello:

$$\underset{\beta\in\mathbb{R}^p}{\arg\min}\left\{\frac{1}{N}\sum_{i=1}^{N}(y_i-(\beta_0+\sum_{j=1}^{J}x_{ij}\beta_j))^2+\lambda\sum_{j=1}^{J}|\beta_j|\right\},$$

$\lambda$ è il *parametro* di penalizzazione che deve essere fissato (o può essere determinato con tecniche di cross validaztion). Se $\lambda\approx 0$, le stime LASSO dei coefficienti tenederanno a essere coincidenti con le stime classiche dei minimi quadrati; se invece $\lambda$ è elevato molte stime dei coeffficienti vengono spinte verso zero.

Per esemplificare consideriamo `Prostate` un dataset che è inserito nel R package `lasso2` usato da Stamey, Kabalin, and Ferrari (1989) and Tibshirani (1996) per esaminare la relazione fra il valore dell'antigene specifico della prostata (PSA) oe altre covariate per soggetti che stavano per essere sottoposti a interventi di prostatectomia. Questa la lista delle covariate

| NOME VARIABILE | DESCRIZIONE |
|---|---|
| *lpsa* | level of prostate-specific antigen |
| *lcavol* | log(cancer volume) |
| *lweight* | log(prostate weight) |
| *age* | age |
| *lbph* | log(benign prostatic hyperplasia amount) |
| *svi* | seminal vesicle invasion |
| *lcp* | log(capsular penetration) |
| *gleason* | Gleason score |

| NOME VARIABILE | DESCRIZIONE |
|---|---|
| *pgg45* | percentage Gleason scores 4 or 5 |

Assumiamo che sia adeguato un modello lineare per la variabile dipendente livello dell'antigene (`lpsa`):

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2), \quad i = 1, \ldots, n, \tag{1}$$

dove il generico $x_{ij}$ è la $j$-ma covariata per $i$-ma unità, e ogni $\beta_j$ rappersenta quindi il coefficiente di regressione relativo alla j-ma variabile.

Potremmo usare la regressione LASSO (come sviluppata in Tibshirani (1996)) per schiacciare verso lo 0 un sottoinsieme di coefficienti con legami deboli con la variabile dipendemte.

Leggiamo i dati e estraiamo in appositi oggetti matrice delle variabili dipendenti, variabile risposta e dimensione campionaria.

```
library(lasso2)
library(glmnet)
library(arm)
library(car)
# i dati
data(Prostate)
# La dimensione campionaria
N <- dim(Prostate)[1]
# la matrice dei predittori
X <- as.matrix(cbind(rep(1,N), Prostate[,1:8]))
# la dimensione del vettore dei predittori
p <- dim(Prostate)[2]
# la variabie dipendente
y <- Prostate[, p]
# e guardiamo i primi casi
head(Prostate)
```

```
##       lcavol  lweight age      lbph svi       lcp gleason pgg45       lpsa
## 1 -0.5798185 2.769459  50 -1.386294   0 -1.386294       6     0 -0.4307829
## 2 -0.9942523 3.319626  58 -1.386294   0 -1.386294       6     0 -0.1625189
## 3 -0.5108256 2.691243  74 -1.386294   0 -1.386294       7    20 -0.1625189
## 4 -1.2039728 3.282789  58 -1.386294   0 -1.386294       6     0 -0.1625189
## 5  0.7514161 3.432373  62 -1.386294   0 -1.386294       6     0  0.3715636
## 6 -1.0498221 3.228826  50 -1.386294   0 -1.386294       6     0  0.7654678
```

Vediamo cosa accade se usiamo `lm` e otteniamo anche i fattori di inflazione della varianza

```
mod.lin <- lm(lpsa~., data = Prostate)
mod1 <- summary(mod.lin)
mod1$coefficients
```

```
##                Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept)  0.669399027 1.296381277  0.5163597 6.068984e-01
## lcavol       0.587022881 0.087920374  6.6767560 2.110634e-09
## lweight      0.454460641 0.170012071  2.6731081 8.956206e-03
## age         -0.019637208 0.011172743 -1.7575995 8.229321e-02
## lbph         0.107054351 0.058449332  1.8315753 7.039819e-02
## svi          0.766155885 0.244309492  3.1360054 2.328823e-03
```

```
## lcp         -0.105473570 0.091013484 -1.1588785 2.496408e-01
## gleason      0.045135964 0.157464467  0.2866422 7.750601e-01
## pgg45        0.004525324 0.004421185  1.0235545 3.088513e-01
```

```
vif(mod.lin)
```

```
##   lcavol  lweight      age     lbph      svi      lcp  gleason    pgg45
## 2.054113 1.363706 1.323600 1.375537 1.956882 3.097954 2.473403 2.974362
```

## il pacchetto lasso2

Il pacchetto `lasso2`` contiene la funzionel1ce“‘ (stima vincolata con norma L1) :

```
mod.lasso <- l1ce(lpsa~., data = Prostate)
summ <- summary(mod.lasso)
```

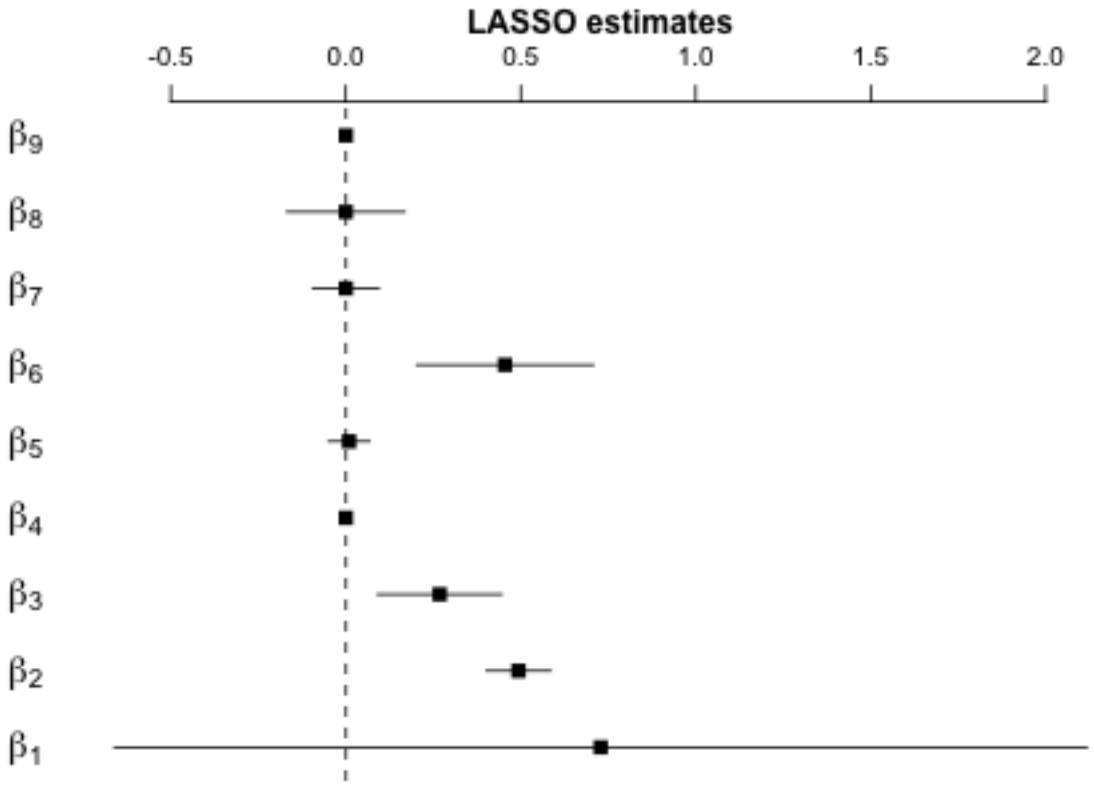
Possiamo dare un'occhiata ai coefficienti e poi vedere un grafico degli stessi con `coefplot` function

```
summ$coefficients
```

```
##                    Value  Std. Error    Z score     Pr(>|Z|)
## (Intercept) 0.7284810757 1.389770396 0.52417369 6.001577e-01
## lcavol      0.4936540169 0.091909166 5.37110758 7.825451e-08
## lweight     0.2681863403 0.177428510 1.51151774 1.306566e-01
## age         0.0000000000 0.011141425 0.00000000 1.000000e+00
## lbph        0.0092825881 0.058711797 0.15810431 8.743746e-01
## svi         0.4550584943 0.252482732 1.80233512 7.149270e-02
## lcp         0.0000000000 0.094671449 0.00000000 1.000000e+00
## gleason     0.0000000000 0.168512625 0.00000000 1.000000e+00
## pgg45       0.0001812107 0.004638956 0.03906282 9.688403e-01
```

```
coefplot(summ$coefficients[,1], summ$coefficients[,2],
         varnames=(beta_names_expr),
         cex.pts=1.2, pch.pts=15, col="black",CI=1, cex.var=1.4,
         main = "LASSO estimates")
```

---

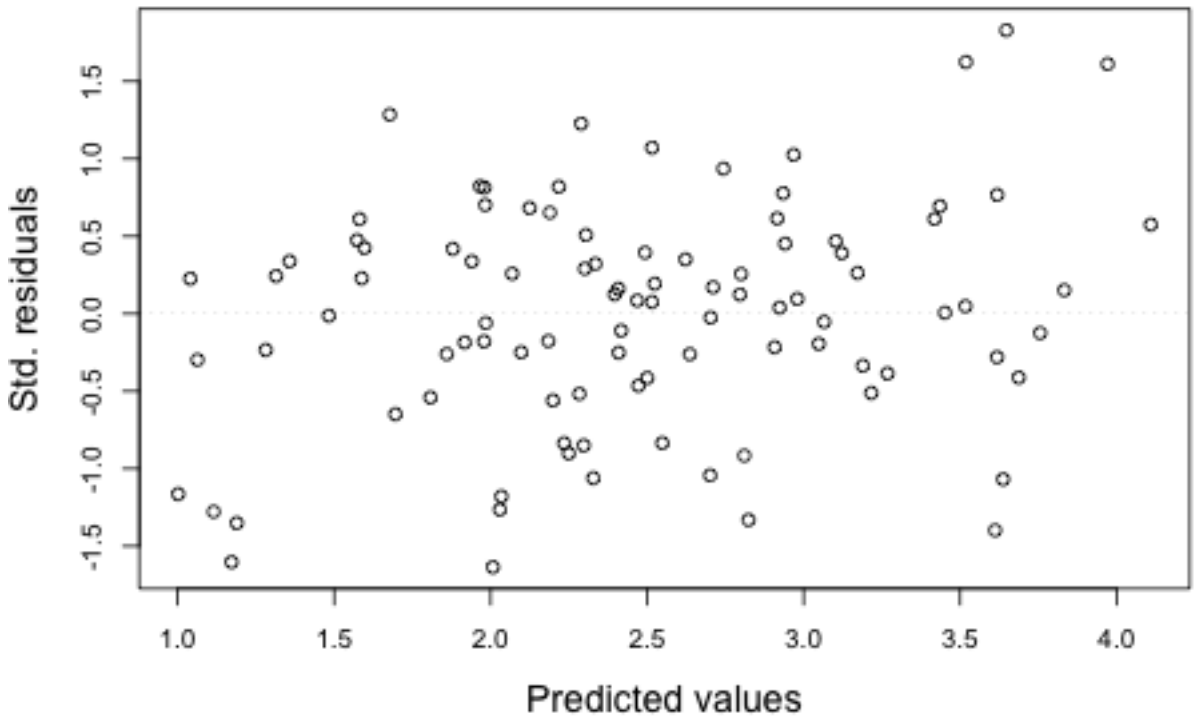



Coem si vede, alcuni coefficienti (`age`, `lcp`, `gleason` ) vengono schiacciati verso zero.

Possiamo calcolare i residui:

```
plot(resid(mod.lasso) ~ fitted(mod.lasso),
     cex.lab = 1.4, xlab = "Predicted values",
     ylab = "Std. residuals")
abline(h = 0, lty = 3, lwd = .2)
```

---




## il pacchetto glmnet

Un altro pacchetto, più recente, è `glmnet` che consente di stimare un modello con regolarizzazione tramite la funzione `glmnet`:

```
mod_glmnet <- glmnet(x = X[,-1], y = y, family = "gaussian")
```

La funzione obiettivo che viene minimizzata è più complessa e permette di ottenere sia Regressione Ridge che il LASSO:

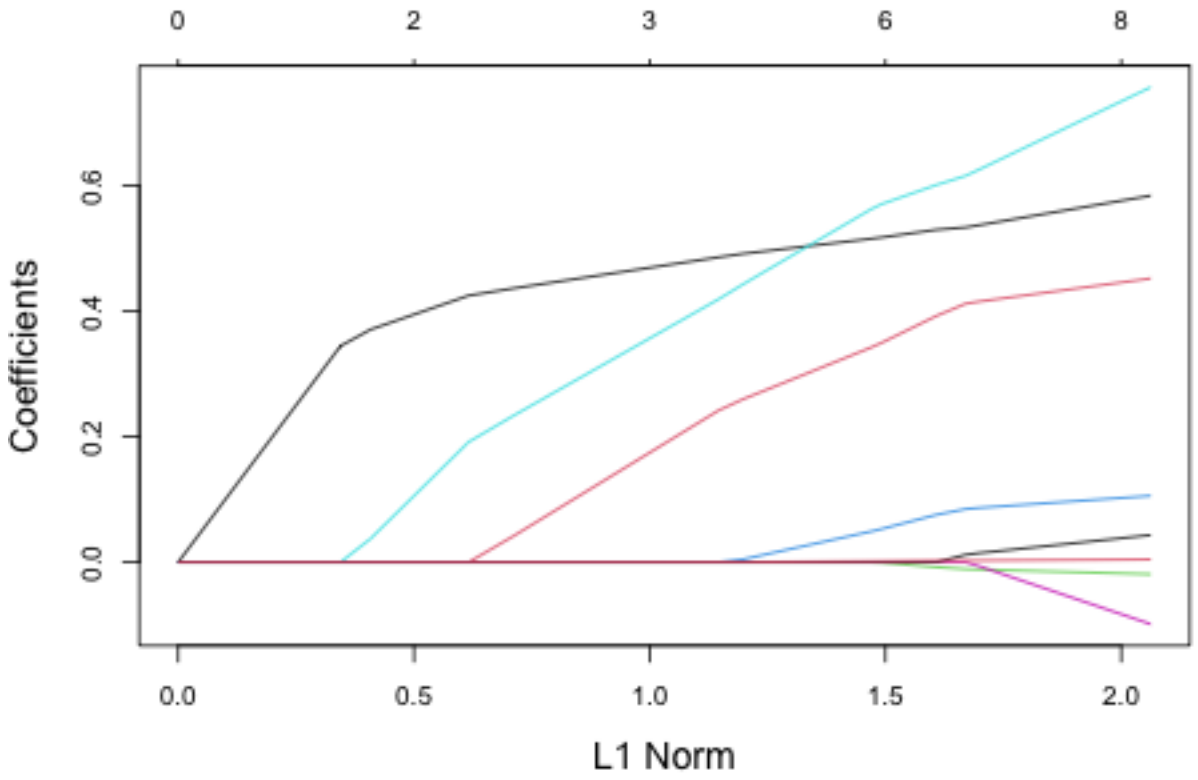
$$\hat{\boldsymbol{\beta}}_{LASSO} = \arg\min_{\beta}[(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda[(1-\alpha)(\sum_{j=2}^{p}\beta_j^2)/2 + \alpha(\sum_{j=2}^{p}|\beta_j|)]$$

Il default è $\alpha = 1$ che significa LASSO, con $\alpha = 0$ si ha regressione Ridge.

La funzione `plot` permette di visualizzare i coefficienti:

```
plot(mod_glmnet, cex.lab =1.4)
```

---




Ciascuna curva visualizza il valore di ciascun parametro al variare del fattore di penalizzazione e la norma $L_1$-norm dell'intero vettore di coefficienti ottenuta al variare del fattore $\lambda$. Sopra è indicato il numero di coefficienti non nulli al variare di $\lambda$ che sono denominati *gradi di libertà effettivi* (df) per il LASSO.

$\lambda$ andrebbe scelto in relazione a quale preferenza si ha fra adattamento e parsimonia: $\lambda$ può essere scelto guardando alla capacità predittica del modello per diversi valori. La funzione `glmnet` produce una sequenza di modelli da cui l'utente può scegliere. ma è possibile lasciare che sia il software a usare la validazione incrociata e definire il valore di $\lambda$ opportuno.

`cv.glmnet` è la funzione per la validazione incrociata. Se consideriamo i dati già visti si ha

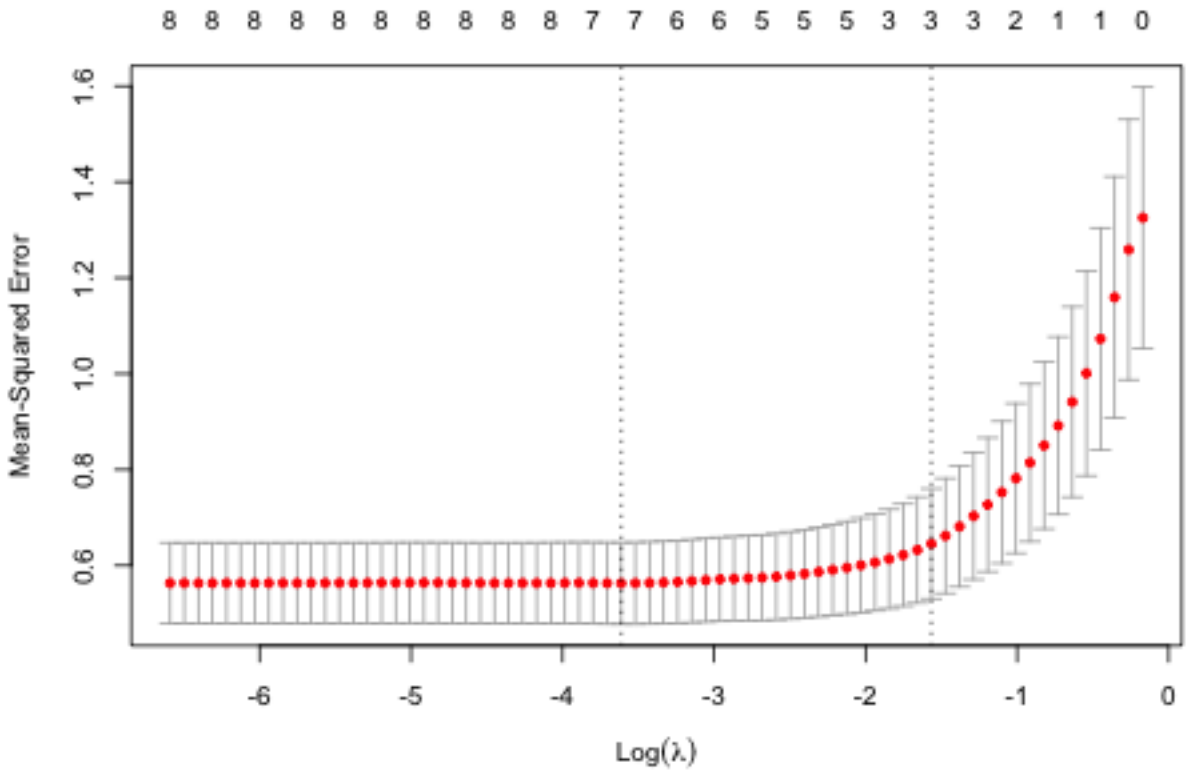
```
cvfit <- cv.glmnet(x = X[,-1], y = y)
cvfit
```

```
##
## Call:  cv.glmnet(x = X[, -1], y = y)
##
## Measure: Mean-Squared Error
##
##      Lambda Measure      SE Nonzero
## min 0.02698  0.5629 0.08517       7
## 1se 0.20892  0.6454 0.11479       3
```

`cv.glmnet` restituisce un oggetto della classe `cv.glmnet` (che abbiano salvato come `cvfit`), che è una lista con tutti gli ingredienti dell'adattamento ottenuto dopo la validazione incrociata. Vengono riportati due valori di $\lambda$: `lambda.min` è il valore di $\lambda$ che fornisce il più piccolo errore a seguito della validazione, `lambda.1se` da il modello più regolarizzato (con meno parametri) ma tale che l'errore non si discosti più di un errore standard dal minimo. L'output mostra anche il numero di coefficienti non zero per i vaolri di $\lambda$ selezionati.

Possiamo anche fare un plot dell'oggetto `cvfit`.

```
plot(cvfit)
```




Il garfico comprende i vari valori dell'errore per la sequanza di $\lambda$ (la curva di validazione incrociata) (i punti rossi),e i due valori di $\lambda'$ selezionati nel summary (in corriponedenza delle linee verticali).

Se vogliamo avere i due valori $\lambda'$ e i coefficienti ottenuti possiamo, per esempio, estrarre i coefficienti stimati per quel valore usando la funzione `coef`:

```
cvfit$lambda.min
```

```
## [1] 0.0269835
```

```
coef(cvfit, s = "lambda.min")
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept)  0.652235465
## lcavol       0.531450368
## lweight      0.402780524
## age         -0.009500636
## lbph         0.080529233
## svi          0.608595937
## lcp          .
## gleason      0.006484200
## pgg45        0.002485511
```

Come già visto in precedenza, i coefficienti per `age`, `lcp` e `gleason` sono stati schiacciati a zero.

Se scegliamo `alpha = 0` in `glmnet` otteniamo la regressione Ridge:

```
mod_ridge <- glmnet(x = X[,-1], y = y, alpha = 0)
cvfit_ridge <- cv.glmnet(x = X[,-1], y = y, alpha = 0)
```

Ora possiamo confrontare i coefficienti ottenutio nei diveri casi:

| | lm | l1ce (lasso) | glmnet (lasso) | glmnet (ridge) |
|---|---|---|---|---|
| (Intercept) | 0.669 | 0.728 | 0.652 | 0.477 |
| lcavol | 0.587 | 0.494 | 0.531 | 0.512 |
| lweight | 0.454 | 0.268 | 0.403 | 0.442 |
| age | -0.020 | 0.000 | -0.010 | -0.015 |
| lbph | 0.107 | 0.009 | 0.081 | 0.095 |
| svi | 0.766 | 0.455 | 0.609 | 0.691 |
| lcp | -0.105 | 0.000 | 0.000 | -0.038 |
| gleason | 0.045 | 0.000 | 0.006 | 0.062 |
| pgg45 | 0.005 | 0.000 | 0.002 | 0.003 |

# Bibliografia


Stamey, Thomas A, John N Kabalin, and Michelle Ferrari. 1989. "Prostate Specific Antigen in the Diagnosis and Treatment of Adenocarcinoma of the Prostate. Ii. Radiation Treated Patients." *The Journal of Urology* 141 (5). Wolters Kluwer Philadelphia, PA: 1084–7.

Tibshirani, Robert. 1996. "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society: Series B (Methodological)* 58 (1). Wiley Online Library: 267–88.