An Introduction to `islasso`

Gianluca Sottile Giovanna Cilluffo Vito M.R. Muggeo

Maggio 24, 2021

Contents

Abstract

In this short note we present and briefly discuss the R package islasso dealing with regression models having a large number of covariates. Estimation is carried out by penalizing the coefficients via a quasi-lasso penalty, wherein the nonsmooth lasso penalty is replaced by its smooth counterpart determined iteratively by data according to the induced smoothing idea. The package includes functions to estimate the model and to test for linear hypothesis on linear combinations of relevant coefficients. We illustrate R code throughout a worked example, by avoiding intentionally to report details and extended bibliography.

Introduction

Let $\mathbf{y} = \mathbf{X}\beta + \epsilon$ be the linear model of interest with usual zero-means and homoscedastic errors. As usual, $\mathbf{y} = (y_1, \ldots, y_n)^T$ is the response vector, $\mathbf{X}$ is the $n \times p$ design matrix (having p quite large) with regression coefficients β. When interest lies in selecting the non-noise covariates and estimating the relevant effect, one assumes the lasso penalized objective function (Tibshirani, 1996),

$$\frac{1}{2}||\mathbf{y} - \mathbf{X}\beta||_2^2 + \lambda||\beta||_1$$

The R functions

The main function of the package is *islasso()* where the user supplies the model formula as in the usual *lm* or *glm* functions, i.e.

```
islasso(formula, family = gaussian, lambda, alpha = 1, data, weights, subset, offset,
    unpenalized, contrasts = NULL, control = is.control())
```

family accepts specification of family and link function as in Table 1, *lambda* is the tuning parameter and *unpenalized* allows to indicate covariates with unpenalized coefficients.

Table 1. Families and link functions allowed in islasso

family	link
gaussian	identity
binomial	logit, probit
poisson	log
gamma	identity, log, inverse

The fitter function is `is.lasso.fit()` which reads as

```
islasso.fit(X, y, family = gaussian, lambda, alpha = 1, intercept = FALSE, weights = NULL,
    offset = NULL, unpenalized = NULL, control = is.control())
```

which actually implements the estimating algorithm as described in the paper. The *lambda* argument in *islasso.fit* and *islasso* specifies the positive tuning parameter in the penalized objective. Any non-negative value can be provided, but if missing, it is computed via K-fold cross validation by the function *cv.glmnet()* from package **glmnet**. The number of folds being used can be specified via the argument *nfolds* of the auxiliary function *is.control()*.

A worked example: the Diabetes data set

We use the well-known **diabetes** dataset available in the **lars** package. The data refer to $n = 442$ patients enrolled to investigate a measure of disease progression one year after the baseline. There are ten covariates, (age, sex, bmi (body mass index), map (average blood pressure) and several blood serum measurements (tc, ldl, hdl, tch, ltg, glu). The matrix *x2* in the dataframe also includes second-order terms, namely first-order interactions between covariates, and quadratic terms for the continuous variables.

To select the important terms in the regression equation we apply the lasso

```
library(lars)
library(glmnet)

data("diabetes", package = "lars")

a1 <- with(diabetes, cv.glmnet(x2, y))
n <- nrow(diabetes)
a1$lambda.min * n
```

```
> [1] 1344.186
```

```
b <- drop(coef(a1, "lambda.min", exact = TRUE))
length(b[b != 0])
```

```
> [1] 15
```

Ten-fold cross validation "selects" $\lambda = 1344.186$. corresponding to 15 non null coefficients

```
names(b[b != 0])
```

```
>  [1] "(Intercept)" "sex"         "bmi"         "map"         "hdl"
>  [6] "ltg"         "glu"         "age^2"       "bmi^2"       "glu^2"
> [11] "age:sex"     "age:map"     "age:ltg"     "age:glu"     "bmi:map"
```

The last six estimates are

```
tail(b[b != 0])
```

```
>      glu^2    age:sex    age:map    age:ltg    age:glu    bmi:map
>  69.599081 107.479925  29.970061   8.506032  11.675332  85.530937
```

A reasonable question is if all the "selected" coefficients are significant in the model. Unfortunately lasso regression does not return standard errors due to nonsmoothness of objective, and some alternative approaches have been proposed., including the (Lockhart et al., 2013). Among the (few) strategies, including the 'covariance test', the 'post-selection inference' and the '(modified) residual bootstrap', here we illustrate the R package **islasso** implementing the recent 'quasi' lasso approach based on the induced smoothing idea (Brown and Wang, 2005) as discussed in Cilluffo et al. (2019)

While the optimal lambda could be selected (without supplying any value to *lambda*), we use optimal value minimizing the AIC

```
library(islasso)
out <- islasso(y ~ x2, data = diabetes, lambda = a1$lambda.min * n)
```

The **summary** method quickly returns the main output of the fitted model, including point estimates, standard errors and p-values. Visualizing estimates for all covariates could be somewhat inconvenient, especially when the number of covariates is large, thus we decide to print estimates only if the pvalue is less than a threshold value. We use *0.50*

```
summary(out, pval = 0.1)
```

```
>
> Call:
> islasso(formula = y ~ x2, lambda = a1$lambda.min * n, data = diabetes)
>
> Residuals:
>     Min      1Q  Median      3Q     Max
> -127.05  -64.99  -11.63   59.25  193.43
>
>             Estimate Std. Error    Df z value Pr(>|z|)
> (Intercept)   152.13      11.63  1.00   13.08   <2e-16 ***
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> (Dispersion parameter for gaussian family taken to be 5922.876)
>
>     Null deviance: 2621009  on 441  degrees of freedom
> Residual deviance: 2611776  on 441  degrees of freedom
> AIC: 5096.8
> Lambda: 1344.2
>
> Number of Newton-Raphson iterations: 29
```

In addition to the usual information printed by the summary method, the output also includes the column *Df* representing the degrees of freedom of each coefficient. Their sum is used to quantify the model complexity

```
sum(out$internal$hi)
```

```
> [1] 1.035904
```

and the corresponding residual degrees of freedom (440.9640965) as reported above. The Wald test (column *z value*) and p-values can be used to assess important or significant covariates. Results suggest that the value of the minimum lambda choosn by the cross validation procedure of glmnet was too high, hence as an alternative, it is also possible to select the tuning parameter λ by means the Bayesian or Akaike Information Criterion. The function *aic.islasso*, requires an islasso fit object and specification of the criterion to be used (AIC/BIC). Hence

```
lmb.bic <- aic.islasso(out, method = "bic", interval = c(1, 100))
```

```
>
> Optimization through bic
>
> lambda =   38.8146 bic =  4927.40166
> lambda =   62.1854 bic =  4951.85099
> lambda =   24.3707 bic =  4914.23394
> lambda =   15.4439 bic =  4912.90440
> lambda =   17.6255 bic =  4914.22572
> lambda =    9.9268 bic =  4918.40832
> lambda =   13.3366 bic =  4915.13480
> lambda =   16.2772 bic =  4912.93746
> lambda =   15.8133 bic =  4913.02305
> lambda =   14.6390 bic =  4913.39457
> lambda =   15.1364 bic =  4913.86615
> lambda =   15.5850 bic =  4912.67395
> lambda =   15.6722 bic =  4913.06249
> lambda =   15.5311 bic =  4913.01898
> lambda =   15.6183 bic =  4913.06886
> lambda =   15.5644 bic =  4912.80222
> lambda =   15.5977 bic =  4912.61239
> lambda =   15.6056 bic =  4912.65762
> lambda =   15.5961 bic =  4912.63357
> lambda =   15.6002 bic =  4912.60666
> lambda =   15.5994 bic =  4912.60170
> lambda =   15.5992 bic =  4912.60172
> lambda =   15.5993 bic =  4912.60171
> lambda =   15.5997 bic =  4912.59977
> lambda =   15.5999 bic =  4912.59981
> lambda =   15.5998 bic =  4912.59981
> lambda =   15.5996 bic =  4912.59978
> lambda =   15.5997 bic =  4912.59978
> lambda =   15.5997 bic =  4912.59977
> lambda =   15.5997 bic =  4912.59977
```

```
out1 <- update(out, lambda = lmb.bic)
summary(out1, pval = 0.05)
```

```
>
> Call:
> islasso(formula = y ~ x2, lambda = lmb.bic, data = diabetes)
>
> Residuals:
>     Min      1Q  Median      3Q     Max
> -119.26  -42.00   -5.70   39.74  155.35
>
>             Estimate Std. Error     Df z value Pr(>|z|)
> (Intercept)  152.133      2.642  1.000  57.581  < 2e-16 ***
> x2sex        -77.942     30.163  0.515  -2.584 0.009766 **
> x2bmi        297.945     28.477  0.439  10.463  < 2e-16 ***
> x2map        201.966     29.675  0.479   6.806 1.00e-11 ***
> x2hdl       -140.918     23.491  0.373  -5.999 1.99e-09 ***
> x2tch        114.273     19.765  0.292   5.782 7.40e-09 ***
> x2ltg        276.944     27.841  0.423   9.947  < 2e-16 ***
> x2glu        101.928     29.960  0.476   3.402 0.000669 ***
> x2bmi^2       92.623     28.392  0.454   3.262 0.001105 **
```

```
> x2age:sex     67.201     29.382  0.494   2.287 0.022188 *
> x2age:ldl    -47.115     23.642  0.380  -1.993 0.046280 *
> x2bmi:map     67.011     27.969  0.431   2.396 0.016579 *
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> (Dispersion parameter for gaussian family taken to be 3085.404)
>
>     Null deviance: 2621009  on 441.0  degrees of freedom
> Residual deviance: 1299791  on 421.3  degrees of freedom
> AIC: 4827.8
> Lambda: 15.6
>
> Number of Newton-Raphson iterations: 22
```

Comparisons between methods to select the tuning parameter and further discussions We conclude this short note by emphasizing that **islasso** also accepts the so-called elastic-net penalty, such that

$$\frac{1}{2}||\mathbf{y} - \mathbf{X}\beta||_2^2 + \lambda\{\alpha||\beta||_1 + \frac{1}{2}(1-\alpha)||\beta||_2^2\}$$

where $0 \leq \alpha \leq 1$ is the mixing parameter to be specified in *islasso()* and *islasso.fit()* via the argument *alpha*, e.g.

```
out2 <- update(out, alpha = 0.5)
lmb.bic <- aic.islasso(out2, method = "bic", interval = c(1, 100))
```

```
>
> Optimization through bic
>
> lambda =   38.8146 bic =  5052.27556
> lambda =   62.1854 bic =  5069.14549
> lambda =   24.3707 bic =  5030.54552
> lambda =   15.4439 bic =  5005.21015
> lambda =    9.9268 bic =  4980.02217
> lambda =    6.5171 bic =  4956.43776
> lambda =    4.4097 bic =  4938.04911
> lambda =    3.1073 bic =  4929.80041
> lambda =    2.3024 bic =  4924.67504
> lambda =    1.8049 bic =  4923.18374
> lambda =    1.4744 bic =  4923.61784
> lambda =    1.7658 bic =  4923.15119
> lambda =    1.7289 bic =  4923.15297
> lambda =    1.7494 bic =  4923.15145
> lambda =    1.7625 bic =  4923.15082
> lambda =    1.7584 bic =  4923.15076
> lambda =    1.7599 bic =  4923.15074
> lambda =    1.7598 bic =  4923.15074
> lambda =    1.7599 bic =  4923.15074
> lambda =    1.7599 bic =  4923.15074
> lambda =    1.7599 bic =  4923.15074
```

```
out3 <- update(out, lambda = lmb.bic)
summary(out3, pval = 0.05)
```

```
>
> Call:
```

```
> islasso(formula = y ~ x2, lambda = lmb.bic, data = diabetes)
>
> Residuals:
>      Min       1Q   Median       3Q      Max
> -141.650  -34.631   -3.122   32.394  145.504
>
>             Estimate Std. Error     Df z value Pr(>|z|)
> (Intercept)  152.133      2.515  1.000  60.486  < 2e-16 ***
> x2sex       -207.978     54.116  0.892  -3.843 0.000121 ***
> x2bmi        450.162     62.865  0.841   7.161 8.03e-13 ***
> x2map        307.253     58.109  0.873   5.287 1.24e-07 ***
> x2ldl        -82.460     31.473  0.528  -2.620 0.008793 **
> x2hdl       -203.142     58.769  0.632  -3.457 0.000547 ***
> x2tch        127.294     62.136  0.500   2.049 0.040497 *
> x2ltg        460.799     61.877  0.787   7.447 9.54e-14 ***
> x2age:sex    143.481     57.309  0.874   2.504 0.012292 *
> x2bmi:map    138.289     63.699  0.831   2.171 0.029932 *
> x2tc:tch    -122.083     55.402  0.372  -2.204 0.027554 *
> x2ldl:ltg    148.727     58.411  0.638   2.546 0.010890 *
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> (Dispersion parameter for gaussian family taken to be 2796.16)
>
>     Null deviance: 2621009  on 441.0  degrees of freedom
> Residual deviance: 1117810  on 399.8  degrees of freedom
> AIC: 4804.1
> Lambda: 1.7599
>
> Number of Newton-Raphson iterations: 7
```

References

- Tibshirani R. *Regression shrinkage and selection via the lasso.* J R Stat Soc: Series B 1996; 58: 267–288
- Cilluffo, G, Sottile, G, La Grutta, S and Muggeo, VMR (2019) *The Induced Smoothed lasso: A practical framework for hypothesis testing in high dimensional regression.* Statistical Methods in Medical Research, online doi: 10.1177/0962280219842890.
- Brown B and Wang Y. *Standard errors and covariance matrices for smoothed rank estimators.* Biometrika 2005; 92: 149–158.