# An Introduction to `islasso`

Gianluca Sottile Giovanna Cilluffo Vito M.R. Muggeo

Ottobre 15, 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 $\beta$. 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 are *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())
```

and *islasso.path* used to fit the regularization path via the induced smoothed lasso framework, i.e.

```
islasso.path(formula, family = gaussian, lambda = NULL, nlambda = 100, lambda.min.ratio = ifelse(nobs <
    nvars, 0.001, 1e-05), 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, *alpha* is elastic-net mixing parameter, *nlambda* is the number of lambda values, *lambda.min.ratio* is the smallest value for lambda (as a fraction of lambda.max), 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 functions are `islasso.fit()` and `islasso.path.fit()` which reads as

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

and

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

whose 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()*. The *lambda* argument in *islasso.path.fit* and *islasso.path* specifies the sequence of positive tuning parameters, user supplied or automatically computed based on *nlambda* and *lambda.min.ratio*.

# 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 could simply apply the lasso using the R package **glmnet**

```
library(islasso)

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

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 a specific criterion chosen between AIC, BIC, AICc, BIC, GCV or GIC. From version 1.4.0 of the R package **islasso** optimal strategy is to built the regularization path

```
out <- islasso.path(y ~ x2, data = diabetes, nlambda = 30L)
```

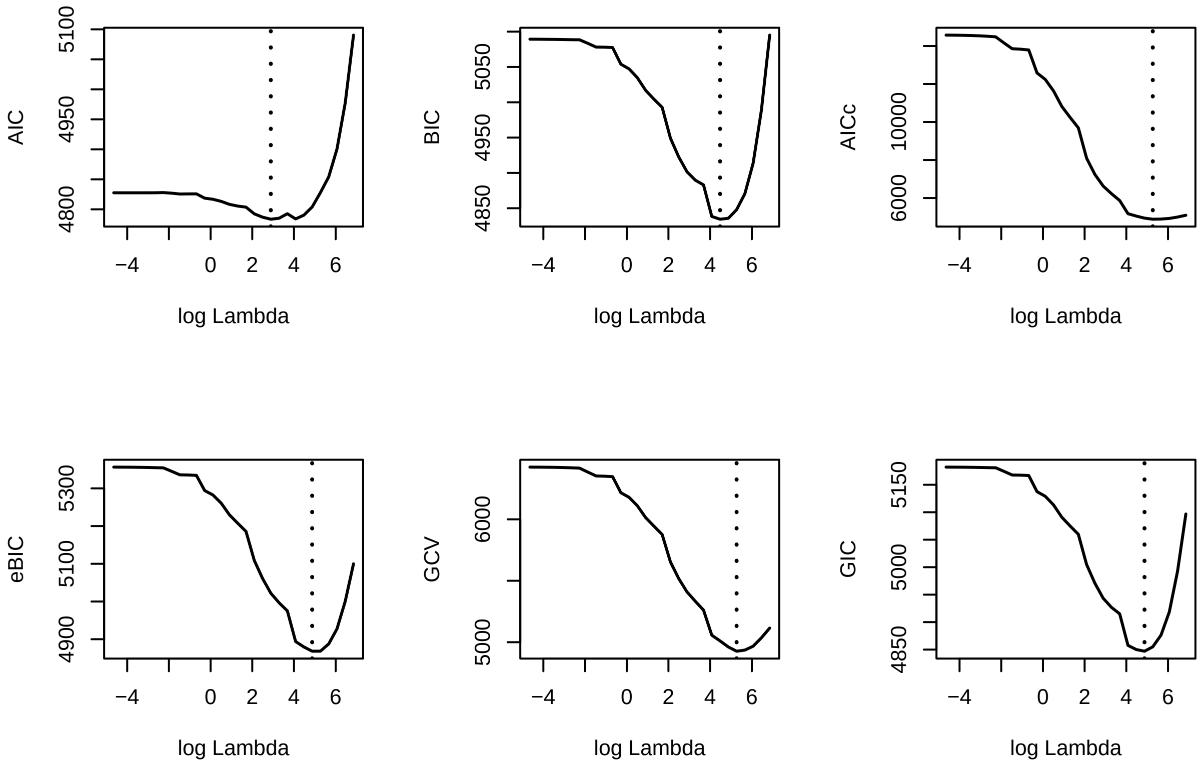
```
>
>
>  Executed in  6.405189 (s)
```

```
out
```

```
>
> Call:
> islasso.path(formula = y ~ x2, nlambda = 30L, data = diabetes)
>
> Coefficients:
>      lambda      df       phi     deviance    logLik
> 1    0.0096 63.9746 2825.8687 1068250.0676 4699.6200
> 2    0.0143 63.9617 2825.8380 1068275.0565 4699.6304
> 3    0.0212 63.9418 2825.8335 1068329.5460 4699.6529
> 4    0.0315 63.9110 2825.9155 1068447.6038 4699.7017
> 5    0.0469 63.8632 2826.2291 1068701.1497 4699.8066
> 6    0.0697 63.7908 2827.1134 1069240.3444 4700.0296
> 7    0.1036 63.6954 2829.4514 1070394.4412 4700.5064
> 8    0.1540 62.7071 2828.4990 1072829.4824 4701.5108
> 9    0.2290 61.7457 2824.6327 1074078.8282 4702.0252
> 10   0.3405 61.6696 2825.9734 1074803.4980 4702.3233
> 11   0.5063 61.5412 2827.3134 1075676.3710 4702.6821
> 12   0.7528 57.5339 2802.9233 1077628.8689 4703.4837
> 13   1.1193 56.3378 2797.8701 1079032.8088 4704.0591
> 14   1.6643 54.1428 2787.5392 1081167.2561 4704.9326
> 15   2.4745 51.0020 2773.3184 1084361.9345 4706.2367
> 16   3.6792 48.6396 2768.5883 1089053.0990 4708.1448
> 17   5.4705 46.2344 2770.5195 1096476.4133 4711.1473
> 18   8.1337 38.2895 2745.5009 1108387.5452 4715.9229
> 19  12.0935 33.1709 2739.9429 1120168.4502 4720.5961
> 20  17.9812 28.8615 2743.3999 1133403.9865 4725.7880
> 21  26.7351 25.5801 2772.6927 1154604.4952 4733.9793
> 22  39.7509 22.0806 2841.3985 1193158.5046 4748.4973
> 23  59.1033 13.2864 2839.7891 1217456.2627 4757.4079
> 24  87.8773 10.7891 2896.6327 1249059.5289 4768.7351
> 25 130.6596  7.7723 3007.4992 1305939.5747 4788.4182
> 26 194.2702  4.8935 3195.3471 1396706.8584 4818.1182
> 27 288.8491  4.0324 3396.8386 1487705.1679 4846.0161
> 28 429.4729  3.2663 3779.2988 1658105.9148 4893.9471
> 29 638.5583  2.8155 4502.7338 1977530.9126 4971.8156
> 30 949.4353  1.1618 5834.8145 2572209.1688 5088.0245
```

and then to choose the best tuning parameter through the one of the criteria listed above using the function *GoF.islasso.path*, e.g.,

```
lmb.best <- GoF.islasso.path(out)
```

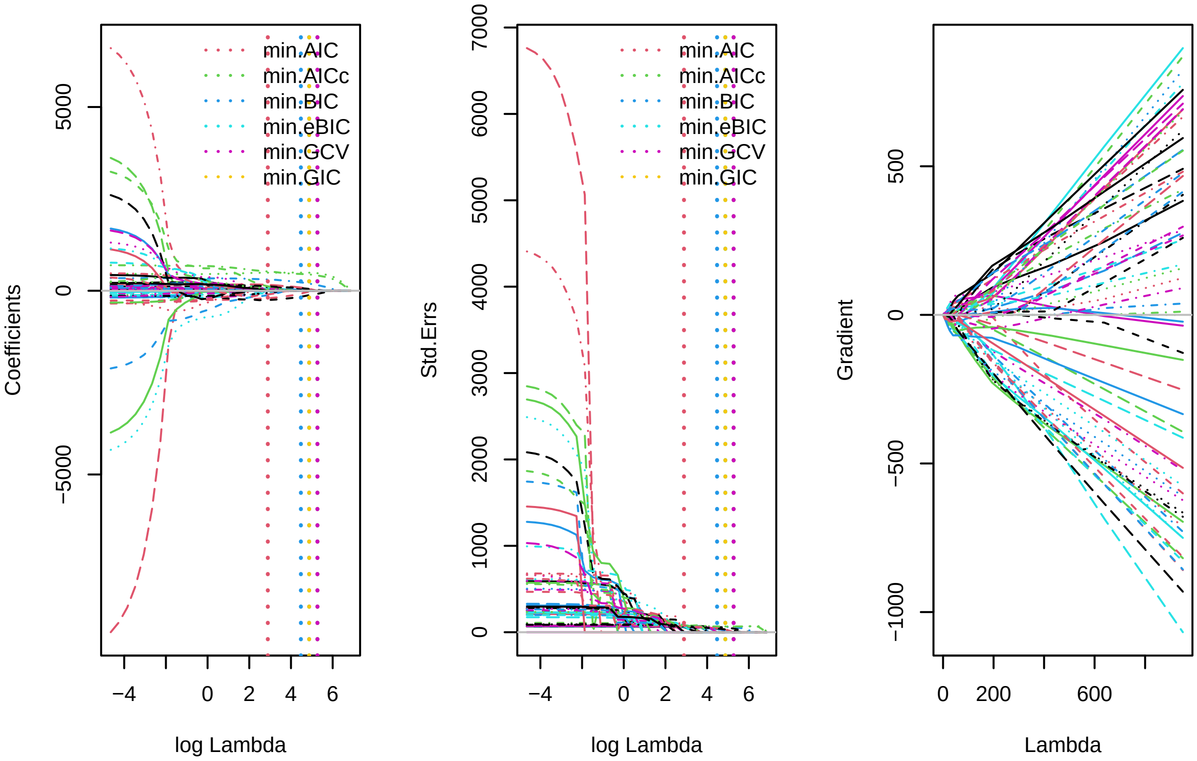


```
lmb.best$lambda.min
```

```
>       AIC       BIC      AICc      eBIC       GCV       GIC
>  17.98115  87.87729 194.27022 130.65964 194.27022 130.65964
```

Using also the regularization path is very usefull to have more insights about coefficients, standard errors and gradient profile

```
par(mfrow = c(1, 3))
plot(out, yvar = "coefficients")
plot(out, yvar = "se")
plot(out, yvar = "gradient")
```



Once selected the best lambda value minimizing for example the BIC criterion, the last step of the strategy consists on fitting a new islasso model.

```
lambda.bic <- lmb.best$lambda.min["BIC"]
out2 <- islasso(y ~ x2, data = diabetes, lambda = lambda.bic)
out2
```

```
>
> Call:
> islasso(formula = y ~ x2, lambda = lambda.bic, data = diabetes)
>
> Coefficients:
> (Intercept)        x2age        x2sex        x2bmi        x2map         x2tc
>      152.13         0.00       -76.08       494.73       228.28         0.00
>       x2ldl        x2hdl        x2tch        x2ltg        x2glu      x2age^2
>        0.00      -160.32         0.00       457.12        23.46         0.00
>     x2bmi^2      x2map^2       x2tc^2      x2ldl^2      x2hdl^2      x2tch^2
>       40.60         0.00         0.00         0.00         0.00         0.00
>     x2ltg^2      x2glu^2    x2age:sex    x2age:bmi    x2age:map     x2age:tc
>        0.00        56.15        85.50         0.00        33.05         0.00
>   x2age:ldl    x2age:hdl    x2age:tch    x2age:ltg    x2age:glu    x2sex:bmi
>        0.00         0.00         0.00         0.00        19.41         0.00
>   x2sex:map     x2sex:tc    x2sex:ldl    x2sex:hdl    x2sex:tch    x2sex:ltg
>        0.00         0.00         0.00         0.00         0.00         0.00
>   x2sex:glu    x2bmi:map     x2bmi:tc    x2bmi:ldl    x2bmi:hdl    x2bmi:tch
>        0.00        71.62         0.00         0.00         0.00         0.00
>   x2bmi:ltg    x2bmi:glu     x2map:tc    x2map:ldl    x2map:hdl    x2map:tch
>        0.00         0.00         0.00         0.00         0.00         0.00
>   x2map:ltg    x2map:glu     x2tc:ldl     x2tc:hdl     x2tc:tch     x2tc:ltg
>       0.00         0.00         0.00         0.00         0.00         0.00
>    x2tc:glu     x2ldl:hdl    x2ldl:tch    x2ldl:ltg    x2ldl:glu    x2hdl:tch
>       0.00         0.00         0.00         0.00         0.00         0.00
>   x2hdl:ltg     x2hdl:glu    x2tch:ltg    x2tch:glu    x2ltg:glu
>       0.00         0.00         0.00         0.00         0.00
>
> Degrees of Freedom: 441 Total (i.e. Null);  431.6 Residual
> Null Deviance: 2621000
> Residual Deviance: 1248000
> AIC: 4789
> Lambda: 87.88
```

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 *0.10*

```
summary(out2, pval = 0.1)
```

```
>
> Call:
> islasso(formula = y ~ x2, lambda = lambda.bic, data = diabetes)
>
> Residuals:
>      Min       1Q   Median       3Q      Max
> -136.951  -40.268   -4.937   36.309  144.918
>
>             Estimate Std. Error     Df z value Pr(>|z|)
> (Intercept)  152.133      2.558  1.000  59.472  < 2e-16 ***
> x2sex        -76.079     45.233  0.754  -1.682 0.092582 .
> x2bmi        494.732     67.792  1.000   7.298 2.92e-13 ***
> x2map        228.284     62.552  0.998   3.649 0.000263 ***
> x2hdl       -160.323     60.826  0.957  -2.636 0.008395 **
> x2ltg        457.117     66.243  1.000   6.901 5.18e-12 ***
> x2age:sex     85.503     45.663  0.828   1.872 0.061144 .
> x2bmi:map     71.616     42.846  0.754   1.671 0.094632 .
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> (Dispersion parameter for gaussian family taken to be 2892.361)
>
>     Null deviance: 2621009  on 441.0  degrees of freedom
> Residual deviance: 1248265  on 431.6  degrees of freedom
> AIC: 4789.3
> Lambda: 87.877
>
> Number of Newton-Raphson iterations: 117
```

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(out2$internal$hi)
```

```
> [1] 10.42713
```

and the corresponding residual degrees of freedom () as reported above. The Wald test (column *z value*) and *p*-values can be used to assess important or significant covariates. Results suggest that variables bmi, map, hdl and ltg to predict the measure of diabetes progression, while sex and two interactions age:sex and bmi:map are borderline informative. Just to be clear, another way to obtain a similar result without computing the regularization path, is to use the function *aic.islasso* which requires a preliminary islasso fit object and a specification of the criterion to be used. Hence

```
lambda.bic2 <- aic.islasso(out2, method = "BIC", interval = c(1, 100))
```

```
>
> Optimization through BIC
>
> lambda =   38.8146 BIC =  1186626.45400
> lambda =   62.1854 BIC =  1220663.88103
> lambda =   24.3707 BIC =  1144936.65608
> lambda =   15.4439 BIC =  1126543.04772
> lambda =    9.9268 BIC =  1114860.56096
> lambda =    6.5171 BIC =  1100616.98252
> lambda =    4.4097 BIC =  1092862.42141
> lambda =    3.1073 BIC =  1088140.54315
> lambda =    2.3024 BIC =  1084683.65193
> lambda =    1.8049 BIC =  1082735.35189
> lambda =    1.4975 BIC =  1080720.95417
> lambda =    1.3075 BIC =  1079691.05472
> lambda =    1.1900 BIC =  1079203.46894
> lambda =    1.1174 BIC =  1078942.96627
> lambda =    1.0726 BIC =  1078750.14705
> lambda =    1.0449 BIC =  1078631.58087
> lambda =    1.0277 BIC =  1078560.13858
> lambda =    1.0171 BIC =  1078516.67689
> lambda =    1.0106 BIC =  1078490.07846
> lambda =    1.0065 BIC =  1078473.73931
> lambda =    1.0040 BIC =  1078463.67928
> lambda =    1.0025 BIC =  1078457.47648
> lambda =    1.0015 BIC =  1078453.64822
> lambda =    1.0010 BIC =  1078451.28431
> lambda =    1.0006 BIC =  1078449.82435
> lambda =    1.0004 BIC =  1078448.92205
> lambda =    1.0002 BIC =  1078448.36480
> lambda =    1.0001 BIC =  1078448.02037
> lambda =    1.0001 BIC =  1078447.80740
> lambda =    1.0000 BIC =  1078447.64462
> lambda =    1.0000 BIC =  1078447.64462
```

```
out3 <- update(out2, lambda = lambda.bic2)
summary(out3, pval = 0.1)
```

```
>
```

```
> Call:
> islasso(formula = y ~ x2, lambda = lambda.bic2, data = diabetes)
>
> Residuals:
>      Min       1Q   Median       3Q      Max
> -154.098  -29.745   -2.456   31.680  149.904
>
>             Estimate Std. Error     Df z value Pr(>|z|)
> (Intercept)  152.133      2.515  1.000  60.492  < 2e-16 ***
> x2sex       -252.843     63.893  1.000  -3.957 7.58e-05 ***
> x2bmi        454.267     81.812  1.000   5.553 2.81e-08 ***
> x2map        337.043     69.942  1.000   4.819 1.44e-06 ***
> x2ltg        669.918     98.963  1.000   6.769 1.29e-11 ***
> x2ltg^2      400.530    242.474  0.987   1.652   0.0986 .
> x2age:sex    165.673     70.763  0.999   2.341   0.0192 *
> x2bmi:map    160.168     84.589  0.997   1.893   0.0583 .
> x2tc:ltg    -775.816    412.762  0.984  -1.880   0.0602 .
> x2ldl:ltg    666.173    317.295  0.991   2.100   0.0358 *
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> (Dispersion parameter for gaussian family taken to be 2795.592)
>
>     Null deviance: 2621009  on 441.0  degrees of freedom
> Residual deviance: 1078104  on 385.6  degrees of freedom
> AIC: 4816.4
> Lambda: 1
>
> Number of Newton-Raphson iterations: 331
```

Comparisons between methods to select the tuning parameter and further discussions are out of the scope of this short note. We conclude this 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.path()* via the argument *alpha*, e.g.

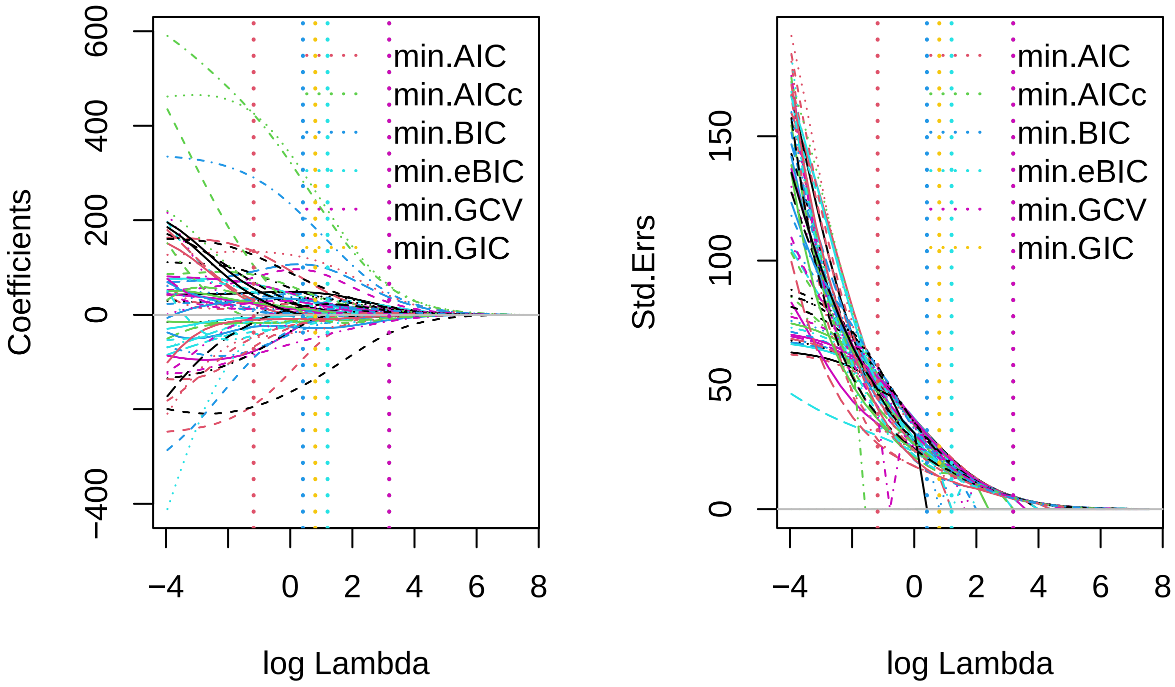
```
# update the islasso path to fit an elastic-net model
out4 <- update(out, alpha = 0.5)
```

```
>
>
>  Executed in  4.58151 (s)
```

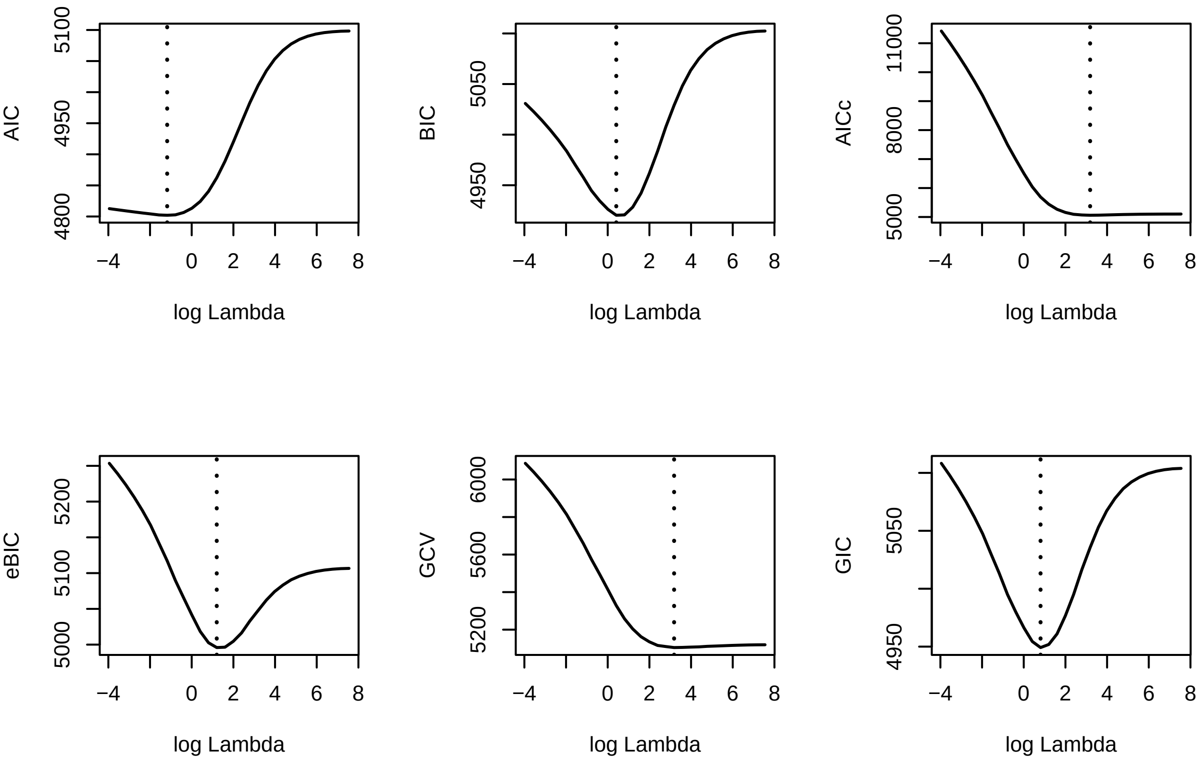
```
out4
```

```
>
> Call:
> islasso.path(formula = y ~ x2, nlambda = 30L, alpha = 0.5, data = diabetes)
>
> Coefficients:
>       lambda      df       phi     deviance    logLik
> 1     0.0192 53.3378 2788.1530 1083649.6469 4705.9463
> 2     0.0285 51.8074 2785.2072 1086767.2213 4707.2160
> 3     0.0424 50.1275 2783.2995 1090698.4552 4708.8120
> 4     0.0630 48.2721 2782.6748 1095616.6721 4710.8006
> 5     0.0937 46.2059 2783.6291 1101743.9506 4713.2657
> 6     0.1394 43.8866 2786.8431 1109479.6886 4716.3583
> 7     0.2072 41.0769 2792.4528 1119558.9740 4720.3556
> 8     0.3081 38.1354 2806.0099 1133248.0879 4725.7272
> 9     0.4581 34.7083 2830.2195 1152724.9908 4733.2592
> 10    0.6811 31.3060 2874.6662 1180608.0556 4743.8234
> 11    1.0127 27.5587 2943.2851 1219818.8439 4758.2648
> 12    1.5057 23.4888 3042.6670 1273390.0639 4777.2621
> 13    2.2387 19.6313 3182.1778 1344052.3655 4801.1330
> 14    3.3286 16.1734 3368.4809 1434388.7081 4829.8849
> 15    4.9491 13.0060 3603.1211 1545717.4257 4862.9241
> 16    7.3585 10.2915 3884.5006 1676971.8784 4898.9478
> 17   10.9409  7.8529 4197.9048 1822508.3980 4935.7329
> 18   16.2674  6.0660 4525.2435 1972707.3606 4970.7362
> 19   24.1870  4.4757 4836.5190 2116094.6451 5001.7493
> 20   35.9623  3.3819 5114.2550 2243204.8073 5027.5326
> 21   53.4703  2.5663 5344.8466 2348705.8621 5047.8465
> 22   79.5018  1.9574 5525.4481 2431432.4045 5063.1468
> 23  118.2066  1.5965 5661.8544 2493500.2320 5074.2882
> 24  175.7546  1.3227 5760.3452 2538453.4770 5082.1857
> 25  261.3193  1.1559 5829.6745 2569977.4924 5087.6409
> 26  388.5404  1.0717 5877.0761 2591369.1051 5091.3047
> 27  577.6982  1.0271 5907.8711 2605211.0298 5093.6594
> 28  858.9459  1.0138 5926.8625 2613664.4353 5095.0913
> 29 1277.1167  1.0051 5938.4909 2618844.1267 5095.9664
> 30 1898.8705  1.0003 5943.1517 2620928.3993 5096.3180
```

```
# some diagnostic plot
par(mfrow = c(1, 2))
plot(out4, yvar = "coefficients")
plot(out4, yvar = "se")
```



```
# select the best tuning parameter
lmb.best2 <- GoF.islasso.path(out4)
```



```
lmb.best2$lambda.min
```

```
>          AIC          BIC         AICc         eBIC          GCV          GIC
>  0.3080847  1.5056683 24.1870426  3.3285790 24.1870426  2.2386907
```

```
# fit a new islasso model with elastic-net penalty
lambda.bic3 <- lmb.best2$lambda.min["BIC"]
out5 <- update(out2, alpha = 0.5, lambda = lambda.bic3)
summary(out5, pval = 0.1)
```

```
>
> Call:
> islasso(formula = y ~ x2, lambda = lambda.bic3, alpha = 0.5,
>     data = diabetes)
>
> Residuals:
>      Min       1Q   Median       3Q      Max
> -120.696  -40.098   -5.179   39.143  154.979
>
>             Estimate Std. Error     Df z value Pr(>|z|)
> (Intercept)  152.133      2.623  1.000  58.009  < 2e-16 ***
> x2sex        -92.139     30.576  0.524  -3.013 0.002583 **
> x2bmi        300.885     28.831  0.443  10.436  < 2e-16 ***
> x2map        209.877     30.044  0.484   6.986 2.84e-12 ***
> x2hdl       -148.335     23.292  0.365  -6.369 1.91e-10 ***
> x2tch        121.563     19.401  0.280   6.266 3.71e-10 ***
> x2ltg        281.328     27.947  0.417  10.066  < 2e-16 ***
> x2glu        106.431     30.327  0.481   3.509 0.000449 ***
> x2bmi^2       95.651     28.761  0.455   3.326 0.000882 ***
> x2glu^2       45.792     27.564  0.396   1.661 0.096655 .
> x2age:sex     75.138     29.930  0.495   2.510 0.012058 *
> x2age:ldl    -54.473     22.136  0.336  -2.461 0.013861 *
> x2age:ltg     48.020     27.292  0.395   1.760 0.078491 .
> x2sex:hdl     46.184     25.423  0.404   1.817 0.069278 .
> x2bmi:map     76.316     28.571  0.434   2.671 0.007559 **
> x2tch:glu     38.543     20.768  0.273   1.856 0.063466 .
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> (Dispersion parameter for gaussian family taken to be 3040.087)
>
>     Null deviance: 2621009  on 441  degrees of freedom
> Residual deviance: 1270727  on 418  degrees of freedom
> AIC: 4824.4
> Lambda: 1.5057
>
> Number of Newton-Raphson iterations: 72
```

```
# or select the best tuning parameter using BIC with an islasso object
lambda.bic4 <- aic.islasso(out5, method = "BIC", interval = c(1, 100))
```

```
>
> Optimization through BIC
>
> lambda =    38.8146 BIC =   2261495.87590
> lambda =    62.1854 BIC =   2380157.56610
> lambda =   24.3707 BIC =  2113946.44497
> lambda =   15.4439 BIC =  1947986.39280
> lambda =    9.9268 BIC =  1780948.88514
> lambda =    6.5171 BIC =  1630499.94033
> lambda =    4.4097 BIC =  1507563.23945
> lambda =    3.1073 BIC =  1414117.85187
> lambda =    2.3024 BIC =  1347003.02173
> lambda =    1.8049 BIC =  1300801.56235
> lambda =    1.4975 BIC =  1270032.16194
> lambda =    1.3075 BIC =  1250101.26692
> lambda =    1.1900 BIC =  1237370.22027
> lambda =    1.1174 BIC =  1229350.89849
> lambda =    1.0726 BIC =  1224335.64044
> lambda =    1.0449 BIC =  1221211.65175
> lambda =    1.0277 BIC =  1219272.18324
> lambda =    1.0171 BIC =  1218070.18296
> lambda =    1.0106 BIC =  1217326.02809
> lambda =    1.0065 BIC =  1216865.62680
> lambda =    1.0040 BIC =  1216580.89661
> lambda =    1.0025 BIC =  1216404.85242
> lambda =    1.0015 BIC =  1216296.02390
> lambda =    1.0010 BIC =  1216228.75378
> lambda =    1.0006 BIC =  1216187.17458
> lambda =    1.0004 BIC =  1216161.47571
> lambda =    1.0002 BIC =  1216145.59235
> lambda =    1.0001 BIC =  1216135.77568
> lambda =    1.0001 BIC =  1216129.70856
> lambda =    1.0000 BIC =  1216125.06737
> lambda =    1.0000 BIC =  1216125.06737
```

```
out6 <- update(out5, lambda = lambda.bic4)
summary(out6, pval = 0.1)
```

```
>
> Call:
> islasso(formula = y ~ x2, lambda = lambda.bic4, alpha = 0.5,
>     data = diabetes)
>
> Residuals:
>      Min       1Q   Median       3Q      Max
> -125.658  -39.099   -4.608   37.297  153.141
>
>             Estimate Std. Error     Df z value Pr(>|z|)
> (Intercept)  152.133      2.579  1.000  59.000  < 2e-16 ***
> x2sex       -120.092     35.922  0.617  -3.343 0.000828 ***
> x2bmi        340.191     35.261  0.528   9.648  < 2e-16 ***
> x2map        235.494     36.028  0.576   6.536 6.30e-11 ***
> x2hdl       -161.981     27.771  0.419  -5.833 5.45e-09 ***
> x2tch        126.745     23.542  0.318   5.384 7.29e-08 ***
> x2ltg        321.544     34.103  0.497   9.428  < 2e-16 ***
> x2glu        105.758     36.457  0.574   2.901 0.003721 **
> x2bmi^2       95.478     34.957  0.540   2.731 0.006309 **
> x2age:sex     91.408     35.669  0.585   2.563 0.010386 *
> x2age:ldl    -61.187     26.062  0.383  -2.348 0.018886 *
> x2age:ltg     57.926     33.629  0.472   1.722 0.084980 .
> x2sex:hdl     53.354     30.348  0.469   1.758 0.078728 .
> x2bmi:map     89.346     35.001  0.518   2.553 0.010691 *
> x2tc:tch     -33.509     19.898  0.212  -1.684 0.092176 .
> x2tch:glu     43.080     25.766  0.318   1.672 0.094528 .
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> (Dispersion parameter for gaussian family taken to be 2938.198)
>
>     Null deviance: 2621009  on 441.0  degrees of freedom
> Residual deviance: 1215954  on 413.8  degrees of freedom
> AIC: 4813.2
> Lambda: 1
>
> Number of Newton-Raphson iterations: 43
```

# 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.