LINselect: R package for estimator selection

Yannick Baraud, Annie Bouvier, Christophe Giraud, and Sylvie Huet,

INRA, MIA, UR341
Jouy-en-Josas F-78352
e-mail: Sylvie.Huet@jouy.inra.fr

Université de Nice Sophia-Antipolis, Laboratoire J-A Dieudonné, UMR 6621
Parc Valrose, Nice F-06108
e-mail: yannick.baraud@unice.fr

Université Paris Sud, Laboratoire de Mathématiques, UMR 8628
Orsay Cedex F-91405
e-mail: christophe.giraud@math.u-psud.fr

Contact: Annie.Bouvier@jouy.inra.fr

Contents

1. Introduction

We consider the linear Gaussian regression framework

$$Y_i = f_i + \boldsymbol{\epsilon}_i,\ i = 1, \ldots, n.$$

The vector $f = (f_1, \ldots, f_n)$ is assumed to be of the form

$$f = X\beta \tag{1}$$

where X is a $n \times p$ matrix, β is an unknown p-dimensional vector and p some integer larger than 1 (and possibly larger than n). The errors $\boldsymbol{\epsilon}_i$ are independent centered Gaussian random variables with unknown variance σ^2. In what follows we denote by $\|\cdot\|$ the euclidean norm in $\mathbb{R}^n$.
The package LINselect deals with two problems :

1. Choosing the tuning parameter in the Lasso procedure.

   The lasso estimator The Lasso estimator defined by

   $$f^{\texttt{lasso}}(\lambda) = X\beta^{\texttt{lasso}}(\lambda) \text{ with } \beta^{\texttt{lasso}}(\lambda) = \arg\min_{\beta \in \mathbb{R}^p} \left\{ \|Y - X\beta\|^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{2}$$

   depends on the choice of the parameter $\lambda \geq 0$. Selecting this parameter among $\Lambda \subset \mathbb{R}_+$ amounts to selecting an estimator among the family $\mathcal{F} = \{f^{\texttt{lasso}}(\lambda),\ \lambda \in \Lambda\}$.

The Gauss-lasso estimator If one is interested in estimating the support of β, which is the set of indices j corresponding to the non zero coefficients $\beta_j, j = 1, \ldots, p$, one may prefer to consider the Gauss-lasso estimator defined as follows. For $J \subset \{1, \ldots, p\}$, let X_J be the matrix obtained by keeping the columns of X with index in J. For each λ, let $J^{\texttt{lasso}}(\lambda)$ be the support of $\beta^{\texttt{lasso}}(\lambda)$, and define the Gauss-lasso estimator of f as the orthogonal projection of Y onto the columns of $X_{J^{\texttt{lasso}}(\lambda)}$, denoted

$$f^{\texttt{Glasso}}(\lambda) = \Pi_{J^{\texttt{lasso}}(\lambda)} Y.$$

Selecting the parameter among $\Lambda \subset \mathbb{R}_+$ amounts to selecting an estimator among the family $\mathcal{F} = \{f^{\texttt{Glasso}}(\lambda),\ \lambda \in \Lambda\}$.

2. Variable selection. Several procedures are available for estimating the support of β. One may choose for example a procedure based on ridge regression, random forest or PLS. For a given procedure, denoted `proc`, we get a collection of subsets J indexed by tuning parameters peculiar to the procedure. This collection is denoted $\mathcal{J}^{\texttt{proc}}$:

$$\mathcal{J}^{\texttt{proc}} = \{J^{\texttt{proc}}(\lambda), \lambda \in \Lambda^{\texttt{proc}}\}.$$

Selecting the subset J among all subsets in the union of collections $\mathcal{J}^{\texttt{proc}}$ comes to selecting an estimator among the family

$$\mathcal{F} = \left\{\widehat{f}_J = \Pi_J Y, \text{ for } J \in \mathcal{J}^{\texttt{proc}}, \texttt{proc} \in \{\texttt{lasso}, \texttt{ridge}, \texttt{pls}, ...\}\right\}. \tag{3}$$

The package LINselect gives answers to these problems using the criteria proposed by Baraud et al. [2], called the `LINselect` criteria, and the V-fold Cross-Validation criteria. The `LINselect` criteria is specially designed to handle the case where the sample size n is smaller than the number of variables p. Its theoretical performances have been assessed in [2] and [3]. It is an alternative to V-fold Cross-Validation for which little is known in a high-dimensional setting from a theoretical point of view [1].

2. Choice of the tuning parameter in the Lasso procedure

2.1. The `LINselect` *criteria*

Let $\lambda_1 > \lambda_2 > \ldots > \lambda_L$ be the lasso regularization path such that the cardinal of $J^{\texttt{lasso}}(\lambda_L) \leq d_{\max}$ for some $d_{\max} \leq \min\{n, p\}$. Let d_ℓ be the rank of the matrix $X_{J^{\texttt{lasso}}(\lambda_\ell)}$.
The `LINselect` criteria is defined as follows: for each ℓ,

$$\begin{aligned}\mathrm{Cr}(f^{\texttt{lasso}}(\lambda_\ell)) \quad = \quad & \inf_{1 \leq m \leq L} \left\{\|Y - \Pi_{J^{\texttt{lasso}}(\lambda_m)} f^{\texttt{lasso}}(\lambda_\ell)\|^2 + \alpha \|f^{\texttt{lasso}}(\lambda_\ell) - \Pi_{J^{\texttt{lasso}}(\lambda_m)} f^{\texttt{lasso}}(\lambda_\ell)\|^2 + \right. \\ & \left. + K \mathrm{pen}(d_m) \widehat{\sigma}_m^2 \right\}\end{aligned}$$

where

$$\widehat{\sigma}_m^2 = \frac{\|Y - \Pi_{J^{\texttt{lasso}}(\lambda_m)} Y\|^2}{n - d_m},$$

pen is a penalty function defined in Section 4, K a constant greater than 1, that can be chosen equal to 1.1, and α a positive constant that can be chosen equal to 0.5. Finally, we choose $\lambda^{\texttt{lasso}}_{\texttt{LINselect}} = \lambda_{\widehat{\ell}}$, the minimizer of $\mathrm{Cr}(f^{\texttt{lasso}}(\lambda_\ell))$ with respect to $\ell \in \{1, \ldots, L\}$.

2.2. The Gauss-lasso estimator

For the Gauss-lasso estimator, the `LINselect` criteria is simplified: for each ℓ,

$$\mathrm{Cr}(f^{\texttt{Glasso}}(\lambda_\ell)) = \|Y - f^{\texttt{Glasso}}(\lambda_\ell)\|^2 + K \mathrm{pen}(d_\ell) \widehat{\sigma}_\ell^2$$

and $\lambda^{\texttt{Glasso}}_{\texttt{LINselect}} = \lambda_{\widehat{\ell}}$, the minimizer of $\mathrm{Cr}(f^{\texttt{Glasso}}(\lambda_\ell))$ with respect to $\ell \in \{1, \ldots, L\}$.

2.3. The V-fold CV criteria

For each $\lambda_\ell, \ell = 1, \ldots, L$ given by the regularization path of the lasso algorithm, see Equation (**??**), the V-fold cross-validated mean squared prediction error of $f^{\texttt{lasso}}(\lambda_\ell)$ is estimated and $\lambda^{\texttt{lasso}}_{\texttt{CV}}$ is the minimizer of this criteria. The `tunelasso` function uses the function `cv.enet` from the package `elasticnet`.
For the `Glasso` estimator, the `tuneLasso` function calculates the V-fold cross-validated mean squared prediction error on a grid of values for λ constructed as the concatenation of the regularization paths of the lasso algorithm applied to the regression of Y versus X and applied to each of the V learning data subsets .

2.4. The function `tuneLasso`

The function `tuneLasso` takes as inputs:

`Y`	the response vector Y with n components.
`X`	the covariate matrix X of size $n \times p$.
`dmax`	the maximum number of variables that will be considered for estimating the support of β. Let $$\begin{aligned} D &= \min\{3 * p/4, n-5\} \text{ if } p \geq n \\ D &= \min\{p, n-5\} \text{ if } p < n. \end{aligned}$$ If `dmax` $\geq D$, then `dmax` will be set to D. Default value is `dmax` $= D$.
`method`	the estimation method. Default value is `c("lasso","Glasso")`.
`LINselect`	if `LINselect=TRUE` the choice of the tuning parameter is based on the `LINselect` criteria.
`Vfold`	if `Vfold=TRUE` the choice of the tuning parameter is based on the V-fold CV criteria. Default value is `Vfold=TRUE`.
`V`	the value of V is the V-fold procedure. Default value is `V` $= 10$.
`K`	a dimensionless tuning parameter, `K` > 1. Default value is `K` $= 1.1$.
`a`	a dimensionless tuning parameter, `a` > 0. Default value is `a` $= 0.5$.
`max.steps`	maximum number of steps in the lasso procedure. Default value is `max.steps` $= 2\min(n, p)$.

The output of the function `tuneLasso` is a list with one or two components according to the argument `method`.

`lasso`	Only present if `method` contains `lasso`. A list with one or two components according to `Vfold` and `LINselect`.
`lasso$Ls`	Only present if `LINselect`=TRUE. A list with components
`lasso$Ls$support`	The estimated support of β, $J^{\texttt{lasso}}(\lambda^{\texttt{lasso}}_{\texttt{LINselect}})$.
`lasso$Ls$coef`	The estimation of β, $\beta^{\texttt{lasso}}(\lambda^{\texttt{lasso}}_{\texttt{LINselect}})$. The first component of this vector is the estimated intercept.
`lasso$Ls$fitted`	The fitted value of the response, $f^{\texttt{lasso}}(\lambda^{\texttt{lasso}}_{\texttt{LINselect}})$.
`lasso$Ls$crit`	Values of the `LINselect` criteria, $\mathrm{Cr}(f^{\texttt{lasso}}(\lambda_\ell))$ for $\lambda_1 > \lambda_2 > \ldots > \lambda_L$ the regularization path of the lasso algorithm such that the support of $\beta^{\texttt{lasso}}(\lambda_L)$ has cardinality smaller than `dmax`.
`lasso$Ls$lambda`	Values of the tuning parameters corresponding to the regularization path of the lasso algorithm.
`lasso$CV`	Only present if `Vfold`=TRUE. A list with the same components as `lasso$Ls`, where `Ls` is replaced by `CV`.
`Glasso`	Only present if `method` contains `Glasso`. A list with the same components as `lasso`, where `lasso` is replaced by `Glasso`.

3. Variable selection

The problem is to estimate the support of β by selecting an estimator of f in the collection $\mathcal{F}$ defined in (2).
The first task is to construct the set of subsets J of $\{1,\dots,p\}$ that will be considered. For this purpose, we gather variable selection procedures based on the lasso, ridge regression, elactic-net, PLS1 regression, adaptive lasso, random forest and on an exhaustive research among the subsets of $\{1,\dots,,p\}$ with small cardinality.

- The lasso procedure gives the collection of the active sets of cardinality less than $d_{\max}$:

  $$\mathcal{J}^{\texttt{lasso}} = \left\{J^{\texttt{lasso}}(\lambda_\ell), \ell = 1,\dots d_{\max}\right\}$$

  as defined in Section 2.1.
- The ridge procedure is based on the minimisation of

  $$\|Y - X\beta\|^2 + \mu\sum_{j=1}^{p}\beta_j^2$$

  with respect to β, for some positive μ. For a fixed μ, let $\beta^{\texttt{ridge}}(\mu)$ be this estimator and let $j_1,\dots,j_p$ be such that $|\beta^{\texttt{ridge}}_{j_1}(\mu)| > \dots > |\beta^{\texttt{ridge}}_{j_p}(\mu)|$. Define

  $$\mathcal{J}^{\texttt{ridge}}(\mu) = \left\{\{j_1,\dots,j_k\}, k = 1,\dots,d_{\max}\right\}.$$

  For some values of μ, say $\mu_1,\dots,\mu_M$,

  $$\mathcal{J}^{\texttt{ridge}} = \left\{J^{\texttt{ridge}}(\mu_m), m = 1,\dots,M\right\}.$$

- The elastic-net procedure mixes the penalities of the lasso and the ridge procedures : $\beta^{\texttt{en}}(\lambda,\mu)$ is the minimizer of

  $$\|Y - X\beta\|^2 + \lambda\sum_{j=1}^{p}|\beta_j| + \mu\sum_{j=1}^{p}\beta_j^2.$$

  For a fixed μ, $\mathcal{J}^{\texttt{en}}(\mu)$ is the collection of the active sets of cardinality less than $d_{\max}$ given by the elasticnet procedure. For some values of μ, say $\mu_1,\dots,\mu_M$,

  $$\mathcal{J}^{\texttt{en}} = \{\mathcal{J}^{\texttt{en}}(\mu_m), m = 1,\dots,M\}.$$

- The partial least squares regression (PLSR1) aims to reduce the dimensionality of the regression problem by calculating a small number of components that are usefull for predicting Y. For a fixed h, let $t_1,\dots,t_h$ be the first uncorrelated latent components that are highly correlated with Y. Doing as for the ridge procedure we deduce a collection of indices $\mathcal{J}^{\texttt{pls}}(h)$ from the PLS regression coefficients $\beta^{\texttt{pls}}(h)$ calculated with the first h components. The total collection of sets is thus

  $$\mathcal{J}^{\texttt{pls}} = \left\{\mathcal{J}^{\texttt{pls}}(h), h = 1,\dots,H\right\}.$$

- The adaptive-lasso procedure starts with a preliminary estimator $\widetilde{\beta}$. Then the lasso procedure is applied replacing $|\beta_j|$ in the penalty by $|\beta_j|/|\widetilde{\beta}_j|$, for each $j = 1,\dots p$. We consider two different preliminary estimator
  - the ridge estimator $\beta^{\texttt{ridge}}(\mu)$. For a fixed μ, $\mathcal{J}^{\texttt{ALridge}}(\mu)$ is the collection of active sets of cardinality less than $d_{\max}$ given by the lasso procedure. For some values $\mu_1,\dots,\mu_M$,

    $$\mathcal{J}^{\texttt{ALridge}} = \left\{\mathcal{J}^{\texttt{ALridge}}(\mu_m), m = 1,\dots,M\right\}.$$

  - the PLSR1 estimator $\beta^{\texttt{pls}}(h)$. For a fixed h, $\mathcal{J}^{\texttt{ALpls}}(h)$ is the collection of active sets of cardinality less than $d_{\max}$ given by the lasso procedure, and

$$\mathcal{J}^{\texttt{ALpls}} = \left\{\mathcal{J}^{\texttt{ALpls}}(h), h = 1, \ldots, H\right\}.$$

- The random forest algorithm returns measures of variable importance. For a fixed s, let $j_1, \ldots, j_p$ be the ranks of the importance variable measures given by random forest when the number of variables randomly chosen at each split equals s. Then

$$\mathcal{J}^{\texttt{rF}}(s) = \{\{j_1, \ldots, j_k\}\, k = 1, \ldots, d_{\max}\}.$$

  For some values of s, say $s_1, \ldots, s_S$,

$$\mathcal{J}^{\texttt{rF}} = \left\{\mathcal{J}^{\texttt{rF}}(s_\ell), \ell = 1, \ldots, S\right\}.$$

  We consider two importance measures. The first one is based on the decrease in the mean square error of prediction after permutation of each of the variables. It leads to the collection $\mathcal{J}^{\texttt{rFmse}}$. The second one is based on the decrease in node impurities, and leads similarly to the collection $\mathcal{J}^{\texttt{rFpur}}$.
- The exhaustive procedure considers the collection of all subsets of $\{1, \ldots, p\}$ with dimension smaller than $d_{\max}$. We denote the corresponding collection $\mathcal{J}^{\texttt{exhaustive}}$.

3.1. The `LINselect` *criteria*

We describe here the LINselect criteria for performing variable selection. For each procedure `proc` we minimize with respect to $J \in \mathcal{J}^{\texttt{proc}}$ the criteria defined in Section 2.2, which can be re-written as follows:

$$\texttt{crit}(J) = \|Y - \Pi_J Y\|^2 \left(1 + K\mathrm{pen}(d_J)/(n - d_J)\right), \tag{4}$$

where d_J is the rank of the matrix X_J. We get thus the best choice $J^{\texttt{proc}}$ for each procedure. Then we choose $\widehat{J}$ as the minimizer of $\texttt{crit}(J^{\texttt{proc}})$ over all procedures.

3.2. The function `VARselect`

The function `VARselect` takes as main inputs:

`Y`, `X`, `dmax`, `K`, `max.steps` : the same inputs as in `tuneLasso`.

`method`	the procedure. Default value is `c("lasso","ridge","pls","en","ALridge","ALpls","rF","exhaustive")`.
`en.lambda`	the values of μ in the elastic-net algorithm. Default value is `en.lambda` $= c(0.01, 0.1, 0.5, 1, 2, 5)$.
`ridge.lambda`	the same as `en.lambda`.
`rF.lmtry`	determines the tuning parameter s in the random forest algorithm: $s = p/$`rF.lmtry`. Default value is `rF.lmtry` $= 2$.
`pls.ncomp`	the parameter H is the PLS1 procedure. Default value is `pls.ncomp` $= 5$.
`ALridge.lambda`	the values of μ in the adaptive lasso procedure using the ridge method for calculationg the preliminary estimator. Default value is the same as `en.lambda`.
`ALpls.ncomp`	the parameter H is the adaptive lasso procedure using the PLS1 method for calculationg the preliminary estimator. Default value is the same as `pls.ncomp`.
`pen.crit`	the penalty values. By default the values are calculated using the function `penalty`, see Section 4.

For the other inputs, see the help file.

The output of the function `VARselect` is a list whose components are: for each `proc` $\in$ `method`:

`proc`	A list with components:
`proc$support`	The estimated support of β: $J^{\texttt{proc}}$.
`proc$crit`	Value of the `LINselect` criteria calculated in $J^{\texttt{proc}}$.
`proc$fitted`	The fitted value of the response: $\Pi_{J^{\texttt{proc}}} Y$.
`proc$coef`	The estimation of β: the minimizer with respect to β of $\|Y - X_{J^{\texttt{proc}}}\beta\|^2$ (the first component of β is the estimated intercept).
`summary`	Only present if more than one procedure is wanted. A list with the same components as before where $J^{\texttt{proc}}$ is replaced by $\widehat{J}$. An additional component named `summary.method` lists the procedures for which the minimum is achieved.

More ouputs are available, see the help file.

4. The penalty function

The penalty function occuring in the `LINselect` criteria is defined as follows. For all integer d satisfying $0 \leq d < n-2$, let us first define a weight function Δ as follows:

$$\Delta(d) = \log \left(\begin{array}{c} p \\ d \end{array} \right) + 2\log(d+1).$$

Then we set

$$\mathrm{pen}(d) = \frac{n-d}{n-d-1} \phi^{-1}\left(\exp[-\Delta(d)]\right)$$

with

$$\phi(x) = P\left(U > \frac{x}{d+3}\right) - \frac{x}{d+1} P\left(V > \frac{(n-d+1)x}{(n-d-1)(d+1)}\right)$$

where U and respectively V are Fisher variables with $d+3$ and $n-d-1$, respectively $d+1$ and $n-d+1$, degrees of freedom.

The function `penalty` takes as inputs

`Delta`	The vector of weights $\Delta(d)$ for $d = 0, \ldots,$ `dmax`.
`n`	The number of observations, n.
`p`	The number of variables, p.
`K`	The constant K in the penalty function.

The output of the function `penalty` is a vector with the same dimension as `Delta` containing the values $K\mathrm{pen}(d)$ for $d = 0, \ldots,$ `dmax`.

Other penalty functions It may be of interest to change the penalty function. In the case considered by the function `VARselect`, if we denote by $|J|$ the cardinality of the subset J and by $\mathcal{J}$ the collection of all subsets of $\{1, \ldots, p\}$ with maximum cardinality $d_{\max}$, the weights Δ are chosen so that

$$\sum_{d=0}^{d_{\max}} \sum_{J \in \mathcal{J}, |J|=d} \exp\left(-\Delta(d)\right) \leq 1.$$

In some cases, we may be interested by other penalty function. For exemple, one could consider the AMDL penalty where $K\mathrm{pen}(d)$ in Equation (3) is replaced by $(n-d)\left[\exp(3d\log(n)/n) - 1\right]$. The argument `pen.crit` of the function `VARselect` allows to give values for the penalty.

References

[1] S. Arlot and A. Célisse. A survey of cross-validation procedures for model selection. *Stat. Surv.*, 4:40–79, 2010.

[2] Y. Baraud, C. Giraud, and S. Huet. Estimator selection in the gaussian setting, 2010. arXiv:1007.2096v2.

[3] C. Giraud, S. Huet, and N. Verzelen. High-dimensional regression with unknown variance. *Statist. Sci.*, 27(4):500–518, 2012.