Linear Regression in High Dimension and/or for Correlated Inputs

Article *in* EAS Publications Series · January 2015

DOI: 10.1051/eas/1466011

2 authors, including:

Julien Jacques
Université Lumiere Lyon 2
78 PUBLICATIONS **868** CITATIONS

SEE PROFILE

Title : will be set by the publisher
Editors : will be set by the publisher
EAS Publications Series, Vol. ?, 2014

LINEAR REGRESSION IN HIGH DIMENSION AND/OR FOR CORRELATED INPUTS

Julien JACQUES[1] and Didier FRAIX-BURNET[2]

Abstract. Ordinary least square is the common way to estimate linear regression models. When inputs are correlated or when they are too numerous, regression methods using derived inputs directions or shrinkage methods can be efficient alternatives. Methods using derived inputs directions build new uncorrelated variables as linear combination of the initial inputs, whereas shrinkage methods introduce regularization and variable selection by penalizing the usual least square criterion. Both kinds of methods are presented and illustrated thanks to the **R** software on a astronomy dataset.

1 Introduction

Multivariate linear regression assumes the following model

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i \tag{1.1}$$

where y_i is the dependent variable to predict and the x_j's are the explanatory variables (covariates, inputs, etc.). The residuals ϵ_i's are usually assumed to be independent and identically distributed:

$$\epsilon_i \sim \mathcal{N}(0, \sigma^2).$$

In matrix notation, this model can be written:

$$\begin{array}{rcl} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} & = & \beta_0 + \begin{bmatrix} x_{11} & \dots & x_{1p} \\ \vdots & \vdots & \vdots \\ x_{n1} & \dots & x_{np} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix} \\ \boldsymbol{y} & = & \beta_0 + \boldsymbol{X} \boldsymbol{\beta} + \boldsymbol{\epsilon} \end{array}$$

[1] Université Lille 1 & CNRS & Inria
[2] Institut de Planétologie et d'Astrophysique de Grenoble (IPAG)

DOI: (will be inserted later)

The model parameters $\boldsymbol{\beta}$ (and σ^2) can be equivalently estimated by maximum likelihood or by ordinary least squares (OLS), what leads to $\hat{\beta}_0^{ols} = \frac{1}{n}\sum_{i=1}^{n} y_i = \bar{y}$ and:

$$\hat{\boldsymbol{\beta}}^{ols} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}.$$

This estimator is unbiased ($E[\hat{\boldsymbol{\beta}}^{ols} - \boldsymbol{\beta}]$=0) and of smallest variance among linear estimators:

$$V(\hat{\boldsymbol{\beta}}^{ols}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}.$$

In that sense, ordinary least squares estimation of linear regression is often the best choice. Nevertheless, in several situations alternatives to OLS can be preferable. Two examples are described below.

Correlated inputs. When the inputs are correlated, $\mathbf{X}'\mathbf{X}$ is ill-conditioned (some eigenvalues are closed to 0). Consequently, the variance of $\hat{\boldsymbol{\beta}}^{ols}$ is very large and the estimators can take artificially large values.
As an example, let consider the model $y_i = 3 + x_{i1} + x_{i2} + \epsilon_i$ in which the covariates are highly correlated:

```
R> x1 = rnorm(20)
R> x2 = rnorm(20,mean=x1,sd=.01)
R> y = rnorm(20,mean=3+x1+x2)
```

In this simulation, the first covariate x_1 is simulated according to a centered Gaussian of unit variance, whereas the second ones x_2 is centered in x_1 with standard deviation equal to .01. With such a simulation scenario, the inputs are highly correlated and thus the model is approximately equivalent to $y_i \simeq 3 + 2x_{i1} + \epsilon_i \simeq 3 + 2x_{i2} + \epsilon_i$. Consequently, parameters β_1 and β_2 can take any large values until they approximately sum to 2. Figure 1 illustrates this phenomenon by plotting the estimated coefficients for 50 simulations.

High number of inputs. In some situations, the number p of predictors x_j can be very large. In such situation, the model interpretation is difficult due to the large number of parameters to interpret, and we often would like to determine a smaller subset that exhibit the strongest effects. Additionally, when p is larger than n, ordinary least square has no unique solution and the regression model is useless.

On way to solve these problems is to select a small number of uncorrelated inputs, using for instance forward or stepwise variable selection algorithms (see **??**). In this chapter, two other alternatives to ordinary least square are presented: the first one define new uncorrelated variables as combination of the initial ones (methods using derived input directions) and the second ones estimate the regression coefficients by penalized least square (shrinkage methods).

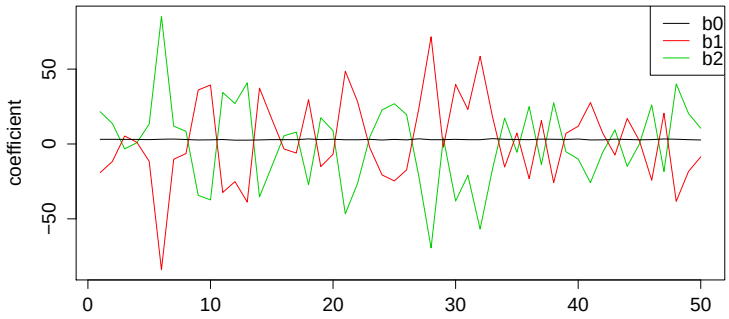

Fig. 1. Values of the estimated OLS coefficients (true values $\beta_0 = 3$, $\beta_1 = 1$ and $\beta_2 = 1$) when inputs are correlated)

2 Methods using Derived Input Directions

2.1 Regression on Principal Components

The idea is to regress the output y on the principal components $z_1, \ldots, z_p$ resulting from a principal component analysis (PCA) of the observation matrix $\boldsymbol{X}$.

Principal component analysis consists in defining new uncorrelated variables (principal components), linear combination of the initial ones, explaining a maximum of the information contained in the initial dataset $\boldsymbol{X}$. If the initial variables have been standardized (centered and with unit variance), the first principal component, maximizing the variance of the variables after projection, is defined as the eigenvector of $\frac{1}{n}\boldsymbol{X}'\boldsymbol{X}$ associated to the larger eigenvalue. The second principal component is defined as maximizing the variance after projection under the constraint to be orthogonal to the first principal component: it is the eigenvector of $\frac{1}{n}\boldsymbol{X}'\boldsymbol{X}$ associated to the second larger eigenvalue. And so on.

Choosing to regress y on a number M ($M < p$) of principal components leads to a reduced regression model, build on a reduced number M of uncorrelated principal components.

PCA and SVD To the factor $\frac{1}{n}$, performing PCA consists in diagonalizing $\boldsymbol{X}'\boldsymbol{X}$:

$$\boldsymbol{X}'\boldsymbol{X} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}'$$

where $\boldsymbol{V}$ is the orthogonal matrix of eigenvectors and $\boldsymbol{D}$ the diagonal matrix of eigenvalues.
The Singular Value Decomposition (SVD) relies on decomposing $\boldsymbol{X}'\boldsymbol{X}$ as follows:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}'$$

where $\boldsymbol{U}$ is the orthogonal matrix of (left-)singular vectors, $\boldsymbol{S}$ the diagonal matrix of singular values and $\boldsymbol{V}$ the orthogonal matrix of eigenvectors (or right-singular vectors). With this decomposition we have (thanks to the orthogonality of $\boldsymbol{U}$)

$$\boldsymbol{X}'\boldsymbol{X} = \boldsymbol{V}\boldsymbol{S}\boldsymbol{U}'\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}' = \boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}'.$$

The square root of the eigenvalues of $\boldsymbol{X}'\boldsymbol{X}$ are then the singular values of $\boldsymbol{X}$. Note finally that the left-singular vector are the eigenvectors of $\boldsymbol{X}\boldsymbol{X}'$.

2.2 Partial Least Square regression

Partial Least Square regression constructs also new variables as linear combinations of the covariates, but unlike PCA it uses Y for this construction. We assume that the covariates have zero mean and unit variance.

PLS begin by computing the first PLS direction Z_1 as a linear combination of the x_j where the coefficient are the covariance of the x_j's with y. Then, y is regressed on Z_1 and then the inputs x_j are orthogonalized with respect to Z_1. The process is continued until M directions have been obtained. As for regression on PCA components, if $M = p$ the OLS regression is obtained, whereas if $M < p$ produces a reduced regression.

2.3 Application on astronomy data

2.3.1 The data

The illustrative data come from the NYU Value-Added Galaxy Catalog (NYU-VAGC) which is a cross-matched collection of galaxy catalogs maintained for the study of galaxy formation and evolution (`http://sdss.physics.nyu.edu/vagc/`, Blanton et Hogg (2005)). Two subsets of 100 galaxies each were extracted. Each galaxy is described by 49 parameters, observables or derived quantities such as the stella formation rate (sfr) or the specific stellar formation rate (specsfr). In this chapter we propose to explain the star formation rate (*sfr*, variable number 43) in function of the following covariates:

var. n°	names	description
2	vdisp	estimated velocity dispersion from spectrum
3	uabs	u absolute magnitude (log of intensity)
4	Jabs	J absolute magnitude
5	sersicampr	The best fit to the variable "A" in r(nanomaggies/arcsec∘2): describes the radial distribution of light
6	sersicnr	The best fit to the Sersic index "n" in r
7	sigmabalmer	Velocity dispersion (sigma not FWHM)measured simultaneously in all of the Balmer lines in kms
8	sigmaforb	Velocity dispersion (sigma not FWHM)measured simultaneously in all the forbidden lines in kms
9-23		equivalent width of absorption or emission lines in the spectrum
24-41		indices (mostly Likc) that are medium band measurements in the spectrum
42	d4000n	The break in the spectrum at 4000 Angstroem
45	sersicr0r	The best fit to the variable "r_0" in r (arcsec)
46	sersicr50r	light radius of best fit model in r (arcsec) encompassing 50% of the total luminosity
47	sersicr90r	light radius of best fit model in r (arcsec) encompassing 90% of the total luminosity
48	umz	magnitude u minus magnitude z (u-z)
49	JmH	J-H
50	HmK	H-K

The data are loaded with the following **R** command:

```
R> data=read.table('vagc.txt',header=TRUE)
R> data=na.omit(data)
R> y=as.numeric(data[,43])
R> x=data[,c(2:42,45:50)]
R> dat=data.frame(y,x)
```

The function `na.omit()` deletes the observation with missing data.

2.3.2 The **R** package

The `pls` package for **R** proposed Partial Least Squares and Principal Component regression.

2.3.3 The analysis

Figure 2 plots the observations and the initial variables in the first principal plan. We can see for instance that variables `uabs` is positively correlated with the second axis whereas variables `oiii4363seqw` and `sersicr0r` are negatively correlated with the first axis. Principal component regression is performed with the following **R** command:

```
R> model_pcr=pcr(y∼.,data=dat,scale=TRUE,validation='CV')
```

The option `scale=TRUE` allows to standardized the variables and `validation='CV'`

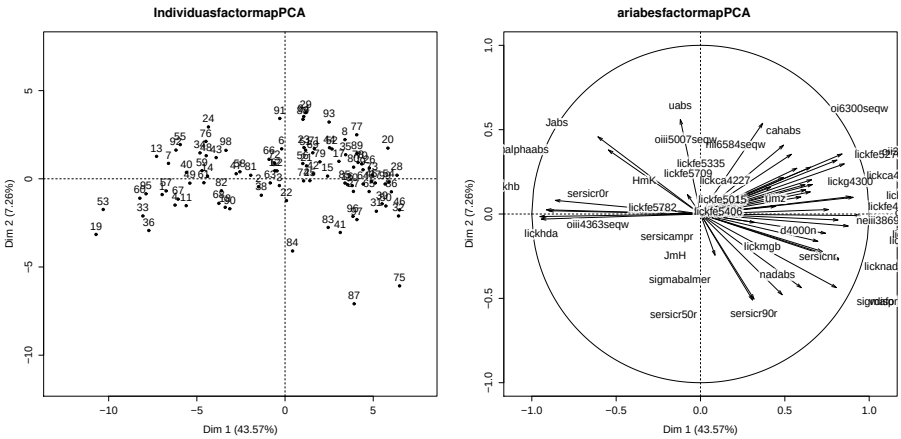

Fig. 2. PCA of the VAGC data

estimates the cross-validation root mean square error (RMSEP) for different number of components. The following command plots the RMSEP in function of the number of components (Figure 3):

```
R> plot(RMSEP(model_pcr), legendpos = "topright")
```

It appears that about 5 components are sufficient to lead to an interesting prediction accuracy.

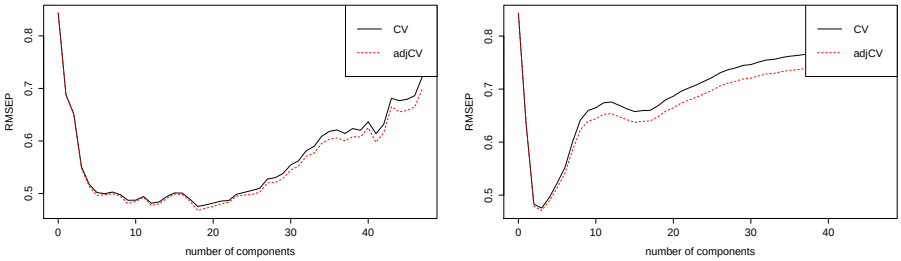

Fig. 3. Cross-validation RMSEP for PCR and PLS

Partial least square regression is then performed with the following **R** command:

```
R> model_pls=plsr(y∼.,data=dat,scale=TRUE,validation='CV')
R> plot(RMSEP(model_pls), legendpos = "topright")
```

For PLS, 2 components seems to be sufficient to provide a regression model with the same performance than PCR with 5 components.
Figure 4 plots the predicted values versus observed ones for PCR and PLS.

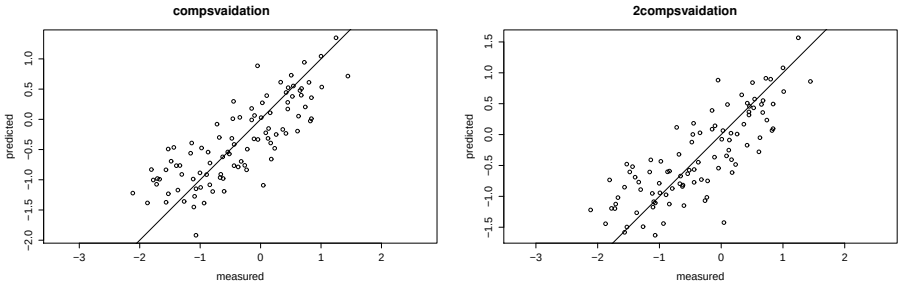

Fig. 4. Predicted versus observed values for PCR and PLS

In order to compare both regression models, they are used in prediction on a new independent dataset:

```
R> datanew=read.table('vagc2.txt',header=TRUE)
R> ynew=as.numeric(datanew[,43])
R> xnew=as.matrix(datanew[,c(2:42,45:50)])
R> datnew=data.frame(ynew,xnew)
R> model_pcr=pcr(y~.,data=dat,scale=TRUE,ncomp=5)
R> ynew_pcr=predict(model_pcr,newdata=data.frame(xnew))
R> print(sqrt(mean((ynew_pcr-ynew)^2)))
0.5530745
R> model_pls=plsr(y~.,data=dat,scale=TRUE,ncomp=2)
R> ynew_pls=predict(model_pls,newdata=data.frame(xnew))
R> print(sqrt(mean((ynew_pls-ynew)^2)))
0.5329118
```

The performance on this test dataset is approximately equivalent for both methods, with a slight advantage for PLS regression (RMSEP of 0.5329118 for PCR and 0.5530745 for PLS).

3 Shrinkage methods

Shrinkage methods consists of estimating the linear regression model (1.1) by penalized least square, imposing some constraints on the regression coefficients.

3.1 Ridge regression

The first shrinkage method, introduced in the 1970's, is ridge regression, which imposed a L_2 penalty on the regression coefficients (Hoerl & Kennard (1970)):

$$\begin{aligned}\hat{\boldsymbol{\beta}}^{ridge} &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{\sum_{i=1}^{n}(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij})^2+\lambda\sum_{j=1}^{p}\beta_j^2\}\\ &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{\sum_{i=1}^{n}(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij})^2\} \text{ subject to } \sum_{j=1}^{p}\beta_j^2 < t_\lambda\end{aligned}$$

where λ is a shrinkage coefficient. The size constraint on the regression coefficients avoid to obtain artificially high values for the regression coefficients related to correlated covariates, by shrinking the coefficients toward 0. Larger is λ, greater is the shrinkage and then the regularization of the model.

This penalized least square estimation leads to biased estimation, contrary to OLS coefficients, but with lower variances. This regularization of the model generally leads to better prediction accuracy. Let note also that when covariates are orthonormal (case in which it is preferable to use OLS), the values of the coefficients are $\hat{\boldsymbol{\beta}}^{ridge} = \hat{\boldsymbol{\beta}}^{ols}/(1+\lambda)$.

Ridge solutions are not equivariant to predictors scaling, and then the predictors have usually to be standardized before the analysis. Moreover, no penalty is imposed on the intercept β_0. Consequently, the ridge coefficients are estimated as follows:

1. $\hat{\beta}_0 = \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$,
2. x_{ij} are replaced by $\frac{x_{ij}-\bar{x}_j}{s_j}$ (standardization),
3. $\hat{\boldsymbol{\beta}}^{ridge} = (\hat{\beta}_1^{ridge},\ldots,\hat{\beta}_p^{ridge})'$ is now obtained by

$$\hat{\boldsymbol{\beta}}^{ridge} = (\boldsymbol{X}'\boldsymbol{X}+\lambda I_p)^{-1}\boldsymbol{X}'\boldsymbol{y}.$$

We have seen in the Introduction that, when inputs are correlated, the matrix $\boldsymbol{X}'\boldsymbol{X}$ is ill-conditioned, and the estimated coefficients $\hat{\boldsymbol{\beta}}^{ols}$ are unstable due to their high variances $V(\hat{\boldsymbol{\beta}}^{ols}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$. Adding a positive term λ on the diagonal of $\boldsymbol{X}'\boldsymbol{X}$ make it well-conditioned, and then leads to a regularized solution.

The shrinkage coefficient λ must be chosen, and several techniques allows to choose the best shrinkage coefficient in a given list $\{\lambda_1,\ldots,\lambda_K\}$:

- Leave One Out Cross-Validation (LOOCV): choose λ_k minimizing

$$\text{PRESS}_k = \sum_{i=1}^{n}(y_i-\hat{y}_{\lambda_k}^{-i})^2,$$

where $\hat{y}_{\lambda_k}^{-i}$ is the prediction of y_i by ridge regression (with λ_k as shrinkage coefficient) obtained without using the ith observation.

- Since LOOCV is computationally expensive (the regression model has to be estimated n times), Generalized Cross-Validation (GCV) produces an approximation of the PRESS coefficient:

$$\text{PRESS}_k \simeq GCV_k = \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}^i_{\lambda_k}}{1 - tr(H)/n} \right)^2,$$

with $H = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X} + \lambda I_p)^{-1}\boldsymbol{X}'$ the *hat* matrix of ridge regression.

On the other side, Hoerl & Kennard (1970) give an estimation of the optimal shrinkage parameter $\lambda_{HKB} = \frac{p\hat{\sigma}^2}{\hat{\boldsymbol{\beta}}^{ols\prime}\hat{\boldsymbol{\beta}}^{ols}}$ as well as Lawless & Wang (1976): $\lambda_{LW} = \frac{p\hat{\sigma}^2}{\sum_{j=1}^{p} a_i \hat{\beta}_j^{ols2}}$ with a_i the ith eigenvalue of $\boldsymbol{X}'\boldsymbol{X}$.

3.2 Lasso regression

Lasso regression considers a L_1 penalty on the regression coefficients (Tibshirani (1996)):

$$\begin{aligned}
\hat{\boldsymbol{\beta}}^{lasso} &= \underset{\boldsymbol{\beta}}{\text{argmin}} \{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \} \\
&= \underset{\boldsymbol{\beta}}{\text{argmin}} \{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 \} \text{ subject to } \sum_{j=1}^{p} |\beta_j| < t
\end{aligned}$$

Using L_1 norm rather than L_2 ones makes the problem non linear in y_i, and then there does not exist closed form for the solution. Nevertheless the problem is convex and a quadratic optimization program can be used to solve it. Figure 5 presents the contours of the least squares error function (red ellipses) and the constraint regions (blue areas). The lasso/ridge solutions are the points of the ellipses which hit the blue areas. Unlike the L_2 constraint area, the L_1 one has corner, and the elliptical contours have many chance to hit the area on a corner. If the solution is at a corner, then one parameter is null. When the dimension p increases, the L_1 area has many corners and there are many opportunities for the lasso solution to have regression coefficients equal to zero. This leads to a selection of a subset of variables.

Let notice that when covariates are orthonormal, the lasso coefficients are equal to $\hat{\boldsymbol{\beta}}^{lasso} = sgn(\hat{\boldsymbol{\beta}}^{ols})(|\hat{\boldsymbol{\beta}}^{ols}| - \lambda/2)_+$.

Ridge and lasso solutions can be interpreted in a Bayesian paradigm: ridge regression is equivalent to the following regression model:

$$\begin{aligned}
y_i|\mathbf{x}_i &\sim \mathcal{N}(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}, \sigma^2) \\
\beta_j|\sigma^2 &\sim \mathcal{N}(0, \gamma^2),
\end{aligned}$$

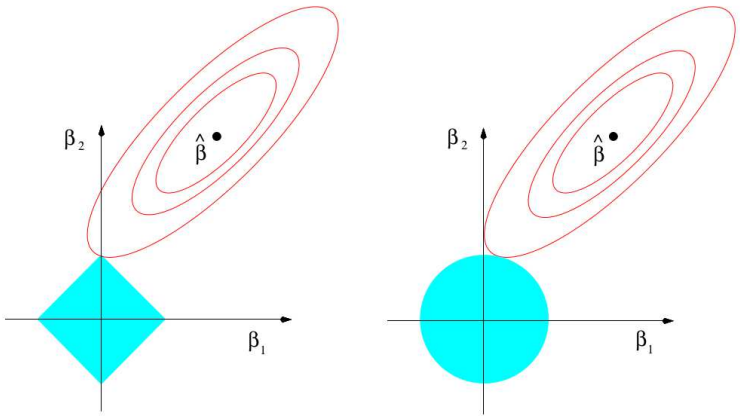

Fig. 5. Lasso versus ridge penalties (source Hastie et al. (2009)).

whereas lasso regression is equivalent to:

$$\begin{aligned} y_i|\mathbf{x}_i &\sim \mathcal{N}(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}, \sigma^2) \\ \beta_j|\sigma^2 &\sim Laplace. \end{aligned}$$

3.2.1 Least Angle Regression

Least Angle Regression (LAR) is a variable selection algorithm which performs a kind of continuous variables selection (Efron et al. (2004)):

1. standardized predictors (mean 0 and variance 1)
   Start with $\mathbf{r} = \mathbf{y} - \bar{\mathbf{y}}$ and $\beta_1 = \ldots = \beta_p = 0$,
2. Find the $\mathbf{x}_j$ most correlated with $\mathbf{r}$,
3. Move β_j from 0 to β_j^{ols} (reg. of $\mathbf{y}$ on $\mathbf{x}_j$) until some some $\mathbf{x}_k$ has better correlation with the current residual $\mathbf{r}$,
4. Move β_j and β_k to their joint OLS values (reg. of $\mathbf{y}$ on $\mathbf{x}_j$ and $\mathbf{x}_k$) until $\mathbf{x}_l$ has better correlation with the current residual $\mathbf{r}$,
5. Continue until all predictors have been entered.

This algorithm can be adapted to compute the entire path of lasso solutions as λ is varied, with the same computational cost as for ridge regression. For this, the following step is inserted between the fourth and fifth ones:

4a. If any β_j hits 0, drops its variable from the active set.

As for ridge regression, the shrinkage coefficient λ can be chosen by cross-validation, but unlike ridge regression, no closed-form exists for generalized cross-validation. Model selection criteria as Mallows C_p can also be used

$$C_p = \frac{SSR}{(\hat{\sigma}^{ols})^2} + 2d - n$$

with SSR the residual sum of square and d the number of active predictors. The model to select is the one leading to the C_p the most closest to $d+1$ (and lower than $d+1$).

3.3 Elastic-net regression

Elastic Net mixes the ridge and lasso penalties in order to combine the advantages of both methods (Zhou & Hastie (2006)): sparsity with the L_1 penalty and regularization with the L_2 penalty:

$$\hat{\boldsymbol{\beta}}^{EN} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}\alpha\beta_j^2 + (1-\alpha)|\beta_j|\}$$

The parameter α, which controls the mixing between L_1 and L_2 penalties, can be selected by cross-validation as for the shrinkage coefficient λ.

3.4 Application on astronomy data

Shrinkage methods are applied on the dataset presented in Section 2.3

3.4.1 The **R** packages

- Ridge:
  - function `lm.ridge` of the package `MASS`.
- Lasso:
  - function `glmnet` of the package `glmnet`,
  - function `lars` of the package `lars`,
  - function `HDlars` of the package `HDPenReg`* (R-forge).
- Elastic-net:
  - function `glmnet` of the package `glmnet`,
  - function `cv.glmnet` computes cross-validation MSE for a grid of values for λ.

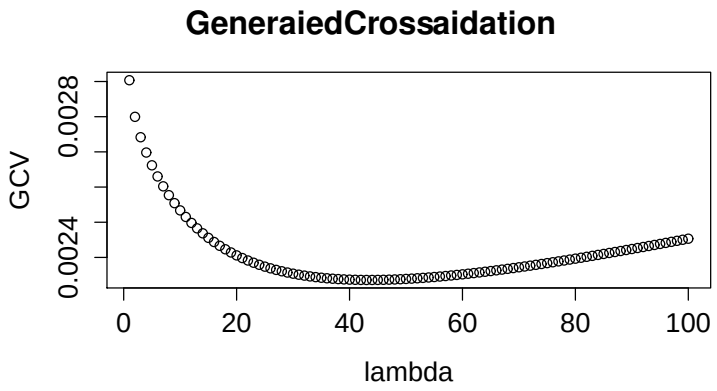

Fig. 6. Values of the generalized cross validation criterion in function of λ.

3.4.2 The analysis

We start by considering ridge regression with $\lambda = 1 : 100$:

```
R> model <- lm.ridge(y~.,data=dat,lambda=1:100)
```

Figure 6 plots the generalized cross validation criterion for each value of λ. This criterion seems to be minimized in $\lambda = 43$.

The LAR algorithm is then applied in order to estimate the lasso model for any values of λ:

```
R> model=lars(as.matrix(x),y,type="lasso",trace=TRUE,normalize=TRUE)
LASSO sequence
Computing X'X .....
LARS Step 1:  Variable 41 added
LARS Step 2:  Variable 2 added
...
LARS Step 35:  Variable 31 added
Lasso Step 36:  Variable 7 dropped
Lasso Step 37:  Variable 29 dropped
LARS Step 38:  Variable 25 added
Lasso Step 39:  Variable 9 dropped
LARS Step 40:  Variable 11 added
LARS Step 41:  Variable 13 added
LARS Step 42:  Variable 36 added
Lasso Step 43:  Variable 17 dropped
LARS Step 44:  Variable 29 added
...
LARS Step 54:  Variable 17 added
LARS Step 55:  Variable 44 added
```

```
Computing residuals, RSS etc .....
```

Figure 7 plots the values of lasso regression coefficients in function of the number of active variables (Df).

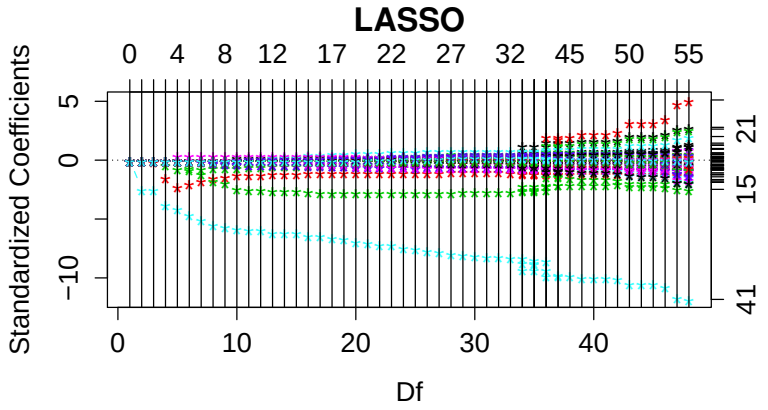

Fig. 7. Values of lasso regression coefficients in function of the number of active variables (Df).

Figure 8 plots the values of the RSS (residual sum of square) and C_p criteria in function of the number of active variables. As expected, the RSS always decreases when the number of actives variables increases, whereas the C_p tends to select a shrinkage coefficient λ corresponding to a lasso model with 8 active variables. The regression coefficients for these 8 variables are:

```
R> object1$beta[9,object1$beta[9,]!=0]
 uabs          -0.145737772  cahabs     -0.022519262
 Jabs          -0.187254744  d4000n     -1.979278908
 oiii4363seqw  -0.025279438  sersicr0r  0.0220510031
 nii6584seqw   -0.009438988  JmH        0.018713420
```

Predicting the stellar formation rate requires eight observables. Among these, broad band photometry is required, with uabs which tends to reflect young stars, and Jabs and JmH that are respectively related to the total mass of the galaxy and to its infrared color. Four observables are related to atomic lines, OIII, Ca, NII and D4000, globally indicators of the state of the gas. The last observable is related to morphology. These eight parameters are thus not unexpected to be indicators of sfr and clearly they are all needed to get a good prediction. The other parameters must thus be considered either non-informative or redundant, which is not entirely surprising.

Elastic-net is now applied, fixing the L_1/L_2 mixing proportion α to .5 (it has been to estimate by cross-validation in practice):

```
R> model=cv.glmnet(as.matrix(x),y,alpha=0.5)
```

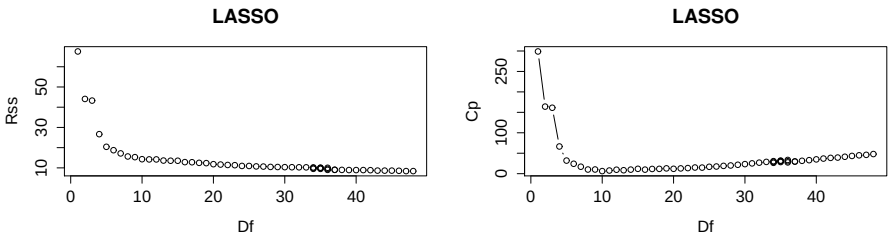

Fig. 8. Values of the RSS (residual sum of square) and C_p criteria in function of the number of active variables (Df).

Figure 9 plots the values of the estimated MSE (mean square error) criterion (and confidence bounds) in function of the logarithm of the shrinkage parameter λ.

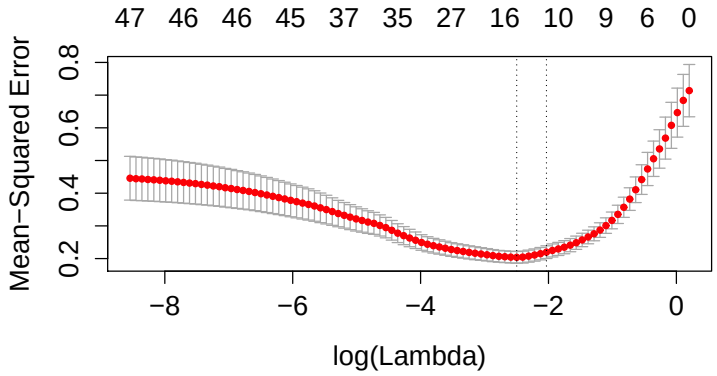

Fig. 9. Values of the MSE (mean square error) criterion in function of the logarithm of λ.

Using the optimal value of λ (0.0823), the estimated coefficients of the elastic-net model are given below:

```
R> model_en=glmnet(as.matrix(x),y,alpha=0.5,lambda=model$lambda.min)
R> model_en$beta
 uabs          -0.1502943963  lickcn2     -0.5632779741
 Jabs          -0.1848840457  lickg4300   -0.0008900572
 oiii4363seqw  -0.0357001056  lickca4455  -0.0027187415
 nii6584seqw   -0.0147642763  lickc4668   -0.0043912528
 hdeltaabs      0.0162437194  d4000n      -1.3868108517
 hbetaabs       0.0731784848  sersicr0r    0.0403275927
 cahabs        -0.0405834357  JmH          0.1387162965
```

Here 14 variables are selected, including the 8 found with LASSO. The supplementary ones are hdeltaabs and hbetaabs, the latter being considered as an indicator of the average stellar age, and several lick indices which are generally difficult to relate directly to stellar formation processes. However, one should not be fooled by physics at this stage, since the variables implied in a good prediction of the parameter sfr are selected from a pure statistical point of vue, and only the quality of this prediction prevails.

Finally, in order to compare these regression model from a prediction point of view, they are used in prediction on a the test dataset, and the RMSEP is computed:

- Linear regression
```
R> model_lm=lm(y~.,data=dat)
R> ynew_lm=predict(model_lm,newdata=data.frame(xnew))
R> print(sqrt(mean((ynew_lm-ynew)^2)))
0.556675
```
- Ridge *(no `predict` function for `lm.ridge`)*
```
R> ynew_ridge= scale(xnew, center = model_ridge$xm, scale
= model_ridge$scales) %*% model_ridge$coef[, which.min(model_ridge$GCV)]
+ model_ridge$ym
R> print(sqrt(mean((ynew_ridge-ynew)^2)))
0.4687964
```
- Lasso
```
R> ynew_lasso=predict(model_lasso,newx=xnew,s=10,mode="step")
R> print(sqrt(mean((ynew_lasso$fit-ynew)^2)))
0.4189243
```
- Elastic-net
```
R> ynew_en=predict(model_en,newx=xnew)
R> print(sqrt(mean((ynew_en-ynew)^2)))
0.4359127
```

Lasso regression is the methods which leads to the best prediction (better than those obtained by PCR or PLS). From an interpretation point of view, lasso is also the most interesting method since it selects a small subset of significant predictors (8 over 47).

3.5 Extensions

Other penalized regression methods have been proposed, among which fused and group lasso can be interesting in Astronomy.

Fused lasso encourages the regression coefficients β_j to be sparse and smooth in j (Tibshirani et al. (2005)), what means that the estimated value for β_j will be close to those of β_{j-1} and β_{j+1}:

$$\hat{\boldsymbol{\beta}}^{FL} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda_1 \sum_{j=1}^{p}|\beta_j| + \lambda_2 \sum_{j=1}^{p-1}|\beta_{j+1} - \beta_j|\}.$$

Such model is interesting when the order of the variables is significant, as for instance in genetics where for instance the variables correspond to a physical position on a chromosome.
The function `EMfusedlasso` of the **R** package `HDPenReg` implements the fused lasso.

When groups of inputs are known *a priori*, Yuan & Lin (2007) define the group lasso as follows:

$$\hat{\boldsymbol{\beta}}^{GL} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{||(y - \beta_0 \mathbb{1} - \sum_{\ell=1}^{L} \boldsymbol{X}_\ell \beta_\ell)||^2 + \lambda \sum_{\ell=1}^{L} \sqrt{p_\ell}||\beta_\ell||_2\}.$$

Such penalty encourages sparsity at the group level. The function `grplasso` of the **R** package `grplasso` implements this model.
When a group structure is assumed but not known, Yengo et al. (2013) propose the clere regression model which simultaneously performs variable clustering and regression.

References

Blanton, M.R. and Hogg, D.W. *New York University Value-Added Galaxy Catalog: A Galaxy Catalog Based on New Public Surveys.* The Astronomical Journal, 129, 2562-2578, 2005.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. *Least Angle Regression*, Annals of Statistics, 35(6), 2358–2364, 2004.

Hastie, T., Tibshirani, R. and Friedman F. *The Elements of Statistical Learning. Data Mining, Inference and Prediction.* Second Edition, 2009, Springer.

Hoerl, A.E. and Kennard, R. *Ridge regression: biaised estimation for nonorthogonal problems.* Technometrics, 12, 55–67, 1970.

Lawless, J.F. and Wang, P. *A simulation study of ridge and other regression estimators*, Communications in Statistics, Theory and Methods, 14, 1589–1604, 1976.

Tibshirani, R. *Regression shrinkage and selection via the lasso.* Journal of the Royal Statistical Society, Series B, 58, 267–288, 1996.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. and Knight, K. *Sparsity and smoothness via the fused lasso.* Journal of the Royal Statistical Society, Series B, 67, 91–108, 2005.

Yengo, L., Jacques, J. and Biernacki, C. *Variable clustering in high dimensional linear regression.* Journal de la Société Française de Statistique, in press, 2014.

Yuan, M. and Lin, Y. *Model selection and estimation in regression with grouped variables.* Journal of the Royal Statistical Society, Series B, 68(1), 49–67, 2007.

Zhou, H. and Hastie, T. *Regularization and variable selection via the elastic net.* Journal of the Royal Statistical Society, Series B, 101, 1418–1429, 2006.