Regularization and variable selection via the Elastic Net

Hui Zou and Trevor Hastie (2005)

Report by Miguel Biron Lattes

Abstract

The purpose of this report is to give a thorough assessment of the Elastic Net (EN) method. We start by describing some of its properties in detail. Next, we evaluate two R implementations of the EN in for a problem whose analytical solution is known. Then, we show how it performs against alternative approaches using both simulated and real datasets, which will help demonstrate the advantages as well as the pitfalls of the algorithm. Finally, we offer a critical review of the paper based on our findings.

1 Description of the method

Consider the standard linear regression setting:

$$y = X\beta + \epsilon \tag{1}$$

such that $y \in \mathbb{R}^n$ is the response vector, $X \in \mathbb{R}^{n \times p}$ is matrix of covariates, $\beta \in \mathbb{R}^p$ is an unknown parameter, and ϵ has n iid components such that $\mathbb{E}[\epsilon_i] = 0$. In the case when $n > p$ and $\text{rank}(X) = p$, there exists a unique ordinary least squares (OLS) solution to Equation 1:

$$\hat{\beta}^{OLS} = \arg\min_{\beta} \|y - X\beta\|_2^2 = (X^T X)^{-1} X^T y \tag{2}$$

where $\|\cdot\|_2$ is the L2 norm. On the other hand, the OLS is not unique when $n < p$ or when $\text{rank}(X) < p$ (and thus it can't be obtained by 2). But even in the case where it is unique, it might not be a satisfactory solution. There are two main reasons [1]:

- **Prediction accuracy**: the OLS solution is known to have low bias but large variance. Recall the decomposition of the mean squared error (MSE) as the sum of the variance of an estimator and its bias squared. By changing the estimator in a way that the increase in squared bias were compensated by a larger decrease in variance, we would obtain lower prediction MSE.

- **Interpretation**: $\hat{\beta}^{OLS}$ can have an arbitrary number of non-zero components. When p is large, it becomes difficult to extract meaningful information or patterns from this estimator.

Variable subset selection tackles the above problem by iteratively obtaining a collection of OLS estimates by selectively eliminating some of the columns of X, and then choosing a particular estimate based on a given criteria or stopping rule. One classical method of this kind is the best-subset selection procedure. In its most naive form, we calculate all the 2^p OLS estimators associated with each possible configuration for the p predictors; i.e, either in or out of the model (the case where all variables are out of the model is simply $\hat{\beta}^{OLS} = 0$). The best of such models is then selected using a model selection criterion, like the Akaike Information Criterion (AIC).

Although there are slightly more efficient ways of carrying out best subset selection than computing all possible models, it still becomes unfeasible for even modest p. A classical alternative to best subset selection is called forward stepwise selection. It starts with the model containing only the intercept, and then at each stage it decides which variable to add to the model by evaluating all these augmented models and selecting one using again a criterion like the AIC. This procedure can accommodate high-dimensional settings, but because of its greedy nature, it tends to select sub-optimal models.

The variable selection problem has also been described in the Bayesian literature, with the most notable example being the "spike-and-slab" model for sparseness, first introduced in [2], later improved by [3] and [4]. Basically, in the spike-and-slab framework the prior distribution of every β_k has a point mass at 0 (the spike), and is very flat elsewhere (the slab). This approach leads to posterior distributions that put high probability on sparse solutions, and by their Bayesian nature, offer a direct way of doing inference on the model parameters (e.g., obtaining standard errors) accounting by the implied search process. Alas, its high computational cost means that it does not scale well to large datasets.

More recently, a broader class of procedures have been proposed by generalizing and extending the least squares criterion [5]:

$$\hat{\beta}_\theta = \arg\min_\beta \|y - X\beta\|_2^2 + P_\theta(\beta) \tag{3}$$

where θ is a parameter vector that requires tuning, and $P_\theta(\beta)$, known as the penalization, is increasing in the absolute value of every component of β. Because of this property, if the least squares term of 3 was deleted, then the solution to the problem would just become $\beta = 0$. Therefore, the intuition behind 3 is that we look for β that achieves a trade-off between agreeing with the data, while at the same time having only a few non-zero components. Because we want $P_\theta(\cdot)$ to apply the same way to any component of β, regardless of the different magnitudes of the predictors, we generally assume that X is standardized:

$$\forall i \in \{1...p\} : 1_n^T x_i = 0, x_i^T x_i = 1$$

where $1_n \in \mathbb{R}^n$ is a vector of ones. We will further assume the response is centered ($1_n^T y = 0$) in order to avoid having an intercept term (which shouldn't be penalized).

One of the most successful examples of an estimator of this form is the Lasso [6]:

$$\hat{\beta}(\lambda) = \arg\min_{\beta} L(\lambda, \beta) = \arg\min_{\beta} \|y - X\beta\|_2^2 + \lambda\|\beta\|_1 \tag{4}$$

where $\|z\|_1 = \sum_{i=1}^{p} |z_i|$ is the L1 norm, and $\lambda \geq 0$. Clearly, when $\lambda = 0$, we recover the OLS estimator. However, for $\lambda > 0$ the Lasso yields interesting properties. Indeed, in this case, the Lasso has a parsimony property, such that the solution to 4 only has a handful of non-zero coefficients [7]. The number of such components decreases as λ increases, eventually setting $\beta = 0$. This shrinking of the coefficients towards 0, besides helping with model interpretability, often improves prediction accuracy, by achieving a positive trade-off between bias and variance.

In spite of the above, the Lasso has some undesirable properties. First, when $n < p$, the Lasso can deliver at most n non-zero coefficients [8]. This does not occur with other penalized estimators, like Ridge regression. We will investigate this behavior further in Section 3. Second, the Lasso estimator is not guaranteed to be unique in all situations [8]. Third, the Lasso tends to introduce too much bias (towards 0) in coefficients where the true parameter is large [5]. Fourth, the tuning process of the λ parameter (usually cross-validation) tends to deliver unstable solutions [9]. Finally, it has been empirically shown that the Lasso underperforms in setups where the true parameter has many small but non-zero components [10]. In order to address some of these situations, [11] proposed the Elastic Net (EN) estimator[1]:

$$\hat{\beta}(\lambda_1, \lambda_2) = \arg\min_{\beta} L(\lambda_1, \lambda_2, \beta) = \arg\min_{\beta} \|y - X\beta\|_2^2 + \lambda_2\|\beta\|_2^2 + \lambda_1\|\beta\|_1 \tag{5}$$

with $\lambda_1, \lambda_2 \geq 0$. It is clear that when $\lambda_2 = 0$ we recover the Lasso, and that when $\lambda_1 = 0$ we obtain the Ridge estimator. In fact, viewed in terms of Equation 3, the EN penalty $P_{\lambda_1,\lambda_2}(\beta)$ can be understood as a scaled weighted average of the L2 and L1 norms:

$$P_{\lambda_1,\lambda_2}(\beta) = (\lambda_1 + \lambda_2)\left(\frac{\lambda_2}{\lambda_1 + \lambda_2}\|\beta\|_2^2 + \frac{\lambda_1}{\lambda_1 + \lambda_2}\|\beta\|_1\right) \tag{6}$$

Therefore, one could theorize that, with appropriate choices of (λ_1, λ_2), the EN can achieve a balance between the good properties of the Lasso and the Ridge estimator, while ameliorating their inconveniences. Indeed, the authors argue that the EN does not have the limit of n non-zero coefficients when $n < p$ (we will offer a proof of this in Section 3). Additionally, the authors state that the EN improves over the Lasso in another way in the situation $n < p$. Indeed, in presence of highly correlated predictors, the Lasso will usually choose one of them and drop the other, which translates into unstable solutions. In contrast, the authors show that the EN will select both, assigning to them similar coefficients.

[1] The authors actually call this the naive elastic net. We will drop this distinction as it has been deprecated in the literature.

The rest of the report is structured in such a way that we first go over the theoretical details of the EN, discussing their relevance, and then move on to test this procedure both in simulated and real datasets, against alternatives approaches.

2 On Lemma 1 and its relevance

2.1 Proof of Lemma 1

Lemma (Lemma 1 in [11]). *Given data set (y, X), and (λ_1, λ_2), define an artificial data set (y^*, X^*) by:*

$$X^* = \frac{1}{\sqrt{1+\lambda_2}} \begin{pmatrix} X \\ \sqrt{\lambda_2} I_p \end{pmatrix}, \qquad y^* = \begin{pmatrix} y \\ 0_p \end{pmatrix}$$

where I_p is the identity matrix of size p, and $0_p \in \mathbb{R}^p$ is a vector of zeroes. Let $\gamma = \lambda_1/\sqrt{1+\lambda_2}$, and $\beta^* = \sqrt{1+\lambda_2}\beta$. Then, the EN criterion can be written as a Lasso problem with modified data:

$$L(\gamma, \beta^*) = \|y^* - X^*\beta\|_2^2 + \gamma\|\beta^*\|_1$$

Moreover, let:

$$\hat{\beta}^* = \arg\min_{\beta^*} L(\gamma, \beta^*)$$

Then, the EN solution can be obtained as:

$$\hat{\beta} = \frac{1}{\sqrt{1+\lambda_2}}\hat{\beta}^*$$

Proof. Starting from 5:

$$\begin{aligned} L(\lambda_1, \lambda_2, \beta) &= \|y - X\beta\|_2^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1 \\ &= y^T y - 2y^T X\beta + \beta^T X^T X\beta + \lambda_2\beta^T\beta + \lambda_1\|\beta\|_1 \\ &= y^T y - 2y^T X\beta + \beta^T(X^T X + \lambda_2 I_p)\beta + \lambda_1\|\beta\|_1 \end{aligned}$$

Now, consider the matrix of size $(n+p) \times p$:

$$\tilde{X} = \begin{pmatrix} X \\ \sqrt{\lambda_2} I_p \end{pmatrix}$$

We can verify that

$$\tilde{X}^T\tilde{X} = X^TX + \lambda_2 I_p$$

However, $\tilde{X}$ is not standardized, since $\forall j \in \{1...p\}$:

$$\tilde{x_j}^T\tilde{x_j} = \underbrace{x_j^T x_j}_{1} + \lambda_2$$

Thus, we now consider the standardized version of $\tilde{X}$:

$$X^* = \frac{1}{\sqrt{1+\lambda_2}}\tilde{X} = \frac{1}{\sqrt{1+\lambda_2}}\begin{pmatrix} X \\ \sqrt{\lambda_2} I_p \end{pmatrix}$$

Likewise, consider the vector of length $n+p$ defined by $y^* = (y, 0_p)^T$. Note that, since y is centered, y^* is also trivially centered. We can verify that:

$$y^{*T}y^* = y^Ty, \qquad\qquad y^{*T}X^* = \frac{1}{\sqrt{1+\lambda_2}}y^TX$$

With these definitions, we can rewrite $L(\lambda_1, \lambda_2, \beta)$:

$$\begin{aligned} L(\lambda_1, \lambda_2, \beta) &= y^Ty - 2y^TX\beta + \beta^T(X^TX + \lambda_2 I_p)\beta + \lambda_1\|\beta\|_1 \\ &= y^{*T}y^* - 2\sqrt{1+\lambda_2}y^{*T}X^*\beta + (1+\lambda_2)\beta X^{*T}X^*\beta + \frac{\sqrt{1+\lambda_2}}{\sqrt{1+\lambda_2}}\lambda_1\|\beta\|_1 \\ &= y^{*T}y^* - 2y^{*T}X^*(\sqrt{1+\lambda_2}\beta) + (\sqrt{1+\lambda_2}\beta)X^{*T}X^*(\sqrt{1+\lambda_2}\beta) + \frac{\lambda_1}{\sqrt{1+\lambda_2}}\|\sqrt{1+\lambda_2}\beta\|_1 \end{aligned}$$

Defining:

$$\beta^* = \sqrt{1+\lambda_2}\beta, \qquad\qquad \gamma = \frac{\lambda_1}{\sqrt{1+\lambda_2}}$$

We arrive to the following expression:

$$\begin{aligned} L(\lambda_1, \lambda_2, \beta) = L(\gamma(\lambda_1, \lambda_2), \beta^*(\beta)) &= y^{*T}y^* - 2y^{*T}X^*\beta^* + \beta^{*T}X^{*T}X^*\beta^* + \frac{\lambda_1}{\sqrt{1+\lambda_2}}\|\beta^*\|_1 \\ &= \|y^* - X^*\beta^*\|_2^2 + \gamma\|\beta^*\|_1 \end{aligned}$$

which we identify as the Lasso criterion. Therefore, if $\hat{\beta}^*$ solves the Lasso problem for the transformed variables, i.e.:

$$\hat{\beta}^* = \arg\min_{\beta^*} L(\gamma, \beta^*)$$

then

$$\hat{\beta} = \frac{1}{\sqrt{1+\lambda_2}}\hat{\beta}^*$$

is a solution for 5. □

2.2 Relevance of Lemma 1

Lemma 1 is paramount to the method proposed by the authors, because it provides a straightforward way to implement a solver for 5. Indeed, since we can pose the EN as a Lasso problem on transformed data, then any method for obtaining solutions to the Lasso will be also a valid method for obtaining EN solutions.

In particular, the authors propose a modified version of the Least Angle Regression with the Lasso modification (LARS) [7]. LARS is an efficient way of computing Lasso paths $\lambda \mapsto \hat{\beta}(\lambda)$, for λ between 0 (OLS estimate) and the smallest value that makes all the coefficients equal to 0, with the computational complexity of a single OLS estimate.

Now, the authors correctly point out that naively applying LARS to the transformed data would be inefficient. Recall that the size of X^* is $(n+p) \times p$, with most of the bottom of the matrix (the $\sqrt{\lambda_2}I_p$ part) being sparse. Of course, the problem becomes even worse when $p \gg n$. A similar thing happens with y^*, which is just y padded with 0's at the bottom.

Because of the above reasons, the authors propose a modified LARS algorithm to solve the Elastic Net problem, which they name LARS-EN. These modifications take advantage of the sparsity of the transformed the data.

2.3 Relationship between the solutions of the original and transformed problems

Given the association between the original and transformed problems, one could ask:

Is the solution to the transformed EN problem $\hat{\beta}^*$ a Lasso solution for the original problem?

We will show that in general, and especially in the interesting case $\lambda_2 > 0$, this is not true, by focusing in the case where $n > p$ and the design matrix X is orthonormal; i.e., when $X^T X = I_p$. We start by showing the exact solution for the Lasso in this particular case.

Proposition 1. *Let (y, X) such that $X^T X = I_p$, and $\lambda > 0$. Also, assume that $n > p$. Then, the solution to 4 is given by:*

$$\hat{\beta}_k = \text{sgn}(\hat{\beta}_k^{OLS}) \left(|\hat{\beta}_k^{OLS}| - \frac{\lambda}{2} \right)_+, \quad \forall k = 1...p \tag{7}$$

where $(a)_+ = \max\{0, a\}$. *Moreover, this solution is unique.*

Proof. Expanding the Lasso criterion:

$$\begin{aligned} L(\lambda, \beta) &= \|y - X\beta\|_2^2 + \lambda\|\beta\|_1 \\ &= y^T y - 2y^T X\beta + \beta^T \underbrace{X^T X}_{I_p} \beta + \lambda\|\beta\|_1 \\ &= y^T y - 2y^T X\beta + \beta^T \beta + \lambda\|\beta\|_1 \end{aligned}$$

We know that, since $X^T X = I_p$, an unique OLS solution exists:

$$\hat{\beta}^{OLS} = (X^T X)^{-1} X^T y = X^T y$$

Putting this back in the above expression:

$$\begin{aligned} L(\lambda, \beta) &= y^T y - 2\hat{\beta^{OLS}}^T \beta + \beta^T \beta + \lambda\|\beta\|_1 \\ &= y^T y + \sum_{k=1}^{p} \underbrace{(-2\hat{\beta}_k^{OLS}\beta_k + \beta_k^2 + \lambda|\beta_k|)}_{f(\hat{\beta}_k^{OLS}, \beta_k, \lambda)} \\ &= \text{const.} + \sum_{k=1}^{p} f(\hat{\beta}_k^{OLS}, \beta_k, \lambda) \end{aligned}$$

This means that the criterion is a sum of univariate functions, which can be optimized separately in order to get the optimal vector $\hat{\beta}$. In other words:

$$\arg\min_{\beta} L(\lambda, \beta) = \begin{pmatrix} \arg\min_{\beta_1} f(\hat{\beta}_1^{OLS}, \beta_1, \lambda) \\ \vdots \\ \arg\min_{\beta_p} f(\hat{\beta}_p^{OLS}, \beta_p, \lambda) \end{pmatrix}$$

Pick $k \in \{1...p\}$, and let us focus on:

$$\min_{\beta_k} f(\hat{\beta}_k^{OLS}, \beta_k, \lambda) = -2\hat{\beta}_k^{OLS}\beta_k + \beta_k^2 + \lambda|\beta_k|$$

The first thing we notice is that $\text{sgn}(\hat{\beta}_k) = \text{sgn}(\hat{\beta}_k^{OLS})$, because the first term is negative (and thus lower) when this happens, while the second and last terms are not affected by the sign. Next, we examine the three possible cases:

Case 1: Assume $\hat{\beta}_k^{OLS} > 0$. Then

$$f(\hat{\beta}_k^{OLS}, \beta_k, \lambda) = -2\hat{\beta}_k^{OLS}\beta_k + \beta_k^2 + \lambda\beta_k$$

Since this is a strictly convex function of β_k, the first order condition is sufficient to yield the global minimizer:

$$\frac{df}{d\beta_k} = 0 = -2\hat{\beta}_k^{OLS} + 2\beta_k + \lambda \Rightarrow \beta_k = \hat{\beta}_k^{OLS} - \frac{\lambda}{2}$$

However, we need to enforce the equality of the signs. Therefore:

$$\hat{\beta}_k = \left(\hat{\beta}_k^{OLS} - \frac{\lambda}{2}\right)_+$$

Case 2: Assume $\hat{\beta}_k^{OLS} < 0$. Now, the first order condition becomes (f is again strictly convex):

$$\frac{df}{d\beta_k} = 0 = -2\hat{\beta}_k^{OLS} + 2\beta_k - \lambda \Rightarrow \beta_k = \hat{\beta}_k^{OLS} + \frac{\lambda}{2}$$

Again, we must enforce the sign equivalence:

$$\hat{\beta}_k = -\left(-\hat{\beta}_k^{OLS} - \frac{\lambda}{2}\right)_+$$

Case 3: Assume $\hat{\beta}_k^{OLS} = 0$. Then $f(0, \beta_k, \lambda) = \beta_k^2 + \lambda|\beta_k|$ is minimized at $\hat{\beta}_k = 0$.

Finally, assuming that the sign function can be defined as:

$$\text{sgn}(z) = \begin{cases} 1 & , z > 0 \\ -1 & , z < 0 \\ 0 & , z = 0 \end{cases}$$

then, we can consolidate the three cases into one formula:

$$\hat{\beta}_k = \text{sgn}(\hat{\beta}_k^{OLS})\left(|\hat{\beta}_k^{OLS}| - \frac{\lambda}{2}\right)_+$$

The uniqueness is given by the fact that the OLS solution is unique in this case.

□

Going back to the question, we know that the unique Lasso estimator for the original variables is given by Proposition 1. We will now find an expression for $\hat{\beta}^*$, which we can then compare to the above

and assess their equivalence. Indeed, as Lemma 2.1 shows, $\hat{\beta}^*$ is the solution to a Lasso problem with transformed data (y^*, X^*) and parameter γ.

Note that, if $X^T X = I_p$, then:

$$X^{*T} X^* = \frac{1}{1+\lambda_2}(\underbrace{X^T X}_{I_p} + \lambda_2 I_p) = \frac{1+\lambda_2}{1+\lambda_2} I_p = I_p$$

Hence, X^* is also orthonormal. Thus, we can also apply Proposition 1 to obtain the solution. Indeed, for all $k = 1...p$:

$$\hat{\beta}^*_k = \text{sgn}(\hat{\beta^*}^{OLS}_k) \left(|\hat{\beta^*}^{OLS}_k| - \frac{\gamma}{2} \right)_+$$

Now,

$$\hat{\beta^*}^{OLS}_k = X^{*T} y^*$$

However,

$$X^{*T} y^* = \frac{1}{\sqrt{1+\lambda_2}} X^T y = \frac{1}{\sqrt{1+\lambda_2}} \hat{\beta}^{OLS}$$
$$\gamma = \frac{\lambda_1}{\sqrt{1+\lambda_2}}$$

Therefore, $\hat{\beta^*}^{OLS} = \frac{1}{\sqrt{1+\lambda_2}} \hat{\beta}^{OLS}$, and:

$$\hat{\beta}^*_k = \frac{1}{\sqrt{1+\lambda_2}} \text{sgn}(\hat{\beta}^{OLS}_k) \left(|\hat{\beta}^{OLS}_k| - \frac{\lambda_1}{2} \right)_+ \tag{8}$$

We can now ascertain that, for all $\lambda_2 > 0$, $\hat{\beta}^* \neq \hat{\beta}$, so $\hat{\beta}^*$ is not the unique Lasso solution. In fact, $\hat{\beta}^*$ is a shrunken version of the Lasso estimate (all coefficients are divided by $\sqrt{1+\lambda_2}$). Of course, when $\lambda_2 = 0$ both are equal, because the EN becomes equivalent to the Lasso.

3 Behavior of the Elastic Net and the Lasso when $p > n$

In what follows, we will give a formal proof that, in the case $p > n$, the Lasso is incapable of delivering solutions with more than n non-zero coefficients. However, there is a simpler way of showing this.

Indeed, as Theorem 1 in [7] shows, the LARS algorithm with Lasso modification is capable of delivering the entire path of Lasso solutions, provided the "one-at-a-time" condition holds. This condition means that at each iteration of LARS, no more than 1 variable enters or leaves the active set.

Therefore, since we know that when $p > n$ all versions of LARS stop at an iteration with at most n variables in the active set, we know this will also apply when we use LARS for the transformed data.

However, because the above won't necessarily apply when the one-at-a-time condition doesn't hold (the authors of LARS specifically state that their algorithm is not designed to handle that case), we now give a more general proof.

Proposition 2. *Suppose the design matrix X is of size $n \times p$, with $p > n$. Then, the Lasso can select at most n variables.*

Proof. Consider the Lasso criterion in Equation 4. Since this is the sum of a differentiable convex function ($\|y - X\beta\|_2^2$) and a subdifferentiable convex function ($\|\beta\|_1$), the result is a subdifferentiable convex function [12], whose subdifferential set is given by:

$$\partial_\beta L(\lambda, \beta) = \left\{ -2X^T y + 2X^T X\beta + \lambda\eta : \eta \in \mathbb{R}^p, \eta_k \in \begin{cases} \{\text{sgn}(\beta_k)\} & , \beta_k \neq 0 \\ [-1, 1] & , \beta_k = 0 \end{cases} \right\}$$

Suppose we have a solution $\hat{\beta}$ for 4, and let us consider the active set $A = \{k \in \mathbb{N} : \hat{\beta}_k \neq 0\}$. For the components of $\hat{\beta}$ indexed by A, the subdifferential contains only one element, which means that the stationarity Karush-Kuhn-Tucker (KKT) conditions [13] become:

$$0 \in \partial_\beta L(\lambda, \hat{\beta}_A) \Leftrightarrow -2X_A^T y_A + 2X_A^T X_A \hat{\beta}_A + \lambda\eta_A = 0$$

Rearranging the terms, we obtain a linear system of equations that $\hat{\beta}_A$ must satisfy:

$$X_A^T X_A \hat{\beta}_A = X_A^T y_A - \frac{1}{2}\lambda\eta_A$$

Note that this is in fact a linear system, because η_A does not depend on $\hat{\beta}_A$, but only on A. Indeed, the signs of the non-zero elements $\hat{\beta}_A$ must be equal to the signs of $X_A^T y_A$. To see this, recall that

$$L(\lambda, \beta) = y^T y - 2y^T X\beta + \beta^T X^T X\beta + \lambda\|\beta\|_1$$

We can see that the only term that is affected by the signs of $\hat{\beta}_A$ is $-2y^T X\beta$. If we fix the magnitudes of $\hat{\beta}_A$, then we can always make $L(\lambda, \beta)$ lower by aligning the signs of $\hat{\beta}_A$ and $X_A^T y_A$.

On the other hand, notice that:

$$\text{rank}(X_A^T X_A) \leq \text{rank}(X^T X) = \text{rank}(X) \leq \min\{n, p\} = n$$

By the rank-nullity theorem, we know that

$$\dim(\ker(X_A^T X_A)) = |A| - \text{rank}(X_A^T X_A) \geq |A| - n$$

where $|A|$ is the size of A. Now, let us proceed by contradiction and assume that $|A| > n$. Then, $\dim(\ker(X_A^T X_A)) \geq 1$. This means that we know there is at least one vector $v \neq 0$ in the basis of the subspace $\ker(X_A^T X_A)$. Therefore, there are infinite vectors satisfying the linear system above, which can be described by:

$$\hat{\beta}_A(t) = \hat{\beta}_A + tv$$

where $t \in \mathbb{R}$. Moreover, because of the assumption that A is the active set, we must have that $\forall t \in \mathbb{R}$, all the components of $\hat{\beta}_A(t)$ are non-zero. However, we can always find at least one t such that at least one component of $\hat{\beta}_A(v^*)$ becomes exactly zero. To do this, we just need to pick any non-zero component of v, and then choose t such that the corresponding component in $\hat{\beta}_A(t)$ becomes 0. Thus, we have arrived to a contradiction, and hence, $|A| \leq n$.

□

Now, it is relatively easy to show why the above argument does not apply to the EN. Indeed, the EN criterion from Equation 5 is again the sum of 2 differentiable convex functions and a subdifferentiable convex function, so its subdifferential exists and is equal to:

$$\partial_\beta L(\lambda_1, \lambda_2, \beta) = \left\{ -2X^T y + 2(X^T X + \lambda_2 I_p)\beta + \lambda_1 \eta : \eta \in \mathbb{R}^p, \eta_k \in \begin{cases} \{\text{sgn}(\beta_k)\} & , \ \beta_k \neq 0 \\ [-1, 1] & , \ \beta_k = 0 \end{cases} \right\}$$

Again, let us assume that we have a solution $\hat{\beta}$ for the above, and let A be the active set. The KKT condition becomes:

$$0 \in \partial_\beta L(\lambda_1, \lambda_2, \hat{\beta}_A) \Leftrightarrow -2X_A^T y_A + 2(X_A^T X_A + \lambda_2 I_{|A|})\hat{\beta}_A + \lambda_1 \eta_A = 0$$

which translates to the following linear system of equations:

$$(X_A^T X_A + \lambda_2 I_{|A|})\hat{\beta}_A = X_A^T y_A - \frac{1}{2}\lambda_1 \eta_A$$

We will show that $\text{rank}(X_A^T X_A + \lambda_2 I_{|A|}) = |A|$. We know that, since $X_A^T X_A$ is a Gram matrix, it is symmetric and positive semi-definite. Thus, its real eigenvalues $\{\epsilon_i\}_{i=1...|A|}$ are non-negative. Let

$\{e_i\}_{i=1...|A|}$ be their associated eigenvectors. Then

$$\begin{aligned} & X_A^T X_A e_i = \epsilon_i e_i \\ \Leftrightarrow & X_A^T X_A e_i + \lambda_2 e_i = \epsilon_i e_i + \lambda_2 e_i \\ \Leftrightarrow & (X_A^T X_A + \lambda_2 I_{|A|}) e_i = (\epsilon_i + \lambda_2) e_i \end{aligned}$$

We see that the eigenvalues of $(X_A^T X_A + \lambda_2 I_{|A|})$ are $\{\epsilon_i + \lambda_2\}_{i=1...|A|}$. Therefore, since $\lambda_2 > 0$, we have that these eigenvalues are all positive. Hence, the matrix is non-singular, which means that the solution to the linear system above is unique for every $|A| \leq p$.

4 R implementations of the Elastic Net

The LARS-EN algorithm mentioned in 2.2 was implemented by the authors in the R package `elasticnet`. However, since the original paper came out, there have been proposed many other alternative ways to solve 5. The R package `glmnet` [14] is possibly the most popular of these.

For a standardized X and arbitrary y, `glmnet` solves an equivalent EN formulation:

$$\min_{\beta_0, \beta} \frac{1}{2n} \|y - \beta_0 1_n - X\beta\|_2^2 + \lambda \left((1-\alpha) \frac{1}{2} \|\beta\|_2^2 + \alpha \|\beta\|_1 \right) \tag{9}$$

Note that this form of the EN uses the intuition of the weighted average of penalties from Equation 6. In order to compare it with Equation 5, let us further assume that y is centered, so that $\beta_0 = 0$. Then, by multiplying 9 by $2n$, we get:

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

which is just 5 with $\lambda_1 = 2n\lambda\alpha$ and $\lambda_2 = n\lambda(1-\alpha)$. We can invert these equations to obtain a mapping $(\lambda_1, \lambda_2) \mapsto (\alpha, \lambda)$:

$$\alpha(\lambda_1, \lambda_2) = \frac{\lambda_1}{\lambda_1 + 2\lambda_2} \qquad\qquad \lambda(\lambda_1, \lambda_2) = \frac{\lambda_1 + 2\lambda_2}{2n} \tag{10}$$

Hence, using 10, we will be able to obtain solutions to 5 for arbitrary (λ_1, λ_2) with package `glmnet`.

4.1 Checking implementations with an orthonormal design matrix

Because of the existence of an exact analytical solution, the case of an orthonormal X is specially useful to check the validity of EN algorithmic implementations. Indeed, Equation 8 gives us the solution for

the transformed data. By inverting the identity $\beta^* = \sqrt{1+\lambda_2}\beta$, we can derive the exact solution of the EN for an orthonormal design matrix:

$$\hat{\beta}_k = \frac{1}{1+\lambda_2} \text{sgn}(\hat{\beta}_k^{OLS}) \left(|\hat{\beta}_k^{OLS}| - \frac{\lambda_1}{2} \right)_+ \tag{11}$$

With this in mind, we draw a matrix X of size $n = 1000$ and $p = 20$ from iid standard normals. To get an orthogonal design matrix, we get the $X = QR$ decomposition of X, and then replace X by Q. Finally, we standardize this orthogonal X by imposing $1_n^T x_j = 0$ and $x_j^T x_j = 1$ for all $j = 1...p$, which means that X becomes orthonormal (i.e, $X^T X = I_p$).

Likewise, we draw a sparse random β vector of size p with iid standard normal components at just 3 random locations, and the rest of the components being exactly 0. With this, we can get the response vector $y = X\beta$. We have purposefully avoided adding noise to the response, because we are interested in assessing the algorithmic implementations and want to avoid additional sources of variation. Also, note that $1_n^T y = (1_n^T X)\beta = 0_p^T \beta = 0$, so y is centered as required.

Now, for `glmnet` to give the adequate solutions, we must set the options `standardize` and `intercept` to `FALSE`. Because of this, we also need to standardize y so that $y^T y/n = 1$ (note the extra $1/n$ term with respect to the standardization of X). Therefore, we obtain $\text{sd}_y = \sqrt{y^T y/n}$, and then divide both β and y by this quantity to achieve the desired standardization, while ensuring the compatibility of y with $y = X\beta$. Note that this change in y does not affect the original formulation of the EN, so that 11 is still valid. Finally, in order for `elasticnet` to give the desired output for an EN fit, we must divide the `beta.pure` matrix that is produced by $1 + \lambda_2$ (which means that `beta.pure` actually gives the Lasso fit for the transformed data)[2].

Figure 1 shows the results of solving the EN using 11, `glmnet` and `elasticnet`, for 4 distinct values of λ_2. For the first two algorithms, we show paths of length 15 from $\lambda_1 = 0$ to $\lambda_1 = 2\max_k |\hat{\beta}_k^{OLS}|$, which is the minimum λ_1 that sets $\hat{\beta} = 0$. In the case of `elasticnet`, because of how LARS works, only the values of λ_1 at which a variable leaves the active set are available. Note that only the paths for the non-zero components are shown. The horizontal dotted lines represent the true β values.

Thus, we can see that using all the modifications mentioned in the preceding paragraphs, we are able to obtain the true paths (i.e, the ones given by the analytical formula) both with `glmnet` and `elasticnet`.

[2]For `elasticnet`, it is not necessary to turn off `normalize` and `intercept`, since the data is already in the format it requires, and thus those options won't change anything.

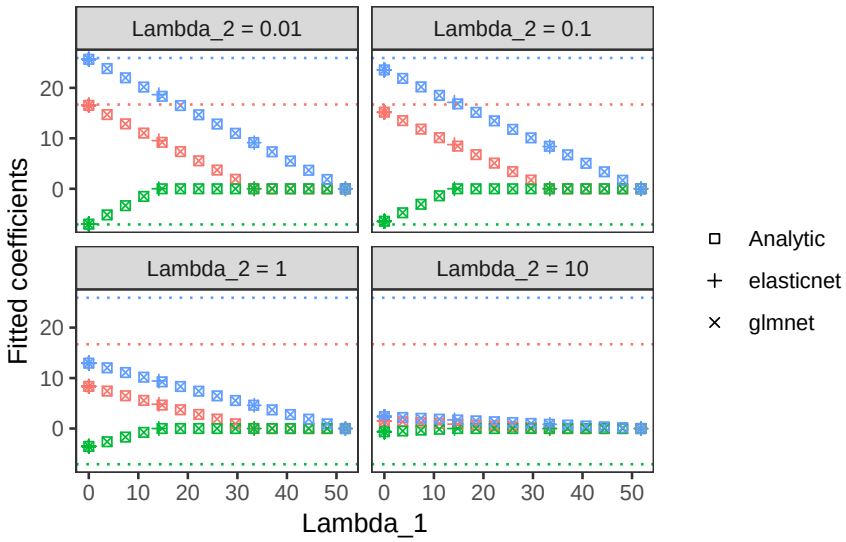


Figure 1: Comparison between the EN paths for an orthonormal design matrix X of size $1,000 \times 20$, and a β with just 3 non-zero components (only the paths for these are shown). The paths were generated using the analytical formula, the `elasticnet` and the `glmnet` packages. The λ_1 grid is of length 15 and goes from 0 to the minimum value that makes $\hat{\beta} = 0$. We used 4 different values for λ_2 that roughly represent the whole range of shrinkage allowed. The horizontal dotted lines represent the true parameters' values.

5 Simulations

In this section, we perform a simulation analysis to assess the out-of-sample predictive performance of the EN. We compare it to the Lasso, and also to two other methods: the Minimiax Concave Penalty (MCP), and the Smoothly Clipped Absolute Deviation (SCAD) [15]. All of these algorithms can be written in the form of a penalized regression from Equation 3. It is known that, although the Lasso has good out-of-sample predictive performance, the shrinkage it induces causes large effects to be biased towards 0. MCP and SCAD propose penalization functions that for small effects behave like Lasso, but for larger effects they monotonically diminish the shrinkage, and hence they are non-convex.

Both MCP and SCAD are implemented in R in package `ncvreg`. They have the same tuning parameters (λ, γ). For each γ, `ncvreg` efficiently delivers complete paths for a sequence of λ, so it behaves much like `glmnet`. For MCP, it is required that $\gamma > 1$, while for SCAD we need $\gamma > 2$. Moreover, for both of them, we have that in the limit $\gamma \to \infty$, they converge to the Lasso.

Inspired by the simulations performed both in the original EN paper, and also the ones on [15], we propose two types of tasks:

Tasks type A: a slight generalization of task a) in [11]. We allow for arbitrary parameters n (sample size), σ (error std. dev.), p (number of coefficients), and a correlation parameter ρ such that $\text{corr}(x_i, x_j) =$

$\rho^{|i-j|}$. We can also turn this decay off. Additionally, we let $p_{nz} \in [0,1]$ the proportion of non-zero coefficients in β. We then choose $\lceil p_{nz} p \rceil$ components at random to be non-zero. Finally, we sample the non-zero coefficients iid from a Gamma(α, β) distribution, parameterized in terms of a mean μ_b and a standard deviation σ_b:

$$\alpha = \frac{\mu_b^2}{\sigma_b^2} \qquad\qquad \beta = \frac{\mu_b}{\sigma_b^2}$$

Note that tasks a) to c) in [11] can be almost exactly replicated using this set-up if we allow the decay in correlation to be optional, and the limit $\sigma_b \to 0$.

Tasks type B: these correspond to a modified task d) in the original paper. Recall that this experiment was designed to emphasizes the grouping effect. However, we can see that there is a problem with its definition. Let x_i and x_k two variables in group 1. By the definition of that task, we have:

$$x_{ij} = Z_1 + \epsilon_{ij} \qquad\qquad x_{kj} = Z_1 + \epsilon_{kj}$$

where $Z_1 \sim N(0,1)$ and $\epsilon_{ij}, \epsilon_{kj} \overset{iid}{\sim} N(0, \sigma_\epsilon^2)$. Now, we note that:

$$\begin{aligned}
\widehat{Cov}(x_i, x_k) &= \frac{1}{n}\sum_{j=1}^{n}(x_{ij} - \underbrace{\bar{x}_i}_{\approx Z_1})(x_{kj} - \underbrace{\bar{x}_k}_{\approx Z_1}) \\
&\approx \frac{1}{n}\sum_{j=1}^{n}(Z_1 + \epsilon_{ij} - Z_1)(Z_1 + \epsilon_{kj} - Z_1) \\
&= \frac{1}{n}\sum_{j=1}^{n}\epsilon_{ij}\epsilon_{kj} \\
&\approx Cov(\epsilon_{ij}, \epsilon_{kj}) = 0
\end{aligned}$$

The above is true for any $\sigma_\epsilon > 0$. The intuition behind is that , for small σ_ϵ, the task is actually creating groups of mildly disturbed intercepts. Because the algorithms center and scale all covariates, these become almost equal to 0. And for large σ_ϵ, they just become random white noise. Therefore, there actually is no grouping effect under this design.

Given the above, in order to introduce the grouping effect we redefine the variables as

$$\text{Group u: } x_{ij} = \sin(uj) + \epsilon_{ij}^u \tag{12}$$

for $u \in \{1, 2, 3\}$, and $\epsilon_{ij}^u \overset{iid}{\sim} N(0, \sigma_\epsilon^2)$. With this setup, we now have:

$$\begin{aligned}
\widehat{Cov}(x_i, x_k) &= \frac{1}{n}\sum_{j=1}^{n}(x_{ij} - \underbrace{\bar{x_i}}_{\approx 0})(x_{kj} - \underbrace{\bar{x_k}}_{\approx 0}) \\
&\approx \frac{1}{n}\sum_{j=1}^{n}(\sin(uj) + \epsilon_{ij}^u)(\sin(uj) + \epsilon_{kj}^u) \\
&= \frac{1}{n}\sum_{j=1}^{n}[\sin(uj)^2 + \underbrace{\epsilon_{ij}^u \sin(uj)}_{\approx 0} + \underbrace{\sin(uj)\epsilon_{kj}^u}_{\approx 0} + \underbrace{\epsilon_{ij}^u \epsilon_{kj}^u}_{\approx 0}] \\
&\approx \frac{1}{n}\sum_{j=1}^{n}\sin(uj)^2 \neq 0
\end{aligned}$$

Moreover, since $\sin(u_1 x)$ is orthogonal to $\sin(u_2 x)$ for any $u_1, u_2 \in \mathbb{N}$, we know that variables from different groups will be uncorrelated. Hence, with this new formulation, we obtain a true grouping effect.

Keeping in mind that we want to try to test the EN in the setups it was designed for, we will emphasize tasks with very sparse coefficients. Also, we will assess how increasing the correlation between predictors affects its performance. Lastly, we will investigate if the EN stands out in the situation where there are grouping effects.

Given the above, we designed a total of sixteen tasks based in modifications of the types described. In each of these tasks we fit the four methods using the same methodology of splitting the data in train/validate/test subgroups used in the original paper. For all tasks, we use a splitting of sizes 200/100/100. Note that this is very different from what the authors did in their simulations, where they assign almost all of the data to the test set and very little to both training and validation. We think that this can make tuning the parameters in the models more difficult, and therefore we adjusted it. Additionally, for tasks of type A, we set $p_{nz} = 0.1$, so as to ensure that there is quite enough sparsity in the parameter vector. Also, we set $\mu_b = 2$ and $\sigma_b = 0.5$. Finally, we allow the correlation to decay in all tasks of type A.

The sixteen tasks are composed of 8 type A tasks and 8 type B tasks, each obtained by taking all the combinations of:

- $p \in \{50, 300\}$. Note that when $p = 300$, since $n = 200$, we get a $p > n$ situation, which the authors did not include in their simulations.
- $\sigma \in \{1, 15\}$. The latter case was included in the tests of the original paper. We include the former because [15] show that MCP and SCAD work better when the effects are large, or alternatively, when the signal-to-noise ratio (SNR) is low.
- $\rho \in \{0.5, 0.9\}$. This settings only affect tasks of type A. The authors of the original paper only

tried $\rho = 0.5$.

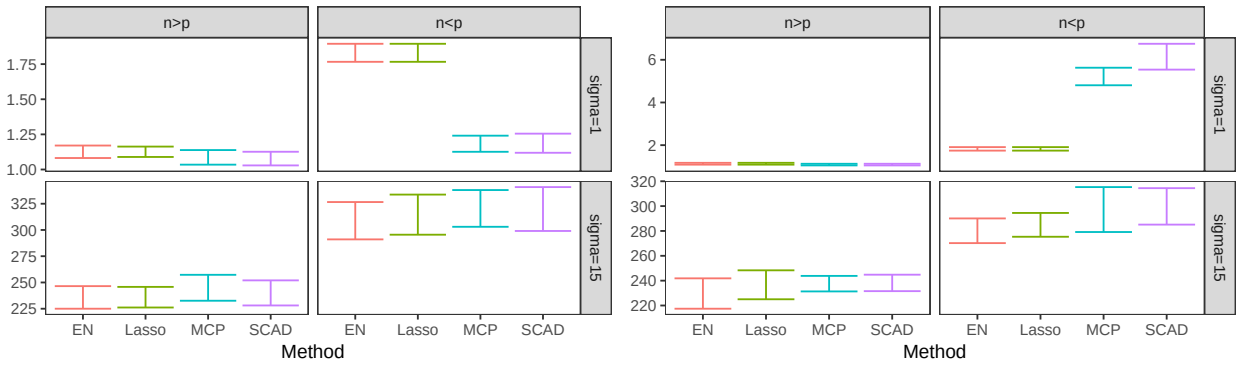
- $\sigma_\epsilon \in \{0.01, 1\}$: This setting only affects tasks of type B. The latter case is interesting because, from Equation 12, it is easy to check that as σ_ϵ grows, the noise dominates the deterministic effect, and therefore the variables become less correlated.

In all of the tasks, we use `glmnet` and `ncvreg`. We use their default settings to scan for values of λ. On the other hand, in the case of EN, we set a grid of size 10 for α with the following values:

$$\alpha \in \{0, 0.16, 0.32, 0.48, 0.64, 0.8, 0.9, 0.99, 0.999, 1\}$$

We chose the grid finer when close to 1 because [14] states that these are usually where the optimal value lies. Note that we also allow for the extreme cases where the EN becomes the Lasso and the Ridge regression. In particular, this means that EN should always show the same or better performance than the Lasso.

For MCP and SCAD we also used the default settings for the λ sequence. For the γ parameter, we set grids based on the most extreme values used in the tests presented in [15]. In particular, for MCP we used an equally spaced grid of size 10 between 1.2 and 20, while for SCAD we used a grid of the same size between 2.1 and 20.

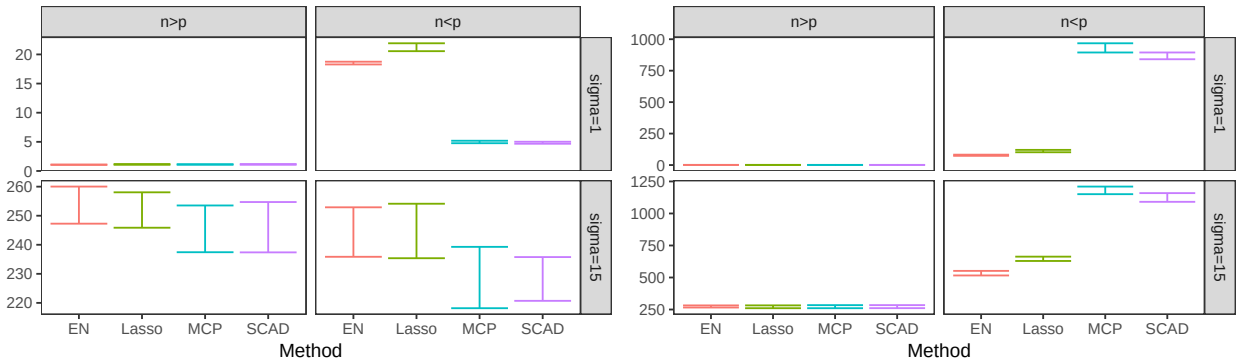


(a) $\rho = 0.5$ **(b)** $\rho = 0.9$

Figure 2: Comparison between the 4 methods under Tasks of type A. The performance metric is the median MSE in the test set for 100 independent realizations of each configuration. The bars represent the median ± 2 standard errors obtained by 500 bootstrap samples.

Figure 2 shows the results of the simulations for tasks of type A. The bars represent the median of the Mean Squared Error (MSE) in the test set across 100 replications of each task, ± 2 standard errors obtained using 500 bootstrap samples. We can see that all the methods produce very similar results, except for the case $n < p$ (i.e, when $p = 300$ and $n = 200$ for the training set). Indeed, in the case where $\rho = 0.5$, both MCP and SCAD obtain much better performances than EN and Lasso, which

behave similarly. This agrees with the results of the simulations in [15], which find that MCP and SCAD perform better than the Lasso with datasets of sparse but uncorrelated regressors. On the other hand, when ρ is increased to 0.9, the opposite is true. In fact, note that the scales have changed, and that MCP and SCAD incur in much more error than EN and Lasso had for $\rho = 0.5$. It is also noteworthy that, even though the bars overlap, the medians tend to be lower for EN and Lasso also in the case $n < p$ and $\sigma = 15$. Both of these facts seem to suggest that the EN is a good choice for $n < p$, especially when one assumes the existence of clusters of variables with high correlation.



(a) $\sigma_\epsilon = 0.01$ **(b)** $\sigma_\epsilon = 1$

Figure 3: Comparison between the 4 methods under Tasks of type B. The performance metric is the median MSE in the test set for 100 independent realizations of each configuration. The bars represent the median ± 2 standard errors obtained by 500 bootstrap samples.

The results for the tests under grouping (type B) are shown in Figure 3. We start by focusing on the case $\sigma_\epsilon = 0.01$. The resulting within-group correlation in this case is higher than 0.99. We note that there only are significant differences for the case $n < p$. Indeed, both MCP and SCAD outperform by a great margin EN and Lasso, both under the low and high noise setup. However, we do note that EN improves significantly over Lasso in the low noise case ($\sigma = 1$).

Now we turn our attention to the case $\sigma_\epsilon = 1$. Here, the within-group empirical correlation is lower and closer to 0.3. Again, the relevant differences appear only in the case $n < p$. We note the conclusions are now inverted: MCP and SCAD perform worse than EN and Lasso. Also, EN performs significantly better than Lasso in both the low and high noise settings.

The conclusions from experiments of type A and B seem to contradict each another. On the one hand, for tasks type A, we see that EN performs better when ρ is increased. Note that, because of the decay in correlation of task A, there is a kind of grouping effect, which is actually similar to what arises from an $AR(1)$ time series process: observations close in time are correlated (serial correlation). On the other hand, for tasks type B, we see that EN performs better when the within-group correlation is milder. This contradiction highlights the fact that the interplay between different factors of a task (the level of noise, the amount of predictors, the correlation and grouping of variables, etc.) yields complicated

interactions that result in non-intuitive performances for different methods. This is closely related to the results known as the "no-free-lunch" theorems in statistical learning [16]. Therefore, one should always apply and compare more than one method when making inferences about real data.

6 Analysis of a real dataset

We now turn our attention towards the assessment of the performance of the EN in a real dataset. The data corresponds to the "Gene Expression Cancer RNA-Seq Data Set" [17, 18]. It contains $n = 801$ measurements of $p = 20,531$ RNA-Seq gene expression levels. Clearly, this is a setting where $p \gg n$. Additionally, the dataset contains a response variable which is a label for each of the samples that categorizes the type of tumor that the individual presented: BRCA (breast carcinoma), KIRC (kidney renal clear cell carcinoma), COAD (colon adenocarcinoma), LUAD (lung adenocarcinoma) and PRAD (prostate adenocarcinoma).

The categorical nature of the response variable suggests that this data can be analyzed using classification models. Even though we have only studied the case of the EN for linear regression, the setup can be extended to handle generalized linear models (GLM) by the following modification of Equation 5:

$$L(\beta, \lambda_1, \lambda_2) = -2\mathcal{L}(X, y, \beta) + \lambda_2\|\beta\|_2^2 + \lambda_1\|\beta\|_1 \tag{13}$$

where $\mathcal{L}(X, y, \beta)$ is the log-likelihood of the model. The quantity $-2\mathcal{L}(X, y, \beta)$ is also known as the deviance.

Using this approach, the package `glmnet` supports both binomial and multinomial regressions, as well as other GLMs. On the other hand, the package `ncvreg` only supports binomial regression. Nevertheless, we can restrict our task to one of finding whether a patient had one particular type of tumor ($y_i = 1$) or not ($y_i = 0$). This becomes a binomial classification task, which we can now tackle using both packages. We chose the tumor type BRCA because it is the most common in the dataset. Recall that the binomial deviance can be simply stated as:

$$-2\mathcal{L}(X, y, \beta) = -2\left(\sum_{i:y_i=1} \log \pi(x_i^T\beta) + \sum_{i:y_i=0} \log(1 - \pi(x_i^T\beta))\right)$$

where π here is the logistic or sigmoid function.

We proceeded first by dividing the dataset in a training and a test set, with a proportion of 70% for the first. We did stratified sampling within the levels of the response variable to achieve better balance. Next, in the training set we defined 10 equally sized subsets or folds, in order to perform 10-fold cross-validation (CV) to tune the model parameters (again, we use stratified sampling to randomly assign the

individuals to each fold). Then, we use the built-in functionalities of each package to perform the CV process given the pre-specified folds. We used the same grids for α and γ from Section 5.

Table 1: Models obtained after tuning by 10-fold CV

Method	Lambda	Alpha / Gamma	Non-zero coefficients	CV mean deviance
EN	0.005	0.900	32	0.021
MCP	0.022	9.556	5	0.065
SCAD	0.022	6.078	14	0.066

The results of the parameter tuning process are summarized in Table 1. We excluded the Lasso in order to reduce the computational burden and because, with the grid being used, it is a particular case of the EN. Now, the first thing we notice is that the EN achieves the lowest CV mean deviance. Additionally, this method includes the most non-zero coefficients (more than twice than the one that follows, which is SCAD), which is consistent with the value of $\alpha = 0.9$. This value represents a slight departure from the pure Lasso estimator, which is itself consistent with what the authors from [14] note are the optimal values for α. In contrast, the MCP and SCAD select much less covariates, possibly due to their similarity to the Lasso with such large values for γ.

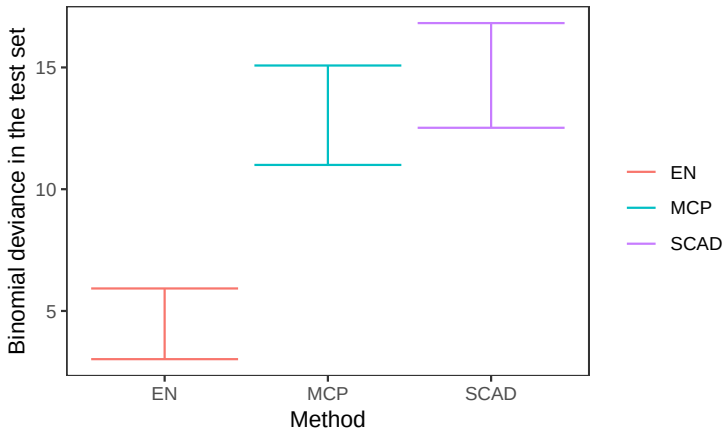


Figure 4: Binomial deviances for the test set achieved by the EN, MCP and SCAD methods, which were previously tuned using 10-fold CV. The center of the error bars are the deviances achieved by each method, while the endpoints correspond to these values $\pm$ 2 times an standard error estimated by 500 bootstrap replications of the test set.

Having obtained appropriately tuned models, we apply them to the test set to assess their out-of-sample performance. These results are summarized in Figure 4. Again, the center of the error bars are the actual deviances achieved by each method, while the endpoints correspond to these values $\pm$ 2 times an standard error estimated by 500 bootstrap replications of the test set. It is clear that this plot shows the same pattern observed in Table 1: EN achieves a significantly better performance than MCP and SCAD.

Again, the larger amount of predictors used by EN is probably playing an important role in achieving the best score, because of the intuition that averaging correlated predictors gives more robust out-of-sample predictions. This is consistent with the fact that SCAD, which uses 14 predictors, achieves slightly better results than MCP, which only uses 5.

7 Discussion

In this report we have attempted to do a thorough examination of [11], by situating their proposed method within the literature of variable selection, and verifying the author's claims that EN actually improves (both theoretically and experimentally) over alternatives, particularly the Lasso, but also others. Indeed, our own experiments have shown that not only the EN is at least as good as Lasso (this is trivially true because the Lasso is a special case), but it is actually significantly better in some setups. As the authors anticipated, these are the cases when $n < p$ and when there are grouping effects (recall the discussion of the results shown in Figure 3b).

In fact, after 13 years since its publication, the EN is still a highly popular method in many scientific fields, especially in genomics [19, 20, 21], where the problem of $p \gg n$ is commonplace. Given the success of the EN, it is a daunting task to try to find ways of improving it.

In spite of the above, we could focus on the paper itself, and ask ourselves if there are aspects that be upgraded. And, indeed, we actually showed that there was a problem with task d) in their simulation's section, because their setup did not achieve grouping effects, and therefore we had to introduce a modification to our simulations. Similarly, we objected the usage of small training and validation set sizes, compared to test set, because it introduced unnecessary variation in the parameter search process. Moreover, the test sets could be made smaller because we only used them to get an error estimate, and we replicated that estimated many times to obtain the median value. This is why we used the 200/100/100 rule, which is more similar to what is typically seen in the literature.

But in a broader sense, there is one important issue not addressed in the paper that should have been mentioned at least: the calculation of standard errors for the parameters. As it is known, simply obtaining the OLS standard errors for the selected model is wrong, because they do not account for the degrees of freedom lost in the search process [1]. Hence, lacking a theory for obtaining reasonable uncertainty estimates for a given variable selection method, the researcher is forced to use bootstrapping in order to get meaningful inferences on the coefficients of the selected model. But, carrying out hundreds of bootstrap replications on large datasets, each one consisting of a full CV tuning of parameters, can be prohibitive in terms of computation time.

The original paper for the Lasso actually proposed a cleverly simple strategy to obtain better standard errors, by interpreting the Lasso solution as the optimum of a Ridge regression problem with a particular anisotropic diagonal penalty matrix. Then, a closed form estimate of the matrix $Var(\hat{\beta})$ can be obtained.

However, this method is not entirely satisfactory, as it gives standard errors equal to 0 for coefficients not in the active set.

Still, the authors of [11] could have proposed an alternative similar to the above, or even assessed the appropriateness of directly applying it to the EN, with perhaps minimum modifications. Intuitively this should not be difficult, because the L2 penalty that the EN adds on top of the Lasso should play nicely with the Ridge regression interpretation.

Nevertheless, since the publication of [11], there have been considerable advances in the literature related to post-selection inference; that is, inference on the model parameters that takes into account the search process. One notable example is [22], which develops a general framework for exact post-selection inference, and actually gives the solution for the case of the EN. This will probably enable the adoption of the EN by more scientific communities, especially the ones where explanatory analyses are more relevant than predictive modeling [23], so that meaningful uncertainty estimates about the fitted coefficients are crucial.

References

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*, 2nd ed., ser. Springer series in statistics. Springer-Verlag New York, 2009.

[2] T. J. Mitchell and J. J. Beauchamp, "Bayesian variable selection in linear regression," *Journal of the American Statistical Association*, vol. 83, no. 404, pp. 1023–1032, 1988.

[3] E. I. George and R. E. McCulloch, "Variable selection via gibbs sampling," *Journal of the American Statistical Association*, vol. 88, no. 423, pp. 881–889, 1993.

[4] H. Ishwaran, J. S. Rao *et al.*, "Spike and slab variable selection: frequentist and bayesian strategies," *The Annals of Statistics*, vol. 33, no. 2, pp. 730–773, 2005.

[5] J. Fan and J. Lv, "A selective overview of variable selection in high dimensional feature space," *Statistica Sinica*, vol. 20, pp. 101–148, 2010.

[6] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[7] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani *et al.*, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[8] R. J. Tibshirani, "The Lasso Problem and Uniqueness," *ArXiv e-prints*, Jun. 2012.

[9] T. Park and G. Casella, "The bayesian lasso," *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 681–686, 2008.

[10] W. J. Fu, "Penalized regressions: the bridge versus the lasso," *Journal of computational and graphical statistics*, vol. 7, no. 3, pp. 397–416, 1998.

[11] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[12] R. T. Rockafellar, *Convex analysis*. Princeton university press, 1970.

[13] A. P. Ruszczyński, *Nonlinear optimization*. Princeton university press, 2006, vol. 13.

[14] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of statistical software*, vol. 33, no. 1, p. 1, 2010.

[15] P. Breheny and J. Huang, "Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection," *The annals of applied statistics*, vol. 5, no. 1, p. 232, 2011.

[16] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[17] J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. M. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, J. M. Stuart, C. G. A. R. Network *et al.*, "The cancer genome atlas pan-cancer analysis project," *Nature genetics*, vol. 45, no. 10, p. 1113, 2013.

[18] D. Dua and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[19] E. E. Schadt, C. Molony, E. Chudin, K. Hao, X. Yang, P. Y. Lum, A. Kasarskis, B. Zhang, S. Wang, C. Suver *et al.*, "Mapping the genetic architecture of gene expression in human liver," *PLoS biology*, vol. 6, no. 5, p. e107, 2008.

[20] J. Barretina, G. Caponigro, N. Stransky, K. Venkatesan, A. A. Margolin, S. Kim, C. J. Wilson, J. Lehár, G. V. Kryukov, D. Sonkin *et al.*, "The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity," *Nature*, vol. 483, no. 7391, p. 603, 2012.

[21] M. J. Garnett, E. J. Edelman, S. J. Heidorn, C. D. Greenman, A. Dastur, K. W. Lau, P. Greninger, I. R. Thompson, X. Luo, J. Soares *et al.*, "Systematic identification of genomic markers of drug sensitivity in cancer cells," *Nature*, vol. 483, no. 7391, p. 570, 2012.

[22] J. D. Lee, D. L. Sun, Y. Sun, J. E. Taylor *et al.*, "Exact post-selection inference, with application to the lasso," *The Annals of Statistics*, vol. 44, no. 3, pp. 907–927, 2016.

[23] G. Shmueli *et al.*, "To explain or to predict?" *Statistical science*, vol. 25, no. 3, pp. 289–310, 2010.