Don’t fall for tuning parameters: Tuning-free variable selection in high dimensions with the TREX

Johannes Lederer  
Department of Statistical Science  
Cornell University  
Ithaca, NY 14853  
johanneslederer@cornell.edu

Christian Müller  
Courant Institute  
New York University  
New York, NY 10012  
cml92@nyu.edu

Abstract

Lasso is a seminal contribution to high-dimensional statistics, but it hinges on a tuning parameter that is difficult to calibrate in practice. A partial remedy for this problem is Square-Root Lasso, because it inherently calibrates to the noise variance. However, Square-Root Lasso still requires the calibration of a tuning parameter to all other aspects of the model. In this study, we introduce TREX, an alternative to Lasso with an inherent calibration to all aspects of the model. This adaptation to the entire model renders TREX an estimator that does not require any calibration of tuning parameters. We show that TREX can outperform cross-validated Lasso in terms of variable selection and computational efficiency. We also introduce a bootstrapped version of TREX that can further improve variable selection. We illustrate the promising performance of TREX both on synthetic data and on a recent high-dimensional biological data set that considers riboflavin production in B. subtilis.

1 Introduction

In recent years, statistical tools that can deal with high-dimensional data and models have become pivotal in many areas of science and engineering. The advent of high-throughput technologies, for example, has transformed biology into a data-driven science that requires mathematical models with many variables. The need to analyze and reduce the complexity of these models has triggered an enormous interest in high-dimensional statistical methods that are able to separate relevant variables from irrelevant ones \[1,2,3,4]. Among the many existing methods, Lasso \[5\] and Square-Root Lasso (or Scaled Lasso) \[6,7,8,9\] have become very popular representatives.

In practice, however, high-dimensional variable selection turns out to be a difficult task. A major shortcoming of Lasso, in particular, is its need for a tuning parameter that is properly adjusted to all aspects of the model \[10\] and therefore difficult to calibrate in practice. Using cross-validation to adjust the tuning parameter is not a satisfactory approach to this problem, because cross-validation is computational inefficient and does not suit variable selection. Replacing Lasso by Square-Root Lasso is also not a satisfactory approach, because Square-Root Lasso resolves only the adjustment of the tuning parameter to the variance of the noise \[11\] but does not address the adjustment to the tail behavior of the noise and to the design. In conclusion, neither of these two approaches provides at the same time readily applicable, accurate, and computational feasible variable selection.

Our contribution: In this study, we present a novel approach for high-dimensional variable selection. First, in Section\[2\] we convey new insight into high-dimensional regression and reveal how a systematic development of the Square-Root Lasso approach leads to TREX, an estimator without...
any tuning parameter. For optimal variable selection, we then combine the with a bootstrapping scheme. Next, in Section 3 we detail on implementations and demonstrate in a thorough numerical study that TREX is both accurate and computationally efficient. Finally, in Section 5 we discuss the findings and indicate directions for subsequent studies.

2 Methodology

In this section, we state the methodology for our study: After specifying the framework, we introduce TREX and its bootstrapped sibling, B-TREX, and describe how they relate to Lasso and Square-Root Lasso.

2.1 Framework for our study

In this study, we aim at variable selection in linear regression. We therefore consider models of the form

\[ Y = X\beta^* + \sigma \epsilon, \quad \text{(Model)} \]

where \( Y \in \mathbb{R}^n \) is a response vector, \( X \in \mathbb{R}^{n \times p} \) a design matrix, \( \sigma > 0 \) a constant, and \( \epsilon \in \mathbb{R}^n \) a noise vector. We allow in particular for high-dimensional settings, where \( p \) rivals or exceeds \( n \), and undisclosed distributions of the noise \( \sigma \epsilon \). Statistical methods for models of the above form typically target \( \beta^* \) (estimation), the support of \( \beta^* \) (variable selection), \( X\beta^* \) (prediction), or \( \sigma^2 \) (variance estimation). In this study, we mainly focus on variable selection.

To ease the exposition of the sequel, we append some conventions and notation: We allow for fixed and for random design matrices \( X \) but assume in either case the normalization \( (X^\top X)_{jj} = n \) for all \( j \in \{1, \ldots, p\} \). Moreover, we assume that the distribution of the noise vector \( \epsilon \) has variance 1 so that \( \sigma \) is the standard deviation of the entire noise \( \sigma \epsilon \). Finally, we denote the support (the index set of the nonzero entries) of a vector \( v \) by \( \text{support}(v) \) and the \( \ell_q \)-norm and the maximum norm of \( v \) by \( \|v\|_q \) and \( \|v\|_\infty \), respectively.

2.2 TREX and B-TREX

We now introduce two novel estimators for high-dimensional linear regression: TREX and B-TREX. To motivate these estimators, let us first detail on the calibration of Lasso. Recall that for a fixed tuning parameter \( \lambda > 0 \), Lasso is a minimizer of a least-squares criterion with \( \ell_1 \)-penalty:

\[ \hat{\beta}_{\text{Lasso}}(\lambda) \in \arg\min_{\beta \in \mathbb{R}^p} \left\{ \frac{\|Y - X\beta\|_2^2}{n} + \lambda \|\beta\|_1 \right\}. \quad \text{(Lasso)} \]

The tuning parameter \( \lambda \) determines the intensity of the regularization and is therefore highly influential, and it is well understood that a reasonable choice is of the order \( \lambda \sim \frac{\sigma\|X^\top \epsilon\|_\infty}{n} \).

For example, this becomes apparent when looking at the following prediction bound for Lasso (cf. [12][13], see also [14] for an overview of Lasso prediction).

**Lemma 1.** If \( \lambda \geq 2\sigma\|X^\top \epsilon\|_\infty/n \), it holds

\[ \frac{\|X\hat{\beta}_{\text{Lasso}}(\lambda) - X\beta^*\|_2^2}{n} \leq 2\lambda\|\beta^*\|_1. \]

This suggests a tuning parameter \( \lambda \) that is small (since the bound is proportional to \( \lambda \)) but not too small (to satisfy the condition \( \lambda \geq \sigma\|X^\top \epsilon\|_\infty/n \)). In practice, however, the corresponding calibration is very difficult, because it needs to incorporate several, often unknown, aspects of the model:

(a) the design matrix \( X \);
(b) the standard deviation of the noise \( \sigma \);
(c) the support of the noise \( \sigma \epsilon \).

References:

[12][13][14]
While one line of research approaches (a) and describes the calibration of Lasso to the design matrix \[15,10,14\], Square-Root Lasso approaches (b) and eliminates the calibration to the standard deviation of the noise. To elucidate the latter approach, we first recall that for a fixed tuning parameter \(\gamma > 0\), Square-Root Lasso is defined similarly as the Lasso:

\[
\hat{\beta}_{\text{Square-Root Lasso}}(\gamma) \in \arg\min_{\beta \in \mathbb{R}^p} \left\{ \frac{\|Y - X\beta\|^2}{\sqrt{n}} + \gamma \|\beta\|_1 \right\} \quad \text{(Square-Root Lasso)}
\]

Similarly as Lasso, Square-Root Lasso requires a tuning parameter \(\gamma\) to determine the intensity of the regularization. However, the tuning parameter should here be of the order (see, for example, [6])

\[
\gamma \sim \frac{\|X^T\epsilon\|_\infty}{n},
\]

so that Square-Root Lasso does not require a calibration to the standard deviation of the noise. The origin of this feature can be readily located: Reformulating the definition of Square-Root Lasso as

\[
\hat{\beta}_{\text{Square-Root Lasso}}(\gamma) \in \arg\min_{\beta \in \mathbb{R}^p} \left\{ \frac{\|Y - X\beta\|^2}{\|Y - X\beta\|_2^2} + \gamma \|\beta\|_1 \right\}
\]

identifies the factor \(\|Y - X\beta\|_2/\sqrt{n}\) in the denominator of the first term as the distinction to Lasso. This additional factor acts as an inherent estimator of the standard deviation of the noise \(\sigma\) and makes therefore the calibration to \(\sigma\) obsolete. On the other hand, Square-Root Lasso still contains a tuning parameter that needs to be adjusted to (a) the design matrix and (c) the tail behavior of the noise vector.

We now develop the Square-Root Lasso approach further to address all aspects (a), (b), and (c). For this, we aim at incorporating an inherent estimation not of \(\sigma\) but rather of the entire quantity of interest \(\|X^T\epsilon\|_\infty/n\). Indeed, if \(\tilde{\beta}\) is a consistent estimator of \(\beta^*\), then \(\sigma\|X^T(Y - X\tilde{\beta})\|_\infty/n\) is a consistent estimator of \(\sigma\|X^T\epsilon\|_\infty/n\) and, using the continuity of the Lasso path, we can derive the following.

**Lemma 2.** Assume that \(c > 0\) is an arbitrary positive constant, \(\hat{\beta}\) a consistent estimator of \(\beta^*\), and

\[
\tilde{\beta} \in \arg\min_{\beta \in \mathbb{R}^p} \left\{ \frac{\|Y - X\beta\|^2}{c\|X^T(Y - X\beta)\|_\infty} + \|\beta\|_1 \right\}.
\]

Then, \(\tilde{\beta}\) is close to a Lasso solution with tuning parameter \(\lambda = c\|X^T\epsilon\|_\infty/n\), that is,

\[
\min_{\beta \in \Omega} \|\beta - \tilde{\beta}\|_2 = o(1)
\]

for \(\Omega = \arg\min_{\beta \in \mathbb{R}^p} \left\{ \|Y - X\beta\|^2 + c\|X^T\epsilon\|_\infty \|\beta\|_1 \right\}\).

In this spirit, we aim at estimating the optimal tuning parameter by incorporating the factor \(\frac{1}{2}\|X^T(Y - X\beta)\|_\infty\) and define TREX\(^1\) according to

\[
\hat{\beta}_{\text{TREX}} \in \arg\min_{\beta \in \mathbb{R}^p} \left\{ \frac{\|Y - X\beta\|^2}{\frac{1}{2}\|X^T(Y - X\beta)\|_\infty} + \|\beta\|_1 \right\} \quad \text{(TREX)}
\]

As Lasso and Square-Root Lasso, TREX consists of two terms that balance the fit and the \(\ell_1\)-norm. The distinctive feature of TREX, however, is the absence of tuning parameters. We also note that TREX can be formulated in terms of convex optimization problems, but the above formulation as a non-convex optimization problem is commensurate with our fast implementation, see Section 3.

An interesting relation to Lasso-type penalties is finally given by the following result (we omit all proofs for the sake of brevity).

\(^1\)We call this new approach TREX to emphasize that it aims at Tuning-free Regression that adapts to the Entire noise \(\sigma\epsilon\) and the design matrix \(X\).
Theorem 1. Let $c > 0$ be an arbitrary positive constant, then

$$
\min_{\beta \in \mathbb{R}^p} \left\{ \frac{\|Y - X \beta\|^2}{c\|X^\top(Y - X \beta)\|_\infty} + \|\beta\|_1 \text{ such that } \|X^\top(Y - X \beta)\|_\infty \leq \|X^\top Y\|_\infty \right\}
$$

$$
= \min_{0 \leq u \leq 2\|X^\top Y\|_\infty/n} \left\{ \min_{\beta \in \mathbb{R}^p} \left\{ \frac{\|Y - X \beta\|^2}{u} + \|\beta\|_1 \text{ such that } c\|X^\top(Y - X \beta)\|_\infty = u \right\} \right\}.
$$

Corollary 1. It holds in particular that the solutions for $c = 2$ correspond to the Lasso solutions with tuning parameter

$$
\arg\min_{0 \leq \lambda \leq 2\|X^\top Y\|_\infty/n} \left\{ \frac{\|Y - X \hat{\beta}_{\text{lasso}}(\lambda)\|^2}{n\lambda} + \frac{p\hat{\beta}_{\text{lasso}}(\lambda)}{\lambda} \right\}.
$$

The key for the latter result is that the norms $\| \cdot \|_\infty$ and $\| \cdot \|_1$ are dual; extensions to other pairs of dual norms are straightforward.

Equipped with TREX to estimate the regression vector $\beta^*$, we can tackle a broad spectrum of tasks including estimation, prediction, and variance estimation. For variable selection, however, we advocate an additional refinement based on sequential bootstrapping [16]. More specifically, we advocate B-TREX for a fixed number of bootstraps $b \in \{1, 2, \ldots\}$:

Data: $(Y, X)$;

Result: $\hat{S}_{b,\text{TREX}} \subset \{1, \ldots, p\}$;

for $i = 1$ to $b$ do

Generate a sequential bootstrap sample $(\tilde{Y}, \tilde{X})$;

Compute $\hat{\beta}_{\text{TREX}}$ on $(\tilde{Y}, \tilde{X})$ according to TREX;

Set $\hat{S}_i := \text{support}(\hat{\beta}_{\text{TREX}})$;

end

Set $\hat{S}_{b,\text{TREX}} := \{j : j \text{ is in more than half of the sets } \hat{S}_1, \ldots, \hat{S}_b\}$;

Algorithm 1: B-TREX with $b$ sequential bootstraps.

B-TREX is the majority vote over the TREX solutions for $b$ sequential bootstrap samples. Note that related bootstrapping schemes (based on traditional bootstrapping and different selection rules, however) have already been applied to Lasso [17] [18]. In practice, it can also be illustrative to report the selection frequencies of each parameter over the bootstrap samples (cf. table in Figure 4 in Section 4.2). We finally note that for prediction and estimation with B-TREX, we perform a least-squares refitting on the set $\hat{S}_{b,\text{TREX}}$. This refitting can improve the prediction and estimation accuracy if the set $\hat{S}_{b,\text{TREX}}$ is a good estimator of the true support of $\beta^*$ [19] [20].

3 Implementation of TREX

To compute TREX we solve a non-smooth, non-convex optimization problem. The associated objective function

$$
f_{\text{TREX}} : \beta \mapsto L(\beta) + \|\beta\|_1
$$

comprises the non-smooth, non-convex data-fitting term $L(\beta) := \frac{\|Y - X \beta\|^2}{\frac{1}{2}\|X^\top(Y - X \beta)\|_\infty}$ and the non-smooth, convex $\ell_1$-regularization term $\|\beta\|_1$. In the literature, convex objective functions are often preferred, because they typically allow for the application of established optimization procedures. However, since the advent of novel optimization procedures [21], non-convex regularization terms such as the Smoothly Clipped After Deviation (SCAD) regularization [22] enjoy increasing popularity. More recently, also objective functions with non-convex data-fitting terms have been proved both statistically valuable and efficiently computable [23] [24] [25].

To facilitate the optimization of the TREX objective function, we exploit that the non-smooth data-fitting term $L(\beta)$ can be well approximated by a smooth data-fitting term. For this, we note that for all vectors $a \in \mathbb{R}^p$ and positive integers $q \in \{1, 2, \ldots\}$, it holds that

$$
\|a\|_\infty \leq \|a\|_q \leq p^{\frac{1}{q}} \|a\|_\infty,
$$
and the data-fitting term \( L(\beta) \) can therefore be approximated by the smooth data-fitting term

\[
\tilde{L}(\beta) = \frac{\|Y - X\beta\|_2^2}{2\|X^\top (Y - X\beta)\|_q^q}.
\]

In practice, we find that any \( q \in [20, 100] \) works well for our purposes. We can calculate the gradient of the smooth approximation \( \tilde{L}(\beta) \) and obtain

\[
\nabla \tilde{L}(\beta) = \frac{2\|Y - X\beta\|_2^2 X^\top X (X^\top (Y - X\beta))^{q-1}}{\|X^\top (Y - X\beta)\|_q^{q+1}} - \frac{4X^\top (Y - X\beta)}{\|X^\top (Y - X\beta)\|_q^q}.
\]

The approximation \( \tilde{L}(\beta) + \|\beta\|_1 \) of the criterion \( f_{\text{TREX}} \) is now amenable to effective (local) optimization with projected scaled sub-gradient (PSS) algorithms \[26\]. PSS schemes are specifically tailored to objective functions with smooth, possibly non-convex data-fitting terms and \( \ell_1 \)-regularization terms. PSS algorithms only require zeroth- and first-order information about the objective function, have a linear time and space complexity per iteration, and are especially effective for problems with sparse solutions. Several PSS algorithms that fit our framework are described in \[26\], and the corresponding implementations are available at [http://www.di.ens.fr/˜mschmidt/Software/L1General.html](http://www.di.ens.fr/~mschmidt/Software/L1General.html). Among these algorithms, the Gafni-Bertsekas variant was particularly effective for our purposes. For further information about PSS algorithms, we refer to \[26\] Chapter 2.3.1).

4 Numerical examples

We demonstrate the performance of TREX and B-TREX on two numerical examples. We first consider a synthetic example inspired by \[6\]. We then consider a high-dimensional biological data set that has been recently studied in great detail in \[27\].

For both examples, we perform the numerical computations in MATLAB. To compute Lasso, we use lasso.m, which follows the popular glmnet R code. To compute TREX, we use Schmidt’s PSS algorithm implemented in l1General2PSSgb.m to optimize the approximate TREX objective function with \( q = 40 \). We use the PSS algorithm with standard parameter settings and the initial solution to the parsimonious all-zeros vector \( \beta_{\text{init}} = (0, \ldots, 0)^\top \in \mathbb{R}^{p} \). We finally use the following PSS stopping criteria: minimum relative progress tolerance \( \text{optTol}=1e^{-7} \), minimum gradient tolerance \( \text{progTol}=1e^{-9} \), and maximum number of iterations \( \text{maxIter}=\max(0.2p, 200) \).

4.1 Synthetic example

We first evaluate the performance of TREX and B-TREX on synthetic data. The method of comparison is Lasso with tuning parameter that leads to minimal 10-fold cross-validated mean squared error (Lasso-CV). We generate data according to the linear regression model (Model) with parameters inspired by the Monte Carlo simulations in \[6\]. We set the sample size to \( n = 100 \), the number of variables to \( p = 500 \), and the true regression vector to \( \beta^* = (1, 1, 1, 1, 1, 0, \ldots, 0)^\top \); we sample standard normal errors \( \epsilon \sim \mathcal{N}(0, I_n) \) and multiply them by a fixed standard deviation \( \sigma \in \{0.1, 0.5, 1, 3\} \); and we sample the rows of \( X \) from the \( p \)-dimensional normal distribution \( \mathcal{N}(0, \Sigma) \), where \( \Sigma \) is the covariance matrix with diagonal entries \( \Sigma_{ii} = 1 \) and off-diagonal entries \( \Sigma_{ij} = \kappa \) for \( i, j \in \{1, \ldots, p\} \) and a fixed correlation \( \kappa \in \{0, 0.5, 0.9\} \), and then normalized them to Euclidean norm \( \sqrt{n} \). We report for each of these settings the prediction error \( \|X\beta^* - X\hat{\beta}\|_2^2/n \), estimation error \( \|\beta^* - \hat{\beta}\|_2 \), and variable selection performance of Lasso-CV, TREX, and B-TREX (with \( b = 31 \)) averaged over 51 repetitions.

Figure 1 summarizes the numerical results for the settings with \( \kappa = 0 \). The runtimes for the computation of a Lasso path, as disclosed in Figure 1, are between one and five seconds and increase with \( \sigma \). The runtimes for the computation of TREX are around 0.1 seconds for all \( \sigma \). The prediction and estimation errors are disclosed in Figures 1a and 1c: For weak noise (\( \sigma = 0.1 \)), all estimators provide near-perfect prediction and estimation. For intermediate noise (\( \sigma \in \{0.5, 1\} \)), all predictions and estimations are still very accurate: The prediction errors of TREX are between 1.5 and 2 and the ones of both Lasso-CV and B-TREX considerably less than 1; the estimation errors of all
estimators are between 0.1 and 1.1. Lasso-CV and B-TREX outperform TREX for both prediction and estimation. For strong noise ($\sigma = 3$), the performance of all methods deteriorates, resulting in prediction errors around four to six and estimation errors around 2. Lasso-CV and B-TREX again outperform TREX for both prediction and estimation. The variable selection results are disclosed in Figures 1b, 1e, and 1f: TREX provides near-perfect variable selection for $\sigma \in \{0.1, 0.5\}$ and B-TREX for $\sigma \in \{0.1, 0.5, 1\}$; for stronger noise, the Hamming distance of these two estimators to $\beta^*$ increase due to increasing numbers of false positives (for TREX) or false negatives (for B-TREX). Lasso-CV, on the other hand, consistently selects too many variables (around 20 to 30 false positives across all $\sigma$). For $\kappa \in \{0.5, 0.9\}$ (data not shown), the estimators relate similarly in terms of prediction and estimation. In terms of variable selection, the performance of TREX deteriorates as compared to Lasso-CV. B-TREX, on the other hand, provides excellent variable selection for all considered parameter settings. A summary of the results for the correlated settings as well as for additional synthetic examples will be provided in an extended version of this manuscript.

In summary, the numerical results for the standard synthetic example considered here provide first evidence that TREX and B-TREX can rival Lasso-CV in terms of prediction and estimation and can outmatch Lasso-CV in terms of variable selection.

### 4.2 Biological example: Riboflavin production in B. subtilis

We next consider a recently published high-dimensional biological data set for the production of riboflavin (vitamin $B_2$) in B. subtilis (Bacillus subtilis) \[27\]. The data set comprises expression profiles of $p = 4088$ genes of different B. subtilis strains for a total of $n = 71$ experiments with varying settings. The corresponding expression profiles are stored in the matrix $X \in \mathbb{R}^{71 \times 4088}$. Along with these expression profiles, the associated standardized riboflavin log-production rates $Y \in \mathbb{R}^{71}$ have been measured. The main objective is now to identify a small set of genes that is highly predictive for the riboflavin production rate. Since B. subtilis is one of the main industrially exploited sources for riboflavin, subsequent genetic modifications of the identified genes may then improve the industrial production of riboflavin. Bühlmann et al. \[27\] analyze the data set with various tools from high-dimensional statistics, including causal modeling, covariance selection, and Lasso based regression. We reproduce their regression results and compare them with the result provided by TREX and B-TREX.
We first report the outcome of the Lasso based approaches detailed in [27]. The runtime for the computation of a single Lasso path with the MATLAB routine is approximately 19 seconds. Lasso-CV selects 38 genes, that is, its solution has 38 non-zero coefficients; the 20 genes with largest coefficients and the associated coefficient are listed in the table in Figure 2. We also depict the fit of the Lasso-CV solution to the standardized riboflavin log-production rates in Figure 3. To obtain a smaller set of genes, Bühlmann et al. apply a stability selection scheme [28] based on 500 subsamples of size \( \frac{n}{2} \) and the 20 coefficients that enter the corresponding Lasso paths first. This stability selection scheme selects three genes: \( \text{LYSC}_{at} \), \( \text{YOAB}_{at} \), and \( \text{YXLD}_{at} \).

We now approach the biological example with TREX and B-TREX. For this, we apply TREX and B-TREX with the same parameters as in the synthetic example. The runtime for a single TREX computation is then approximately 18 seconds. TREX selects 20 genes and therefore provides a considerably sparser solution than Lasso-CV; the corresponding genes and the associated coefficients are listed in the table in Figure 2. B-TREX with \( b = 31 \) sequential bootstraps and the standard majority vote selects three genes: \( \text{YXLE}_{at} \), \( \text{YOAB}_{at} \), and \( \text{YXLD}_{at} \). The outcomes of B-TREX with selection rules different from majority vote can be deduced from the table in Figure 2, where we list the selection frequencies of the 20 genes that are selected most frequently across the bootstraps. We finally depict the fit of the TREX and B-TREX solutions to the standardized riboflavin log-production rates in Figure 2.

A joint comparison of the solutions of Lasso-CV, TREX, and B-TREX reveals four key insights: First, the set of genes selected by TREX and the set of the 20 genes corresponding to the highest coefficients in the Lasso-CV solution are distinct but share a common subset of 12 genes (see the table in Figure 2). Second, the sets of genes selected by B-TREX and Lasso-CV stability selection have two genes in common, \( \text{YOAB}_{at} \) and \( \text{YXLD}_{at} \), which also correspond to the top-ranked Lasso-CV and TREX genes (see the table in Figure 2). On the other hand, the gene \( \text{YXLE}_{at} \) is associated with the highest frequency in the B-TREX solution but is not selected by stability selection. We argue that the B-TREX solution is biologically more plausible: Since the genes \( \text{YXLD}_{at} \) and \( \text{YXLE}_{at} \) are located in the same operon, both genes are likely to be co-expressed and are thus likely involved in similar cellular functions. Third, the solution of Lasso-CV (with 38 coefficients) and TREX (with 20 coefficients) fit the standardized riboflavin log-production rate very well. The B-TREX solution also provides a reasonable fit with only three coefficients (see the plot in Figure 2). This observation indicates that all three solutions provide good

---

**Figure 2:** Left: Fit to the standardized riboflavin log-production rate in B. subtilis. The fit with Lasso-CV is based on 38 genes, the fit with TREX on 20 genes, and the fit with B-TREX on 3 genes. Right: Gene rankings for riboflavin production in B. subtilis. The left panel contains the 20 genes with largest coefficients in Lasso-CV (out of 38 genes with non-zero coefficients) and the associated coefficients. The center panel contains the 20 genes with non-zero coefficients in TREX and the associated coefficients. The right panel contains the 20 genes with largest frequencies in B-TREX and the associated frequencies.
empirical prediction; however, a comparison in terms of prediction error cannot be made, since the
ground truth $X\beta^*$ is unknown. Fourth, the runtime for the computation of TREX is comparable
to the runtime for the computation of a single Lasso path, implying that Lasso-CV needed about
10 times longer to be computed than TREX. Note also that the present stability selection scheme
involves 500 Lasso computations, whereas B-TREX involves only $b = 31$ TREX computations.

5 Conclusions

We have introduced TREX, a simple, fast, and accurate method for high-dimensional variable selec-
tion. We have shown in Section 2 that TREX avoids tuning parameters and, therefore, challenging
calibrations. Moreover, we have shown in Section 4.1 that TREX can outmatch a cross-validated
Lasso in terms of speed and accuracy.

To further improve variable selection, we propose B-TREX, a combination of TREX with a boot-
strapping scheme. This proposition is supported by the numerical results in Section 4.1 and in line
with earlier claims that bootstrapping can improve variable selection \[17\] \[18\]. Moreover, we argue
in Section 4.2 that the solution of B-TREX on the recent riboflavin data set in \[27\] is supported by
biological insights.

Our contribution therefore suggests that TREX and B-TREX can challenge standard methods such as
cross-validated Lasso and can be valuable in a wide range of applications. We expect that fur-
ther theoretical guarantees and optimized implementations can be obtained in subsequent studies.
Finally, we note that TREX can easily be generalized (to incorporate group structures, for example)
and varied (to include non-convex penalties, for example).

References

[1] A. Belloni and V. Chernozhukov. High dimensional sparse econometric models: an introduc-
tion. In P. Alquier, E. Gautier, and G. Stoltz, editors, Inverse Problems and High-Dimensional
Estimation, volume 203 of Lect. Notes Stat. Proc. Springer, 2011.
[2] P. Bühlmann and S. van de Geer. Statistics for high-dimensional data: Methods, theory and
applications. Springer Series in Statistics. Springer, 2011.
[3] T. Cai and X. Shen, editors. High-dimensional data analysis, volume 2 of Frontiers of Statis-
tics. World Scientific Publishing, 2011.
[4] T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning: Data mining,
inference, and prediction. Springer Series in Statistics. Springer, 2001.
[5] R. Tibshirani. Regression shrinkage and selection via the lasso. J. Roy. Statist. Soc. Ser. B,
58(1):267–288, 1996.
[6] A. Belloni, V. Chernozhukov, and L. Wang. Square-root lasso: pivotal recovery of sparse
signals via conic programming. Biometrika, 98(4):791–806, 2011.
[7] A. Owen. A robust hybrid of lasso and ridge regression. In Prediction and discovery, volume
443 of Contemp. Math., pages 59–71. Amer. Math. Soc., 2007.
[8] N. Städler, P. Bühlmann, and S. van de Geer. $\ell_1$-penalization for mixture regression models.
Test, 19(2):209–256, 2010.
[9] T. Sun and C.-H. Zhang. Scaled sparse linear regression. Biometrika, 99(4):879–898, 2012.
[10] M. Hebiri and J. Lederer. How correlations influence Lasso prediction. IEEE Trans. Inform.
Theory, 59(3):1846–1854, 2013.
[11] F. Bunea, J. Lederer, and Y. She. The group square-root lasso: Theoretical properties and fast
algorithms. IEEE Trans. Inform. Theory, 60(2):1313–1325, 2014.
[12] V. Koltchinskii, K. Lounici, and A. Tsybakov. Nuclear-norm penalization and optimal rates
for noisy low-rank matrix completion. Ann. Statist., 39(5):2302–2329, 2011.
[13] P. Rigollet and A. Tsybakov. Exponential Screening and optimal rates of sparse estimation.
Ann. Statist., 39(2):731–771, 2011.
[14] A. Dalalyan, M. Hebiri, and J. Lederer. On the prediction performance of the lasso. *preprint, arXiv:1402.1700*, 2014.

[15] S. van de Geer and J. Lederer. The Lasso, correlated design, and improved oracle inequalities. *IMS Collections*, 9:303–316, 2013.

[16] C. Rao, P. Pathak, and V. Koltchinskii. Bootstrap by sequential resampling. *J. Statist. Plann. Inference*, 64(2):257–281, 1997.

[17] F. Bach. Bolasso: Model consistent lasso estimation through the bootstrap. In *Proceedings of the 25th International Conference on Machine Learning*, ICML ’08, pages 33–40, 2008.

[18] F. Bunea, Y. She, H. Ombao, A. Gongvatana, K. Devlin, and R. Cohen. Penalized least squares regression methods and applications to neuroimaging. *Neuroimage*, 55, 2011.

[19] A. Belloni and V. Chernozhukov. Least squares after model selection in high-dimensional sparse models. *Bernoulli*, 19(2):363–719, 2013.

[20] J. Lederer. Trust, but verify: benefits and pitfalls of least-squares refitting in high dimensions. *preprint, arxiv/1306.0113*, 2013.

[21] R. Mazumder, J. Friedman, and T. Hastie. Sparsenet: Coordinate descent with nonconvex penalties. *J. Amer. Statist. Assoc.*, 106(495):1125–1138, 2011.

[22] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.*, 96:1348–1360, 2001.

[23] P-L. Loh and M. Wainwright. Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. In *NIPS*, pages 476–484, 2013.

[24] Z. Wang, H. Liu, and T. Zhang. Optimal computational and statistical rates of convergence for sparse nonconvex learning problems. *preprint, arXiv/1306.4960*, 2013.

[25] Y. Nesterov. Gradient methods for minimizing composite objective function. CORE Discussion Papers 2007076, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2007.

[26] M. Schmidt. *Graphical Model Structure Learning with LI-Regularization*. PhD thesis, University of British Columbia, 2010.

[27] P. Bühlmann, M. Kalisch, and L. Meier. High-dimensional statistics with a view toward applications in biology. *Annual Review of Statistics and Its Application*, 1(1):255–278, 2014.

[28] N. Meinshausen and P. Bühlmann. Stability selection. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 72(4):417–473, 2010.