Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

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 L. Müller
NYU Courant Institute of Mathematical Sciences and
Center for Genomics and Systems Biology
New York, NY 10012
cm192@nyu.edu

Abstract

Lasso is a popular method for high-dimensional variable selection, but it hinges on a tuning parameter that is difficult to calibrate in practice. 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 two biological data sets from the fields of genomics and proteomics.

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 (Belloni and Chernozhukov 2011; Bühlmann and van de Geer 2011; Hastie, Tibshirani, and Friedman 2001). Among the many existing methods, Lasso (Tibshirani 1996) and Square-Root Lasso (or Scaled Lasso) (Belloni, Chernozhukov, and Wang 2011; Owen 2007; Städler, Bühlmann, and van de Geer 2010; Sun and Zhang 2012) 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 (Hebiri and Lederer 2013) 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 computationally inefficient and provides unsatisfactory variable selection performance. 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 but does not address the adjustment to the tail behavior of the noise and to the design. Similarly, more advanced Lasso-based procedures such as the Uncorrelated Lasso (Chen et al. 2013) or the Trace Lasso (Grave, Obozinski, and Bach 2011) also comprise tuning parameters that need proper calibration. In conclusion, none of the present approaches simultaneously provides parameter-free, accurate, and computationally attractive variable selection.

Our contribution: In this study, we present a novel approach for high-dimensional variable selection. First, we 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 TREX with a bootstrapping scheme. Next, we detail on implementations and demonstrate in a thorough numerical study that TREX is both accurate and computationally efficient. Finally, we discuss the findings and indicate directions for subsequent studies.

Methodology

Framework for our study

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

$$Y = X\beta^* + \sigma\epsilon, \qquad \text{(Model)}$$

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 β^* (estimation), the support of β^* (variable selection), $X\beta^*$ (prediction), or σ^2 (variance estimation). In this study, we 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 $\left(X^\top X\right)_{jj} = n$ for all $j \in \{1,\dots,p\}$. Moreover, we assume that the distribution of the noise vector ϵ has variance 1 so that σ is the standard deviation of the entire noise $\sigma\epsilon$. Finally, we denote the support (the index set of the non-zero entries) of a vector v by $\mathrm{support}(v)$ and the ℓ_q−norm and the maximum norm of v by $\|v\|_q$ and $\|v\|_\infty$, respectively.

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 ℓ_1-penalty:

$$\widehat{\beta}_{\text{Lasso}}(\lambda) \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{\|Y - X\beta\|_2^2}{n} + \lambda\|\beta\|_1 \right\}. \quad \text{(Lasso)}$$

The tuning parameter λ 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. (Koltchinskii, Lounici, and Tsybakov 2011; Rigollet and Tsybakov 2011), see also (Dalalyan, Hebiri, and Lederer 2014) for an overview of Lasso prediction).

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

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

This suggests a tuning parameter λ that is small (since the bound is proportional to λ) but not too small (to satisfy the condition $\lambda \gtrsim \sigma\|X^\top\epsilon\|_\infty/n$). 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 σ;

(c) the tail behavior of the noise vector ϵ.

While one line of research approaches (a) and describes the calibration of Lasso to the design matrix (van de Geer and Lederer 2013; Hebiri and Lederer 2013; Dalalyan, Hebiri, and Lederer 2014), 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 Lasso:

$$\widehat{\beta}_{\sqrt{\text{Lasso}}}(\gamma) \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{\|Y - X\beta\|_2}{\sqrt{n}} + \gamma\|\beta\|_1 \right\}. \quad \text{(Square-Root Lasso)}$$

Square-Root Lasso also requires a tuning parameter γ to determine the intensity of the regularization. However, the tuning parameter should here be of the order (see, for example, (Belloni, Chernozhukov, and Wang 2011))

$$\gamma \sim \frac{\|X^\top\epsilon\|_\infty}{n},$$

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

$$\widehat{\beta}_{\sqrt{\text{Lasso}}}(\gamma) \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{\frac{\|Y-X\beta\|_2^2}{n}}{\frac{\|Y-X\beta\|_2}{\sqrt{n}}} + \gamma\|\beta\|_1 \right\}$$

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 σ and makes therefore the calibration to σ 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 σ but rather of the entire quantity of interest $\sigma\|X^\top\epsilon\|_\infty/n$. For this, note that if $\widehat{\beta}$ is a consistent estimator of β^*, then $\sigma\|X^\top(Y - X\widehat{\beta})\|_\infty/n$ is a consistent estimator of $\sigma\|X^\top\epsilon\|_\infty/n$. In this spirit, we define TREX[1] according to

$$\widehat{\beta}_{\text{TREX}} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{\|Y - X\beta\|_2^2}{\frac{1}{2}\|X^\top(Y - X\beta)\|_\infty} + \|\beta\|_1 \right\}. \quad \text{(TREX)}$$

Square-Root Lasso and Lasso are equivalent families of estimators (there is a one-to-one mapping between the tuning parameter paths of Square-Root Lasso and Lasso); in contrast, TREX is a *single, tuning-free* estimator, and its solution is in general not on the tuning parameter paths of Lasso and Square-Root Lasso. However, we can establish an interesting relationship between these paths and TREX (we omit all proofs for sake of brevity):

Theorem 1. *It holds that*

$$\begin{aligned} &\min_{\beta\in\mathbb{R}^p} \left\{ \frac{\|Y - X\beta\|_2^2}{\frac{1}{2}\|X^\top(Y - X\beta)\|_\infty} + \|\beta\|_1 \right. \\ &\qquad\qquad \left. \textit{such that } \|X^\top(Y - X\beta)\|_\infty \leq \|X^\top Y\|_\infty \right\} \\ &= \min_{0\leq u\leq 2\|X^\top Y\|_\infty/n} \left\{ \min_{\beta\in\mathbb{R}^p} \left\{ \frac{\|Y - X\beta\|_2^2}{u} + \|\beta\|_1 \right. \right. \\ &\qquad\qquad \left. \left. \textit{such that } \frac{1}{2}\|X^\top(Y - X\beta)\|_\infty = u \right\} \right\}. \end{aligned}$$

In view of the Karush-Kuhn-Tucker conditions for Lasso, the latter formulation strongly resembles the Lasso path. This resemblance is no surprise: In fact, any consistent estimator $\widehat{\beta}$ of β^* is related to a Lasso solution with an optimal (but in practice *unknown*) tuning parameter $\lambda \sim \sigma\|X^\top\epsilon\|_\infty/n$ via the formulation of TREX:

Lemma 2. *Assume that $\widehat{\beta}$ a consistent estimator of β^* and*

$$\widetilde{\beta} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{\|Y - X\beta\|_2^2}{\frac{1}{2}\|X^\top(Y - X\widehat{\beta})\|_\infty} + \|\beta\|_1 \right\}.$$

[1] We call this new approach TREX to emphasize that it aims at *T*uning-free *R*egression that adapts to the *E*ntire noise $\sigma\epsilon$ and the design matrix X.

Then, $\widetilde{\beta}$ is close to a Lasso solution with tuning parameter $\lambda = \frac{1}{2}\|X^\top \epsilon\|_\infty / n$, that is,

$$\min_{\beta \in \Omega} \|\widetilde{\beta} - \beta\|_2 = o(1)$$

for $\Omega = \operatorname{argmin}_{\beta \in \mathbb{R}^p} \left\{ \|Y - X\beta\|_2^2 + \frac{1}{2}\|X^\top \epsilon\|_\infty \|\beta\|_1 \right\}$.

Equipped with TREX to estimate the regression vector β^*, we can tackle a broad spectrum of tasks including estimation, prediction, and variance estimation. In this paper, however, we focus on variable selection. For this task, we advocate an additional refinement based on sequential bootstrapping (Rao, Pathak, and Koltchinskii 1997). More specifically, we advocate B-TREX for a fixed number of bootstraps $b \in \{1, 2, \dots\}$:

```
**Data**: $(Y, X)$;
**Result**: $\widehat{S}_{\text{B-TREX}} \subset \{1, \dots, p\}$;
**for** $i = 1$ *to* $b$ **do**
    Generate a sequential bootstrap sample $(\widetilde{Y}, \widetilde{X})$;
    Compute $\widehat{\beta}_{\text{TREX}}$ on $(\widetilde{Y}, \widetilde{X})$ according to (TREX);
    Set $\widehat{S}_i := \text{support}(\widehat{\beta}_{\text{TREX}})$;
**end**
Set $\widehat{S}_{\text{B-TREX}} := \{j :$
$j$ is in more than half of the sets $\widehat{S}_1, \dots, \widehat{S}_b\}$;
```

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 (Bach 2008; Bunea et al. 2011). In practice, it can also be illustrative to report the selection frequencies of each parameter over the bootstrap samples (cf. Figure 3). We finally note that B-TREX readily provides estimation and prediction if a least-squares refitting on the set $\widehat{S}_{\text{B-TREX}}$ is performed. This refitting can improve the prediction and estimation accuracy if the set $\widehat{S}_{\text{B-TREX}}$ is a good estimator of the true support of β^* (Belloni and Chernozhukov 2013; Lederer 2013).

We point out that the norms $\|\cdot\|_\infty$ and $\|\cdot\|_1$ in the formulation of TREX are dual and that extensions to other pairs of dual norms are straightforward.

A theoretical analysis of TREX is beyond the scope of this paper but is the subject of a forthcoming theory paper (with different authors). Note also that theoretical results for standard variable selection methods are incomplete: in particular, there are currently *no finite sample guarantees for approaches based on Lasso and Square-Root Lasso*: Finite sample bounds (“Oracle inequalities”) for Lasso (Bühlmann and van de Geer 2011) and Square-Root Lasso (Bunea, Lederer, and She 2014) require that the tuning parameters are properly calibrated to the model; yet, there are no guarantees that standard calibration schemes such as Cross-Validation or BIC-type criteria provide such tuning parameters.

Implementation of TREX

To compute TREX, we consider the objective function

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

that comprises the data-fitting term $L(\beta) := \frac{\|Y - X\beta\|_2^2}{\frac{1}{2}\|X^\top(Y - X\beta)\|_\infty}$ and the ℓ_1-regularization term $\|\beta\|_1$. To make this objective function amenable to standard algorithms (Nesterov 2007; Schmidt 2010), we invoke a smooth approximation of the data-fitting term. For this, we note that for all vectors $a \in \mathbb{R}^p$ and positive integers $q \in \{1, 2, \dots\}$, 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

$$\overline{L}(\beta) = \frac{\|Y - X\beta\|_2^2}{\frac{1}{2}\|X^\top(Y - X\beta)\|_q}.$$

We find that any $q \in [20, 100]$ works well in practice (see supplementary material). We can calculate the gradient of the smooth approximation $\overline{L}(\beta)$ and obtain

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

The approximation $\overline{L}(\beta) + \|\beta\|_1$ of the criterion f_{TREX} is now amenable to effective (local) optimization with projected scaled sub-gradient (PSS) algorithms (Schmidt 2010). PSS schemes are specifically tailored to objective functions with smooth, possibly non-convex data-fitting terms and ℓ_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 (Schmidt 2010, Chapter 2.3.1)[2]. Among these algorithms, the Gafni-Bertsekas variant was particularly effective for our purposes.

The smooth formulation of the TREX criterion remains non-convex; therefore, convergence to the global minimum cannot be guaranteed. Neverthless, we show that the above implementation is fast, scalable, and provides estimators with excellent statistical performance.

Note also that the advent of novel optimization procedures (Breheny and Huang 2011; Mazumder, Friedman, and Hastie 2011) lead to an increasing popularity of non-convex regularization terms such as the Smoothly Clipped After Deviation (SCAD) (Fan and Li 2001) and Minimax Concave Penality (MCP) (Zhang 2010). More recently, also objective functions with non-convex data-fitting terms have been proved both statistically valuable and efficiently computable (Loh and Wainwright 2013; Nesterov 2007; Wang, Liu, and Zhang 2013).

[2] http://www.di.ens.fr/%7Emschmidt/Software/L1General.html provides the implementations

Numerical Examples

We demonstrate the performance of TREX and B-TREX on three numerical examples. We first consider a synthetic example inspired by (Belloni, Chernozhukov, and Wang 2011). We then consider two high-dimensional biological data sets that involve riboflavin production in B. subtilis (Bühlmann, Kalisch, and Meier 2014) and mass spectrometry data from melanoma patients (Mian et al. 2005).

We perform the numerical computations in MATLAB 2012b on a standard MacBook Pro with dual 2GHz Intel Core i7 and 4GB 1333MHz DDR3 memory. To compute Lasso and its cross-validated version, we use the MATLAB-internal procedure `lasso.m` (with standard values), which follows the popular glmnet R code. To compute TREX, we use Schmidt's PSS algorithm implemented in `L1General2_PSSgb.m` to optimize the approximate TREX objective function with $q = 40$. We use the PSS algorithm with standard parameter settings and set the initial solution to the parsimonious all-zeros vector $\beta_{\text{init}} = (0, \ldots, 0)^\top \in \mathbb{R}^p$. We use the following PSS stopping criteria: minimum relative progress tolerance optTol=1e-7, minimum gradient tolerance progTol=1e-9, and maximum number of iterations maxIter $= \max(0.2p, 200)$. As standard for the number of bootstrap samples in B-TREX we set $b = 31$.

Synthetic Example

We first evaluate the scalability and the variable selection performance of TREX and B-TREX on synthetic data. The method of comparison is Lasso with the 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 (Belloni, Chernozhukov, and Wang 2011): We set the sample size to $n = 100$, the number of variables to $p = 500$ (or vary over p), and the true regression vector to $\beta^* = (1, 1, 1, 1, 1, 0, \ldots, 0)^\top$; we sample standard normal errors $\epsilon \sim \mathcal{N}(0, \mathrm{I}_n)$ 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 Σ 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 scalability and variable selection results averaged over 51 repetitions (thick, colored bars) and the corresponding standard deviations (thin, black bars). More precisely, we report the runtime of plain Lasso and of TREX as a function of p (for $n = 100$, $\sigma = 0.5$, $\kappa = 0$) in Figure 1, and we report the runtime and the variable selection performance of Lasso-CV, TREX, and B-TREX in Hamming distance for fixed $p = 500$ in Figure 2.

The data shown in Figure 1 suggest that the runtime for TREX is between quadratic and cubic in p (a least-squares fit results in $\mathcal{O}(p^{2.4})$) and, thus, illustrates the scalability of TREX at least up to $p = 4000$. In comparison, the runtime for a single Lasso path (*without* Cross-Validation or any other calibration scheme), shown in Figure 1, reveals a near-linear dependence of p (a least-squares fit results in $\mathcal{O}(p^{1.1})$), though with a higher offset and slope.

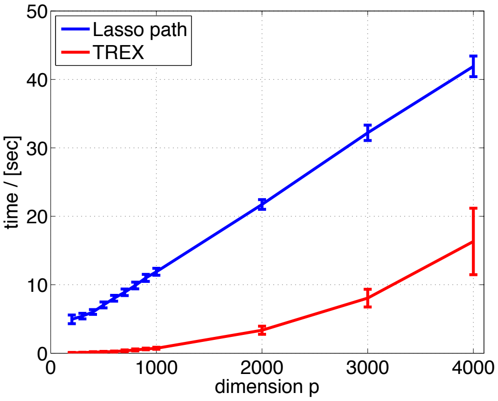

Figure 1: Runtime (in seconds) of TREX and *plain* Lasso as a function of p.

Figure 2 summarizes the numerical results for the settings with $\kappa = 0$. The runtimes disclosed in Figure 2 indicate that both TREX and B-TREX can rival Lasso-CV in terms of speed. The variable selection results show that 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 β^* increases. Lasso-CV, on the other hand, consistently selects too many variables. For $\kappa \in \{0.5, 0.9\}$ (see supplementary material), 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. In summary, the numerical results for the standard synthetic example considered here provide first evidence that TREX and B-TREX can outmatch Lasso-CV in terms of variable selection.

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) (Bühlmann, Kalisch, and Meier 2014). 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.

We first report the outcomes of standard Lasso-based approaches, which can be obtained along the lines of (Bühlmann, Kalisch, and Meier 2014). The runtime for the computation of a single Lasso path with the MATLAB routine is approximately 58 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 values are listed in Table 1. For variable selection, Bühlmann *et al.* specifically propose stability selection (Meinshausen and Bühlmann 2010). The stan-

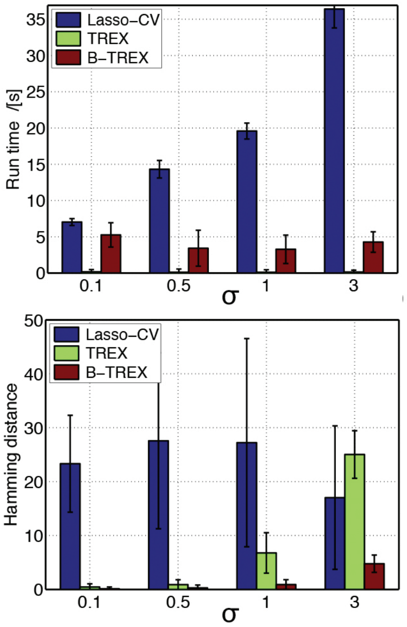

Figure 2: Runtimes (in seconds) and variable selection errors in Hamming distance on the synthetic example with $\kappa = 0$ and $p = 500$.

dard stability selection approach is based on 500 Lasso computations on subsamples of size $\lfloor \frac{n}{2} \rfloor$ and the 20 coefficients that enter the corresponding Lasso paths first. This approach yields three genes: LYSC_at, YOAB_at, and YXLD_at (Bühlmann, Kalisch, and Meier 2014).

We next apply TREX and B-TREX. The runtime for a single TREX computation is approximately 30 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 Table 1. B-TREX with the standard majority vote selects three genes: YXLE_at, YOAB_at, and YXLD_at. The outcomes of B-TREX with selection rules different from majority vote can be deduced from Table 1, where we list the selection frequencies of the 20 genes that are selected most frequently across the bootstraps.

The numerical results reveal three 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. Second, the sets of genes selected by B-TREX and Lasso-CV stability selection have the two top-ranked Lasso-CV and TREX genes in common. On the other hand, the gene associated with the highest frequency in the B-TREX solution is not selected by stability selection. The B-TREX solution is biologically plausible: Since the genes YXLD_at and YXLE_at are located in the same operon, both genes are likely to be co-expressed and involved in similar cellular functions. Third, the runtime for a single Lasso path is about two times larger than for a single TREX solution.

Lasso-CV genes	$\hat{\beta}$	TREX genes	$\hat{\beta}$	B-TREX genes	frequencies
YOAB_at	-0.232	YXLD_at	-0.219	YXLE_at	0.58
YXLD_at	-0.206	YOAB_at	-0.168	YOAB_at	0.52
ARGF_at	-0.191	ARGF_at	-0.112	YXLD_at	0.52
XHLB_at	0.138	YEBC_at	-0.088	YCKE_at	0.45
YXLE_at	-0.105	YCKE_at	0.069	LYSC_at	0.42
YEBC_at	-0.105	YCGO_at	-0.065	XTRA_at	0.42
LYSC_at	-0.066	YEZB_at	0.049	YFHE_r_at	0.42
YDDK_at	-0.063	YFHE_r_at	0.041	YPGA_at	0.39
SPOVAA_at	0.057	YHZA_at	-0.030	YDDK_at	0.35
YCLB_at	0.053	YDDK_at	-0.024	YEBC_at	0.35
YHDS_r_at	0.051	LYSC_at	-0.024	XLYA_at	0.32
DNAJ_at	-0.049	RPLL_at	-0.022	YHDS_r_at	0.29
YFHE_r_at	0.045	YXLE_at	-0.019	YTGB_at	0.29
YKBA_at	0.043	YYDA_at	-0.015	YYDA_at	0.29
YQJU_at	0.041	YCDH_at	-0.015	ARGF_at	0.26
GAPB_at	0.035	YBFI_at	0.007	RPLL_at	0.26
YYDA_at	-0.033	YHDS_r_at	0.006	XKDS_at	0.26
YTGB_at	-0.030	SPOVAA_at	0.003	YHCL_at	0.26
YBFI_at	0.022	PKSA_at	0.003	YRVJ_at	0.26
YFIO_at	0.021	YDDH_at	-0.001	YURQ_at	0.26

Table 1: 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.

The model complexities differ considerably, ranging from 3 parameters for B-TREX to 38 for Lasso-CV, and in applications, simple models are often preferred. We evaluate, therefore, the Leave-One-Out Cross-Validation errors (LOOCV-errors) of the methods under consideration for fixed numbers of parameters. As a reference, we report the LOOCV-errors of Lasso-CV (with the cross-validations performed on the training sets of size $n - 1$) in the first row of Table 2. In the three subsequent rows, we then show the LOOCV-errors of TREX, of TREX with least-squares refitting (TREX-LS), and of Lasso with tuning parameter such that the number of non-zero entries equals the number of non-zero entries of TREX (Lasso-T). Finally, we give the LOOCV-errors of B-TREX and of Lasso with tuning parameter such that the number of non-zero entries equals the number of non-zero entries of B-TREX (Lasso-BT). The computations for Stability Selection are very intensive and therefore omitted. We observe that for fixed model complexity, the solutions of TREX (with least-squares refitting) and B-TREX have lower LOOCV-error than their Lasso-based counterparts.

We conclude that the genes selected by B-TREX are commensurate with biological knowledge and that B-TREX can provide small models with good predictive performance.

Classification of Melanoma Patients

We also demonstrate the usefulness of the ranked B-TREX list for a proteomics data set from a study on melanoma patients (Mian et al. 2005). The data[3] consist of $n = 205$ mass spectrometry scans of serum samples from 101 patients with Stage I melanoma (moderately severe) and 104 patients with Stage IV melanoma (very severe). Each scan measures the intensities for 18 856 mass over charge (m/Z) values. The objective is to find m/Z values that are indicators for the

[3] see http://www.maths.nottingham.ac.uk/%7Eild/mass-spec

	LOOCV-error	# of coefficients
Lasso-CV	0.42	39
TREX	0.51	21
TREX-LS	0.45	21
Lasso-T	0.47	21
B-TREX	0.50	4
Lasso-BT	0.62	4

Table 2: Means of the Leave-One-Out Cross-Validation errors and median of the corresponding numbers of non-zero coefficients on the riboflavin dataset.

stage of the disease, eventually leading to proteins that can serve as discriminative biomarkers (Mian et al. 2005).

We want to compare outcomes of our estimators with results described in (Vasiliu, Dey, and Dryden 2014). For this, we use the same linear regression framework (even though one could also argue in favor of a logistic regression framework) and the same data pre-processing: We apply an initial peak filtering step that yields the $p = 500$ most relevant m/Z values. The resulting data are then normalized and stored in the matrix $X \in \mathbb{R}^{205\times 500}$. Next, the class labels in $Y \in \mathbb{R}^{205}$ are set to $Y_i = -1$ for $i = 1, \ldots, 101$ (Stage I patients) and to $Y_i = 1$ for $i = 102, \ldots, 205$ (Stage IV patients).

We now demonstrate the usefulness of the ranked list of predictors provided by B-TREX. For this, we first report in Figure 4 the parameter values of the least-squares refitted versions of the three estimators 10-fold cross-validated Lasso (Lasso-CV), TREX, and B-TREX. Lasso-CV selects 43 predictors, TREX selects 8 predictors, and B-TREX selects 2 predictors. We now use the signs of the (least-squares refitted) responses to estimate the class labels, cf. (Vasiliu, Dey, and Dryden 2014). We depict in Figure 4 averaged 10-fold classification errors of Sure-Independence Screening (SIS), Iterative SIS (ISIS), Elastic Net, and Penalized Euclidean Distance (PED) (all taken from (Vasiliu, Dey, and Dryden 2014)) and of TREX, B-TREX, and Lasso-CV. TREX shows almost identical classification error/model complexity as SIS and ISIS and outperforms Elastic net in terms of model complexity. PED and Lasso-CV have lower classification error but higher model complexity. B-TREX with standard majority vote results in a very sparse model with moderate error. More importantly, classification based on the top predictors from B-TREX is insensitive with respect to the threshold: For *any* number of predictors from 6 up to 23, B-TREX outperforms *all* other estimators. We conclude that the ranked list of B-TREX predictors can lead to very robust and accurate model selection and, in particular, can outperform on this data set all other standard estimators.

Conclusions

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

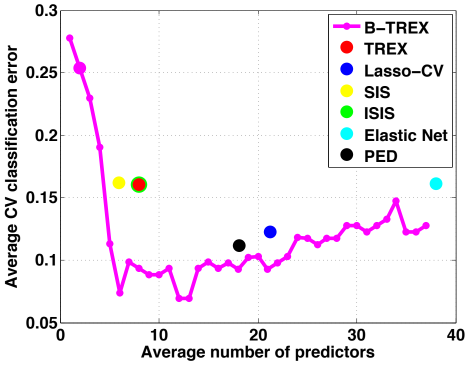

Figure 3: Mean 10-fold CV classification errors vs. average number of predictors.

To further improve variable selection, we proposed B-TREX, a combination of TREX with a bootstrapping scheme. This proposition is supported by the numerical results and in line with earlier claims that bootstrapping can improve variable selection (Bach 2008; Bunea et al. 2011). Moreover, we argue that the solution of B-TREX on the recent riboflavin data set in (Bühlmann, Kalisch, and Meier 2014) is supported by biological insights. Finally, the results on the melanoma data show that TREX can yield robust classification.

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 will provide further theoretical guarantees, optimized implementations, and tests for prediction and estimation performance in a forthcoming paper. A TREX MATLAB-toolbox as well as all presented numerical data will be made publicly available at the authors' websites.

Acknowledgments

We sincerely thank the reviewers for their insightful comments and Jacob Bien, Richard Bonneau, and Irina Gaynanova for the valuable discussions.

References

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

Belloni, A., and Chernozhukov, V. 2011. High dimensional sparse econometric models: an introduction. In Alquier, P.; Gautier, E.; and Stoltz, G., eds., *Inverse Problems and High-Dimensional Estimation*, volume 203 of *Lect. Notes Stat. Proc.* Springer.

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

Belloni, A.; Chernozhukov, V.; and Wang, L. 2011. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika* 98(4):791–806.

Breheny, P., and Huang, J. 2011. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The annals of applied statistics* 5(1):232.

Bühlmann, P., and van de Geer, S. 2011. *Statistics for high-dimensional data: Methods, theory and applications*. Springer Series in Statistics.

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

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

Bunea, F.; Lederer, J.; and She, Y. 2014. The Group Square-Root Lasso: Theoretical Properties and Fast Algorithms. *IEEE Trans. Inform. Theory* 60(2):1313–1325.

Chen, S.; Ding, C.; Luo, B.; and Xie, Y. 2013. Uncorrelated Lasso. In *AAAI*.

Dalalyan, A.; Hebiri, M.; and Lederer, J. 2014. On the Prediction Performance of the Lasso. *preprint, arXiv:1402.1700*.

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

Grave, E.; Obozinski, G.; and Bach, F. 2011. Trace Lasso: a trace norm regularization for correlated designs. In *Advances in Neural Information Processing Systems*, 2187–2195.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2001. *The elements of statistical learning: Data mining, inference, and prediction*. Springer Series in Statistics.

Hebiri, M., and Lederer, J. 2013. How Correlations Influence Lasso Prediction. *IEEE Trans. Inform. Theory* 59(3):1846–1854.

Koltchinskii, V.; Lounici, K.; and Tsybakov, A. 2011. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *Ann. Statist.* 39(5):2302–2329.

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

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

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

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

Mian, S.; Ugurel, S.; Parkinson, E.; Schlenzka, I.; Dryden, I.; Lancashire, L.; Ball, G.; Creaser, C.; Rees, R.; and Schadendorf, D. 2005. Serum proteomic fingerprinting discriminates between clinical stages and predicts disease progression in melanoma patients. *Journal of clinical oncology* 23(22):5088–5093.

Nesterov, Y. 2007. Gradient methods for minimizing composite objective function. CORE Discussion Papers 2007076, Université catholique de Louvain.

Owen, A. 2007. A robust hybrid of lasso and ridge regression. In *Prediction and discovery*, volume 443 of *Contemp. Math.* Amer. Math. Soc. 59–71.

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

Rigollet, P., and Tsybakov, A. 2011. Exponential Screening and optimal rates of sparse estimation. *Ann. Statist.* 39(2):731–771.

Schmidt, M. 2010. *Graphical Model Structure Learning with L1-Regularization*. Ph.D. Dissertation, University of British Columbia.

Städler, N.; Bühlmann, P.; and van de Geer, S. 2010. ℓ_1-penalization for mixture regression models. *Test* 19(2):209–256.

Sun, T., and Zhang, C. 2012. Scaled sparse linear regression. *Biometrika* 99(4):879–898.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B* 58(1):267–288.

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

Vasiliu, D.; Dey, T.; and Dryden, I. L. 2014. Penalized Euclidean Distance Regression. *preprint, arxiv:1405.4578*.

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

Zhang, C.-H. 2010. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics* 894–942.