Path Thresholding: Asymptotically Tuning-Free High-Dimensional Sparse Regression

Divyanshu Vats and Richard G. Baraniuk
Rice University

Abstract

In this paper, we address the challenging problem of selecting tuning parameters for high-dimensional sparse regression. We propose a simple and computationally efficient method, called path thresholding (PaTh), that transforms *any* tuning parameter-dependent sparse regression algorithm into an asymptotically tuning-free sparse regression algorithm. More specifically, we prove that, as the problem size becomes large (in the number of variables and in the number of observations), PaTh performs accurate sparse regression, under appropriate conditions, without specifying a tuning parameter. In finite-dimensional settings, we demonstrate that PaTh can alleviate the computational burden of model selection algorithms by significantly reducing the search space of tuning parameters.

1 Introduction

Sparse regression is a powerful tool used across several domains for estimating a sparse vector β^* given linear observations $y = X\beta^* + w$, where X is the known measurement matrix and w is the observation noise. Examples of applications include the analysis of gene expression data [1], fMRI data [2], and imaging data [3]. Furthermore, sparse regression forms the basis of other important machine learning algorithms including dictionary learning [4] and graphical model learning [5].

Several efficient algorithms now exist in the literature for solving sparse regression; see Lasso [6], OMP [7], and their various extensions [8–11]. Several works have also analyzed the conditions required for reliable identification of the sparse vector β^*; see [12] for a comprehensive review. However, the performance of most sparse regression algorithms depend on a *tuning parameter*, which in turn depends either on the statistics of the unknown noise in the observations or on the unknown sparsity level of the regression coefficients.

Examples of methods to select tuning parameters include the Bayesian information criterion (BIC) [13], the Akaike information criterion (AIC) [14], cross-validation (CV) [15, 16], and methods based on minimizing the Stein's unbiased risk estimate (SURE) [17–19]. All of the above methods are only suitable for low-dimensional settings, where the number of observations n is much larger than the number of variables p. In high-dimensional settings, where $p > n$, all of the above methods typically overestimate the locations of the non-zero elements in β^*. Although the overestimation could be empirically corrected using multi-stage algorithms [8, 9, 20, 21], this process is computationally demanding with no known theoretical guarantees for reliable estimation of β^*. Stability selection [22] is popular for high dimensional problems, but it is also computationally demanding.

In this paper, we develop an algorithm to select tuning parameters that is (i) computationally efficient, (ii) agnostic to the choice of the sparse regression method, and (iii) asymptotically reliable. Our proposed algorithm, called PaTh, computes the solution path of a sparse regression method and thresholds a quantity computed at each point in the solution path. We prove that, under appropriate conditions, PaTh is *asymptotically tuning-free*, i.e., when the problem size becomes large (in the number of variables p and the number of observations n), PaTh reliably estimates the location of the non-zero entries in β^* independent of the choice of the threshold. We compare PaTh to algorithms in the literature that use the Lasso to jointly estimate β^* and the noise variance [23–25]. We compliment our theoretical results with numerical simulations and also demonstrate the potential benefits of using PaTh in finite-dimensional settings.

Appearing in Proceedings of the 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33. Copyright 2014 by the authors.

The rest of the paper is organized as follows. Section 2 formulates the sparse regression problem. Section 3 presents the PaTh algorithm. Section 4 proves the asymptotic tuning-free property of PaTh. Section 5 compares PaTh to scaled Lasso [25], which is similar to square-root Lasso [23]. Section 6 presents numerical simulations on real data. Section 7 summarizes the paper and outlines some future research directions.

2 Problem Formulation

In this section, we formulate the sparse linear regression problem. We assume that the observations $y \in \mathbb{R}^n$ and the measurement matrix $X \in \mathbb{R}^{n \times p}$ are known and related to each other by the linear model

$$y = X\beta^* + w\,, \tag{1}$$

where $\beta^* \in \mathbb{R}^p$ is the *unknown sparse regression vector* that we seek to estimate. We assume the following throughout this paper:

(A1) The matrix X is fixed with normalized columns, i.e., $\|X_i\|_2^2/n = 1$ for all $i \in \{1, 2, \ldots, p\}$.

(A2) The entries of w are i.i.d. zero-mean Gaussian random variables with variance σ^2.

(A3) The vector β^* is k-sparse with support set $S^* = \{j : \beta^*_j \neq 0\}$. Thus, $|S^*| = k$.

(A4) The number of observations n and the sparsity level k are all allowed to grow to infinity as the number of variables p grows to infinity. In the literature, this is referred to as the *high-dimensional framework.*

For any set S, we associate a loss function, $\mathcal{L}(S; y, X)$, which is the cost associated with estimating S^* by the set S. An appropriate loss function for the linear problem in (1) is the least-squares loss, which is defined as

$$\mathcal{L}(S; y, X) := \min_{\alpha \in \mathbb{R}^{|S|}} \|y - X_S\alpha\|_2^2 = \left\|\Pi^\perp[S]y\right\|_2^2\,, \tag{2}$$

where X_S is an $n \times |S|$ matrix that only includes the columns indexed by S and $\Pi^\perp[S] = I - \Pi[S] = I - X_S(X_S^T X_S)^{-1}X_S^T$ is the orthogonal projection onto the kernel of the matrix X_S.

In this paper, we mainly study the problem of estimating S^*, since, once S^* has been estimated, an estimate of β^* can be easily computed by solving a constrained least-squares problem. Our main goal is to devise an algorithm for estimating S^* that is *asymptotically tuning-free* so that when $p \to \infty$, S^* can be estimated with high probability without specifying a tuning parameter.

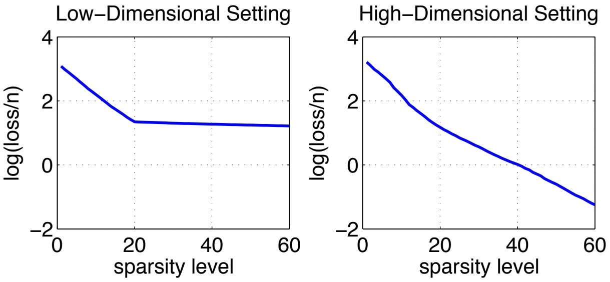

Figure 1: Plot of a function of the loss versus the sparsity level when using the forward-backward (FoBa) algorithm in the low-dimensional setting (left) and the high-dimensional setting (right).

3 Path Thresholding (PaTh)

In this section, we develop PaTh. Recall that we seek to estimate the support of a sparse vector β^* that is observed through y and X by the linear model in (1). Let Alg be a generic sparse regression algorithm with the following structure:

$$\begin{aligned} \widehat{S}_s &= \mathsf{Alg}(y, X, s) \\ y, X &= \text{Defined in (1)} \\ s &= \text{Desired sparsity level} \\ \widehat{S}_s &= \text{Estimate of } S^* \text{ s.t. } |\widehat{S}_s| = s\,. \end{aligned} \tag{3}$$

The tuning parameter in Alg is the sparsity level s, which is generally unknown. All standard sparse regression algorithms can be defined using Alg with appropriate modifications. For example, algorithms based on sparsity, including OMP [7], marginal regression [26,27], FoBa [9], and CoSaMP [28], can be written as (3). Algorithms based on real valued tuning parameters, such as the Lasso [6], can be written as (3) after mapping the tuning parameter to the sparsity level. We refer to Remark 3.3 for more details about this transformation.

3.1 Motivating Example

Before presenting PaTh, we discuss an example to illustrate the challenges in selecting tuning parameters for high-dimensional problems. Let $p = 1000$, $k = 20$, and $X_{ij} \sim \mathcal{N}(0, \sigma^2)$, where $\sigma = 2$. Furthermore, let the non-zero entries in β^* be sampled uniformly between $[0.5, 1.0]$. Figure 1 plots the $\log(\text{loss}/n)$ versus the sparsity level. The loss at sparsity level s is equal to $\|\Pi^\perp[\widehat{S}_s]y\|_2^2$ and $\widehat{S}_s$ is computed using the Forward-Backward (FoBa) sparse regression algorithm [9]. We refer to the sequence of estimates $\widehat{S}_1, \widehat{S}_2, \ldots, \widehat{S}_s$ as the *solution path* of FoBa. Furthermore, we consider a low-dimensional setting ($n = 2000$) and a high-dimensional setting ($n = 200$). In both settings, $\widehat{S}_k = S^*$, i.e., FoBa outputs the true support.

In the low-dimensional setting, we clearly see that Figure 1 has a visible change at the sparsity level $s = 20$. This suggests that the unknown sparsity level could be inferred by appropriately thresholding some quantity computed over the solution path of a sparse regression algorithm. However, in the high-dimensional setting, no such change is visible. Thus, it is not clear if the unknown sparsity could be detected from the solution path. As it turns out, we show in the next Section that an appropriate algorithm could be devised on the solution path to infer the sparsity level in an asymptotically reliable manner.

3.2 Overview of PaTh

Algorithm 1: Path Thresholding (PaTh)

```
Inputs: Observations $y$, measurement matrix $X$,
  and a parameter $c$
1 for $s = 0, 1, 2, \ldots, \min\{n, p\}$ do
2     $\widehat{S}_s \leftarrow \mathsf{Alg}(y, X, s)$
3     $\widehat{\sigma}_s^2 \leftarrow \|\Pi^\perp[\widehat{S}_s]y\|_2^2/n$
4     $\Delta_s \leftarrow \max_{j\in(\widehat{S}_s)^c} \left\{ \|\Pi^\perp[\widehat{S}_s]y\|_2^2 - \|\Pi^\perp[\widehat{S}_s \cup j]y\|_2^2 \right\}$
5     if $\Delta_s < 2c\widehat{\sigma}_s^2 \log p$ then
6         Return $\widehat{S} = \widehat{S}_s$.
```

Algorithm 1 presents path thresholding (PaTh) that uses the generic sparse regression algorithm Alg in (3) to estimate S^*. Besides y and X, the additional input to PaTh is a parameter c. We will see in Section 4 that as $p \to \infty$, under appropriate conditions, PaTh reliably identifies the true support as long as $c > 1$.

From Algorithm 1, it is clear that PaTh evaluates Alg for multiple different values of s, computes Δ_s defined in Line 4, and stops when Δ_s falls bellow a threshold. The quantity Δ_s is the *maximum* possible decrease in the loss when adding an additional variable to the support computed at sparsity level s. The threshold in Line 5 of Algorithm 1 is motivated from the following proposition.

Proposition 3.1. *Consider the linear model in (1) and assume that (A1)–(A4) holds. If $\widehat{S}_k = S^*$, then $\mathbb{P}(\Delta_k < 2c\sigma^2 \log p) \geq 1 - (p-k)/p^c$.*

Proof. Using simple algebra, Δ_s in Line 4 of Algorithm 1 can be written as

$$\Delta_s = \max_{j\in(\widehat{S}_s)^c} \frac{|X_j^T \Pi^\perp[\widehat{S}_s]y|^2}{\|\Pi^\perp[\widehat{S}_s]X_i\|_2^2} \,. \tag{4}$$

Under the conditions in the proposition, it is easy to see that $\Delta_k = \max_{j\in(S^*)^c} \|P_j w\|_2^2$, where P_j is a rank one projection matrix. The result follows from the Gaussian tail inequality and properties of projection matrices. □

Proposition 3.1 says that if σ were known, then with high probability, Δ_k could be upper bounded by $2c\sigma^2 \log p$, where c is some constant. Furthermore, under some additional conditions that ensure that $\Delta_s > 2c\sigma^2 \log p$ for $s < k$, Algorithm 1 could estimate the unknown sparsity level with high probability. For the marginal regression algorithm, the authors in [27] use a variant of this method to interchange the parameter s and σ^2.

Since σ is generally unknown, a natural alternative is to use an estimate of σ to compute the threshold. In PaTh, we use an estimate of σ computed from the loss at each solution of Alg, i.e., $\widehat{\sigma}_s^2 = \|\Pi^\perp[\widehat{S}_s]y\|_2^2/n$. Thus, for each s starting at $s = 0$, PaTh checks if $\Delta_s \leq 2c\widehat{\sigma}_s^2 \log n$ and then stops the first time the inequality holds (see Line 5). We can also use the estimate $\frac{\|\Pi^\perp[\widehat{S}_s]y\|_2^2}{n-s}$ with minimal change in performance since s is generally much smaller than n. Before illustrating PaTh using an example, we make some additional remarks regarding PaTh.

Remark 3.1. (Selecting the parameter c) In Section 4, we prove that the performance of PaTh is independent of c as $p \to \infty$ and $c > 1$. However, in finite-dimensional settings, we need to set c to an appropriate value. Fortunately, the choice of c is independent of the noise variance, which is *not* the case for popular sparse regression algorithms like the Lasso, the OMP, and the FoBa. In our numerical simulations, we observed that the performance of PaTh is insensitive to the choice of c as long as $c \in [0.5, 1.5]$.

Remark 3.2. (Computational Complexity) Besides computing the solution path of Alg, PaTh requires computing Δ_s, which involves taking a maximum over at most p variables. For each s, assuming that that the residual $\Pi^\perp[\widehat{S}_s]y$ and the projection matrix $\Pi^\perp[\widehat{S}_s]$ is computed by the sparse regression algorithm, the additional complexity of PaTh is $O(n^2 p)$. Furthermore, assuming that PaTh stops after computing $O(k)$ solutions, the total additional complexity of PaTh is $O(n^2 kp)$. This complexity can be reduced to $O(nkp)$, with no change in the theoretical results that we present in Section 4, by modifying the Δ_s computations so that $\Delta_s = \max_{j\in(\widehat{S}_s)^c} |X_j^T \Pi^\perp[\widehat{S}_s]y|^2$. This scaling of the additional complexity of PaTh is nearly the same as the complexity of computing the solution path of sparse regression algorithms. For example, assuming that $p > n$, the solution path of the Lasso can be computed in time $O(n^2 p)$ using the LARS algo-

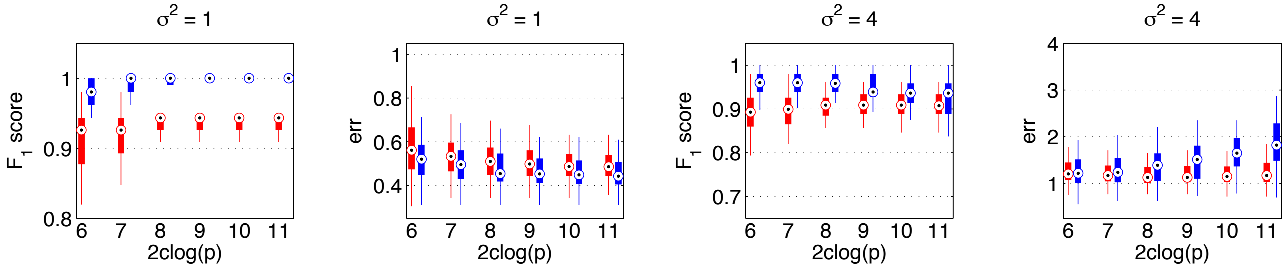

Figure 2: An illustration of the performance of PaTh when $X \in \mathbb{R}^{216\times 84}$ corresponds to stock returns data and $k = 30$. The figures are box plots of error measures over 100 trial. Red is Lasso and blue is FoBa. The first two plots show the F_1 score and the err when $\sigma^2 = 1$. The last two plots show the F_1 score and err when $\sigma^2 = 4$.

rithm [29].

Remark 3.3. (Generalizing Algorithm 1 to real valued tuning parameters) We present PaTh in Algorithm 1 using the sparsity level as a tuning parameter. However, several sparse regression algorithms, such as the Lasso, depend on a real-valued tuning parameter. One simple way to map such algorithms to the sparsity level is to compute the solution path, order the solutions in increasing level of sparsity, compute the loss at each solution, and finally select a solution at each sparsity level with minimal loss. The last step ensures that there is a unique solution for each sparsity level. We use this approach in all the implementations used in this paper. Alternatively, depending on the sparse regression algorithm, we can also directly apply PaTh to the solution path. For example, consider the Lasso that solves $\min_\beta[\|y - X\beta\|_2^2/(2n) + s\|\beta\|_1]$. As s decreases, $|\widehat{S}_s|$ generally increases. Thus, we can use Algorithm 1 with the Lasso by simply replacing Line 1 with "**for** $s = s_1, s_2, s_3, \ldots$", where $s_{i'} > s_{j'}$ for $i' < j'$.

3.3 Illustrative Example

In this section, we illustrate the performance of PaTh. We use data from [30] composed of 216 observations of monthly stock returns from 84 companies. This results in a matrix X. We let $k = 30$ and select a β^* such that the non-zero entries in β^* are uniformly distributed between 0.5 and 1.5. We simulate two sets of observations; one using $\sigma^2 = 1$ and another one using $\sigma^2 = 4$. We apply PaTh using Lasso [6] and FoBa [9]. To evaluate the performance of an estimate $\widehat{\beta}$ with support $\widehat{S}$, we compute the F_1 score and the error in estimating β^*:

$$F_1 \text{ score} = 1/(1/\text{Recall} + 1/\text{Precision})\,, \tag{5}$$

$$\text{Precision} = |S^* \cap \widehat{S}|/|\widehat{S}|\,, \tag{6}$$

$$\text{Recall} = |S^* \cap \widehat{S}|/|S^*|\,, \tag{7}$$

$$\text{err} = \|\widehat{\beta} - \beta^*\|_2\,. \tag{8}$$

Naturally, we want F_1 to be large (close to 1) and err to be small (close to 0). Figure 2 shows box plots of the error measures, where red corresponds to the Lasso and blue corresponds to FoBa. The horizontal axis in the plots refer to the different choices of the threshold $2c\log p$, where $c \in [0.6, 1.3]$. For σ small, we clearly see that the threshold has little impact on the final estimate. For larger σ, we notice some differences, mainly for FoBa, as the threshold is varied. Overall, this example illustrates that PaTh can narrow down the choices of the final estimate of β^* or S^* to a few estimates in a computationally efficient manner. In the next section, we show that when p is large, PaTh can accurately identify β^* and S^*.

4 Tuning-Free Property of PaTh

In this section, we prove that, under appropriate conditions, PaTh is asymptotically tuning-free. We state our result in terms of the generic sparse regression algorithm Alg defined in (3). For a constant $c > 1$, we assume that Alg has the following property:

(A5) Alg can reliably estimate the true support S^*, i.e., $\mathbb{P}(S^* = \mathsf{Alg}(y, X, k)) \geq 1 - 1/p^c$, where c is the input to PaTh.

Assumption (A5), which allows us to separate the analysis of PaTh from the analysis of Alg, says that Alg outputs the true support for $s = k$ with high probability. Under appropriate conditions, this property holds for all sparse regression algorithms under various conditions [12]. The next theorem is stated in terms of the following two parameters:

$$\beta_{\min} = \min_{i \in S^*} |\beta_i|\,, \tag{9}$$

$$\rho_{2k} = \min\left\{\frac{\|X_A v\|_2^2}{n\|v\|_2^2} : S^* \subseteq A, v \in \mathbb{R}^{2k}\right\}. \tag{10}$$

The parameter $\beta_{\min}$ is the minimum absolute value over the non-zero entries in β^*. The parameter ρ_{2k}, referred to as the restricted eigenvalue (RE), is the minimum eigenvalue over certain blocks of the matrix $X^T X/n$.

Theorem 4.1. *Consider the linear model in (1) and assume that (A1)–(A4) holds. Let $\widehat{S}$ be the output of* PaTh *used with the sparse regression method* Alg *defined in (3) that satisfies (A5). Select $\epsilon > k/n + \sqrt{1/k}$ and $c > 1/(1-\epsilon)$. For some constant $C > 0$, if*

$$n \geq \frac{2ck\log p}{\rho_{2k}} + \frac{8\sigma c\sqrt{k}\log p}{\beta_{\min}\rho_{2k}^2} + \frac{8\sigma^2 c\log p}{\beta_{\min}^2\rho_{2k}^2}\,, \qquad (11)$$

then $\mathbb{P}(\widehat{S} = S^*) \geq 1 - Cp^{1-(1-\epsilon)c}$.

The proof of Theorem 4.1, given in Appendix A, simply identifies sufficient conditions under which $\Delta_s \geq 2c\log p$ for $s < k$ and $\Delta_k < 2c\log p$. We now make some remarks regarding Theorem 4.1.

Remark 4.1. (Tuning-Free Property) Recall from (A4) that $k, n \to \infty$ as $p \to \infty$. Furthermore, based on the conditions on n in (11), it is clear that $\lim_{p\to\infty}(k/n + \sqrt{1/k}) = 0$. This means that for any $c > 1$, if (11) holds, then $\lim_{p\to\infty}\mathbb{P}(\widehat{S} = S^*) \to 1$. This shows that PaTh is *asymptotically tuning-free.*

Remark 4.2. (Numerical Simulation) To illustrate the tuning-free property of PaTh, we consider a simple numerical example. Let $p = 1000$, $k = 50$, and $\sigma = 1$. Assume that the non-zero entries in β^* are drawn from a uniform distribution on $[1, 2]$. Next, randomly assign each non-zero entry either a positive or a negative sign. The rows of the measurement matrix X are drawn i.i.d. from $\mathcal{N}(0, \Sigma)$. We consider two cases: $\Sigma = I$ and $\Sigma = 0.8I + 0.2\mathbf{1}\mathbf{1}^T$, where $\mathbf{1}$ is a vector of ones. For both cases, we use PaTh with Lasso and FoBa. Figure 3 plots the mean F_1 score, defined in (5), and mean log(err), defined in (8), over 100 trials as n ranges from 200 to 1000. Note that $F_1 = 1$ corresponds to accurate support recovery. For both Lasso and FoBa, we used PaTh with $c = 1$ and $c = 1.5$. We clearly see that once n is large enough, both algorithms, with different choices of c, lead to accurate support recovery. This naturally implies accurate estimation of β^*.

Remark 4.3. (Superset Recovery) We have analyzed PaTh for reliable support recovery. If Alg can only output a superset of S^*, which happens when using the Lasso under the restricted strong convexity condition [31,32], then (A5) can be modified and the statement of Theorem 2 can be modified to reflect superset recovery. The only change in (11) will be to appropriately modify the definition of ρ_{2k} in (10).

Remark 4.4. (Extension to sub-Gaussian noise) Our analysis only uses tail bounds for Gaussian and chi-squared random variables. Hence, we can easily extend our analysis to sub-Gaussian noise using tail bounds for sub-Gaussian and sub-exponential random variables in [33].

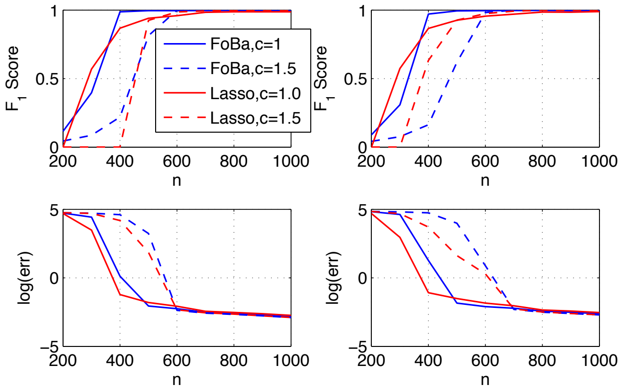

Figure 3: Mean F_1 score and mean log(err) over 100 trials when using PaTh with Lasso and FoBa with $p = 1000$, $k = 10$, $\sigma = 1$, and $\beta_{\min} \geq 1$. (left) X is drawn from $\mathcal{N}(0, I)$ (right) X is drawn from $\sim \mathcal{N}(0, 0.8I + 0.2\mathbf{1}\mathbf{1}^T)$.

Remark 4.5. (Scaling of n) For an appropriate parameter c_2 that depends on ρ_{2k}, σ, and $\beta_{\min}$, (11) holds if $n > c_2 k\log p$. When using the Lasso, under appropriate conditions, in the most general case, $n > c_2' k\log p$ is sufficient for reliable support recovery. Thus, both the Lasso and the PaTh require the same scaling on the number of observations. An open question is to analyze the tightness of the condition (11) and see if the dependence on k in (11) can be improved. One way may be to incorporate prior knowledge that $k \geq k_{\min}$. In this case, the scaling in (11) can be improved to $n > c_2(k - k_{\min})\log p$.

5 Connections to Scaled Lasso

In this Section, we highlight the connections between PaTh and the scaled Lasso algorithm proposed in [25]. Let $\widehat{\sigma}$ be an initial estimate of the noise variance. Consider the following computations on the solution path until an equilibrium is reached:

$$t \leftarrow \min_s\{s : \Delta_s < 2c\widehat{\sigma}^2\log p\} \qquad (12)$$

$$\widehat{S}_t \leftarrow \mathsf{Alg}(y, X, t) \qquad (13)$$

$$\widehat{\sigma}_{new} \leftarrow \|\Pi^{\perp}[\widehat{S}_t]y\|/\sqrt{n} \qquad (14)$$

$$\text{If } \widehat{\sigma}_{new} \neq \widehat{\sigma}, \text{ then let } \widehat{\sigma} \leftarrow \widehat{\sigma}_{new} \text{ and go to (12).} \qquad (15)$$

The next theorem shows that, under an additional condition on the solution path, (12)-(15) is equivalent to implementing PaTh.

Theorem 5.1. *Suppose $\widehat{\sigma}_0 > \widehat{\sigma}_1 > \cdots > \widehat{\sigma}_{n-1} > \widehat{\sigma}_n$, where $\widehat{\sigma}_s = \|\Pi^{\perp}[\widehat{S}_s]y\|_2^2/n$. If $\widehat{\sigma} = \widehat{\sigma}_0 = \|y\|_2/\sqrt{n}$, then the output of* PaTh *is equal to the output of (12)-(15).*

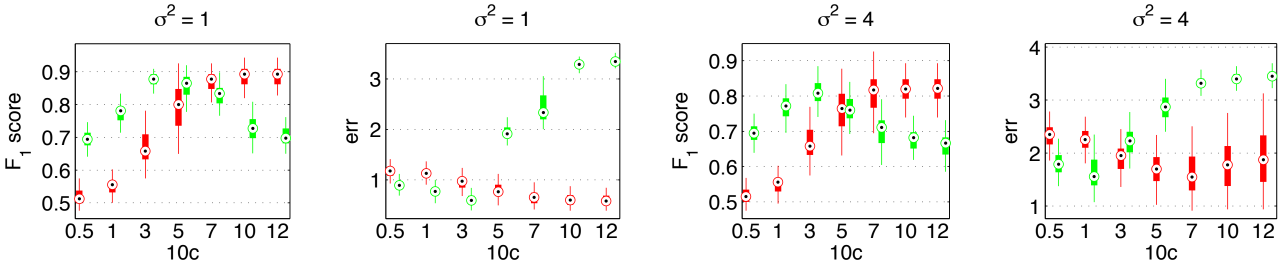

Figure 4: A comparison of Lasso (blue) to scaled lasso (green). See Figure 2 for details about the setup.

Proof. Let $\widehat{S}$ be the output of PaTh and $\widehat{T}$ be the output of (12)-(15). From the properties of PaTh, we know that $\Delta_s > 2c\widehat{\sigma}_s^2 \log n$ for all $s < |\widehat{S}|$. Furthermore, we know that $\widehat{\sigma}_1 > \widehat{\sigma}_2 > \cdots > \widehat{\sigma}_s$. This means that $\Delta_s > 2c\widehat{\sigma}_t^2 \log n$ for any $t \leq s < |\widehat{S}|$. Thus, (12)–(15) reaches an equilibrium when $\widehat{T} = \widehat{S}$. □

The condition in Theorem 5.1 ensures that the loss decreases as the tuning parameter s in Alg increases. This condition easily holds for sparse regression algorithms, including the OMP and the FoBa, that are based on greedy methods.

Interestingly, (12)–(15) resemble the scaled Lasso algorithm [25]. In particular, scaled Lasso starts with an initial estimate of σ, selects an appropriate tuning parameter, selects an estimate from the solution path of Lasso, and repeats until an equilibrium is reached. Furthermore, as shown in [25], scaled Lasso solves the following problem:

$$(\widehat{\beta}, \widehat{\sigma}) = \operatorname*{argmin}_{\beta \in \mathbb{R}^p, \sigma > 0} \left[\frac{\|y - X\beta\|_2^2}{2n\sigma} + \frac{\sigma}{2} + \lambda \|\beta\|_1 \right]. \quad (16)$$

The theoretical results in [25] show that, under appropriate restricted eigenvalue conditions, selecting $\lambda_0 = \sqrt{2c \log p/n}$, where c is a constant, results in accurate estimation of β^*. This choice of λ_0 comes from the upper bound of the random variable $\|X^T w\|_\infty / (\sigma n)$. The motivation for scaled Lasso grew out of the results in [24] and related discussions in [34,35]. Furthermore, as shown in [36], scaled Lasso is equivalent to square-root Lasso [23].

PaTh can be seen as a solution to the more general problem

$$(\widehat{\beta}, \widehat{\sigma}) = \operatorname*{argmin}_{\beta \in \mathbb{R}^p, \|\beta\|_0 \leq s, \sigma > 0} \left[\frac{\|y - X\beta\|_2^2}{2n\sigma} + \frac{\sigma}{2} \right], \quad (17)$$

where $\|\beta\|_0$ is the ℓ_0-norm that counts the number of non-zero entries in β and the parameter s is implicitly related to the threshold $2c \log p$ in Line 4 of PaTh (Algorithm 1). The advantage of PaTh is that it can be used with *any* sparse regression algorithm and the choice of the threshold does not depend on the measurement matrix.

We now empirically compare scaled Lasso (SL) to PaTh used with Lasso (PL). In particular, we want to compare the sensitivity of both methods to the choice of the parameter $\lambda_0 = \sqrt{2c \log p/n}$ in SL and the choice of the parameter $2c \log n$ in PL. We consider the same setting as in Figure 2 (see Section 3.3 for the details). Figure 4 shows the box plots for the F_1 score and err for PL (in red) and SL (in green) as c ranges from 0.05 to 1.2. It is clear that there exists a parameter c for both SL and PL such that their performance is nearly equal. For example, in the first plot, the F_1 score of SL at $c = 0.5$ is nearly the same as the F_1 score of PL at $c = 1.0, 1.2$. Furthermore, it is clear from the plot that PL is relatively insensitive to the choice of c when compared to SL. This suggests the possible advantages of using PaTh with Lasso over the scaled Lasso estimator.

6 Application to Real Data

In this Section, we demonstrate the advantages of using PaTh on real data. Although, we have shown that PaTh is asymptotically tuning-free, this may not be true in finite-dimensional settings. Instead, our main goal is to demonstrate how PaTh can significantly reduce the possible number of solutions for various regression problems by reparametrizing the sparse regression problem in terms of the parameter c in PaTh. We consider the following two real data sets:

- UCI Communities and Crime Data [37,38]: This data set contains information about the rate of violent crimes from 1994 communities. The goal is to find a sparse set of attributes, from the 122 given attributes, that best predict the rate of violent crimes. We removed 22 attributes due to insufficient data, randomly selected 100 communities, and normalized the columns to get a matrix $X \in \mathbb{R}^{100 \times 100}$ and observations $y \in \mathbb{R}^{100}$.
- Prostate cancer data [39]: This data set contains information about gene expression values of 12533

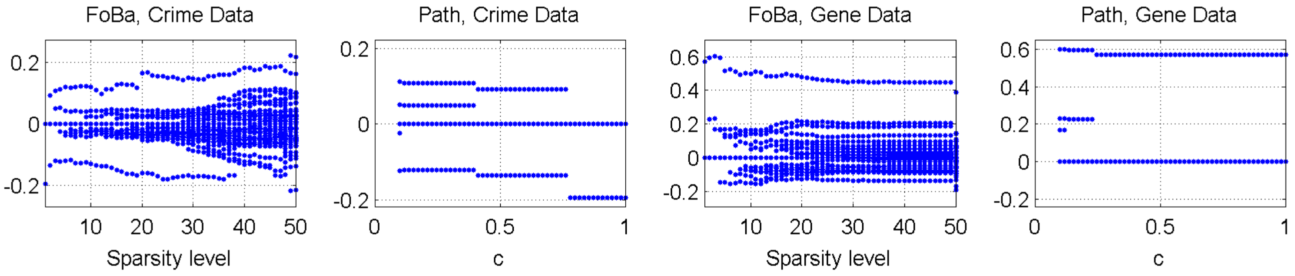

Figure 5: Comparison of the solution paths of FoBa and PaTh+ FoBa. The first figure shows the solution path of FoBa applied to the crime data. The horizontal axis specifies the sparsity level and the vertical axis specifies the coefficient values. The second figure applies PaTh to the solution path in the first figure for 50 different values of c in the range $[0.1, 1]$. PaTh reduces the total number of solutions from 50 to 4. We observe similar trends for the gene data (last two figures).

genes from 50 patients with prostate cancer. We consider the problem of finding relationships between the genes. For this, we randomly select the expression values from one gene, say $y \in \mathbb{R}^{50}$, and regress on the remaining gene expression values $X \in \mathbb{R}^{50\times 12532}$.

Figure 5 presents our results on applying FoBa [9] to both the data sets. As detailed in the caption of Figure 5, PaTh reduces the number of solutions from 50 to 4 for each data set without tuning any parameter. Subsequently, cross-validation or stability selection [22] can be used to further prune the set of estimates. This post-processing stage will require far less computations than applying cross-validation or stability selection without using PaTh to reduce the number of solutions.

7 Conclusions

We have proposed a new computationally efficient algorithm, called path thresholding (PaTh), for selecting tuning parameters in any high-dimensional sparse regression method. Our main contribution shows that (i) PaTh is agnostic to the choice of the sparse regression method, and (ii) PaTh, under appropriate conditions, selects the optimal tuning parameter with high probability as the problem size grows large. Thus, using PaTh with any tuning-dependent regression algorithm leads to an asymptotically tuning-free sparse regression algorithm. In finite-dimensional settings, we have shown that PaTh can drastically reduce the possible number of solutions of a sparse regression problem. Thus, PaTh can be used to significantly reduce the computational costs associated with cross-validation and stability selection.

Our work motivates several avenues of future research. For example, it will be interesting to explore the use of PaTh for estimating approximately sparse vectors. Furthermore, it will also be useful to study the use of PaTh in asymptotically tuning-free learning of graphical models.

Acknowledgment

We thank Prof. Dennis Cox and Ali Mousavi for feedback that improved the quality of the paper. This work was supported by the Grants NSF IIS-1124535, CCF-0926127, CCF-1117939; ONR N00014-11-1-0714, N00014-10-1-0989; and ARO MURI W911NF-09-1-0383.

A Proof of Theorem 4.1

We want to show that PaTh stops when $s = k$, where recall that k is the unknown sparsity level of the sparse vector β^* that we seek to estimate. For this to happen, it is sufficient to find conditions under which the following two equations are true:

$$\Delta_s \geq \widehat{\sigma}_s \tau_p \,, \text{whenver } s < k \,, \tag{18}$$

$$\Delta_k < \widehat{\sigma}_k \tau_p \,, \tag{19}$$

where $\tau_p = 2c\log p$, $\widehat{\sigma}^2_s = \|\Pi^\perp[\widehat{S}_s]y\|_2^2/n$, and Δ_s is defined in Line 4 of Algorithm 1. It is clear that if (18) holds, then Algorithm 1 will not stop for $s < k$. Furthermore, if (19) holds, then Algorithm 1 stops when $s = k$. The rest of the proof is centered around finding conditions under which both (18) and (19) hold.

Finding conditions under which (18) holds. Using (4), we can lower bound Δ_s as follows:

$$\begin{aligned}\Delta_s &= \max_{j\in(\widehat{S}_s)^c} \frac{|X_j^T \Pi^\perp[\widehat{S}_s]y|^2}{\|\Pi^\perp[\widehat{S}_s]X_j\|_2^2} \overset{(a)}{\geq} \frac{1}{n} \max_{j \in S^*\backslash\widehat{S}_s} |X_j^T \Pi^\perp[\widehat{S}_s]y|^2 \\ &\overset{(b)}{=} \frac{1}{n}\left\| X^T_{S^*\backslash\widehat{S}_s} \Pi^\perp[\widehat{S}_s]y \right\|_\infty^2 \\ &\overset{(c)}{\geq} \frac{1}{k_s n}\left\| X^T_{S^*\backslash\widehat{S}_s} \Pi^\perp[\widehat{S}_s]y \right\|_2^2 , \quad k_s = |S^*\backslash\widehat{S}_s| \\ &\overset{(d)}{\geq} \frac{1}{k_s n}\left[\|\mathcal{R}\|_2^2 - 2\left|(\mathcal{R})^T w\right| \right] .\end{aligned} \tag{20}$$

Step (a) restricts the size over which the maximum is taken and uses the equation $\|\Pi^\perp[\widehat{S}_s]X_j\|_2^2 \leq n$. Step (b) uses the ℓ_∞-norm notation. Step (c) uses the fact that $\|v\|_\infty^2 \geq \|v\|_2^2/k_s$ for any $k_s \times 1$ vector v. Step (d) uses (1) and also defines the notation

$$\mathcal{R} = X_{S^*\backslash\widehat{S}_s}^T \Pi^\perp[\widehat{S}_s] X_{S^*\backslash\widehat{S}_s} \beta^*_{S^*\backslash\widehat{S}_s} \,. \tag{21}$$

Next, we use (1) to evaluate $\widehat{\sigma}_s^2$:

$$\begin{aligned} n\widehat{\sigma}_s^2 = \|\Pi^\perp[\widehat{S}_s]y\|_2^2 &\leq \|\Pi^\perp[\widehat{S}_s]X\beta^*\|_2^2 + \|\Pi^\perp[\widehat{S}_s]w\|_2^2 \\ &+ 2|(\Pi^\perp[\widehat{S}_s]X\beta^*)^T w| \,. \end{aligned} \tag{22}$$

Using (20) and (22), (18) holds if the following equation holds for $s < k$:

$$\begin{aligned} &\frac{\|\mathcal{R}\|_2^2}{\tau_p k_s} - \|\Pi^\perp[\widehat{S}_s]X\beta^*\|_2^2 \\ &\geq \frac{2\left|(\mathcal{R})^T w\right|}{\tau_p k_s} + \|\Pi^\perp[\widehat{S}_s]w\|_2^2 + 2|(\Pi^\perp[\widehat{S}_s]X\beta^*)^T w| \,. \end{aligned} \tag{23}$$

We now find a lower bound for the left hand side (LHS) of (23) and an upper bound for the right hand side (RHS) of (23). Using (21), we have

$$\|\mathcal{R}\|_2^2 = \left(\beta^*_{S^*\backslash\widehat{S}_s}\right)^T \left(X_{S^*\backslash\widehat{S}_s}^T \Pi^\perp[\widehat{S}_s] X_{S^*\backslash\widehat{S}_s}\right)^2 \beta^*_{S^*\backslash\widehat{S}_s} \,.$$

Let $0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{k_s}$ be the eigenvalues of $X_{S^*\backslash\widehat{S}_s}^T \Pi^\perp[\widehat{S}_s] X_{S^*\backslash\widehat{S}_s}$. Then, for a unitary matrix M and a diagonal matrix D, where $D_{ii} = \lambda_i$, $X_{S^*\backslash\widehat{S}_s}^T \Pi^\perp[\widehat{S}_s] X_{S^*\backslash\widehat{S}_s} = M^T D M$. If $\xi = M\beta^*_{S^*\backslash\widehat{S}_s}$, then the LHS of (23) can be written as

$$\begin{aligned} \frac{\xi^T D^2 \xi}{\tau_p k_s} - \xi^T D \xi &= \sum_{i=1}^{k_s} \left[\frac{\lambda_i^2}{\tau_p k_s} - \lambda_i\right] \xi_i^2 \\ &\geq \left[\frac{n^2 \rho_{2k}^2}{\tau_p k_s} - n\rho_{2k}\right] \|\beta^*_{S^*\backslash\widehat{S}_s}\|_2^2 \,, \end{aligned} \tag{24}$$

where we use the assumption that $n\rho_{2k} > \tau_p k_s$ and the fact that $\lambda_1 > n\rho_{2k}$, where ρ_{2k} is defined in (10).

Next, we find an upper bound for the RHS in (23). To do so, we make use of standard results for Gaussian random vectors. In particular, we have that for any vector v,

$$\mathbb{P}(|v^T w| > \sigma\|v\|_2\sqrt{\tau_p}) \leq 2e^{-\tau_p/2} \,, \tag{25}$$

$$\begin{aligned} &\mathbb{P}(\|\Pi^\perp[\widehat{S}_s]w\|_2^2/\sigma^2 \geq n - s + \sqrt{(n-s)\tau_p/2} + \tau_p) \\ &\qquad \leq e^{-\tau_p/2}. \end{aligned} \tag{26}$$

Using (25) and (26), we can upper bound the RHS in (23) using the following equation with probability at least $1 - 3e^{-\tau_p/2}$:

$$\begin{aligned} &\frac{2\sigma\|\mathcal{R}\|_2\sqrt{\tau_p}}{\tau_p k_s} + 2\sigma 1\|\Pi^\perp[\widehat{S}_s]X\beta^*\|_2\sqrt{\tau_p} \\ &\qquad + \sigma^2\left(n - s + \sqrt{(n-s)\tau_p/2} + \tau_p\right) \end{aligned} \tag{27}$$

Next, note that $\|\mathcal{R}\|_2 \leq k_s n \|\beta^*_{S^*\backslash\widehat{S}_s}\|_2$ and $\|\Pi^\perp[\widehat{S}_s]X\beta^*\|_2 \leq \sqrt{nk_s}\|\beta^*_{S^*\backslash\widehat{S}_s}\|_2$. Furthermore, choosing $\tau_p/2 < n$, (27) can be upper bounded by

$$\frac{2\sigma n\|\beta^*_{S^*\backslash\widehat{S}_s}\|_2\sqrt{\tau_p}}{\tau_p} + 2\sigma\sqrt{k_s n}\|\beta^*_{S^*\backslash\widehat{S}_s}\|_2\sqrt{\tau_p} + 4\sigma^2 n \,.$$

Rearranging terms, using $\|\beta^*_{S^*\backslash\widehat{S}_s}\|_2^2 \geq k_s\beta_{\min}^2$, $n \geq \tau_p k_s$, and $k_s \leq k$, (23) holds with probability at least $1 - 3e^{-\tau_p/2}$ if the following holds:

$$n \geq \frac{k\tau_p}{\rho_{2k}} + \frac{4\sigma^2\tau_p}{\beta_{\min}^2\rho_{2k}^2} + \frac{4\sigma\sqrt{k}\tau_p}{\beta_{\min}\rho_{2k}^2} \,. \tag{28}$$

Finding conditions under which (19) holds. Assuming that $\widehat{S}_k = S^*$ and using the definition of Δ_k and $\widehat{\sigma}_k$, we can write (19) as

$$\max_{j\in(S^*)^c} \|P_i w\|_2^2/\sigma^2 < \|\Pi^\perp[S^*]w\|_2^2\tau_p/(n\sigma^2) \,, \tag{29}$$

where P_i is a rank one projection matrix. Note that $\|P_i w\|_2^2/\sigma^2$ is the square of a $\mathcal{N}(0,1)$ random variable. Using the Gaussian tail ineqaulity, we have that

$$\mathbb{P}\left(\max_{j\in(S^*)^c} \|P_i w\|_2^2/\sigma^2 \geq \nu_p\right) \leq 2(p-k)e^{-\nu_p/2}/\sqrt{\nu_p} \,.$$

Moreover, using standard bounds for chi-square random variables, we have that

$$\mathbb{P}\left(\|\Pi^\perp[S^*]w\|_2^2/\sigma^2 \leq n - k - 2\sqrt{(n-k)\alpha_p}\right) \leq e^{-\alpha_p} \,.$$

Thus, (19) holds with probability at least $1 - 3(p-k)e^{-\nu_p/2}/\sqrt{\nu_p}$ if the following is true

$$\frac{\nu_p}{\tau_p} < 1 - \frac{k}{n} - 2\frac{\sqrt{(n-k)\nu_p/2}}{n}$$

Let $\nu_p = 2(1-\epsilon)c\log p$. Then, the above conditions holds if $\epsilon > \frac{k}{n} + \sqrt{\frac{2c\log p}{n}}$. Since $n > 2ck\log p$, if $\epsilon > k/n + \sqrt{1/k}$, then (19) holds with probability at least $1 - 3p^{1-(1-\epsilon)c}/\sqrt{2(1-\epsilon)c\log p}$.

Combining both results, and using (A5), we have that under the conditions stated in the theorem, for some constant $C > 0$, $\mathbb{P}(\widehat{S} = S^*) \geq 1 - Cp^{1-(1-\epsilon)c}$.

References

[1] M. Segal, K. Dahlquist, and B. Conklin, "Regression approaches for microarray data analysis," *Journal of Computational Biology*, vol. 10, no. 6, pp. 961–980, 2003.

[2] G. Varoquaux, A. Gramfort, and B. Thirion, "Small-sample brain mapping: sparse recovery on spatially correlated designs with randomization and clustering," in *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, 2012, pp. 1375–1382.

[3] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91, Mar. 2008.

[4] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[5] N. Meinshausen and B. Yu, "Lasso-type recovery of sparse representations for high-dimensional data," *Annals of Statistics*, pp. 246–270, 2009.

[6] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1996.

[7] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[8] H. Zou, "The adaptive Lasso and its oracle properties," *Journal of the American Statistical Association*, vol. 101, pp. 1418–1429, December 2006.

[9] T. Zhang, "Adaptive forward-backward greedy algorithm for learning sparse representations," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4689–4708, 2011.

[10] D. Vats and R. G. Baraniuk, "When in doubt, SWAP: High-dimensional sparse recovery from correlated measurements," in *Advances in Neural Information Processing Systems*, 2013.

[11] D. Vats and R. G. Baraniuk, "Swapping variables for high-dimensional sparse regression with correlated measurements," *arXiv preprint arXiv:1312.1706*, 2013.

[12] P. Bühlmann and S. Van De Geer, *Statistics for High-Dimensional Data: Methods, Theory and Applications*, Springer-Verlag New York Inc, 2011.

[13] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[14] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[15] S. Geisser, "The predictive sample reuse method with applications," *Journal of the American Statistical Association*, vol. 70, no. 350, pp. 320–328, 1975.

[16] D. Homrighausen and D. McDonald, "The lasso, persistence, and cross-validation," in *Proceedings of The 30th International Conference on Machine Learning*, 2013, pp. 1031–1039.

[17] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[18] H. Zou, T. Hastie, and R. Tibshirani, "On the "degrees of freedom" of the lasso," *Annals of Statistics*, vol. 35, no. 5, pp. 2173–2192, 2007.

[19] Y. C. Eldar, "Generalized SURE for exponential families: Applications to regularization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 471–481, 2009.

[20] L. Wasserman and K. Roeder, "High dimensional variable selection," *Annals of statistics*, vol. 37, no. 5A, pp. 2178, 2009.

[21] T. Zhang, "Analysis of multi-stage convex relaxation for sparse regularization," *Journal of Machine Learning Research*, vol. 11, pp. 1081–1107, Mar. 2010.

[22] N. Meinshausen and P. Bühlmann, "Stability selection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 4, pp. 417–473, 2010.

[23] A. Belloni, V. Chernozhukov, and L. Wang, "Square-root Lasso: pivotal recovery of sparse signals via conic programming," *Biometrika*, vol. 98, no. 4, pp. 791–806, 2011.

[24] N. Städler, P. Bühlmann, and S. Van De Geer, "ℓ_1-penalization for mixture regression models," *Test*, vol. 19, no. 2, pp. 209–256, 2010.

[25] T. Sun and C.-H. Zhang, "Scaled sparse linear regression," *Biometrika*, vol. 99, no. 4, pp. 879–898, 2012.

[26] J. Fan and J. Lv, "Sure independence screening for ultrahigh dimensional feature space," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 5, pp. 849–911, 2008.

[27] C. Genovese, J. Jin, L. Wasserman, and Z. Yao, “A comparison of the lasso and marginal regression,” *Journal of Machine Learning Research*, vol. 13, pp. 2107–2143, 2012.

[28] D. Needell and J. A. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples,” *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

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

[30] M. J. Choi, V. Y. Tan, A. Anandkumar, and A. S. Willsky, “Learning latent tree graphical models,” *Journal of Machine Learning Research*, vol. 12, pp. 1771–1812, 2011.

[31] P. J. Bickel, Y. Ritov, and A. B. Tsybakov, “Simultaneous analysis of Lasso and Dantzig selector,” *Annals of Statistics*, vol. 37, no. 4, pp. 1705–1732, 2009.

[32] S. Negahban, P. Ravikumar, M. Wainwright, and B. Yu, “A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers,” *Statistical Science*, vol. 27, no. 4, pp. 538–557, 2012.

[33] R. Vershynin, *Compressed Sensing: Theory and Applications*, chapter Introduction to the non-asymptotic analysis of random matrices, Cambridge University Press, 2010.

[34] T. Sun and C.-H. Zhang, “Comments on: ℓ_1-penalization for mixture regression models,” *Test*, vol. 19, no. 2, pp. 270–275, 2010.

[35] A. Antoniadis, “Comments on: ℓ_1-penalization for mixture regression models,” *Test*, vol. 19, no. 2, pp. 257–258, 2010.

[36] C. Giraud, S. Huet, and N. Verzelen, “High-dimensional regression with unknown variance,” *Statistical Science*, vol. 27, no. 4, pp. 500–518, 2012.

[37] M. Redmond and A. Baveja, “A data-driven software tool for enabling cooperative information sharing among police departments,” *European Journal of Operational Research*, vol. 141, no. 3, pp. 660–678, 2002.

[38] K. Bache and M. Lichman, “UCI machine learning repository,” 2013.

[39] `http://www.biolab.si/supp/bi-cancer/projections/info/prostata.htm`, ,” Accessed May 31, 2013.