No penalty no tears: Least squares in high-dimensional linear models

Xiangyu Wang XW56@STAT.DUKE.EDU
Department of Statistical Science, Duke University

David Dunson DUNSON@STAT.DUKE.EDU
Department of Statistical Science, Duke University

Chenlei Leng C.LENG@WARWICK.AC.UK
Department of Statistics, University of Warwick

Abstract

Ordinary least squares (OLS) is the default method for fitting linear models, but is not applicable for problems with dimensionality larger than the sample size. For these problems, we advocate the use of a generalized version of OLS motivated by ridge regression, and propose two novel three-step algorithms involving least squares fitting and hard thresholding. The algorithms are methodologically simple to understand intuitively, computationally easy to implement efficiently, and theoretically appealing for choosing models consistently. Numerical exercises comparing our methods with penalization-based approaches in simulations and data analyses illustrate the great potential of the proposed algorithms.

1. INTRODUCTION

Long known for its consistency, simplicity and optimality under mild conditions, ordinary least squares (OLS) is the most widely used technique for fitting linear models. Developed originally for fitting fixed dimensional linear models, unfortunately, classical OLS fails in high dimensional linear models where the number of predictors p far exceeds the number of observations n. To deal with this problem, Tibshirani (1996) proposed ℓ_1-penalized regression, a.k.a, *lasso*, which triggered the recent overwhelming exploration in both theory and methodology of penalization-based methods. These methods usually assume that only a small number of coefficients are nonzero (known as the *sparsity* assumption), and minimize the same least squares loss function as OLS by including an additional penalty on the coefficients, with the typical choice being the ℓ_1 norm. Such "penalization" constrains the solution space to certain directions favoring sparsity of the solution, and thus overcomes the non-unique issue with OLS. It yields a sparse solution and achieves model selection consistency and estimation consistency under certain conditions. See Zhao and Yu (2006); Fan and Li (2001); Zhang (2010); Zou and Hastie (2005).

Despite the success of the methods based on regularization, there are important issues that can not be easily neglected. On the one hand, methods using convex penalties, such as *lasso*, usually require strong conditions for model selection consistency (Zhao and Yu, 2006; Lounici, 2008). On the other hand, methods using non-convex penalties (Fan and Li, 2001; Zhang, 2010) that can achieve model selection consistency under mild conditions often require huge computational expense. These concerns have limited the practical use of regularized methods, motivating alternative strategies such as direct hard thresholding (Jain et al., 2014).

In this article, we aim to solve the problem of fitting high-dimensional sparse linear models by reconsidering OLS and answering the following simple question: Can ordinary least squares consistently fit these models with some suitable algorithms? Our result provides an affirmative answer to this question under fairly general settings. In particular, we give a generalized form of OLS in high dimensional linear regression, and develop two algorithms that can consistently estimate the coefficients and recover the support. These algorithms involve least squares type of fitting and hard thresholding, and are non-iterative in nature. Extensive empirical experiments are provided in Section 4 to compare the proposed estimators to many existing penalization methods. The performance of the new estimators is very competitive under various setups in terms of model selection, parameter estimation and computational time.

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

1.1. Related Works

The work that is most closely related to ours is Yang et al. (2014), in which the authors proposed an algorithm based on OLS and ridge regression. However, both their methodology and theory are still within the ℓ_1 regularization framework, and their conditions (especially their C-Ridge and C-OLS conditions) are overly strong and can be easily violated in practice. Jain et al. (2014) proposed an iterative hard thresholding algorithm for sparse regression, which shares a similar spirit of hard thresholding as our algorithm. Nevertheless, their motivation is completely different, their algorithm lacks theoretical guarantees for consistent support recovery, and they require an iterative estimation procedure.

1.2. Our Contributions

We provide a generalized form of OLS for fitting high dimensional data motivated by ridge regression, and develop two algorithms that can consistently fit linear models on weakly sparse coefficients. We summarize the advantages of our new algorithms in three points.

Our algorithms work for highly correlated features under random designs. The consistency of the algorithms relies on a moderately growing conditional number, as opposed to the strong irrepresentable condition (Zhao and Yu, 2006; Wainwright, 2009) required by *lasso*. Our algorithms can achieve consistent identify strong signals for ultra-high dimensional data ($\log p = o(n)$) with only a bounded variance assumption on the noise ε, i.e., $var(\varepsilon) < \infty$. This is remarkable as most methods (c.f. Zhang (2010); Yang et al. (2014); Cai and Wang (2011); Wainwright (2009); Zhang and Huang (2008); Wang and Leng (2015)) that work for $\log p = o(n)$ case rely on a sub-Gaussian tail/bounded error assumption, which might fail to hold for real data. Lounici (2008) proved that *lasso* also achieves consistent model selection with a second-order condition similar to ours, but requires two additional assumptions. The algorithms are simple, efficient and scale well for large p. In particular, the matrix operations are fully parallelizable with very few communications for very large p, while regularization methods are either hard to be computed in parallel in the feature space, or the parallelization requires a large amount of machine communications.

The remainder of this article is organized as follows. In Section 2 we generalize the ordinary least squares estimator for high dimensional problems where $p > n$, and propose two three-step algorithms consisting only of least squares fitting and hard thresholding in a loose sense. Section 3 provides consistency theory for the algorithms. Section 4 evaluates the empirical performance. We conclude and discuss further implications of our algorithms in the last section. All the proofs are provided in the supplementary materials.

2. HIGH DIMENSIONAL ORDINARY LEAST SQUARES

Consider the usual linear model

$$Y = X\beta + \varepsilon,$$

where X is the $n \times p$ design matrix, Y is the $n \times 1$ response vector and β is the coefficient. In the high dimensional literature, β_i's are routinely assumed to be zero except for a small subset $S^* = supp(\beta)$. In this paper, we consider a slightly more general setting, where β is not exactly sparse, but consists of both strong and weak signals. In particular, we defined S^* and S_*

$$S^* = \{k \ : |\beta_k| \geq \tau^*\} \qquad S_* = \{k \ : |\beta_k| \leq \tau_*\}$$

as the strong and weak signal sets and $S^* \cup S_* = \{1, 2, \cdots, p\}$. The algorithms developed in this paper is to recover the strong signal set S^*. The specific relationship between τ^* and τ_* will be detailed later.

To carefully tailor the low-dimensional OLS estimator in a high dimensional scenario, one needs to answer the following two questions: i) What is the correct form of OLS in the high dimensional setting? ii) How to correctly use this estimator? To answer these, we reconsider OLS from a different perspective by viewing the OLS as the limit of the ridge estimator with the ridge parameter going to zero, i.e.,

$$(X^TX)^{-1}X^TY = \lim_{r \to 0} \ (X^TX + rI_p)^{-1}X^TY.$$

One nice property of the ridge estimator is that it exists regardless of the relationship between p and n. A keen observation reveals the following relationship immediately.

Lemma 1. *For any $p, n, r > 0$, we have*

$$(X^TX + rI_p)^{-1}X^TY = X^T(XX^T + rI_n)^{-1}Y. \quad (1)$$

Notice that the right hand side of (1) exists when $p > n$ and $r = 0$. Consequently, we can naturally extend the classical OLS to the high dimensional scenario by letting r tend to zero in (1). Denote this high dimensional version of the OLS as

$$\hat{\beta}^{(HD)} = \lim_{r \to 0} X^T(XX^T + rI_n)^{-1}Y = X^T(XX^T)^{-1}Y.$$

Unfortunately, $\hat{\beta}^{(HD)}$ does not have good general performance in estimating sparse vectors in high-dimensional cases. Instead of directly estimating β as $\hat{\beta}^{HD}$, however, this new estimator of β may be used for dimension reduction by observing $\hat{\beta}^{(HD)} = X^T(XX^T)^{-1}X\beta +$

$X^T(XX^T)^{-1}\varepsilon = \Phi\beta + \eta$. Since η is stochastically small, if Φ is close to a diagonally dominant matrix and β is sparse, then the strong and weak signals can be separated by simply thresholding the small entries of $\hat{\beta}^{(HD)}$. The exact meaning of this statement will be discussed in the next section. Some simple examples demonstrating the diagonal dominance of $X^T(XX^T)^{-1}X$ are illustrated immediately in Figure 1, where the rows of X in the left two plots are drawn from $N(0,\Sigma)$ with $\sigma_{ij} = 0.6$ or $\sigma_{ij} = 0.99^{|i-j|}$. The sample size and data dimension are chosen as $(n,p) = (50, 1000)$. The right plot takes the standardized design matrix directly from the real data in Section 4. A clear diagonal dominance pattern is visible in each plot.

This ability to separate strong and weak signals allows us to first obtain a smaller model with size d such that $|S^*| < d < n$ containing S^*. Since d is below n, one can directly apply the usual OLS to obtain an estimator, which will be thresholded further to obtain a more refined model. The final estimator will then be obtained by an OLS fit on the refined model. This three-stage non-iterative algorithm is termed *Least-squares adaptive thresholding (LAT)* and the concrete procedure is described in Algorithm 1.

Algorithm 1 *The Least-squares Adaptive Thresholding (LAT) Algorithm*

Initialization:

Input $(Y, X), d, \delta$

Stage 1 : Pre-selection

Standardize Y and X to $\tilde{Y}$ and $\tilde{X}$ having mean 0 and variance 1.

Compute $\hat{\beta}^{(HD)} = \tilde{X}^T(\tilde{X}\tilde{X}^T + 0.1 \cdot I_n)^{-1}\tilde{Y}$. Rank the importance of the variables by $|\hat{\beta}_i^{(HD)}|$;

Denote the model corresponding to the d largest $|\hat{\beta}_i^{(HD)}|$ as $\tilde{\mathcal{M}}_d$. Alternatively use extended BIC (Chen and Chen, 2008) in conjunction with the obtained variable importance to select the best submodel.

Stage 2 : Hard thresholding

$\hat{\beta}^{(OLS)} = (X^T_{\tilde{\mathcal{M}}_d} X_{\tilde{\mathcal{M}}_d})^{-1} X^T_{\tilde{\mathcal{M}}_d} Y$;

$\hat{\sigma}^2 = \sum_{i=1}^n (y - \hat{y})^2/(n-d)$;

$\bar{C} = (X^T_{\tilde{\mathcal{M}}_d} X_{\tilde{\mathcal{M}}_d})^{-1}$;

Threshold $\hat{\beta}^{(OLS)}$ by $\text{MEAN}(\sqrt{2\hat{\sigma}^2 \bar{C}_{ii} \log(4d/\delta)})$ or use BIC to select the best submodel. Denote the chosen model as $\hat{\mathcal{M}}$.

Stage 3 : Refinement

$\hat{\beta}_{\hat{\mathcal{M}}} = (X^T_{\hat{\mathcal{M}}} X_{\hat{\mathcal{M}}})^{-1} X^T_{\hat{\mathcal{M}}} Y$;

$\hat{\beta}_i = 0, \forall i \notin \hat{\mathcal{M}}$;

return $\hat{\beta}$.

The input parameter d is the submodel size selected in Stage 1 and δ is the tuning parameter determining the threshold in Stage 2. In Stage 1, we use $\hat{\beta}^{(HD)} = \tilde{X}^T(\tilde{X}\tilde{X}^T + 0.1 \cdot I_n)^{-1}\tilde{Y}$ instead of $\hat{\beta}^{(HD)} = \tilde{X}^T(\tilde{X}\tilde{X}^T)^{-1}\tilde{Y}$ because $\tilde{X}\tilde{X}^T$ is rank deficient (the rank is $n-1$) after standardization. The number 0.1 can be replaced by any arbitrary small number. As noted in Wang and Leng (2015), this additional ridge term is also essential when p and n get closer. Our results in Section 3 mainly focus on $\hat{\beta}^{(HD)} = X^T(XX^T)^{-1}Y$ where X is assumed to be drawn from a distribution with mean zero, so no standardization or ridge adjustment is required. However, the result is easy to generalize to the case where a ridge term is included. See Wang and Leng (2015).

The Stage 1 of Algorithm 1 is very similar to variable screening methods (Fan and Lv, 2008; Wang and Leng, 2015). However, most screening methods require a sub-Gaussian condition the noise to handle the ultra-high dimensional data where $\log(p) = o(n)$. In contrast to the existing theory, we prove in the next section a better result that Stage 1 of Algorithm 1 can produce satisfactory submodel even for heavy-tailed noise.

The estimator $\hat{\beta}^{(OLS)}$ in Stage 2 can be substituted by its ridge counterpart $\hat{\beta}^{(Ridge)} = (X^T_{\tilde{\mathcal{M}}_d} X_{\tilde{\mathcal{M}}_d} + rI_d)^{-1} X^T_{\tilde{\mathcal{M}}_d} Y$ and $\bar{C}$ by $(X^T_{\tilde{\mathcal{M}}_d} X_{\tilde{\mathcal{M}}_d} + rI_d)^{-1}$ to stabilize numerical computation. Similar modification can be applied to the Stage 3 as well. The resulted variant of the algorithm is referred to as the *Ridge Adaptive Thresholding (RAT)* algorithm and described in Algorithm 2.

Algorithm 2 *The Ridge Adaptive Thresholding (RAT) Algorithm*

Initialization:

Input $(Y, X), d, \delta, r$

Stage 1 : Pre-selection

Standardize Y and X to $\tilde{Y}$ and $\tilde{X}$ having mean 0 and variance 1.

Compute $\hat{\beta}^{(HD)} = \tilde{X}^T(\tilde{X}\tilde{X}^T + 0.1 \cdot I_n)^{-1}\tilde{Y}$. Rank the importance of the variables by $|\hat{\beta}_i^{(HD)}|$;

Denote the model corresponding to the d largest $|\hat{\beta}_i^{(HD)}|$ as $\tilde{\mathcal{M}}_d$. Alternatively use eBIC in Chen and Chen (2008) in conjunction with the obtained variable importance to select the best submodel.

Stage 2 : Hard thresholding

$\hat{\beta}^{(Ridge)} = (X^T_{\tilde{\mathcal{M}}_d} X_{\tilde{\mathcal{M}}_d} + rI_d)^{-1} X^T_{\tilde{\mathcal{M}}_d} Y$;

$\hat{\sigma}^2 = \sum_{i=1}^n (y - \hat{y})^2/(n-d)$;

$\bar{C} = (X^T_{\tilde{\mathcal{M}}_d} X_{\tilde{\mathcal{M}}_d} + rI_d)^{-1}$;

Threshold $\hat{\beta}^{(OLS)}$ by $\text{MEAN}(\sqrt{2\hat{\sigma}^2 \bar{C}_{ii} \log(4d/\delta)})$ or use BIC to select the best submodel. Denote the chosen model as $\hat{\mathcal{M}}$.

Stage 3 : Refinement

$\hat{\beta}_{\hat{\mathcal{M}}} = (X^T_{\hat{\mathcal{M}}} X_{\hat{\mathcal{M}}} + rI)^{-1} X^T_{\hat{\mathcal{M}}} Y$;

$\hat{\beta}_i = 0, \forall i \notin \hat{\mathcal{M}}$;

return $\hat{\beta}$.

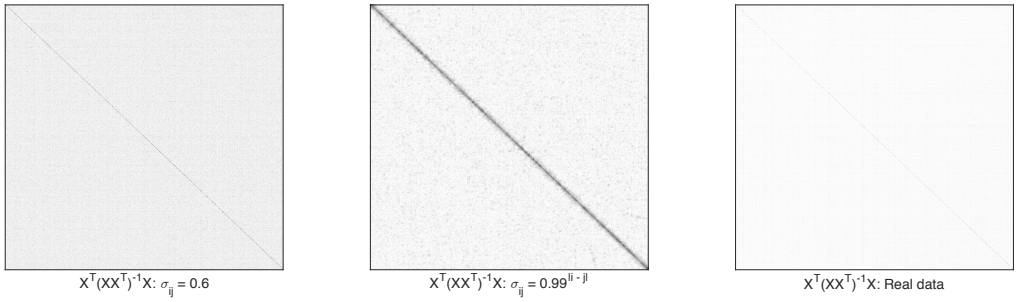

Figure 1. Examples for $X^T(XX^T)^{-1}X$. Left: $X \sim N(0, \Sigma)$ with $\sigma_{ij} = 0.6$ and $\sigma_{ii} = 1$; Middle: $X \sim N(0, \Sigma)$ with $\sigma_{ij} = 0.9^{|i-j|}$; Right: Real data from Section 4.

We suggest to use 10-fold cross-validation to tune the ridge parameter r. Notice that the model is already small after stage 1, so using cross-validation will not significantly increase the computational burden. The computational performance is illustrated in Section 4.

3. THEORY

In this section, we prove the consistency of Algorithm 1 in recoverying S^* and provide concrete forms for all the values needed for the algorithm to work. Recall the linear model $Y = X\beta + \varepsilon$. We consider the random design where the rows of X are drawn from an elliptical distribution with covariance Σ. It is easy to show that x_i admits an equivalent representation as

$$x_i \overset{(d)}{=} L_i \frac{\sqrt{p} z_i}{\|z_i\|_2} \Sigma^{1/2} = \frac{\sqrt{p} L_i}{\|z_i\|_2} z_i \Sigma^{1/2}. \tag{2}$$

where z_i is a p-variate standard Gaussian random variable and L_i is a nonnegative random variable that is independent of z_i. We denote this distribution by $EN(L, \Sigma)$. This random design allows for various correlation structures and contains many distribution families that are widely used (Bickel et al., 2009; Raskutti et al., 2010). The noise ε, as mentioned earlier, is only assumed to have the second-order moment, i.e., $var(\varepsilon) = \sigma^2 < \infty$, in contrast to the sub-Gaussian/bounded error assumption seen in most high dimension literature. See Zhang (2010); Yang et al. (2014); Cai and Wang (2011); Wainwright (2009); Zhang and Huang (2008). This relaxation is similar to Lounici (2008); however we do not require any further assumptions needed by Lounici (2008). In Algorithm 1, we also propose to use extended BIC and BIC for parameter tuning. However, the corresponding details will not be pursued here, as their consistency is straightforwardly implied by the results from this section and the existing literature (Chen and Chen, 2008).

As shown in (2), the variable L controls the signal strength of x_i, we thus need a lower bound on L_i to guarantee a good signal strength. Define $\kappa = cond(\Sigma)$. We state our result in three theorems.

Theorem 1. *Assume $x_i \sim EN(L_i, \Sigma)$ with $E[L_i^{-2}] < M_1$ and ε_i is a random variable with a bounded variance σ^2. We also assume $p > c_0 n$ for some $c_0 > 1$ and $var(Y) \leq M_0$. If $|S^*| \log p = o(n)$, $n > 4c_0/(c_0 - 1)^2$, and $\tau^*/\tau_* \geq 4\kappa^2$, then we can choose γ to be $\frac{2c_1\kappa^{-1}\tau}{3}\frac{n}{p}$, where c_1 is some absolute constant specified in Lemma 2 and for any $\alpha \in (0, 1)$ we have*

$$P\left(\max_{i \in S_*} |\hat{\beta}_i^{(HD)}| \leq \gamma \leq \min_{i \in S^*} |\hat{\beta}_i^{(HD)}|\right) = 1 - O\left(\frac{\sigma^2\kappa^4 \log p}{\tau^{*2} n^\alpha}\right).$$

Theorem 1 guarantees the model selection consistency of the first stage of Algorithm 1. It only requires a second-moment condition on the noise tail, relaxing the sub-Gaussian assumption seen in other literature. The probability term shows that the algorithm requires the strong signals to be lower bounded by a signal strength of $\sigma\sqrt{\frac{\log p}{n^\alpha}}$. In addition, a gap of $\tau^*/\tau_* \geq 4\kappa^2$ is needed between the strong and the weak signals in order for a successful support recovery.

As γ is not easily computable based on data, we propose to rank the $|\hat{\beta}_i^{(HD)}|'s$ and select d largest coefficients. Alternatively, we can construct a series of nested models formed by ranking the largest n coefficients and adopt the extended BIC (Chen and Chen, 2008). Once the submodel $\tilde{\mathcal{M}}_d$ is obtained, we proceed to the second stage by obtaining an estimate via ordinary least squares $\hat{\beta}^{(OLS)}$ corresponding to $\tilde{\mathcal{M}}_d$. The theory for $\hat{\beta}^{(OLS)}$ requires more stringent conditions, as we now need to estimate $\beta_{\hat{\mathcal{M}}_d}$ instead of just obtaining a correct ranking. In particular, we have to impose conditions on the magnitude of β_{S_*} and the moments of L, i.e., for $\hat{\beta}^{(OLS)}$ we have the following result.

Theorem 2. *Assume the same conditions for X and ε as in Theorem 1. We also assume $n \geq 64\kappa d \log p$ and $d - |S^*| \leq \tilde{c}$ for some $\tilde{c} > 0$. If $E[L^{-12}] \leq M_1$, $E[L^{12}] \leq M_2$, $\tau_* \leq \frac{\sigma}{\kappa}\sqrt{\frac{\log p}{n}}$ and there exists some $\iota \in (0, 1)$ such that*

$\sum_{i\in S_*}|\beta_i|^\iota \le R$, *then for any* $\alpha > 0$, *we have*

$$P\Big(\max_{|\hat{\mathcal{M}}|\le d,\ S^*\subset\hat{\mathcal{M}}} \|\hat{\beta}^{(OLS)}-\beta\|_\infty \le 2\sigma\sqrt{\frac{\log p}{n^\alpha}}\Big) = 1 - O\Big(\frac{\lambda_*^{-2}d\log d}{n^{\frac{1}{3}(1-\alpha)}} + \frac{M_1+M_2}{n^{\frac{1}{3}(1-4\alpha)}} + \frac{(M_1+M_2)R^3}{(\log p)^{2\iota}n^{3-4\alpha-2\iota}}\Big),$$

i.e., if $\tau^* \ge 5\sigma\sqrt{\frac{\log p}{n^\alpha}}$, *then we can choose* $\gamma' = 3\sigma\sqrt{\frac{\log p}{n^\alpha}}$ *and*

$$\max_{i\notin S^*}|\hat{\beta}_i^{(OLS)}| \le \gamma' \le \min_{i\in S^*}|\hat{\beta}_i^{(OLS)}|$$

with probability tending to 1.

The moment condition on L is not tight. We use this number just for simplicity. As shown in Theorem 2, the l_ι norm of β_{S_*} is allowed to grow in a rate of $(\log p)^{2\iota/3}n^{1-4\alpha/3-2\iota/3}$, i.e., our algorithms work for weakly sparse coefficients. However, different from Theorem 1, Theorem 2 imposes an upper bound on α. This is mainly due to the different structures between $\hat{\beta}^{(HD)}$ and $\hat{\beta}^{(OLS)}$, i.e., $\hat{\beta}^{(OLS)}$ relies on L for diminishing the weak signals while $\hat{\beta}^{(HD)}$ does not. For ridge regression, we have the following result.

Theorem 3 (Ridge regression)**.** *Assume all the conditions in Theorem 2. If we choose the ridge parameter satisfying*

$$r \le \frac{\sigma n^{(7/9-5\alpha/18)}\sqrt{\log p}}{162\kappa M_0},$$

then we have

$$P\Big(\max_{|\hat{\mathcal{M}}|\le d, S^*\subset\hat{\mathcal{M}}} \|\hat{\beta}^{(ridge)}-\beta\|_\infty \le 3\sigma\sqrt{\frac{\log p}{n^\alpha}}\Big) = 1 - O\Big(\frac{\lambda_*^{-2}d\log d}{n^{\frac{1}{3}(1-\alpha)}} + \frac{2M_1+M_2}{n^{\frac{1}{3}(1-4\alpha)}} + \frac{(M_1+M_2)R^3}{(\log p)^{2\iota}n^{3-4\alpha-2\iota}}\Big),$$

i.e., if $\tau^* \ge 7\sigma\sqrt{\frac{\log p}{n^\alpha}}$, *then we can choose* $\gamma' = 4\sigma\sqrt{\frac{\log p}{n^\alpha}}$ *and*

$$\max_{i\notin S^*}|\hat{\beta}_i^{(Ridge)}(r)| \le \gamma' \le \min_{i\in S^*}|\hat{\beta}_i^{(Ridge)}(r)|$$

with probability tending to 1.

When both the noise ε and X follows Gaussian distribution and $\tau_* = 0$, we can obtain a more explicit form of the threshold γ', as the following Corollary shows.

Corollary 1 (Gaussian noise)**.** *Assume* $\varepsilon \sim N(0,\sigma^2)$, $X \sim N(0,\Sigma)$ *and* $\tau_* = 0$. *For any* $\delta \in (0,1)$, *define* $\gamma' = 8\sqrt{2}\hat{\sigma}\sqrt{\frac{2\kappa\log(4d/\delta)}{n}}$, *where* $\hat{\sigma}$ *is the estimated standard error as* $\hat{\sigma}^2 = \sum_{i=1}^n(y_i-\hat{y}_i)^2/(n-d)$. *For sufficiently large* n, *if* $d \le n - 4K^2\log(2/\delta)/c$ *for some absolute constants* c, K *and* $\tau^* \ge 24\sigma\sqrt{\frac{2\kappa\log(4d/\delta)}{n}}$, *then with probability at least* $1-2\delta$, *we have*

$$|\hat{\beta}_i^{(OLS)}| \ge \gamma' \quad \forall i \in S^* \quad \text{and} \quad |\hat{\beta}_i^{(OLS)}| \le \gamma' \quad \forall i \notin S_*.$$

Write $\bar{C} = (X^T_{\tilde{\mathcal{M}}_d}X_{\tilde{\mathcal{M}}_d})^{-1}$ as in Algorithm 1. In practice, we propose to use $\gamma' = mean(\sqrt{2\hat{\sigma}^2\bar{C}_{ii}\log(4d/\delta)})$ as the threshold (see Algorithm 1), because the estimation error takes a form of $\sqrt{\sigma^2\bar{C}_{ii}\log(4d/\delta)}$. Once the final model is obtained, as in Stage 3 of Algorithm 1, we refit it again using ordinary least squares. The final output will have the same output as if we knew S^* *a priori* with probability tending to 1. As implied by Theorem 1 – 3, *LAT* and *RAT* can consistently identify strong signals in the ultra-high dimensional ($\log p = o(n)$) setting with only the bounded moment assumption $var(\varepsilon) < \infty$, in contrast to most existing methods that require $\varepsilon \sim N(0,\sigma^2)$ or $\|\varepsilon\|_\infty < \infty$.

4. EXPERIMENTS

In this section, we provide extensive numerical experiments for assessing the performance of *LAT* and *RAT*. In particular, we compare the two methods to existing penalized methods including *lasso*, elastic net (*enet* (Zou and Hastie, 2005)), *adaptive lasso* (Zou, 2006), *scad* (Fan and Li, 2001) and *mc+* (Zhang, 2010). As it is well-known that the *lasso* estimator is biased, we also consider two variations of it by combining *lasso* with Stage 2 and 3 of our *LAT* and *RAT* algorithms, denoted as *lasLAT* (*las1* in Figures) and *lasRAT* (*las2* in Figures) respectively. We note that the *lasLat* algorithm is very similar to the thresholded lasso (Zhou, 2010) with an additional thresholding step. We code *LAT* and *RAT* and *adaptive lasso* in *Matlab*, use `glmnet` (Friedman et al., 2010) for *enet* and *lasso*, and `SparseReg` (Zhou et al., 2012; Zhou and Lange, 2013) for *scad* and *mc+*. Since *adaptive lasso* achieves a similar performance as *lasLat* on synthetic datasets, we only report its performance for the real data.

4.1. Synthetic Datasets

The model used in this section for comparison is the linear model $Y = X\beta + \varepsilon$, where $\varepsilon \sim N(0,\sigma^2)$ and $X \sim N(0,\Sigma)$. To control the signal-to-noise ratio, we define $r = \|\beta\|_2/\sigma$, which is chosen to be 2.3 for all experiments. The sample size and the data dimension are chosen to be $(n,p) = (200,1000)$ or $(n,p) = (500,10000)$ for all experiments. For evaluation purposes, we consider four different structures of Σ below.

(i) ***Independent predictors***. The support is set as $S = \{1,2,3,4,5\}$. We generate X_i from a standard multivariate normal distribution with independent components. The coefficients are specified as

$$\beta_i = \begin{cases} (-1)^{u_i}(|N(0,1)|+1),\ u_i \sim Ber(0.5) & i \in S \\ 0 & i \notin S. \end{cases}$$

(ii) ***Compound symmetry***. All predictors are equally correlated with correlation $\rho = 0.6$. The coefficients are set to

be $\beta_i = 3$ for $i = 1, ..., 5$ and $\beta_i = 0$ otherwise.

(iii) ***Group structure***. This example is Example 4 in Zou and Hastie (2005), for which we allocate the 15 true variables into three groups. Specifically, the predictors are generated as

$$x_{1+3m} = z_1 + N(0, 0.01),$$
$$x_{2+3m} = z_2 + N(0, 0.01),$$
$$x_{3+3m} = z_3 + N(0, 0.01),$$

where $m = 0, 1, 2, 3, 4$ and $z_i \sim N(0, 1)$ are independent. The coefficients are set as

$$\beta_i = 3,\ i = 1, 2, \cdots, 15;\ \beta_i = 0,\ i = 16, \cdots, p.$$

(iv) ***Factor models***. This model is also considered in Meinshausen and Bühlmann (2010) and Cho and Fryzlewicz (2012). Let $\phi_j, j = 1, 2, \cdots, k$ be independent standard normal variables. We set predictors as $x_i = \sum_{j=1}^{k} \phi_j f_{ij} + \eta_i$, where f_{ij} and η_i are generated from independent standard normal distributions. The number of factors is chosen as $k = 5$ in the simulation while the coefficients are specified the same as in Example (ii).

To compare the performance of all methods, we simulate 200 synthetic datasets for $(n, p) = (200, 1000)$ and 100 for $(n, p) = (500, 10000)$ for each example, and record i) the **root mean squared error (RMSE):** $\|\hat{\beta} - \beta\|_2$, ii) the **false negatives (# FN)**, iii) the **false positives (# FP)** and iv) the actual **runtime** (in milliseconds). We use the extended BIC (Chen and Chen, 2008) to choose the parameters for any regularized algorithm. Due to the huge computation expense for *scad* and *mc+*, we only find the first $\lceil \sqrt{p} \rceil$ predictors on the solution path (because we know $s \ll \sqrt{p}$). For *RAT* and *LAT*, d is set to $0.3 \times n$. For *RAT* and *larsRidge*, we adopt a 10-fold cross-validation procedure to tune the ridge parameter r for a better finite-sample performance, although the theory allows r to be fixed as a constant. For all hard-thresholding steps, we fix $\delta = 0.5$. The results for $(n, p) = (200, 1000)$ are plotted in Figure 2, 3, 4 and 5 and a more comprehensive result (average values for **RMSE, # FPs, # FNs, runtime**) for $(n, p) = (500, 10000)$ is summarized in Table 1.

As can be seen from both the plots and the tables, *LAT* and *RAT* achieve the smallest RMSE for Example (ii), (iii) and (iv) and are on par with *lasLAT* for Example (i). For Example (iii), *RAT* and *enet* achieve the best performance while all the other methods fail to work. In addition, the runtime of *LAT* and *RAT* are also competitive compared to that of *lasso* and *enet*. We thus conclude that *LAT* and *RAT* achieve similar or even better performance compared to the usual regularized methods.

4.2. A Student Performance Dataset

We look at one dataset used for evaluating student achievement in Portuguese schools (Cortez and Silva, 2008). The data attributes include student grades and school related features that were collected by using school reports and questionnaires. The particular dataset used here provides the students' performance in mathematics. The goal of the research is to predict the final grade based on all the attributes.

The original data set contains 395 students and 32 raw attributes. The raw attributes are recoded as 40 attributes and form 780 features after interaction terms are added. We then remove features that are constant for all students. This gives 767 features for us to work with. To compare the performance of all methods, we first randomly split the dataset into 10 parts. We use one of the 10 parts as a test set, fit all the methods on the other 9 parts, and then record their prediction error (root mean square error, RMSE), model size and runtime on the test set. We repeat this procedure until each of the 10 parts has been used for testing. The averaged prediction error, model size and runtime are summarized in Table 2. We also report the performance of the null model which predicts the final grade on the test set using the mean final grade in the training set.

It can be seen that *RAT* achieves the smallest cross-validation error, followed by *scad* and *mc+*. In the post-feature-selection analysis, we found that two features, the 1st and 2nd period grades of a student, were selected by all the methods. This result coincides with the common perception that these two grades usually have high impact on the final grade.

In addition, we may also be interested in what happens when no strong signals are presented. One way to do this is to remove all the features that are related to the 1st and 2nd grades before applying the aforementioned procedures. The new result without the strong signals removed are summarized in Table 3.

Table 3 shows a few interesting findings. First, under this artificial weak signal scenario, *adaptive lasso* achieves the smallest cross-validation error and *RAT* is the first runner-up. Second, in Stage 1, *lasso* seems to provide slightly more robust screening than OLS in that the selected features are less correlated. This might be the reason that *LAT* is outperformed by *lasLAT*. However, in both the strong and weak signal cases, *RAT* is consistently competitive in terms of performance.

5. CONCLUSION

We have proposed two novel algorithms *Lat* and *Rat* that only rely on least-squares type of fitting and hard threshold-

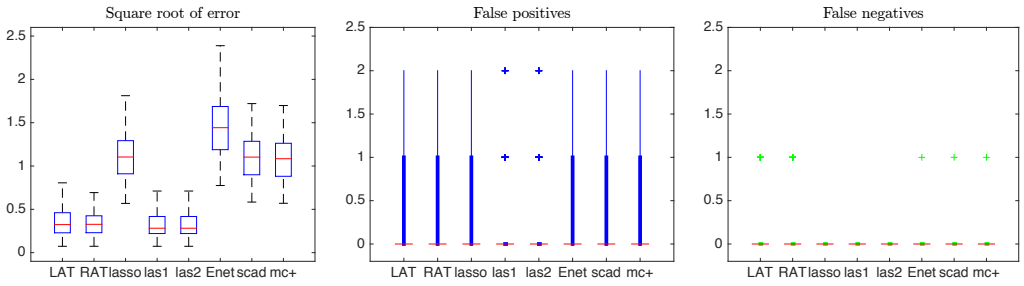

Figure 2. The Boxplots for Example (i). Left: Estimation Error; Middle: False Positives; Right: False Negatives

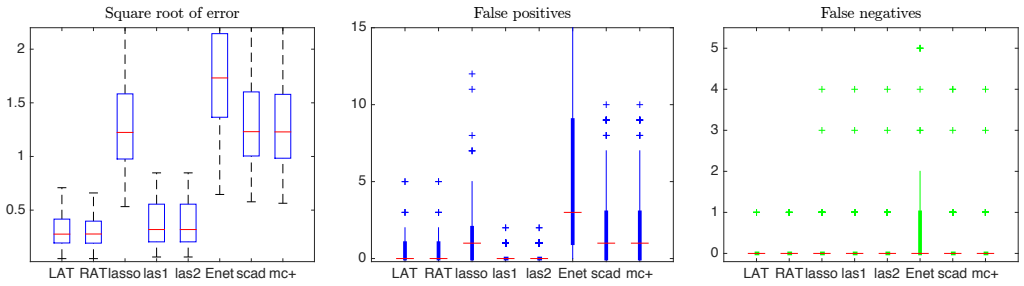

Figure 3. The Boxplots for Example (ii). Left: Estimation Error; Middle: False Positives; Right: False Negatives

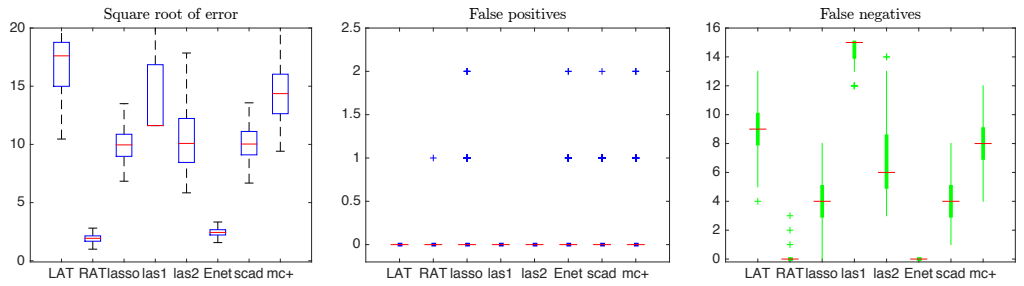

Figure 4. The Boxplots for Example (iii). Left: Estimation Error; Middle: False Positives; Right: False Negatives

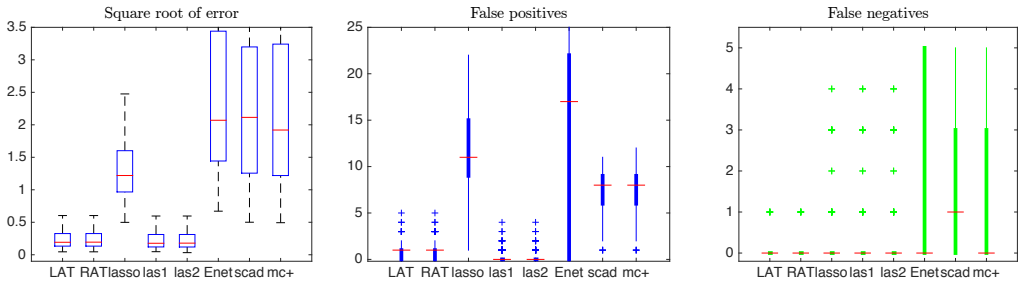

Figure 5. The boxplots for Example (iv). Left: Estimation Error; Middle: False Positives; Right: False Negatives

Table 1. Results for $(n, p) = (500, 10000)$

Example		*LAT*	*RAT*	*lasso*	*lasLAT*	*lasRAT*	*enet*	*scad*	*mc+*
Ex.(i)	RMSE	0.263	0.264	0.781	0.214	0.214	1.039	0.762	0.755
	# FPs	0.550	0.580	0.190	0.190	0.190	0.470	0.280	0.280
	# FNs	0.010	0.010	0.000	0.000	0.000	0.000	0.000	0.000
	Time	36.1	41.8	72.7	72.7	74.1	71.8	1107.5	1003.2
Ex. (ii)	RMSE	0.204	0.204	0.979	0.260	0.260	1.363	0.967	0.959
	# FPs	0.480	0.480	1.500	0.350	0.350	10.820	2.470	2.400
	# FNs	0.000	0.000	0.040	0.040	0.040	0.040	0.020	0.020
	Time	34.8	40.8	76.1	76.1	77.5	82.0	1557.6	1456.1
Ex. (iii)	RMSE	9.738	1.347	7.326	17.621	3.837	1.843	7.285	8.462
	# FPs	0.000	0.000	0.060	0.000	0.000	0.120	0.120	0.090
	# FNs	4.640	0.000	1.440	13.360	1.450	0.000	1.800	2.780
	Time	35.0	41.6	75.6	75.6	77.5	74.4	6304.4	4613.8
Ex. (iv)	RMSE	0.168	0.168	1.175	0.256	0.256	1.780	0.389	0.368
	# FPs	0.920	0.920	21.710	0.260	0.260	37.210	6.360	6.270
	# FNs	0.010	0.010	0.140	0.140	0.140	0.450	0.000	0.000
	Time	34.5	41.1	78.7	78.7	80.8	81.4	1895.6	1937.1

Table 2. Prediction Error of the Final Grades by Different Methods

methods	mean error	Standard error	average model size	runtime (millisec)
LAT	1.93	0.118	6.8	22.3
RAT	**1.90**	0.131	3.5	74.3
lasso	1.94	0.138	3.7	60.7
lasLAT	2.02	0.119	3.6	55.5
lasRAT	2.04	0.124	3.6	71.3
enet	1.99	0.127	4.7	58.5
scad	1.92	0.142	3.5	260.6
mc+	1.92	0.143	3.4	246.0
adaptive lasso	2.01	0.140	3.6	65.5
null	4.54	0.151	0	—

Table 3. Prediction Error of the Final Grades Excluding Strong Signals

methods	mean error	Standard error	average model size	runtime (millisec)
LAT	4.50	0.141	5.3	22.4
RAT	4.26	0.130	4.0	74.0
lasso	4.27	0.151	5.0	318.9
lasLAT	4.25	0.131	2.9	316.5
lasRAT	4.28	0.127	2.8	331.9
enet	4.37	0.171	6.0	265.6
scad	4.30	0.156	4.8	387.5
mc+	4.29	0.156	4.7	340.2
adaptive lasso	**4.24**	0.180	4.8	298.0
null	4.54	0.151	0	—

ing, based on a high-dimensional generalization of OLS. The two methods are simple, easily implementable, and can consistently fit a high dimensional linear model and recover its support. The performance of the two methods are competitive compared to existing regularization methods. It is of great interest to further extend this framework to other models such as generalized linear models and models for survival analysis.

References

Bickel, P. J., Ritov, Y., and Tsybakov, A. B. (2009). Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732.

Cai, T. T. and Wang, L. (2011). Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information Theory*, 57(7):4680–4688.

Chen, J. and Chen, Z. (2008). Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771.

Cho, H. and Fryzlewicz, P. (2012). High dimensional variable selection via tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(3):593–622.

Cortez, P. and Silva, A. M. G. (2008). Using data mining to predict secondary school student performance.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1.

Jain, P., Tewari, A., and Kar, P. (2014). On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems*, pages 685–693.

Lounici, K. (2008). Sup-norm convergence rate and sign concentration property of lasso and dantzig estimators. *Electronic Journal of Statistics*, 2:90–102.

Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473.

Raskutti, G., Wainwright, M. J., and Yu, B. (2010). Restricted eigenvalue properties for correlated gaussian designs. *The Journal of Machine Learning Research*, 11:2241–2259.

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

Wainwright, M. J. (2009). Sharp thresholds for high-dimensional and noisy sparsity recovery using-constrained quadratic programming (lasso). *IEEE Transactions on Information Theory*, 55(5):2183–2202.

Wang, X. and Leng, C. (2015). High dimensional ordinary least squares projection for screening variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*.

Yang, E., Lozano, A., and Ravikumar, P. (2014). Elementary estimators for high-dimensional linear regression. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 388–396.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942.

Zhang, C.-H. and Huang, J. (2008). The sparsity and bias of the lasso selection in high-dimensional linear regression. *The Annals of Statistics*, 36(4):1567–1594.

Zhao, P. and Yu, B. (2006). On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7:2541–2563.

Zhou, H., Armagan, A., and Dunson, D. B. (2012). Path following and empirical bayes model selection for sparse regression. *arXiv preprint arXiv:1201.3528*.

Zhou, H. and Lange, K. (2013). A path algorithm for constrained estimation. *Journal of Computational and Graphical Statistics*, 22(2):261–283.

Zhou, S. (2010). Thresholded lasso for high dimensional variable selection and statistical estimation. *arXiv preprint arXiv:1002.1583*.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.