An adaptive shortest-solution guided decimation approach to sparse high-dimensional linear regression

Xue Yu
Renmin University of China
Yifan Sun (sunyifan@ruc.edu.cn)
Renmin University of China
Hai-Jun Zhou
Chinese Academy of Sciences

Research Article

Keywords: sparse high-dimensional linear regression, applied mathematics, statistical mathematics

Posted Date: June 9th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-598251/v1

Version of Record: A version of this preprint was published at Scientific Reports on December 1st, 2021.
See the published version at https://doi.org/10.1038/s41598-021-03323-7.

An adaptive shortest-solution guided decimation approach to sparse high-dimensional linear regression

Xue Yu[1], Yifan Sun[1,*], and Hai-Jun Zhou[2,3,4,*]

[1]Center for Applied Statistics, School of Statistics, Renmin University of China, Beijing 100872, China
[2]CAS Key Laboratory for Theoretical Physics, Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing 100190, China
[3]School of Physical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China
[4]MinJiang Collaborative Center for Theoretical Physics, MinJiang University, Fuzhou 350108, China
[*]sunyifan@ruc.edu.cn, zhouhj@itp.ac.cn

ABSTRACT

High-dimensional linear regression model is the most popular statistical model for high-dimensional data, but it is quite a challenging task to achieve a sparse set of regression coefficients. In this paper, we propose a simple heuristic algorithm to construct sparse high-dimensional linear regression models, which is adapted from the shortest-solution guided decimation algorithm and is referred to as ASSD. This algorithm constructs the support of regression coefficients under the guidance of the least-squares solution of the recursively decimated linear equations, and it applies an early-stopping criterion and a second-stage thresholding procedure to refine this support. Our extensive numerical results demonstrate that ASSD outperforms LASSO, vector approximate message passing, and two other representative greedy algorithms in solution accuracy and robustness. ASSD is especially suitable for linear regression problems with highly correlated measurement matrices encountered in real-world applications.

Introduction

Detecting the relationship between a response and a set of predictors is a common problem encountered in different branches of scientific research. This problem is referred to as regression analysis in statistics. A major focus of regression analysis has been on linear regression models, which search for a linear relationship between the responses and the predictors. Consider the linear regression model of the following form

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}^0 + \boldsymbol{\varepsilon}\,, \tag{1}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ is the response vector, $\boldsymbol{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_p) \in \mathbb{R}^{n\times p}$ is an $n \times p$ measurement matrix with $\boldsymbol{X}_i = (X_{1i}, X_{2i}, \ldots, X_{ni})^\top \in \mathbb{R}^n$ being the i-th column, $\boldsymbol{\beta}^0 = (\beta_1^0, \cdots, \beta_p^0)^\top \in \mathbb{R}^p$ is the vector of p true regression coefficients, and $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)^\top \in \mathbb{R}^n$ are random errors with $\mathbb{E}(\varepsilon_i) = 0$. The variance of ε_i is $\mathbb{E}(\varepsilon_i^2) = \sigma^2$ with σ^2 being the variance of the noise level (σ being the typical magnitude of the noise). Let s_0 be the number of nonzero entries in $\boldsymbol{\beta}^0$. We focus on the case where $p > n$ and $s_0 < n$, and the goal is to construct a spare vector $\boldsymbol{\beta}$ which serves as the best approximation to the hidden truth vector $\boldsymbol{\beta}^0$, given $\boldsymbol{y}$ (the measurement results) and $\boldsymbol{X}$ (the measurement matrix) but with $\boldsymbol{\varepsilon}$ (the noise vector) unknown.

Such linear regression models are widely adopted in many practical applications because of their simplicity and interpretability. With the advancement in measurement technologies, high-dimensional data are nowadays accumulating with fast speed in a variety of fields such as genomics, neuroscience, systems biology, economics, and social science. In these high-dimensional data, the number p of predictors is often larger than the number n of samples or measurements ($p > n$), making the solution of the linear regression problem far from being unique. Additional criteria need to be imposed to reduce the degeneracy of solutions and to select the most appropriate linear regression model. One of the most important criteria is sparsity. Motivated by empirical findings in genomics and other fields, we usually assume that the high-dimensional regression models are sparse, in the sense that only a relatively small number of predictors are important for explaining the observed data[1]. Associated with this sparsity criterion are two highly nontrivial issues in high-dimensional linear regression: (1) variables selection, namely to specify the most relevant predictors; and (2) parameters or coefficients estimation, namely to determine the individual contributions of the chosen predictors. Sparse high-dimensional linear regression has also been studied from the angle of compressed sensing[2].

In principle, the regression coefficients can be specified by searching for the solution with the least number of nonzero elements, but this non-convex l_0 minimization problem is intractable in practice. Over the years a variety of approaches have been proposed to approximate the optimal l_0 solution. The existing approaches can roughly be divided into three categories: relaxation methods, physics-inspired message-passing methods, and greedy methods. The basic idea of the relaxation methods is to replace the non-smooth l_0-norm penalty with a smooth approximation. Among them the Least Absolute Shrinkage and Selection Operator (LASSO)[3,4], which uses the l_1-norm penalty, is the most popular one. LASSO is a convex optimization problem, which can be solved by methods such as LARS[5–7], coordinate descent[8] and proximal gradient descent[9]. However, due to the over-shrinking of large coefficients, LASSO is known to lead to biased estimates. To remedy this problem some non-convex penalties, e.g., the smoothly clipped absolute deviation (SCAD) penalty[10] and the minimax concave penalty (MCP)[11], have been proposed. The global minimizers of these penalty functions help to eliminate the estimation bias.

An alternative strategy comes from the approximate message-passing (AMP) methods, which are closely related to the Thouless-Anderson-Palmer equation in statistical physics that are capable of dealing with high-dimensional inference problems. They have shown remarkable success in sparse regression and compressed sensing[12–15]. However, the convergence issue limits the practical application of the AMP methods, especially on problems with highly correlated predictors. Recently, several algorithms such as Generalized AMP (GAMP)[16], SwAMP[17], adaptive damping[18], mean removal[18] and direct free-energy minimization[19] were proposed to fix this problem. Especially, the orthogonal or vector AMP (VAMP) algorithm[20,21] offers a robust alternative to the conventional AMP.

Another line of research focuses on greedy methods for l_0 minimization such as orthogonal least squares (OLS)[22] and orthogonal matching pursuit (OMP)[23]. The main idea is to select at each iteration step a single variable vector that has the largest magnitude of (rescaled) inner product with the current residual response vector. A sure-independence-screening (SIS) method based on correlation learning was proposed to improve variable selection[24], and an iterative version of this SIS approach (ISIS) could be adopted to enhance the performance of variable selection[25]. Several more recently developed greedy methods proposed to select several variables at a time, including the iterate hard thresholding (IHT) algorithm[26,27], the primal-dual active set (PDAS) methods[28], and the adaptive support detection and root finding (ASDAR) approach[29].

Most of the above-mentioned approximate methods generally assume that the measurement matrix satisfies some regularity conditions such as the irrepresentable condition and the sparse Riesz condition, for mathematical convenience or for good algorithmic performance. Roughly speaking, these conditions require that the predictors should be fully uncorrelated or only weakly correlated. But these strict conditions are often not met in real-world applications. As such, it is desirable to develop an efficient and robust method applicable for more general correlation structures of the predictors. Recently, the shortest-solution guided decimation (SSD) algorithm[30] is proposed as a greedy method for solving high-dimensional linear regression. Similar to OLS and OMP, at each iteration step SSD selects a single variable as a candidate predictor. The difference is that this selection is based on the dense least-squares (i.e., shortest Euclidean length) solution of the decimated linear equations. Initial simulation results demonstrated that this SSD algorithm significantly outperforms several of the most popular algorithms (l_1-based penalty methods, OLS, OMP, and AMP) when the measurement matrices are highly correlated.

Although the SSD algorithm is highly competitive to other heuristic algorithms both for uncorrelated and correlated measurement matrices, a crucial assumption in its original implementation is that there is no any measurement noise ($\boldsymbol{\varepsilon} = \mathbf{0}$). As we will demonstrate later, when the measurement noise is no longer negligible, the naive noise-free SSD algorithm fails to extract the spare solution of linear regression. To overcome this difficulty, here we extend the SSD algorithm and propose the adaptive SSD algorithm (ASSD) to estimate the sparse high-dimensional regression models. Compared with the original SSD, the new ASSD algorithm adopts a much more relaxed termination condition to allow early stop. Furthermore and significantly, we add a second-stage screening to single out the truly important predictors after the first-stage estimation is completed.

We test the performance of ASSD both on synthetic data (predictors and responses are both simulated) and on semi-synthetic data (real predictors but simulated responses, using the gene expression data from cancer samples). In comparison with the representative algorithms LASSO, VAMP and two greedy methods (ASDAR and SIS-LASSO), our extensive simulation results demonstrate that ASSD outperforms all these competing algorithms in terms of accuracy and robustness of variables selection and coefficients estimation. It appears that ASSD is especially suitable for linear regression problems with highly correlated measurement matrices encountered in real-world applications. On the other hand, ASSD is generally slower than these other algorithms, pointing to a direction of further improvement. It may also be interesting to analyze theoretically the SSD and ASSD algorithms.

Methods

The shortest solution as a guidance vector

Consider the singular value decomposition (SVD) of the measurement matrix $\boldsymbol{X}$: $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$, where $\boldsymbol{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n)$ is an $n \times n$ orthogonal matrix, and $\boldsymbol{V} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_p)$ is a $p \times p$ orthogonal matrix, and $\boldsymbol{D}$ is an $n \times p$ diagonal matrix of the singular

values $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. Here $\{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_n\}$ and $\{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_p\}$ form a complete set of orthonormal basis vectors for the n- and p-dimensional real space respectively, so the vectors $\boldsymbol{u}_i = (u_{1i},\ldots,u_{ni})^\top$ satisfy $\boldsymbol{u}_i^\top \boldsymbol{u}_j = \delta_{ij}$, and vectors $\boldsymbol{v}_i = (v_{1i},\ldots,v_{pi})^\top$ satisfy $\boldsymbol{v}_i^\top \boldsymbol{v}_j = \delta_{ij}$, where δ_{ij} is the Kronecker symbol: $\delta_{ij} = 0$ for $i \neq j$ and $\delta_{ij} = 1$ for $i = j$. We can express the true coefficient vector $\boldsymbol{\beta}^0$ as a linear combination of the basis vectors $\boldsymbol{v}_j$:

$$\boldsymbol{\beta}^0 = \sum_{i=1}^{n} c_i \boldsymbol{v}_i + \sum_{j=n+1}^{p} c_j \boldsymbol{v}_j \,. \tag{2}$$

Substituting the above expression into the regression function $\mathbb{E}(\boldsymbol{y}|\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{\beta}^0$ of model (1), we obtain that

$$\mathbb{E}(\boldsymbol{y}|\boldsymbol{X}) = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top \boldsymbol{\beta}^0 = \sum_{i=1}^{n} \lambda_i c_i \boldsymbol{u}_i \,, \tag{3}$$

with the parameter c_i for $i = 1,2,\ldots,n$ being

$$c_i = \Theta(\lambda_i) \frac{\mathbb{E}(\boldsymbol{y}|\boldsymbol{X})^\top \boldsymbol{u}_i}{\lambda_i} \,. \tag{4}$$

Here $\Theta(x)$ is the Heaviside function: $\Theta(x) = 1$ for $x > 0$ and $\Theta(x) = 0$ for $x \leq 0$. We define a vector $\boldsymbol{\gamma}$ as

$$\boldsymbol{\gamma} := \sum_{i=1}^{n} \Theta(\lambda_i) \frac{\mathbb{E}(\boldsymbol{y}|\boldsymbol{X})^\top \boldsymbol{u}_i}{\lambda_i} \boldsymbol{v}_i \,. \tag{5}$$

Then

$$\boldsymbol{\beta}^0 = \boldsymbol{\gamma} + \sum_{j=n+1}^{p} c_j \boldsymbol{v}_j \,. \tag{6}$$

We call $\boldsymbol{\gamma}$ the guidance vector[30]. This vector $\boldsymbol{\gamma}$ is dense and it is not the true coefficient vector $\boldsymbol{\beta}^0$ we are seeking. However, interestingly, this dense vector $\boldsymbol{\gamma}$ does provide information about the locations of nonzero elements of $\boldsymbol{\beta}^0$ (see Fig. 1 and also the earlier empirical observations[30]). To understand this important property of $\boldsymbol{\gamma}$, firstly, we reformulate the matrices $\boldsymbol{V}$ and $\boldsymbol{D}$ as partitioned matrices: $\boldsymbol{V} = (\boldsymbol{V}_1, \boldsymbol{V}_2)$ with $\boldsymbol{V}_1 \in \mathbb{R}^{p\times n}$ and $\boldsymbol{V}_2 \in \mathbb{R}^{p\times(p-n)}$, and $\boldsymbol{D} = (\boldsymbol{D}_1, \boldsymbol{0})$ with $\boldsymbol{D}_1 = \mathrm{diag}(\lambda_1,\ldots,\lambda_n)$. Then, we have

$$\begin{aligned} \boldsymbol{\gamma} &= \boldsymbol{V}_1 \boldsymbol{D}_1^{-1} \boldsymbol{U}^\top \mathbb{E}(\boldsymbol{y}|\boldsymbol{X}) = \boldsymbol{V}_1 \boldsymbol{D}_1^{-1} \boldsymbol{U}^\top \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top \boldsymbol{\beta}^0 \\ &= \boldsymbol{V}_1 \boldsymbol{V}_1^\top \boldsymbol{\beta}^0 \,. \end{aligned} \tag{7}$$

We define $\boldsymbol{Q} := \boldsymbol{V}_1 \boldsymbol{V}_1^\top$, which is a $p \times p$ symmetric matrix. According to equation (7), each element γ_i of $\boldsymbol{\gamma}$ is

$$\gamma_i = Q_{ii}\beta_i^0 + \sum_{j\neq i} Q_{ij}\beta_j^0 \,, \tag{8}$$

where $Q_{ii} = \sum_{l=1}^{n} v_{il}^2$, and $Q_{ij} = \sum_{l=1}^{n} v_{il}v_{jl}$. Since $||\boldsymbol{v}_l||_2 = 1$, we may expect that $v_{il} \approx \pm 1/\sqrt{p}$, and thus $Q_{ii} \approx n/p \equiv \alpha$ (here α is the compression ratio). Expecting that v_{il} and v_{jl} are almost independent of each other, we get that $Q_{ij} \approx \pm\sqrt{n}/p$, where $\pm$ means that Q_{ij} is positive or negative with roughly equal probability. Define $\rho := s_0/p$ as the sparcity of $\boldsymbol{\beta}^0$. Because $\boldsymbol{\beta}^0$ is a sparse vector with only ρp nonzero entries, the summation in the right hand side of equation (eq:Q) contains at most ρp terms. Neglecting the possible weak correlations among Q_{ij} ($j \neq i$), we have $\sum_{j\neq i} Q_{ij}\beta_j^0 \approx \pm\frac{\sqrt{n}}{p}\sqrt{\rho p}m_0 = \pm\sqrt{\alpha\rho}m_0$, where $m_0 = \sqrt{\frac{1}{\rho p}\sum_{i=1}^{p}(\beta_i^0)^2}$ is the mean magnitude of the β_i^0 coefficients. Putting the above approximations together, we finally get

$$\gamma_i \approx \alpha\beta_i^0 + (\pm\sqrt{\alpha\rho}m_0) \,. \tag{9}$$

Notice that the second term in the right hand side of equation (9) is independent of the index i. When β_i^0 is nonzero and the two terms in the right hand side of equation (9) have the same sign, γ_i will be likely to have relatively large magnitude. It then follows that, for the element γ_k that has the largest magnitude among all the elements of $\boldsymbol{\gamma}$, the corresponding β_k^0 is very likely to be nonzero and also $|\beta_k^0| \gtrapprox m_0$.

The above analysis offers a qualitative explanation on why the guidance vector $\boldsymbol{\gamma}$ can help us to locate the nonzero elements in the sparse vector $\boldsymbol{\beta}^0$. When there is no noise ($\boldsymbol{\varepsilon} = \boldsymbol{0}$) this guidance vector is easy to determine and it is the shortest Eucidean-length solution of an underdetermined linear equation. In the presence of measurement noise ($\boldsymbol{\varepsilon} \neq \boldsymbol{0}$), however, the conditional

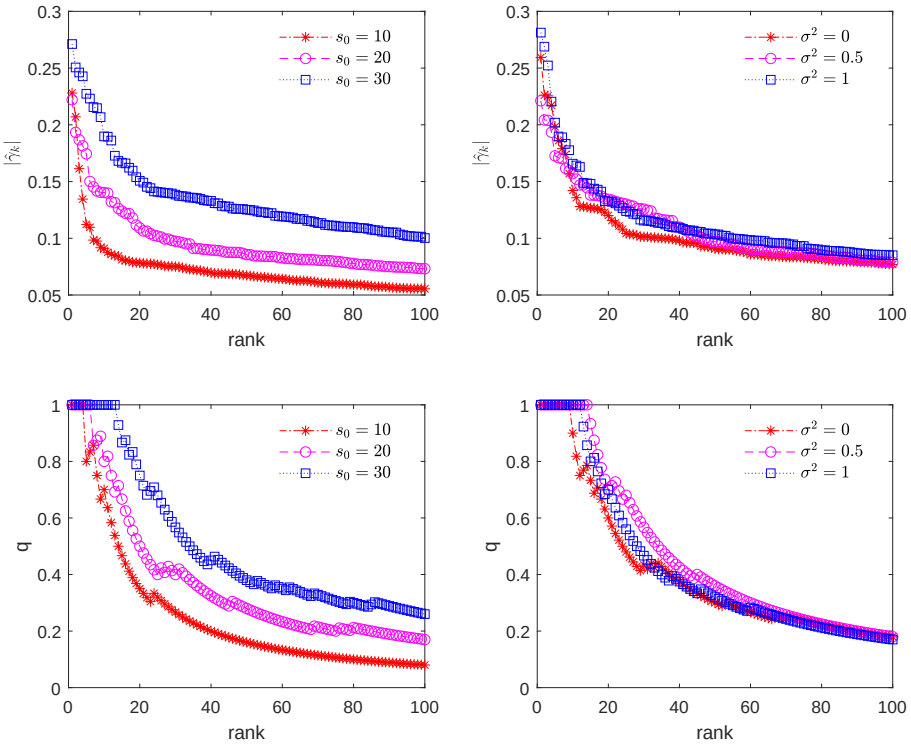

Figure 1. Estimated guidance vector $\hat{\boldsymbol{\gamma}}$ on an uncorrelated Gaussian measurement matrix with $p = 1000$ and $n = 200$. Top: rank curves for the estimated guidance vector. Bottom: proportion $q(r)$ of nonzero elements of $\boldsymbol{\beta}^0$ among the r top-ranked indices i. In the two subfigures on the left panel, $\sigma^2 = 1$ and $s_0 = 10, 20, 30$; and in the two subfigures on the right panel, $s_0 = 20$ and $\sigma^2 = 0, 0.5, 1$.

expectation $\mathbb{E}(\boldsymbol{y}|\boldsymbol{X})$ is unknown. Then we cannot get the exact value of the guidance vector $\boldsymbol{\gamma}$ but can only get an approximate $\boldsymbol{\gamma}$. Consider the estimator

$$\hat{\boldsymbol{\gamma}} = \boldsymbol{V}_1 \boldsymbol{D}_1^{-1} \boldsymbol{U}^\top \boldsymbol{y} = \boldsymbol{X}^+ \boldsymbol{y}\,, \tag{10}$$

where $\boldsymbol{X}^+ = \boldsymbol{V}_1 \boldsymbol{D}_1^{-1} \boldsymbol{U}^\top$ is the Moore-Penrose inverse of $\boldsymbol{X}$. It can be proved that $\hat{\boldsymbol{\gamma}}$ is the best linear unbiased estimator to $\boldsymbol{\gamma}$ (see more details in the Supplementary Information). Combined with the above theoretical analysis, we conjecture that $\hat{\boldsymbol{\gamma}}$ is also helpful for us to guess which elements of the true coefficient vector $\boldsymbol{\beta}^0$ are nonzero. The validity of this conjecture has been confirmed by simulation results. Figure 1 shows the magnitude of elements $\hat{\gamma}_k$ of the estimated guidance vector $\hat{\boldsymbol{\gamma}}$ in descending order (top) and the proportion $q(r)$ of nonzero elements of $\boldsymbol{\beta}^0$ among the r top-ranked indices i (bottom) for datasets generated from models with uncorrelated Gaussian measurement matrices with $n = 200$, $p = 1000$, $s_0 = 10, 20, 30$ and $\sigma^2 = 0, 0.5, 1$. It can be seen that $\hat{\boldsymbol{\gamma}}$ indeed contains important clues about the nonzero elements of $\boldsymbol{\beta}^0$: for the indices k that are ranked in the top in terms of magnitude of $\hat{\gamma}_k$, the corresponding β_k^0 values have high probabilities to be nonzero. In particular, for the 10 top-ranked indices in the examples of $s_0 = 20$, the corresponding entries in $\boldsymbol{\beta}^0$ are nearly all nonzero.

In practice, the estimated guidance vector $\hat{\boldsymbol{\gamma}}$ can be solved through LQ decomposition or convex optimization which are more efficient than SVD. In our simulation studies, we employ the convex optimization method[30].

shortest-solution guided Decimation

Based on the above theoretical analysis and empirical results, we now try to solve the linear model (1) through a shortest-solution guided decimation algorithm. Specifically, let $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$ be a p-dimensional coefficient vector. Assume that the k-th element of the guidance vector $\hat{\boldsymbol{\gamma}}$ has the largest magnitude. If all the other $p-1$ elements β_i of the vector $\boldsymbol{\beta}$ are known,

β_k can be uniquely determined as the solution of the minimization problem

$$\begin{aligned}\beta_k &:= \arg\min_{\beta}\Big(\boldsymbol{y}-\boldsymbol{X}_k\beta-\sum_{i\neq k}\boldsymbol{X}_i\beta_i\Big)^2 \\ &= \frac{\boldsymbol{y}^\top\boldsymbol{X}_k}{\boldsymbol{X}_k^\top\boldsymbol{X}_k}-\sum_{i\neq k}\frac{\boldsymbol{X}_i^\top\boldsymbol{X}_k}{\boldsymbol{X}_k^\top\boldsymbol{X}_k}\beta_i\,. \end{aligned} \tag{11}$$

Plugging equation (11) into model (1), we obtain that

$$\boldsymbol{y}' = \boldsymbol{X}'\boldsymbol{\beta}_{-k}+\boldsymbol{\varepsilon}\,, \tag{12}$$

where $\boldsymbol{\beta}_{-k}=(\beta_1,\ldots,\beta_{k-1},\beta_{k+1},\ldots,\beta_p)^\top$, namely the vector formed by deleting β_k from $\boldsymbol{\beta}$; $\boldsymbol{X}'=(\boldsymbol{X}'_1,\ldots,\boldsymbol{X}'_{k-1},\boldsymbol{X}'_{k+1},\ldots,\boldsymbol{X}'_p)$ is an $n\times(p-1)$ decimated measurement matrix with its column vector $\boldsymbol{X}'_i$ being

$$\boldsymbol{X}'_i = \boldsymbol{X}_i-\frac{\boldsymbol{X}_i^\top\boldsymbol{X}_k}{\boldsymbol{X}_k^\top\boldsymbol{X}_k}\boldsymbol{X}_k\,, \tag{13}$$

and $\boldsymbol{y}'$ is the residual of the original response vector,

$$\boldsymbol{y}' = \boldsymbol{y}-\frac{\boldsymbol{y}^\top\boldsymbol{X}_k}{\boldsymbol{X}_k^\top\boldsymbol{X}_k}\boldsymbol{X}_k\,. \tag{14}$$

Notice that equation (12) has the identical form as that of the original linear model (1). Therefore, we can obtain the corresponding estimated guidance vector through the least-squares solution of equation (12). Then, we repeat the above decimation process (11-14) to further shrink the residual response vector, until certain stopping criterion is met. Suppose that a total number L of elements of $\boldsymbol{\beta}$ have been picked during this whole decimation process. We can uniquely and easily determine the values of these L elements by setting all the other $p-L$ elements to be zero and then backtracking the L constructed equations of the form (11).

Up to now, this above SSD algorithm is the same as the original algorithm[30]. In the original SSD algorithm, the stopping criterion is that the magnitude of the residual response vector becomes less than a prespecified threshold (e.g., 10^{-5}). We test the performance of SSD on a single noisy problem instance, to test if this stopping criterion is still appropriate for the noisy situation. Figure 2 shows the trace of the SSD process on two datasets with noise level $\sigma^2=0$ (left) and $\sigma^2=1$ (right). The two datasets have the identical 200×1000 measurement matrix $\boldsymbol{X}$ and the true coefficient vector $\boldsymbol{\beta}^0$ with $s_0=30$ nonzero elements, which are generated in the same way as that in the simulation experiment in Results. We see that, for the noise-free situation, the decimation stops (i.e., $\frac{1}{n}||\boldsymbol{y}'||_1<10^{-5}$) after $L=30$ steps (top left) with all the nonzero elements of $\boldsymbol{\beta}^0$ being recovered exactly (bottom left). However, once noise is added, there is a significant increase in the number of decimation steps ($L=187$, top right), and the resulting coefficient vector $\boldsymbol{\beta}$ is dense and is dramatically different from $\boldsymbol{\beta}^0$ (bottom right). These results suggest that the stopping criterion used in the original SSD algorithm is no longer appropriate for the linear regression model with noise and needs to be improved.

Adaptive Shortest-Solution guided Decimation (ASSD)

Modified stopping criterion

With an additional examination on the bottom right panel of Figure 2, we find that during the early steps of the decimation process the identification of the nonzero elements of $\boldsymbol{\beta}^0$ is highly accurate. Specifically, there are only four mistakes in identification in the initial 30 decimation steps. In later decimation steps, however, the index k of the largest-magnitude element $\hat{\gamma}_k$ is no longer reliable, in the sense that the true value of β_k^0 may actually be zero. These mis-identified elements are too numerous to be corrected by the subsequent backtracking process of SSD, and the resulting coefficient vector $\boldsymbol{\beta}$ is then quite different from $\boldsymbol{\beta}^0$. These observations indicate the necessity of stopping the decimation process earlier.

Firstly, we set an upper bound $L_{\max}$ for the number of decimation steps. It has been established that the true coefficient vector $\boldsymbol{\beta}^0$ cannot be reconstructed consistently with a sample of size n if there are more than $O\big(n/\ln(n)\big)$ nonzero elements[35]. We therefore take

$$L_{\max}=\frac{n}{\ln n}\,. \tag{15}$$

We repeat the shortest-solution guided decimation only up to $L_{\max}$ steps. Additionally, we estimate $\boldsymbol{\beta}^0$ by the solution of the l_1 minimization problem[36]

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p}\big\|\boldsymbol{\beta}\big\|_1 \quad \text{subject to} \quad \big\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\big\|_2\leq\eta\,. \tag{16}$$

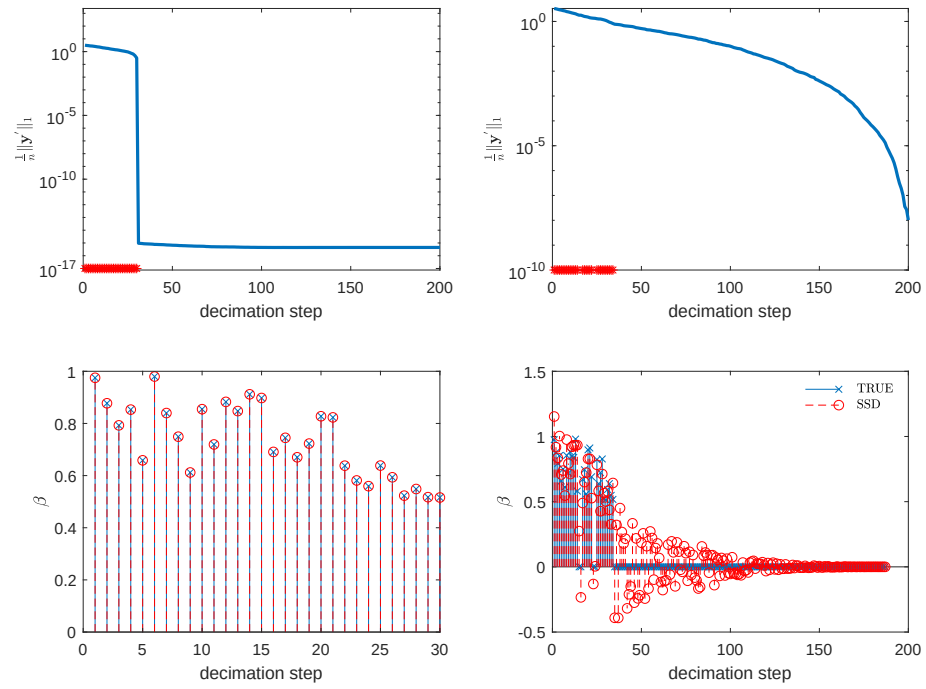

Figure 2. Simulation results of SSD with the naive stopping criterion 10^{-5} on a single uncorrelated Gaussian measurement matrix ($p = 1000$, $n = 200$). Top: trace of l_1-norm of the residual response vector $\boldsymbol{y}'$. The red stars on the horizontal axis signify that the identified element β_k^0 of $\boldsymbol{\beta}^0$, with k being the index of the largest magnitude element $\hat{\gamma}_k$ of $\hat{\boldsymbol{\gamma}}$, is indeed nonzero. Bottom: results of coefficient estimation. The two subfigures on the left panel display the noise-free situation ($\sigma^2 = 0$), and those on the right panel display the noise situation ($\sigma^2 = 1$).

Under certain conditions on the RIP (restricted isometry property) constant of $\boldsymbol{X}$, the estimation error measured in l_2-norm of $\boldsymbol{\beta}^0$ is of the order of $\eta/\sqrt{n}$[36]. Inspired by this insight, we terminate the SSD process earlier than $L_{\max}$ steps once the Euclidean length of the residual response vector (i.e., $\|\boldsymbol{y}'\|_2$) is smaller than a prespecified value η.

Second-stage thresholding after SSD

Even after the early stopping strategy is applied to the decimation process, we find that some of the zero-valued coefficients β_i^0 are still predicted to be nonzero by the algorithm. To reduce this false-positive fraction as much as possible, we propose a second-stage thresholding procedure to the SSD algorithm. The idea is to manually reset some of the coefficients $\beta_i = 0$ if the value predicted by the SSD algorithm is below certain threshold value. This refinement procedure turns out to be rather effective in improving the variable selection accuracy.

Suppose that after early stopping L elements of $\boldsymbol{\beta}$ are assigned with nonzero values, and the indices of all the zero-valued coefficients form a set A (i.e., $\beta_i = 0$ if and only if $i \in A$). We sort the absolute values of these L estimated coefficients in an ascending order (say $|\beta_{r_1}| \leq |\beta_{r_2}| \leq \ldots |\beta_{r_L}|$), and use the first $L/2$ of them to calculate an empirical measure $\hat{\sigma}$ of coefficients uncertainty as

$$\hat{\sigma} = \left(\frac{2}{L} \sum_{j=1}^{L/2} (\beta_{r_j} - m)^2 \right)^{1/2}, \tag{17}$$

where m means the average value of the considered $L/2$ elements, $m = (2/L)\sum_{j=1}^{L/2} \beta_{r_j}$. Notice $\hat{\sigma}$ is distinct in meaning from the the noise magnitude σ of the original model system (1). We adopt a data-driven procedure to determine the optimal thresholding level. First we set

$$\theta_0 = \hat{\sigma}\sqrt{2 \ln p} \tag{18}$$

to be the basic thresholding level[37](see also the initial work on thresholding to wavelet coefficients[38]). Next, we take the actual thresholding level θ to be $\theta = \tau\theta_0$ with τ taking discrete values. As τ increases from zero to a relatively large value

Algorithm 1 Adaptive Shortest-Solution guided Decimation (ASSD) algorithm for the noisy sparse high-dimensional regression problem (1). This algorithm has one adjustable parameter η, which is the prespecified precision level.

Input: $\boldsymbol{X}' = (\boldsymbol{X}'_1, \ldots, \boldsymbol{X}'_p)$: $n \times p$ residual measurement matrix, initially $\boldsymbol{X}' = \boldsymbol{X}$; $\boldsymbol{y}'$: n-dimensional residual response vector, initially $\boldsymbol{y}' = \boldsymbol{y}$; A: index set, initially $A = \{1, 2, \ldots, p\}$; $L_{\max}$: maximum number of decimation steps, $L_{\max} = n/\ln n$; R: a sufficiently large integer bound, e.g., $R = 20$; L: decimation step, initially $L = 0$.

while $\|\boldsymbol{y}'\|_2 > \eta$ and $L < L_{\max}$ **do** ▷ decimation with early-stopping

1. Get the least-squares solution $\hat{\boldsymbol{\gamma}} = \{\hat{\gamma}_i : i \in A\}$ for the linear equation $\sum_{i \in A} \hat{\gamma}_i \boldsymbol{X}'_i = \boldsymbol{y}'$.
2. Get leading index k of $\hat{\boldsymbol{\gamma}}$ by criterion $|\hat{\gamma}_k| \geq |\hat{\gamma}_i|$ $(\forall i \in A)$, and then delete k from set A.
3. Update $L \leftarrow L + 1$, and update $\boldsymbol{X}'_i$ (for every $i \in A$) and $\boldsymbol{y}'$ as

$$\boldsymbol{X}'_i \leftarrow \boldsymbol{X}'_i - \frac{\boldsymbol{X}'^{\top}_i \boldsymbol{X}'_k}{\boldsymbol{X}'^{\top}_k \boldsymbol{X}'_k} \boldsymbol{X}'_k \,, \qquad \boldsymbol{y}' \leftarrow \boldsymbol{y}' - \frac{\boldsymbol{y}'^{\top} \boldsymbol{X}'_k}{\boldsymbol{X}'^{\top}_k \boldsymbol{X}'_k} \boldsymbol{X}'_k \,.$$

end while

Set $\beta_i = 0$ for all $i \in A$, and determine the remaining coefficients β_i $(i \notin A)$ by minimizing $\left\| \boldsymbol{y} - \sum_{i \notin A} \beta_i \boldsymbol{X}_i \right\|_2$.
Sort the L nonzero elements $|\beta_i|$ of $\boldsymbol{\beta}$ in ascending order to compute $\hat{\sigma}$ according to equation (17), and set $\theta_0 = \hat{\sigma}\sqrt{2 \ln p}$.

for $\tau = 0, 0.01, 0.02, \ldots, R$ **do** ▷ second-stage thresholding

1. Set the actual threshold level $\theta = \tau \theta_0$.
2. If $|\beta_i| < \theta$ $(\forall i \notin A)$, then add index i to set A.
3. Update the elements of $\boldsymbol{\beta}$ outside set A by solving the minimization problem (19).
4. Get the BIC index according to equation (20) and check if it is the new minimum (if yes, record the vector $\boldsymbol{\beta}$).

end for

Output: sparse coefficient vector $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^{\top}$ which has the global minimal value of BIC.

R (e.g., $R = 20$), the threshold value θ becomes more and more elevated. At a given value of τ, we first update the index set A by adding some indices i to A if $|\beta_i| < \tau\theta_0$, and then we update the remaining elements of $\boldsymbol{\beta}$ by solving the minimization problem

$$\{\hat{\beta}_i | i \notin A\} = \underset{\{\beta_i | i \notin A\}}{\arg\min} \left\| \boldsymbol{y} - \sum_{i \notin A} \beta_i \boldsymbol{X}_i \right\|_2^2 . \tag{19}$$

Finally, we compute the BIC (Bayesian Information Criterion) index[35] as

$$\mathrm{BIC} = \frac{1}{2} \left\| \boldsymbol{y} - \sum_{i \notin A} \hat{\beta}_i \boldsymbol{X}_i \right\|_2^2 + p_{\mathrm{nz}} \ln n \,, \tag{20}$$

where p_{nz} means the number of nonzero elements in the vector $\boldsymbol{\beta}$, namely $p_{\mathrm{nz}} = p - |A|$. The BIC value is a trade-off between the prediction error and the model complexity. We choose the value of τ such that BIC achieves the minimum value, and consider the corresponding coefficient vector $\boldsymbol{\beta}$ as the final solution of the linear regression problem (1).

An initial demonstration of ASSD performance

We summarize the above ideas in the pseudo-code of Algorithm 1. This ASSD algorithm has two parts: decimation with early stopping, followed by refinement by second-stage thresholding.

Let us work on a small example case to better appreciate the working characteristics of ASSD. We generate an $n \times p$ random Gaussian matrix $\boldsymbol{X}$ with $n = 200$ and $p = 1000$, whose elements are i.i.d. $\mathcal{N}(0, 1)$ distributed. The truth coefficient

vector $\boldsymbol{\beta}^0$ has $s_0 = 30$ nonzero elements, each of which is uniformly distributed in $[0.5, 1]$, and 970 zero elements. The response vector $\boldsymbol{y}$ is generated from the linear regression model (1) with error level $\sigma^2 = 1$. We compare the performance of ASSD with that of the original SSD which does not conduct early-stopping nor the second-stage thresholding, and that of SSD1, which only adopts early-stopping but skips the second-stage thresholding.

The algorithmic results shown in Figure 3 reveal that all these three algorithms assigned good approximate values for the nonzero elements of $\boldsymbol{\beta}^0$. SSD has a high false-positive rate (154 of the zero elements of $\boldsymbol{\beta}^0$ are misclassified as nonzero), and early-stopping dramatically reduces this rate (only 8 false-positive predictions in SSD1). By applying the second-stage thresholding, ASSD achieves zero false-positive rate. In addition, ASSD and SSD1 are more efficient than SSD (SSD, 27.2 s; SSD1, 7.83 s; ASSD, 8.2 s).

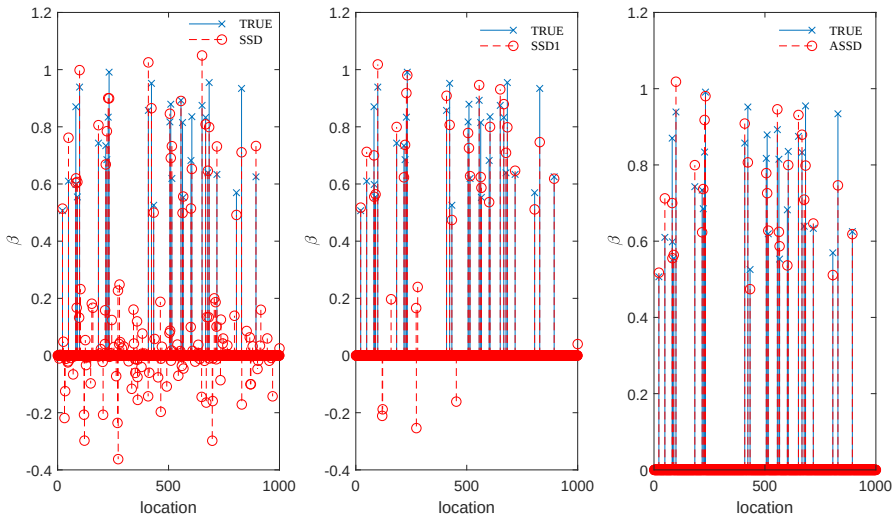

Figure 3. Performance comparison on the original SSD (left), SSD1 which is SSD with early-stopping, and ASSD. The measurement matrix is 200×1000, while the noise level is $\sigma^2 = 1$. The blue crosses mark the 30 nonzero elements of the true coefficient vector β^0, the red circles mark the elements of the estimated coefficient vector β. In ASSD, we set $R = 20$ and $\eta = \sqrt{n}\sigma$.

Results

Model implementation

To better gauge the performance of ASSD, we compare ASSD with four different methods LASSO, VAMP, SIS+LASSO, and ASDAR. We implemented all these methods in Matlab. Our implementation of LASSO uses the function *lasso*. For VAMP, we use the publicly available Matlab package[31]. The algorithm SIS+LASSO first selects $(n \ln n)$ variables based on SIS and then runs LASSO to further reduce the number of falsely identified nonzero coefficients. We implement ASDAR by using the Matlab package *sdar*[29].

For ASSD, we set $R = 20$, $L_{\max} = n/\ln n$ and $\eta = \sqrt{n}\sigma$. For LASSO and SIS+LASSO, the tuning parameters are selected by using 10-fold cross validation. For ASDAR, we set $\tau = 5$ and stop the iteration if the number of identified nonzero elements is greater than $L = 0.5n$, or the residual norm is smaller than $\sqrt{n}\sigma$, or the distance of two subsequent solutions (measured in l_2-norm) is smaller than 1. For VAMP, a small amount of damping is useful when the measurement matrix is ill-conditioned. We set the dampling parameter to be 0.95. Other parameters of VAMP, including the maximum number of iterations, the tolerance for stopping, are the default values in public-domain GAMPmatlab toolbox[31].

We focus on four metrics for algorithmic comparisons: (1) the relative error (RE) of estimation, defined as $\|\boldsymbol{\beta} - \boldsymbol{\beta}^0\|_2 / \|\boldsymbol{\beta}^0\|_2$; (2) the true positive counts (TP) and (3) the false positive counts (FP) of variable selection, and (4) the CPU time in seconds. In each scenario, we calculate the average and standard deviation of these four metrics over 96 independent runs.

Results on three types of measurement matrices

We first consider different types of measurement matrices $\boldsymbol{X}$ of the same size. In our numerical experiments we set $n = 300$ and $p = 2000$. The number of nonzero coefficients in $\boldsymbol{\beta}^0$ is set to be $s_0 = 40$, with each of them being generated from the uniform distribution $\mathscr{U}[0.5, 1]$. Three types of measurement matrix are considered:

- *Correlated Gaussian matrix*: Each row of the matrix $\boldsymbol{X}$ is drawn independently from $\mathcal{N}(0,\boldsymbol{\Sigma})$, where $\Sigma_{ij}=\pi^{|j-i|}$, $1\le i,j\le p$ with $\pi=0$ and 0.7 corresponding to no and strong correlations.
- *Structured matrix*: The matrix $\boldsymbol{X}$ is the product of an $n\times r$ matrix $\boldsymbol{X}_1$ and a $r\times p$ matrix $\boldsymbol{X}_2$. Both X_1 and X_2 are random Gaussian matrix whose elements are independently generated from $\mathcal{N}(0,1)$. The rank r is closely related to the degree of correlation between elements in matrix $\boldsymbol{X}$. When $r\gg n$, the elements in matrix $\boldsymbol{X}$ are weakly correlated or even uncorrelated. As r approaches n from above, the elements in matrix $\boldsymbol{X}$ are more and more correlated. We consider two scenarios: $r=p=2000$ and $r=n+5=305$ corresponding to weakly correlated and highly correlated (or structured) matrices, respectively.
- *Real-world matrix*: We choose the gene expression data from The Cancer Genome Atlas (TCGA) ovarian cancer samples[32] and we use the the dataset provided by two earlier studies[33,34]. The dataset is available at https://bioinformatics.mdanderson.org/Supplements/ResidualDisease/. There are 594 samples and 22,277 genes in the original dataset. We randomly subsample the samples and genes to obtain a 300×2000 measurement matrix $\boldsymbol{X}$.

The response vector $\boldsymbol{y}$ is generated via the linear regression model equation (1), in which the random errors are independently generated from normal distribution with means 0 and variance $\sigma^2=1$.

Correlated Gaussian matrix

Table 1 shows the results on Gaussion matrices. Here and hereinafter, the standard deviations of metrics are shown in the parentheses, and in each column, the numbers in boldface indicate the best performers. It is observed that ASSD has the best performance in variable selection. ASDAR achieve similar performance with ASSD when there is no correlation ($\pi=0$), but it suffers from identifying more false positives when π increases to $\pi=0.7$. For estimation, ASSD again has the best or close-to-the-best performance compared with VAMP. Although VAMP produces a smaller relative error than ASSD when $\pi=0$, its performance deteriorates significantly when the correlation π is high.

Table 1. Simulation results on Gaussian measurement matrices with $p=2000$, $n=300$, $\sigma^2=1$, $s_0=40$, and $\pi=0$ or 0.7.

π	Methods	TP	FP	RE	Time
0	LASSO	**40(0.25)**	109(20.29)	0.34(5.56E-02)	12.74(1.32)
	VAMP	**40(0)**	1(1.87)	**0.07(9.61E-03)**	**0.07(0.04)**
	SIS_LASSO	23(2.19)	26(2.76)	0.69(5.06E-02)	0.56(0.07)
	ASDAR	**40(0)**	**0(0)**	0.08(9.43E-03)	0.13(0.03)
	ASSD	**40(0.52)**	**0(1.35)**	0.10(4.19E-02)	14.94(2.28)
0.7	LASSO	**40(0.45)**	111(22.86)	0.35(6.28E-02)	20.02(3.05)
	VAMP	**40(0.17)**	432(572.29)	6.80(5.83E+01)	**0.16(0.06)**
	SIS_LASSO	18(2.39)	21(3.39)	0.79(5.04E-02)	1.04(0.09)
	ASDAR	39(0.88)	4(3.70)	**0.16(7.68E-02)**	0.20(0.06)
	ASSD	39(1.58)	**1(3.26)**	0.17(1.02E-01)	18.89(3.86)

ASSD shows no advantage in speed. ASSD is similar to LASSO in computation time, but it is much slower than ASDAR and VAMP.

Structured matrices

Results on the structured matrices are reported in Table 2. We see that when the rank number r of the matrix $\boldsymbol{X}$ is large, i.e. $r=p=2000$, VAMP, ASDAR and ASSD are on the top of the list in metrics for variable selection (TP and FP) and the metric for estimation (RE). As the rank number r approaches n ($r=305$), ASDAR becomes less accurate in variable selection and coefficient estimation than VAMP and ASSD. The favorable performance of VAMP is not unexpected because it can achieve Bayes-optimal estimation for a class of structured measurement matrix, namely that of rotationally-invariant matrix. It is encouraging to observe that ASSD performs comparably well in variable selection and estimation, even when the measurement matrix is highly structured.

Real-world matrix

Table 3 shows the results on a real measurement matrix which is a subsample of gene expression data from TCGA ovarian cancer samples. ASSD again has the best performance both in variable selection and in coefficient estimation. LASSO, SIS+LASSO, and VAMP do not work well on the real matrix as they identify too many false positives and produce significantly larger estimation errors. ASDAR is similar to ASSD in terms of estimation error, but it is inferior to ASSD in terms of variable

Table 2. Simulation results on structured measurement matrices with $p = 2000$, $n = 300$, $\sigma^2 = 1$, $s_0 = 40$, and $r = 2000, 305$.

r	Methods	TP	FP	RE	Time
2000	LASSO	**40(0)**	30(11.83)	4.01E-02(3.72E-03)	9.07(2.37)
	VAMP	**40(0)**	**0(0)**	**1.41E-03(1.60E-04)**	**0.08(0.05)**
	SIS+LASSO	22(2.00)	26(3.15)	7.02E-01(4.78E-02)	0.85(0.12)
	ASDAR	**40(0)**	**0(0)**	**1.41E-03(1.61E-04)**	0.12(0.03)
	ASSD	**40(0)**	**0(0)**	**1.41E-03(1.61E-04)**	15.10(2.75)
305	LASSO	**40(0)**	102(20.04)	9.72E-02(2.67E-02)	24.61(8.43)
	VAMP	**40(0)**	**0(0)**	**5.03E-03(7.17E-04)**	**0.10(0.05)**
	SIS+LASSO	14(2.29)	31(3.30)	8.77E-01(4.58E-02)	0.99(0.07)
	ASDAR	38(5.77)	33(45.69)	1.54E-01(3.42E-01)	0.35(0.31)
	ASSD	**40(0)**	**0(0)**	5.04E-03(7.08E-04)	14.09(1.37)

selection. It is indeed quite a remarkable observation that only the ASSD algorithm achieves almost perfect accuracy for this real-world problem instance.

Table 3. Simulation results on a real-world measurement matrix with $p = 2000$, $n = 300$, $\sigma^2 = 1$ and $s_0 = 40$.

Methods	TP	FP	RE	Time
LASSO	**40(0.59)**	131(19.37)	3.46E-01(6.78E-02)	20.76(3.04)
VAMP	**40(0.34)**	98(411.13)	1.07E+02(9.38E+02)	**0.12(0.07)**
SIS+LASSO	7(2.31)	20(3.64)	9.42E-01(3.46E-02)	1.62(0.75)
ASDAR	39(2.04)	8(11.77)	1.59E-01(1.35E-01)	0.22(0.08)
ASSD	**40(0.66)**	**0(0.90)**	**1.18E-01(4.48E-02)**	15.52(1.39)

Influence of model parameters

We now investigate more closely the effect of each of the model parameters (the sample size n, the number of predictors p, and the sparsity level s_0) on the performance of LASSO, VAMP, SIS+LASSO, ASDAR, and ASSD. The same three types of measurement matrices are examined: Gaussion matrix with $\pi = 0.7$, structured matrix with the rank number $r = n+5$, and the real-world matrix. The nonzero elements of $\boldsymbol{\beta}^0$ are i.i.d. random values drawn from the uniform distribution over $[0.5, 1]$. We generate the response vector from the linear regression model (1). The random errors are generated independently from $\mathcal{N}(0, 0.5)$. The simulation results are based on 96 independent repeats.

Figure 4 shows the influence of sample size n on the relative errors (top left panel), true positives (top right panel), false positives (bottom left panel), and probability of exact identification of nonzero coefficients (bottom right panel), when the measurement matrix is a correlated Gaussion one. Results obtained on the other two types of measurement matrices can be found as Supplementary Fig. S1 and Supplementary Fig. S2 online. (For the real-world matrix and the correlated Gaussion matrix, the results of VAMP are too unstable to be shown here and hereinafter.) As expected, the performances of all the methods improve as n increases. For the real-world and the correlated Gaussian matrices, ASDAR and ASSD perform comparably well in estimation accuracy. However, ASSD performs significantly better than ASDAR in accuracy of variable selection. Specifically, ASSD is able to exactly recover the support when $n = 300$, whereas the success probability of ASDAR is only 42%. Similar observations are obtained for the real-world measurement matrix. For the structured measurement matrices, VAMP has the best performance, and ASSD again has close-to-the-best performance compared with ASDAR, LASSO and SIS+LASSO.

Figure 5 shows the influence of number of covariates p on the performances of the four methods for a correlated Gaussion measurement matrix. Data are generated from the model with $s_0 = 30$ and $n = 300$. We see that ASSD always produces the lowest relative errors and FP, and the highest TP. In particular, the probability of exactly recovering the support of the true coefficient vector $\boldsymbol{\beta}^0$ of ASSD is higher than that of the other methods as p increases, which indicates that ASSD is more robust to the number of covariates. Similar observations are also made for the other types of measurement matrices as Supplementary Fig. S3 and Supplementary Fig. S4 online.

The influence of the sparsity level s_0 on the performance of the four methods for a correlated Gaussion measurement matrix is presented in Figure 6. The corresponding results obtained for the other types of matrices are presented as Supplementary Fig. S5 and Supplementary Fig. S6 online. Data are generated with $n = 300$ and $p = 2000$. When the number of nonzero

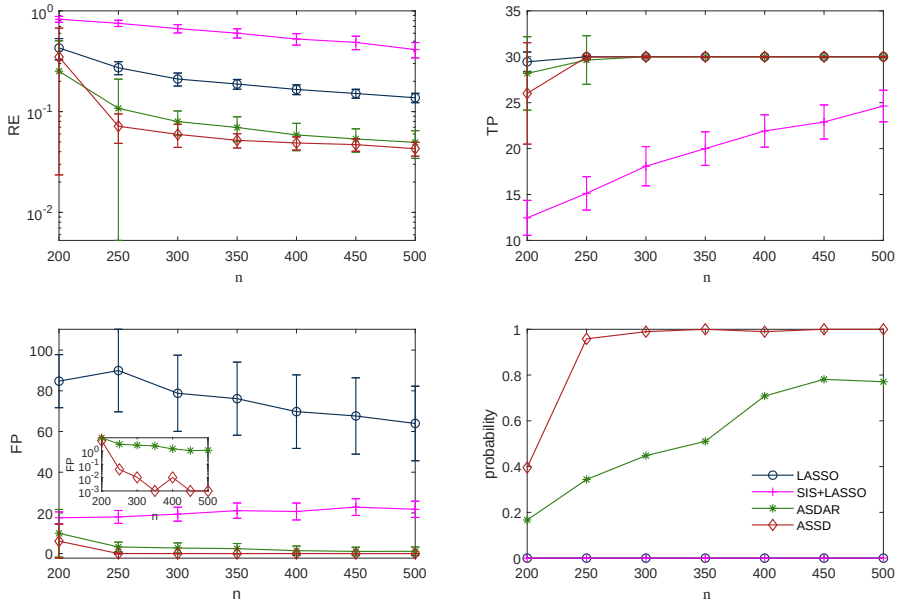

Figure 4. Simulation results on a correlated Gaussion measurement matrix ($\pi = 0.7$, $p = 2000$, $s_0 = 30$, $\sigma^2 = 0.5$): influence of the sample size n on relative errors (top left), true positives (top right), false positives (bottom left) with the inset being a semi-logarithmic plot, and probability of exact identification of nonzero coefficients (bottom right).

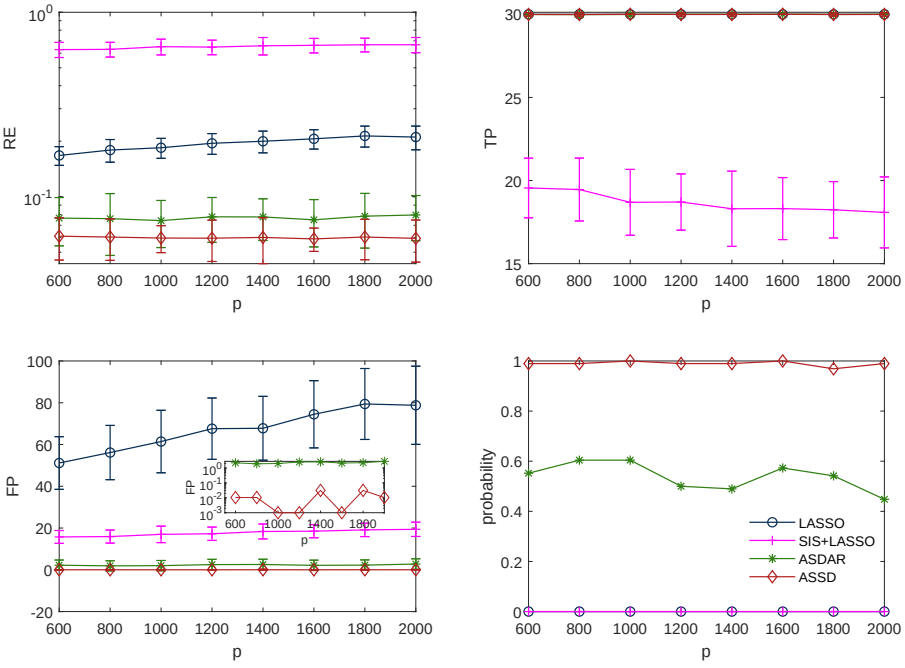

Figure 5. Simulation results on a correlated Gaussion measurement matrix ($\pi = 0.7$, $n = 300$, $s_0 = 30$, $\sigma^2 = 0.5$): influence of the covariates number p on relative errors (top left), true positives (top right), false positives (bottom left) with the inset being a semi-logarithmic plot, and probability of exact identification of nonzero coefficients (bottom right).

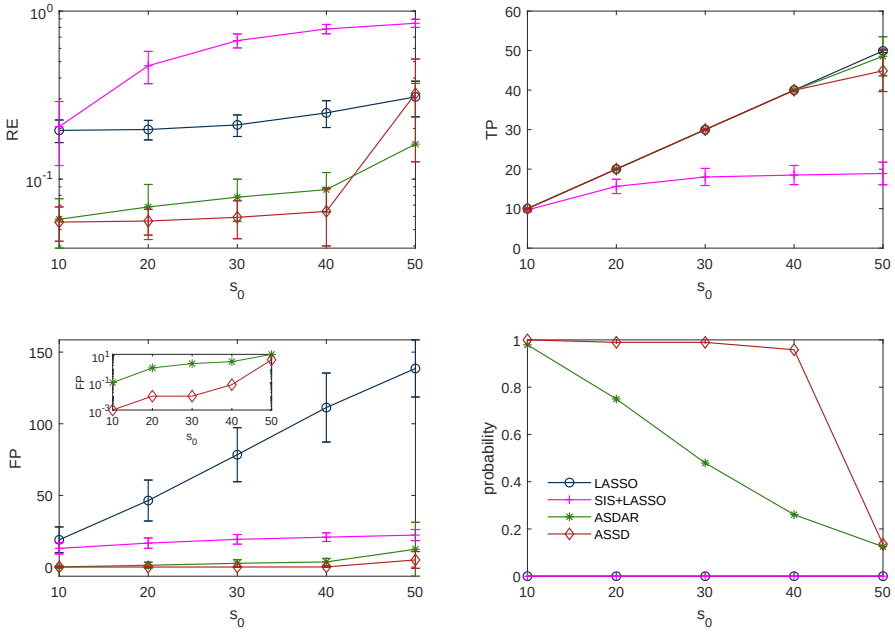

Figure 6. Simulation results on a correlated Gaussion measurement matrix ($\pi = 0.7$, $n = 300$, $p = 2000$, $\sigma^2 = 0.5$): influence of the sparsity level s_0 on relative errors (top left), true positives (top right), false positives (bottom left) with the inset being a semi-logarithmic plot, and and probability of exact identification of nonzero coefficients (bottom right).

elements s_0 increases, the performances of all the four methods become worse. ASSD generally has the best or close-to-the-best performance in accuracy of estimation and variable selection at each value of s_0.

In summary, our simulation results demonstrate that the proposed ASSD is more accurate and robust in variable selection and coefficient estimation than LASSO, VAMP, SIS+LASSO, and ASDAR. This ASSD algorithm is a promising heuristic method for highly correlated random and real-world measurement matrices

Discussion

In this paper, we proposed the adaptive shortest-solution guided decimation (ASSD) algorithm to estimate high-dimensional sparse linear regression models. Compared to the original SSD algorithm which is developed for linear regression models without noise[30], the ASSD algorithm takes into account the effect of measurement noise and adopts an early-stopping strategy and a second-stage thresholding procedure, resulting in significantly better performance in variables selection (which columns $\boldsymbol{X}_i$ are relevant) and coefficients estimation (what are the corresponding regression values β_i). Extensive simulation studies demonstrate that ASSD has favorable performance, and outperforms the comparison methods in variable selection and is competitive with or outperforms VAMP and ASDAR in coefficient estimation. It is robust to the model parameters, and it is especially robust for different types measurement matrices such as those whose entries are highly correlated. These numerical results suggest that ASSD can serve as an efficient and robust tool to real-world sparse estimation problems.

In terms of speed, ASSD is slower than VAMP and ASDAR and this is an issue to be further improved in the future. To accelerate ASSD, on the one hand, we can select a small fraction of elements in coefficient vector instead of just one of them in each decimation step, and on the other hand, we can adopt more delicate early-stopping strategy to further reduce the unnecessary decimation steps. In addition, the rigorous theoretical understanding on ASSD needs to be pursued. We have only considered the linear regression model in this paper. It will be interesting to generalize ASSD to other types of models, such as logistic model and cox model.

Data availability

The data supporting this study are provided within the paper.

References

1. Zhang, Z., Xu, Y., Yang, J., Li, X. & Zhang, D. A survey of sparse representation: algorithms and applications. *IEEE Access* **3**, 490–530 (2015).
2. Rani, M., Dhok, S. B. & Deshmukh, R. B. A systematic review of compressive sensing: Concepts, implementations and applications. *IEEE Access* **6**, 4875–4894 (2018).
3. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. Royal Stat. Soc. (Series B)* **58**, 267–288 (1996).
4. Chen, S. S., Donoho, D. L. & Saunders, M. A. Atomic decomposition by basis pursuit. *SIAM J. on Sci. Comput.* **20**, 33–61 (1998).
5. Osborne, M. R., Presnell, B. & Turlach, B. A. A new approach to variable selection in least squares problems. *IMA J. Numer. Analysis* **20**, 389–403 (2000).
6. Osborne, M. R. *et al.* On the lasso and its dual. *J. Comput. Graph. Stat.* **9**, 319–337 (2000).
7. Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. Least angle regression. *The Annals Stat.* **32**, 385–479 (2004).
8. Friedman, J., Hastie, T., Höfling, H. & Tibshirani, R. Pathwise coordinate optimization. *The Annals Appl. Stat.* **1**, 302–332 (2007).
9. Agarwal, A., Negahban, S. & Wainwright, M. J. Fast global convergence of gradient methods for high-dimensional statistical recovery. *Annals Stat.* **40**, 2452–2482 (2012).
10. Fan, J. & Li, R. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Stat. Assoc.* **96**, 1348–1360 (2001).
11. Zhang, C. H. Nearly unbiased variable selection under minimax concave penalty. *Annals Stat.* **38**, 894–942 (2010).
12. Donoho, D. L., Maleki, A. & Montanari, A. Message-passing algorithms for compressed sensing. *Proc. Natl. Acad. Sci. USA* **106**, 18914–18919 (2009).
13. Ziniel, J. & Schniter, P. Efficient high-dimensional inference in the multiple measurement vector problem. *IEEE Transactions on Signal Process.* **61**, 340–354 (2013).
14. Vila, J. P., Member, S., Schniter, P. & Member, S. Expectation-maximization gaussian-mixture approximate message passing. *IEEE Transactions on Signal Process.* **61**, 4658–4672 (2013).
15. Metzler, C. A., Maleki, A. & Baraniuk, R. G. From denoising to compressed sensing. *IEEE Transactions on Inf. Theory* **62**, 5117–5144 (2016).
16. Rangan, S. Generalized approximate message passing for estimation with random linear mixing. *IEEE Int. Symp. on Inf. Theory - Proc.* 2168–2172 (2011).
17. Manoel, A., Krzakala, F., Tramel, E. W. & Zdeborová, L. Sparse Estimation with the Swept Approximated Message-Passing Algorithm. *arXiv preprint* (2014).
18. Vila, J., Schniter, P., Rangan, S., Krzakala, F. & Zdeborová, L. Adaptive damping and mean removal for the generalized approximate message passing algorithm. *2015 IEEE Int. Conf. on Acoust. Speech Signal Process. (ICASSP)* **2015-Augus**, 2021–2025 (2015).
19. Rangan, S., Fletcher, A. K., Schniter, P. & Kamilov, U. S. Inference for generalized linear models via alternating directions and bethe free energy minimization. *IEEE Transactions on Inf. Theory* **63**, 676–697 (2017).
20. MA, J. & PING, L. Orthogonal amp. *IEEE Access* **5**, 2020–2033 (2017).
21. Rangan, S., Schniter, P. & Fletcher, A. K. Vector approximate message passing. *2017 IEEE Int. Symp. on Inf. Theory (ISIT)* 1588–1592 (2017).
22. Chen, S., Billings, S. A. & Luo, W. Orthogonal least squares methods and their application to non-linear system identification. *Int. J. Control.* **50**, 1873–1896 (1989).
23. Mallat, S. G. & Zhang, Z. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Process.* **41**, 3397–3415 (1993).
24. Fan, J. & Lv, J. Sure independence screening for ultrahigh dimensional feature space. *J. Royal Stat. Soc. Ser. B: Stat. Methodol.* **70**, 849–911 (2008).
25. Fan, J. & Song, R. Sure independence screening in generalized linear models with NP-dimensionality. *Annals Stat.* **38**, 3567–3604 (2010).

26. Blumensath, T. & Davies, M. E. Iterative hard thresholding for compressed sensing. *Appl. Comput. Harmon. Analysis* **27**, 265–274 (2009).

27. Jain, P., Tewari, A. & Kar, P. On iterative hard thresholding methods for high-dimensional m-estimation. *Adv. Neural Inf. Process. Syst.* **27**, 685–693 (2014).

28. Jiao, Y., Jin, B. & Lu, X. A primal dual active set with continuation algorithm for the l0-regularized optimization problem. *Appl. Comput. Harmon. Analysis* **39**, 400–426 (2015).

29. Huang, J., Jiao, Y., Liu, Y. & Lu, X. A constructive approach to l0 penalized regression. *J. Mach. Learn. Res.* **19**, 1–37 (2018).

30. Shen, M., Zhang, P. & Zhou, H. J. Compressed sensing by shortest-solution guided decimation. *IEEE Access* **6**, 5564–5572 (2018).

31. "Generalized approximate message passing". Source-Forge.net project GAMPmatlab available on-line at http://gampmatlab.sourceforge.net/ (2020).

32. The Cancer Genome Atlas Research Network. Integrated genomic analyses of ovarian carcinoma. *Nature* **474**, 609–615 (2011).

33. Wang, F., Mukherjee, S., Richardson, S. & Hill, S. M. High-dimensional regression in practice: an empirical study of finite-sample prediction, variable selection and ranking. *Stat. Comput.* **30**, 697719 (2020).

34. Tucker, S. L. *et al.* Molecular biomarkers of residual disease after surgical debulking of high-grade serous ovarian cancer. *Clin. Cancer Res.* **20**, 3280–3288 (2014).

35. Wang, H., Li, B. & Leng, C. Shrinkage tuning parameter selection with a diverging number of parameters. *J. Royal Stat. Soc. Ser. B: Stat. Methodol.* **71**, 671–683 (2009).

36. Candès, E. J., Romberg, J. K. & Tao, T. Stable signal recovery from incomplete and inaccurate measurements. *Commun. on Pure Appl. Math.* **59**, 1207–1223 (2005).

37. Guo, J., Hu, J., Jing, B. Y. & Zhang, Z. Spline-lasso in high-dimensional linear regression. *J. Am. Stat. Assoc.* **111**, 288–297 (2016).

38. Donoho, D. L. & Johnstone, J. M. Ideal spatial adaptation by wavelet shrinkage. *Biometrika* **81**, 425–455 (1994).

Acknowledgements

This study was partly supported by the Bureau of Statistics of China (2019LZ11), Fund for building world-class universities (disciplines) of Renmin University of China, the National Natural Science Foundation of China Grants No. 11975295 and No. 12047503, and the Chinese Academy of Sciences Grants No. QYZDJ-SSW-SYS018 and XDPD15.

The computer resources were provided by Public Computing Cloud Platform of Renmin University of China.

Author contributions statement

Y.-F.S., H.-J.Z., and X.Y. conceived research; X.Y. performed research; X.Y., Y.-F. S., and H.-J.Z. wrote and reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Accession codes: The ASSD code is available as a Matlab code at Github: `https://github.com/sugar-xue/ASSD`.

Supplementary text and figures: Accompanied with the maintext and are accessible online.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- supplementary.pdf