

**Statistica Sinica Preprint No: SS-2018-0075**

<b>Title</b>	The Broken Adaptive Ridge Procedure and Its Applications
<b>Manuscript ID</b>	SS-2018-0075
<b>URL</b>	<a href="http://www.stat.sinica.edu.tw/statistica/">http://www.stat.sinica.edu.tw/statistica/</a>
<b>DOI</b>	10.5705/ss.202018.0075
<b>Complete List of Authors</b>	Linlin Dai Kani Chen and Gang Li
<b>Corresponding Author</b>	Linlin Dai
<b>E-mail</b>	ldaiab@connect.ust.hk

# The Broken Adaptive Ridge Procedure and Its Applications

Linlin Dai, Kani Chen and Gang Li

*Southwestern University of Finance and Economics,  
Hong Kong University of Science and Technology and  
University of California, Los Angeles*

*Abstract:* In this study, we employ the *broken adaptive ridge method* to estimate the lower-dimensional patterns of the coefficients in regression models. Based on a reweighted  $\ell_2$ -penalization, the new method simultaneously recovers the true sparsity and the inherent structures of the features, making it theoretically and practically appealing. The resulting estimate is shown to enjoy the oracle property. The proposed method also contains a set of variable selection or pattern estimation methods. As a special case, the fused broken adaptive ridge, which penalizes the differences between adjacent coefficients, is thoroughly discussed, with applications to signal approximation and image processing. The associated algorithms are numerically easy to implement. Simulation studies and real-data analyses illustrate the advantages of the proposed method over the fused lasso method.

*Key words and phrases:* Oracle estimator, Linear regression, Re-weighted  $\ell_2$ -

penalization.

## 1. Introduction

Identifying the underlying dynamics of a data set of interest is an important task in many applications, including, for instance, denoising, forecasting, filtering, and even more sophisticated analyses in machine learning research. In a high-dimensional setting, the underlying patterns of the regression coefficients usually have a lower-dimensional structure. In particular, when the candidate variables can be treated individually, the true coefficients are assumed to contain many zeros. Many state-of-the-art variable selection methods have been developed, such as lasso (Tibshirani, 2011), bridge penalty (Fu, 1998; Huang *et al.*, 2008a, 2009), SCAD (Fan and Li, 2001), elastic net (Zou and Hastie, 2005), and MCP (Zhang, 2010), among many others. These methods have gained much attention in recent years and are widely used to find a parsimonious model. In this study, we focus on variables that naturally have some local structures, such as piecewise constancy, a linear trend, or being grouped. Our goal is to reduce the dimension of the covariates and to estimate their underlying structures.

The  $\ell_0$ -penalized regression is one of the most natural methods for variable selection, which directly penalizes the cardinality of a model. Rinaldo (2009) applied the  $\ell_0$ -penalization to identify a piecewise constant function

to approximate a signal. Owing to the lack of convexity, the  $\ell_0$ -penalization procedure is computationally difficult to implement, especially for high-dimensional data sets. A body of literature is devoted to approaches based on penalties such as  $\ell_1$ -norm penalties and  $\ell_2$ -norm penalties. The fused lasso method (Tibshirani *et al.*, 2005), based on the  $\ell_1$ -penalization, simultaneously captures sudden jumps and infers nonzero segments in a noisy signal or gene sequence. This novel approach uses the  $\ell_1$  and the fusion (or total variation) penalties, which favors solutions that are both sparse and piecewise constant. As an extension, the two-dimensional (2D) fused lasso (Tibshirani and Taylor, 2011) is introduced in image denoising. From an algorithmic viewpoint, the fused lasso penalization has its roots in the well-known total variation method (Rudin *et al.*, 1992), which had a significant impact on the modern imaging science. For more recent developments of the fused lasso and its variants in network inferences, see Shen and Huang (2010), Zhu *et al.* (2013), Wang *et al.* (2016), Shin *et al.* (2016), and Tang and Song (2016). On the other hand, for grouped data, such as assayed genes or proteins in biological applications, Yuan and Lin (2006) invented the group lasso methods by imposing the  $\ell_2$ -penalty on the coefficients within each group. Simon *et al.* (2013) studied a sparse group lasso method, which yields solutions that are sparse at both the group and the

individual feature levels. Other advancements that use  $\ell_p$ -penalization to capture local structures of coefficients can be found in Eilers (2003), Rippe *et al.* (2012), Price *et al.* (2015), and Lam *et al.* (2016). Despite their impressive performance in empirical studies, a theoretical justification of the oracle property (Fan and Li, 2001) of many of them remains challenging.

In this study, instead of finding a desirable solution in a single step, we propose an iterative reweighted  $\ell_2$ -penalization procedure, referred to as the *broken adaptive ridge* (BAR) method. The proposed method has several distinctive features compared with other existing methods in the literature. First, it is in a general form, in the sense that it can be used to estimate any local linear structure of regression coefficients. Some special cases of the BAR method, such as the fused BAR method, are introduced, with applications to signal processing, gene detection, trend filtering, or image denoising. Second, the method can simultaneously produce a sparse solution and estimate the underlying pattern of covariates. Moreover, under certain conditions, it is shown that the BAR procedure converges to a fixed point, and that the resulting estimate possesses the oracle property; that is, it performs as well as if the correct underlying model were given in advance. Because the adaptive objective function is strictly convex and differentiable, the iterative procedure is easy to implement with a closed-

form iterative function. To avoid computational overflows in each iterative step, we establish efficient algorithms using the Lagrange multiplier technique. The results of numerical studies demonstrate that, compared with the fused lasso, the fused BAR method exhibits a good performance in terms of variable selection and structure estimation.

The rest of the paper is organized as follows. The BAR method is described in Section 2, along with its special cases. Section 3 presents the oracle property of the proposed method. We establish a general algorithm for the BAR method in Section 4. Numerical studies on signal approximation and image processing are conducted in Sections 5 and 6, respectively. All technical proofs are provided in the Supplementary Material.

## 2. Broken adaptive ridge procedure

Consider the linear model

$$\mathbf{y} = \sum_{j=1}^{p_n} \beta_j \mathbf{x}_j + \boldsymbol{\varepsilon},$$

where  $\mathbf{y} \in \mathbb{R}^n$  is a response variable,  $\mathbf{x}_j \in \mathbb{R}^n$  are feature vectors, and  $\boldsymbol{\varepsilon}$  is a vector of independent and identically distributed random variables with mean zero and finite variance  $\sigma^2$ . Suppose that the response variable  $\mathbf{y} = (y_1, \dots, y_n)$  is centered, and that the covariate matrix  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_{p_n})$  is standardized by column vectors. We wish to recover the sparsity and the underlying patterns of the feature vectors. Throughout this paper,  $\|\cdot\|$

represents the Euclidean norm of a vector and the spectral norm of a matrix.

Let  $\mathbf{d}_k \in \mathbb{R}^{p_n}$ , for  $k = 1, \dots, K_n$ , be nonzero column vectors, implying prior knowledge of the data structure. Define

$$\begin{aligned} \mathbf{g}(\tilde{\boldsymbol{\beta}}) &\equiv \arg \min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda_n \sum_{k=1}^{K_n} \frac{(\mathbf{d}'_k \boldsymbol{\beta})^2}{c_k^2(\tilde{\boldsymbol{\beta}})} \\ &= \arg \min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda_n \boldsymbol{\beta}' \mathbf{D}(\tilde{\boldsymbol{\beta}}) \boldsymbol{\beta}, \end{aligned} \quad (2.1)$$

where

$$\mathbf{D}(\tilde{\boldsymbol{\beta}}) = \sum_{k=1}^{K_n} \frac{\mathbf{d}_k \mathbf{d}'_k}{c_k^2(\tilde{\boldsymbol{\beta}})}, \text{ and } c_k(\tilde{\boldsymbol{\beta}}) = \mathbf{d}'_k \tilde{\boldsymbol{\beta}}.$$

From the convexity and differentiability of the objective function in (2.1),

we have that

$$\mathbf{g}(\tilde{\boldsymbol{\beta}}) = \{\mathbf{X}'\mathbf{X} + \lambda_n \mathbf{D}(\tilde{\boldsymbol{\beta}})\}^{-1} \mathbf{X}'\mathbf{y}. \quad (2.2)$$

In principle, the ridge estimator

$$\hat{\boldsymbol{\beta}}(\text{ridge}) = (\mathbf{X}'\mathbf{X} + \xi \mathbf{I})^{-1} \mathbf{X}'\mathbf{y}$$

is chosen as the initial value, where  $\xi > 0$  is a tuning parameter. The

proposed estimator is thus referred to as the *broken adaptive ridge* (BAR)

estimator, which is defined as the limit of the iterative algorithm  $\hat{\boldsymbol{\beta}}^{(j)} =$

$\mathbf{g}\{\hat{\boldsymbol{\beta}}^{(j-1)}\}$ ; that is,

$$\hat{\boldsymbol{\beta}}(\text{BAR}) = \lim_{j \rightarrow \infty} \hat{\boldsymbol{\beta}}^{(j)}. \quad (2.3)$$

Because the subsequent updates  $\mathbf{d}'_k \hat{\boldsymbol{\beta}}^{(j)}$  usually do not yield any zeros, the

weights  $\{c_k(\tilde{\boldsymbol{\beta}})\}^{-2}$  in each iteration are well defined. Note that the data-

dependent weight  $\{c_k(\tilde{\boldsymbol{\beta}})\}^{-2}$  is more useful than a constant weight  $c^{-2}$ . As the sample size grows, the weights for the zero  $\mathbf{d}'_k\boldsymbol{\beta}$  tend to infinity, whereas those for the nonzero  $\mathbf{d}'_k\boldsymbol{\beta}$  converge to finite constants. In this sense, the proposed BAR procedure and the adaptive lasso (Zou, 2006) are similar in spirit. As pointed out by a reviewer, the BAR method provides us with new insights into the ridge penalty: it can produce a sparse solution and estimate the local structures of predictors using an iterative procedure.

Noting that the term  $\mathbf{d}'_k\boldsymbol{\beta}$  represents any linear combination of  $\boldsymbol{\beta}$ , this allows us to design the vector  $\mathbf{d}_k$  in line with some believed structure or geometry in the feature vectors, such as sparsity, piecewise constancy, and the grouping effect. We present below a set of illustrative examples that motivate our work on the BAR procedure.

**Example 2.1** (*Broken adaptive ridge estimator for variable selection*). Let  $K_n = p_n$  and  $\mathbf{d}_j = \mathbf{e}_j$ , where  $\mathbf{e}_j$  is the standard basis vector with the  $j$ th component equal to one. The design of  $\mathbf{d}_k$  only encourages the sparsity of the coefficients, and virtually ignores any other underlying patterns of the feature vectors. As a result, the BAR method is appropriate for selecting selection for those variables that can be treated individually.

**Example 2.2** (*Fused broken adaptive ridge estimator*). Setting  $\mathbf{X} = \mathbf{I}$  yields an interesting, but highly nontrivial class of problems that includes

signal approximation, gene detection, and image denoising. In signal approximation, a noisy signal is usually approximated by a piecewise constant function. A variety of denoising methods have been developed, including lowess (Cleveland, 1979), kernel estimators (Gasser *et al.*, 1985; Müller and Stadtmüller, 1987), penalized smoothing splines (Ruppert *et al.*, 2009), Markov random field (Geman and Geman, 1984), and wavelets (Donoho and Johnstone, 1994; Chang *et al.*, 2000). To encourage the underlying sparse or blocky structure of  $\mathbf{y}$ , we set  $\mathbf{d}_j = \mathbf{e}_j$ , for  $j = 1, 2, \dots, p_n$ , and the remaining  $\mathbf{d}_j = (0, \dots, -1, 1, \dots, 0)'$ , with the  $(j - p_n)$ th element being -1 and the  $(j - p_n + 1)$ th element being one. In this way, the expression (2.1) can be written as

$$\mathbf{g}(\tilde{\boldsymbol{\beta}}) = \arg \min_{\boldsymbol{\beta}} \|\mathbf{y} - \boldsymbol{\beta}\|^2 + \lambda_1 \sum_{i=1}^n \frac{\beta_i^2}{\tilde{\beta}_i^2} + \lambda_2 \sum_{i=2}^n \frac{(\beta_i - \beta_{i-1})^2}{(\tilde{\beta}_i - \tilde{\beta}_{i-1})^2}, \quad (2.4)$$

where  $\lambda_1 > 0$  and  $\lambda_2 > 0$  are tuning parameters. We refer to the limit of the iterative procedure based on (2.4) as the 1D fused BAR estimator, because the penalties are imposed on both the coefficients and the differences between the adjacent coefficients. A general form of the 1D fused BAR method is induced by allowing the design matrix  $\mathbf{X}$  to be arbitrary. In a similar fashion, the 2D fused BAR estimator has the iterative function

$$\mathbf{g}(\tilde{\boldsymbol{\beta}}) = \arg \min_{\boldsymbol{\beta}} \|\mathbf{y} - \boldsymbol{\beta}\|^2 + \lambda_1 \sum_{i=1}^n \frac{\beta_i^2}{\tilde{\beta}_i^2} + \lambda_2 \sum_{(i,j) \in E} \frac{(\beta_i - \beta_j)^2}{(\tilde{\beta}_i - \tilde{\beta}_j)^2}, \quad (2.5)$$

where  $E$  is the edge set of the graph. It is seen that the second penalty term on the right-hand side of (2.5) favors the flatness of the proximal coefficients. Therefore, the 2D fused BAR estimator is useful for coping with the adjacent pixels in image denoising.

**Example 2.3** (*Broken adaptive ridge trend filter*). Identifying the unknown underlying trend of a given noisy signal or sequence is of great importance for a wide range of applications. In many cases, the signal can be approximated by piecewise linear trends. To both select and estimate the trend's components, we take into account the optimization rule

$$\mathbf{g}(\tilde{\boldsymbol{\beta}}) = \arg \min_{\boldsymbol{\beta}} \|\mathbf{y} - \boldsymbol{\beta}\|^2 + \lambda_1 \sum_{i=1}^n \frac{\beta_i^2}{\tilde{\beta}_i^2} + \lambda_2 \sum_{i=2}^{n-1} \frac{(\beta_{i-1} - 2\beta_i + \beta_{i+1})^2}{(\tilde{\beta}_{i-1} - 2\tilde{\beta}_i + \tilde{\beta}_{i+1})^2}, \quad (2.6)$$

where  $\lambda_1 > 0$  and  $\lambda_2 > 0$  are tuning parameters. It is clear that (2.6) is a special case of the iterative function (2.1). The second penalty on the right-hand side of (2.6) constrains the slopes between two consecutive coefficients, resulting in a solution that has fewer linear segments.

### 3. Oracle properties

In this section, we investigate the oracle property of the BAR estimator.

Assume that the true  $\boldsymbol{\beta}_0$  satisfies

$$\mathbf{d}'_k \boldsymbol{\beta}_0 \neq 0 \text{ for } k = 1, \dots, q_n,$$

$$\mathbf{d}'_k \boldsymbol{\beta}_0 = 0 \text{ for } k = q_n + 1, \dots, K_n,$$

where  $\|\mathbf{d}_k\| \neq 0$ . Let  $\mathfrak{D}$  denote the space spanned by the vectors  $\mathbf{d}_{q_n+1}, \dots, \mathbf{d}_{K_n}$  and the dimensionality of  $\mathfrak{D}$  be  $(p_n - m_n)$ , where  $m_n$  is the dimensionality of the subspace orthogonal to  $\mathfrak{D}$ . There exists an orthonormal basis of  $\mathbb{R}^{p_n}$ ,  $\mathbf{T} = (\mathbf{T}_1; \mathbf{T}_2) = (\mathbf{u}_1, \dots, \mathbf{u}_{m_n}; \mathbf{u}_{m_n+1}, \dots, \mathbf{u}_{p_n})$ , such that  $\mathbf{u}_{m_n+1}, \dots, \mathbf{u}_{p_n} \in \mathfrak{D}$ . Then,

$$\mathbf{u}'_j \boldsymbol{\beta}_0 = 0 \quad \text{for } j = m_n + 1, \dots, p_n,$$

$$\mathbf{u}'_i \mathbf{d}_k = 0 \quad \text{for } i = 1, \dots, m_n \text{ and } k = q_n + 1, \dots, K_n.$$

Let  $\mathbf{X}_1 = \mathbf{X}\mathbf{T}_1$ ,  $\mathbf{X}_2 = \mathbf{X}\mathbf{T}_2$ ,  $\boldsymbol{\Sigma}_n = n^{-1}\mathbf{X}'\mathbf{X}$ , and  $\tilde{\boldsymbol{\Sigma}}_{n1} = n^{-1}\mathbf{X}'_1\mathbf{X}_1$ . For simplicity of notation, we write  $\hat{\boldsymbol{\beta}}(\text{BAR})$  as  $\hat{\boldsymbol{\beta}}$  and omit the tilde on  $\boldsymbol{\beta}$  in (2.2). Define  $b_n \equiv \min_{1 \leq k \leq q_n} |\mathbf{d}'_k \boldsymbol{\beta}_0|$ .

The following regularity conditions are assumed:

(A1)  $0 < 1/C < \lambda_{\min}(\boldsymbol{\Sigma}_n) \leq \lambda_{\max}(\boldsymbol{\Sigma}_n) < C < \infty$ , for some  $C > 1$ ;

(A2) As  $n \rightarrow \infty$ ,

$$\frac{m_n}{n} \rightarrow 0, \quad \frac{\lambda_n}{p_n} \rightarrow \infty, \quad \frac{\lambda_n q_n}{b_n^2 \sqrt{n}} \rightarrow 0, \quad \frac{p_n}{n b_n^2} \rightarrow 0;$$

(A3) For  $1 \leq k \leq q_n$ ,  $0 < \|\mathbf{d}_k\| \leq c_0 < \infty$ , for some constant  $c_0$ .

(A4) The initial estimator satisfies  $\|\hat{\boldsymbol{\beta}}^{(0)} - \boldsymbol{\beta}_0\| = O_p\{(p_n/n)^{1/2}\}$ .

Condition (A1) assumes that the  $\ell_2$ -norm of the covariance matrix  $\boldsymbol{\Sigma}_n$  is bounded away from zero and infinity. Condition (A2) restricts the number of covariates, number of nonzero linear combinations of covariates, tuning

parameter, and smallest nonzero linear combination. It is also made to ensure that the nonzero  $\mathbf{d}'_k \boldsymbol{\beta}_0$  are identifiable. Condition (A3) ensures the simplicity of the proof and is satisfied for many commonly-used penalties, such as the fusion penalty and the trend filter penalty. For high-dimensional data,  $\|\mathbf{d}_k\|$  would be allowed to diverge to infinity at some rate as  $n \rightarrow \infty$ . Such relaxation would not necessarily affect the asymptotic properties of the BAR estimate, because the penalty term in the first line of (2.1) remains the same when its numerator and denominator divided simultaneously by  $\|\mathbf{d}_k\|^2$ . The initial value needs to satisfy condition (A4).

**Lemma 1.** *Suppose that regularity conditions (A1)–(A4) are satisfied. For any positive sequence  $\delta_n \rightarrow \infty$ , such that  $\lambda_n/(\delta_n p_n) \rightarrow \infty$ , define  $\mathfrak{B} \equiv \{\boldsymbol{\beta} \in \mathbb{R}^{p_n} : \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\| \leq \delta_n \sqrt{p_n/n}\}$ . Then, with probability tending to one:*

(a)  $\mathbf{g}(\boldsymbol{\beta})$  is a mapping from  $\mathfrak{B}$  to itself;

(b)

$$\sup_{\boldsymbol{\beta} \in \mathfrak{B}} \frac{\|\mathbf{T}'_2 \mathbf{g}(\boldsymbol{\beta})\|}{\|\mathbf{T}'_2 \boldsymbol{\beta}\|} < \frac{1}{C_0}, \quad \text{for some constant } C_0 > 1. \quad (2.7)$$

**Remark 1.** The statement (2.7) reveals that  $\lim_{k \rightarrow \infty} \mathbf{T}'_2 \hat{\boldsymbol{\beta}}^{(k)} \equiv \mathbf{T}'_2 \hat{\boldsymbol{\beta}} = \mathbf{0}$ , with probability tending to one. In other words, the BAR estimator is zero-consistent in the sense that those zero linear combinations of coefficients are exactly zero as  $n \rightarrow \infty$ . Additionally, the result that  $\mathbf{g}(\cdot)$  is a mapping of  $\mathfrak{B}$  to itself is necessary for the convergence of  $\hat{\boldsymbol{\beta}}^{(k)}$ .

On the other hand, because  $\mathbf{T}'_2\boldsymbol{\beta}_0 = \mathbf{0}$ , the regression model is reduced to

$$\mathbf{y} = \mathbf{X}\mathbf{T}_1\mathbf{T}'_1\boldsymbol{\beta}_0 + \boldsymbol{\varepsilon}. \quad (2.8)$$

Define

$$\begin{aligned} \mathbf{f}(\mathbf{T}'_1\tilde{\boldsymbol{\beta}}) &\equiv \arg \min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}_1\mathbf{T}'_1\boldsymbol{\beta}\|^2 + \lambda_n \sum_{k=1}^{q_n} \frac{(\mathbf{d}'_k\mathbf{T}_1\mathbf{T}'_1\boldsymbol{\beta})^2}{\tilde{c}_k^2(\mathbf{T}'_1\tilde{\boldsymbol{\beta}})}, \\ &= \arg \min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}_1\mathbf{T}'_1\boldsymbol{\beta}\|^2 + \lambda_n \boldsymbol{\beta}'\tilde{\mathbf{D}}(\mathbf{T}'_1\tilde{\boldsymbol{\beta}})\boldsymbol{\beta}, \end{aligned}$$

where

$$\tilde{c}_k(\mathbf{T}'_1\boldsymbol{\beta}) = \mathbf{d}'_k\mathbf{T}_1\mathbf{T}'_1\boldsymbol{\beta} \quad \text{and} \quad \tilde{\mathbf{D}}(\mathbf{T}'_1\boldsymbol{\beta}) = \mathbf{T}'_1 \sum_{k=1}^{q_n} \frac{\mathbf{d}_k\mathbf{d}'_k}{\tilde{c}_k^2(\mathbf{T}'_1\boldsymbol{\beta})} \mathbf{T}_1.$$

Similarly, by the gradient rule, we obtain

$$\mathbf{f}(\mathbf{T}'_1\boldsymbol{\beta}) = \{\mathbf{X}'_1\mathbf{X}_1 + \lambda_n\tilde{\mathbf{D}}(\mathbf{T}'_1\boldsymbol{\beta})\}^{-1}\mathbf{X}'_1\mathbf{y}. \quad (2.9)$$

The asymptotic normality of the nonzero linear combinations  $\mathbf{d}'_k\hat{\boldsymbol{\beta}}$ , for  $k = 1, \dots, q_n$ , are shown in Lemma 2 below.

**Lemma 2.** *Suppose that regularity conditions (A1)–(A4) are satisfied. For any  $q_n$ -vector  $\mathbf{a}_n$  with  $\|\mathbf{a}_n\| \leq 1$ , let  $s_n^2 = \sigma^2\mathbf{a}'_n\tilde{\boldsymbol{\Sigma}}_n^{-1}\mathbf{a}_n$ . Define  $\mathfrak{B}_1 = \{\mathbf{T}'_1\boldsymbol{\beta} \in \mathbb{R}^{m_n} : \|\mathbf{T}'_1\boldsymbol{\beta} - \mathbf{T}'_1\boldsymbol{\beta}_0\| \leq \delta_n\sqrt{p_n/n}\}$ , and assume that  $\inf_{\boldsymbol{\beta} \in \mathfrak{B}_1} (\mathbf{d}'_k\mathbf{T}_1\mathbf{T}'_1\boldsymbol{\beta})^2 \geq c_1(\mathbf{d}'_k\mathbf{T}_1\boldsymbol{\theta}_0)^2$ , for  $1 \leq k \leq q_n$ , where  $\boldsymbol{\theta}_0 = \mathbf{T}'_1\boldsymbol{\beta}_0$ . Then, in region  $\mathfrak{B}_1$ , with probability tending to one, there exists a unique fixed point of  $\mathbf{f}(\cdot)$ , denoted*

by  $\hat{\boldsymbol{\theta}}^\circ$ . Furthermore, as  $n \rightarrow \infty$ ,

$$\sqrt{n}s_n^{-1}\mathbf{a}'_n(\hat{\boldsymbol{\theta}}^\circ - \mathbf{T}'_1\boldsymbol{\beta}_0) \rightarrow \mathcal{N}(0, 1),$$

with probability tending to one.

**Remark 2.** Lemma 2 shows the existence and uniqueness of the fixed point of  $\mathbf{f}(\cdot)$ , defined as (2.9). The asymptotic properties of the fixed point  $\hat{\boldsymbol{\theta}}^\circ$  imply that  $\hat{\boldsymbol{\theta}}^\circ$  is consistent with the true  $\mathbf{T}'_1\boldsymbol{\beta}_0$ . To show that the BAR estimator is asymptotically normal, it suffices to show that  $P(\mathbf{T}'_1\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\theta}}^\circ) \rightarrow 1$  as  $n \rightarrow \infty$ .

**Theorem 1** (Oracle property). *Suppose conditions (A1)–(A4) hold and that  $\inf_{\boldsymbol{\beta} \in \mathfrak{B}_1} (\mathbf{d}'_k \mathbf{T}_1 \mathbf{T}'_1 \boldsymbol{\beta})^2 \geq c_1 (\mathbf{d}'_k \mathbf{T}_1 \mathbf{T}'_1 \boldsymbol{\beta}_0)^2$ , for  $1 \leq k \leq q_n$ , where  $\mathfrak{B}_1$  is defined as in Lemma 2. For any  $q_n$ -vector  $\mathbf{a}_n$ , with  $\|\mathbf{a}_n\| \leq 1$ , set  $s_n^2 = \sigma^2 \mathbf{a}'_n \tilde{\boldsymbol{\Sigma}}_{n1}^{-1} \mathbf{a}_n$ . Then, with probability tending to one:*

- (i) *The BAR estimator  $\mathbf{T}'\hat{\boldsymbol{\beta}}$  exists and is the unique fixed point of  $\mathbf{T}'\mathbf{g}(\cdot)$  in the region  $\mathfrak{B}$ , defined as in Lemma 1;*
- (ii)  $\mathbf{T}'_2\hat{\boldsymbol{\beta}} = \mathbf{0}$ ;
- (iii)  $\sqrt{n}s_n^{-1}\mathbf{a}'_n(\mathbf{T}'_1\hat{\boldsymbol{\beta}} - \mathbf{T}'_1\boldsymbol{\beta}_0) \rightarrow \mathcal{N}(0, 1)$ .

**Remark 3.** For additional insight into the BAR procedure and its oracle properties, recall that the initial value  $\hat{\boldsymbol{\beta}}^{(0)}$  is asymptotically consistent with  $\boldsymbol{\beta}_0$  and  $\mathbf{d}'_k \mathbf{T}_1 = 0$ , for  $(q_n + 1) \leq k \leq K_n$ . The oracle properties of the

BAR estimator are essentially the result of the iterative weight  $(\mathbf{d}'_k \tilde{\boldsymbol{\beta}})^{-2}$ . Specifically, when the true  $\mathbf{d}'_k \boldsymbol{\beta}_0$  is zero or, alternatively,  $\mathbf{d}'_k \mathbf{T}_2 \mathbf{T}'_2 \boldsymbol{\beta}_0 = 0$ , for  $(q_n + 1) \leq k \leq K_n$ , the weight  $\{\mathbf{d}'_k \mathbf{T}_2 \mathbf{T}'_2 \hat{\boldsymbol{\beta}}^{(0)}\}^{-2}$  is large, resulting in a smaller estimate of  $\mathbf{d}'_k \hat{\boldsymbol{\beta}}^{(j)}$  per iteration. This leads to the zero-consistency of the BAR estimator. On the other hand, for those nonzero  $\mathbf{d}'_k \boldsymbol{\beta}_0$ , for  $1 \leq k \leq q_n$ , we have the nonzero weight function

$$\{\mathbf{d}'_k \hat{\boldsymbol{\beta}}^{(j)}\}^{-2} = \{\mathbf{d}'_k (\mathbf{T}_1 \mathbf{T}'_1 + \mathbf{T}_2 \mathbf{T}'_2) \hat{\boldsymbol{\beta}}^{(j)}\}^{-2} \approx \{\mathbf{d}'_k \mathbf{T}_1 \mathbf{T}'_1 \hat{\boldsymbol{\beta}}^{(j)}\}^{-2},$$

where  $\{\mathbf{d}'_k \mathbf{T}_1 \mathbf{T}'_1 \hat{\boldsymbol{\beta}}^{(j)}\}^{-2}$  is the weight function when the true model (2.8) is known in advance. Hence, the asymptotic normality of  $\mathbf{d}'_k \boldsymbol{\beta}_0$  is very closed to the unique fixed point of  $\mathbf{f}(\cdot)$ .

In real applications, when  $\mathbf{d}'_k \boldsymbol{\beta}_0 = 0$ , the denominator  $c_k^2(\hat{\boldsymbol{\beta}}^{(k)})$  will inevitably run into a small value close to zero, causing an arithmetic overflow. In the next section, we attempt to use the Lagrange multiplier to overcome this computational difficulty. The resulting Algorithm 1 for the BAR procedure can be used for signal approximation, image processing, and gene detection.

## 4. Algorithm

### 4.1. 1D fused BAR implementation

We consider the 1D fused BAR method with an arbitrary  $\mathbf{X}$ . Let

$$\mathbf{M} = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ & & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & -1 & 1 \end{pmatrix}$$

of size  $(p_n - 1) \times p_n$ . The iterative function (2.4) can be written as

$$\mathbf{g}(\tilde{\boldsymbol{\beta}}) = \arg \min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda_1 \boldsymbol{\beta}' \mathbf{H}_1(\tilde{\boldsymbol{\beta}}) \boldsymbol{\beta} + \lambda_2 \boldsymbol{\beta}' \mathbf{M}' \mathbf{H}_2(\tilde{\boldsymbol{\beta}}) \mathbf{M} \boldsymbol{\beta}, \quad (4.1)$$

where  $\mathbf{H}_1(\tilde{\boldsymbol{\beta}}) = \text{diag}(\tilde{\beta}_i^{-2})$ , and  $\mathbf{H}_2(\tilde{\boldsymbol{\beta}}) = \text{diag}\{(\mathbf{M}\tilde{\boldsymbol{\beta}})_i^{-2}\}$ , with  $(\mathbf{M}\tilde{\boldsymbol{\beta}})_i$  the  $i$ th component of  $\mathbf{M}\tilde{\boldsymbol{\beta}}$ . Because the objective function in (4.1) is differentiable and strictly convex, there exists a unique global minimum. After a few iterative steps, however, some elements of  $\tilde{\boldsymbol{\beta}}$  and  $\mathbf{M}\tilde{\boldsymbol{\beta}}$  would be close to zero. As a result, the division in the diagonal entries of  $\mathbf{H}_1(\tilde{\boldsymbol{\beta}})$  and  $\mathbf{H}_2(\tilde{\boldsymbol{\beta}})$  will run into an overflow, and the iterative procedure will stop at a suboptimal value. To avoid these divisions in  $\mathbf{H}_1(\tilde{\boldsymbol{\beta}})$  and  $\mathbf{H}_2(\tilde{\boldsymbol{\beta}})$ , we set  $\tilde{\mathbf{z}} = \mathbf{M}\tilde{\boldsymbol{\beta}}$  and

$$\min_{\boldsymbol{\beta}, \mathbf{z}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda_1 \boldsymbol{\beta}' \mathbf{H}_1(\tilde{\boldsymbol{\beta}}) \boldsymbol{\beta} + \lambda_2 \mathbf{z}' \mathbf{H}_1(\tilde{\mathbf{z}}) \mathbf{z} \quad \text{subject to } \mathbf{z} = \mathbf{M}\boldsymbol{\beta},$$

where  $\mathbf{H}_1(\tilde{\mathbf{z}}) = \text{diag}(\tilde{z}_i^{-2})$ , by the preceding definition.

The Lagrange function is

$$L(\boldsymbol{\beta}, \mathbf{z}, \mathbf{u}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda_1 \boldsymbol{\beta}' \mathbf{H}_1(\tilde{\boldsymbol{\beta}}) \boldsymbol{\beta} + \lambda_2 \mathbf{z}' \mathbf{H}_1(\tilde{\mathbf{z}}) \mathbf{z} + \mathbf{u}'(\mathbf{M}\boldsymbol{\beta} - \mathbf{z}),$$

where  $\mathbf{u}$  is the Lagrange multiplier. The Lagrange dual of (4.1) is

$$\max_{\mathbf{u}} \min_{\boldsymbol{\beta}, \mathbf{z}} L(\boldsymbol{\beta}, \mathbf{z}, \mathbf{u}). \quad (4.2)$$

To further solve the problem, we first minimize  $L(\boldsymbol{\beta}, \mathbf{z}, \mathbf{u})$  over  $\boldsymbol{\beta}$  and  $\mathbf{z}$ .

The term of  $L(\boldsymbol{\beta}, \mathbf{z}, \mathbf{u})$  that involves  $\boldsymbol{\beta}$  is

$$\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda_1 \boldsymbol{\beta}' \mathbf{H}_1(\tilde{\boldsymbol{\beta}}) \boldsymbol{\beta} + \mathbf{u}' \mathbf{M} \boldsymbol{\beta}.$$

It follows that

$$\begin{aligned} \min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda_1 \boldsymbol{\beta}' \mathbf{H}_1(\tilde{\boldsymbol{\beta}}) \boldsymbol{\beta} + \mathbf{u}' \mathbf{M} \boldsymbol{\beta} \\ = \mathbf{y}' \mathbf{y} - (\mathbf{X}' \mathbf{y} - \mathbf{M}' \mathbf{u} / 2)' \{ \mathbf{X}' \mathbf{X} + \lambda_1 \mathbf{H}_1(\tilde{\boldsymbol{\beta}}) \}^{-1} (\mathbf{X}' \mathbf{y} - \mathbf{M}' \mathbf{u} / 2), \end{aligned}$$

and the optimal  $\boldsymbol{\beta}$  is  $\hat{\boldsymbol{\beta}}^\circ = \{ \mathbf{X}' \mathbf{X} + \lambda_1 \mathbf{H}_1(\tilde{\boldsymbol{\beta}}) \}^{-1} (\mathbf{X}' \mathbf{y} - \mathbf{M}' \mathbf{u} / 2)$ . Similarly, minimizing  $L(\boldsymbol{\beta}, \mathbf{z}, \mathbf{u})$  over  $\mathbf{z}$ , we have

$$\min_{\mathbf{z}} \lambda_2 \mathbf{z}' \mathbf{H}_1(\tilde{\mathbf{z}}) \mathbf{z} - \mathbf{u}' \mathbf{z} = -\frac{1}{4\lambda_2} \mathbf{u}' \mathbf{H}_1^{-1}(\tilde{\mathbf{z}}) \mathbf{u}.$$

Therefore, the dual problem (4.2) is equivalent to

$$\min_{\mathbf{u}} (\mathbf{X}' \mathbf{y} - \mathbf{M}' \mathbf{u} / 2)' \{ \mathbf{X}' \mathbf{X} + \lambda_1 \mathbf{H}_1(\tilde{\boldsymbol{\beta}}) \}^{-1} (\mathbf{X}' \mathbf{y} - \mathbf{M}' \mathbf{u} / 2) + \frac{1}{4\lambda_2} \mathbf{u}' \mathbf{H}_1^{-1}(\tilde{\mathbf{z}}) \mathbf{u} - \mathbf{y}' \mathbf{y}.$$

It is straightforward to obtain the solution, denoted as

$$\hat{\mathbf{u}}^\circ = 2\{ \mathbf{M} \mathbf{B}(\tilde{\boldsymbol{\beta}}) \mathbf{M}' + \mathbf{H}_1^{-1}(\tilde{\mathbf{z}}) / \lambda_2 \}^{-1} \mathbf{M} \mathbf{B}(\tilde{\boldsymbol{\beta}}) \mathbf{X}' \mathbf{y},$$

where  $\mathbf{B}(\tilde{\boldsymbol{\beta}}) = \{ \mathbf{X}' \mathbf{X} + \lambda_1 \mathbf{H}_1(\tilde{\boldsymbol{\beta}}) \}^{-1}$ . In practice, if the inverse of a matrix does not exist, we suggest using the Moore–Penrose pseudo-inverse, denoted

as  $\text{Pinv}()$ . On the other hand, to avoid the division in  $\mathbf{B}(\tilde{\boldsymbol{\beta}})$ , we instead calculate  $\mathbf{B}(\tilde{\boldsymbol{\beta}}) = \mathbf{H}_1^{-1}(\tilde{\boldsymbol{\beta}})\{\mathbf{X}'\mathbf{X}\mathbf{H}_1^{-1}(\tilde{\boldsymbol{\beta}}) + \lambda_1\mathbf{I}_n\}^{-1}$ . To implement the 1D fused BAR procedure, we take  $\mathbf{M}_0 = \mathbf{M}$  in Algorithm 1. In particular, we set  $\mathbf{X} = \mathbf{I}$  when performing signal approximation.

#### 4.2. 2D fused BAR implementation

We now investigate the implementation of the 2D fused BAR method, with an application to image denoising. In contrast to signals, the adjacent pixels of one image include both the horizontal-level neighbors and vertical-level neighbors. The objective function in (2.5) can be written as

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \boldsymbol{\beta}\|^2 + \lambda_1\boldsymbol{\beta}'\mathbf{H}_1(\tilde{\boldsymbol{\beta}})\boldsymbol{\beta} + \lambda_2\boldsymbol{\beta}'\{\mathbf{M}'_1\mathbf{H}_3(\tilde{\boldsymbol{\beta}})\mathbf{M}_1 + \mathbf{M}'_2\mathbf{H}_4(\tilde{\boldsymbol{\beta}})\mathbf{M}_2\}\boldsymbol{\beta},$$

where  $\mathbf{M}_1$  and  $\mathbf{M}_2$  capture the vertical and horizontal neighbors, respectively in a graph,  $\mathbf{H}_3(\tilde{\boldsymbol{\beta}}) = \text{diag}\{(\mathbf{M}_1\tilde{\boldsymbol{\beta}})_j^{-2}\}$ , and  $\mathbf{H}_4(\tilde{\boldsymbol{\beta}}) = \text{diag}\{(\mathbf{M}_2\tilde{\boldsymbol{\beta}})_j^{-2}\}$ .

To overcome the numerical difficulty in  $\mathbf{H}_3(\tilde{\boldsymbol{\beta}})$  and  $\mathbf{H}_4(\tilde{\boldsymbol{\beta}})$ , we derive the refined iterative procedure based on the Lagrange multiplier. In a similar vein, the Lagrange function is

$$\tilde{L}(\boldsymbol{\beta}, \mathbf{z}, \mathbf{u}) = \|\mathbf{y} - \boldsymbol{\beta}\|^2 + \lambda_1\boldsymbol{\beta}'\mathbf{H}_1(\tilde{\boldsymbol{\beta}})\boldsymbol{\beta} + \lambda_2\mathbf{z}'\mathbf{H}^*(\tilde{\mathbf{z}})\mathbf{z} + \mathbf{u}'(\mathbf{M}^*\boldsymbol{\beta} - \mathbf{z}),$$

where  $\tilde{\mathbf{z}}_1 = \mathbf{M}_1\tilde{\boldsymbol{\beta}}$ ,  $\tilde{\mathbf{z}}_2 = \mathbf{M}_2\tilde{\boldsymbol{\beta}}$ ,  $\mathbf{z} = (\mathbf{z}'_1, \mathbf{z}'_2)'$ ,  $\mathbf{u} = (\mathbf{u}'_1, \mathbf{u}'_2)'$ ,  $\mathbf{M}^* = (\mathbf{M}'_1, \mathbf{M}'_2)'$ ,

and

$$\mathbf{H}^*(\tilde{\mathbf{z}}) = \begin{pmatrix} \mathbf{H}_1(\tilde{\mathbf{z}}_1) & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_1(\tilde{\mathbf{z}}_2) \end{pmatrix}.$$

Note that the above Lagrange function  $\tilde{L}(\boldsymbol{\beta}, \mathbf{z}, \mathbf{u})$  is in the same form of  $L(\boldsymbol{\beta}, \mathbf{z}, \mathbf{u})$  in the 1D fused BAR method. As a result, we set  $\mathbf{M}_0 = \mathbf{M}^*$  in Algorithm 1 for image denoising.

Note that Algorithm 1 is a general algorithm for the BAR method. Thus, its application is not restricted to signal approximation and image processing. In general, any design of  $\mathbf{d}_k$  can be incorporated in  $\mathbf{M}$ . Algorithm 1 is flexible in the sense that it allows for different penalties to be imposed on the various types of structures of  $\mathbf{d}_k$ .

### 4.3. Choice of tuning parameters

To implement the BAR procedure, the initial value  $\hat{\boldsymbol{\beta}}^{(0)}$  and parameters  $\lambda_1$  and  $\lambda_2$  need to be chosen carefully. The BAR method recommends the ridge estimator as the initial  $\hat{\boldsymbol{\beta}}^{(0)}$ , with the tuning parameter  $\xi$  chosen carefully using five-fold cross-validation (CV). However, when the sample size  $n$  is small, the value of  $\xi$  chosen by CV may vary owing to the different partitions of the data. To avoid this problem, we instead use the univariate regression estimator as the initial estimator  $\hat{\boldsymbol{\beta}}^{(0)}$  whenever  $p \gg n$ ; that is,

$$\hat{\beta}_j^{(0)} = \frac{\sum_{i=1}^n x_{ij}y_i}{\sum_{i=1}^n x_{ij}^2}, \quad j = 1, \dots, p_n, \quad (4.3)$$

---

**Algorithm 1:** Fused BAR Algorithm

---

**Result:** Fused BAR estimator  $\hat{\beta}^*$ .

```

1 Input  $\mathbf{y}$ ,  $\mathbf{X}$ ,  $\mathbf{M}_0$ ,  $\hat{\beta}^{(0)}$ ,  $\lambda_1$ ,  $\lambda_2$  and  $\epsilon$ ;
2  $k \leftarrow 0$ ;
3  $\tilde{\beta} \leftarrow \hat{\beta}^{(0)}$ ;
4 while ( $\|\tilde{\beta} - \hat{\beta}^{(k)}\| > \epsilon$  or  $k = 0$ ) do
5    $\tilde{\beta} \leftarrow \hat{\beta}^{(k)}$ ;
6    $\tilde{\mathbf{z}} \leftarrow \mathbf{M}_0 \tilde{\beta}$ ;
7    $\mathbf{H}_1^{-1}(\tilde{\beta}) \leftarrow \text{diag}(\tilde{\beta}_j^2)$ ;
8    $\mathbf{H}_0^{-1}(\tilde{\mathbf{z}}) \leftarrow \text{diag}(\tilde{z}_j^2)$ ;
9    $\mathbf{B}(\tilde{\beta}) \leftarrow \mathbf{H}_1^{-1}(\tilde{\beta})\{\mathbf{X}'\mathbf{X}\mathbf{H}_1^{-1}(\tilde{\beta}) + \lambda_1 \mathbf{I}_n\}^{-1}$ ;
10   $\hat{\mathbf{u}}^\circ \leftarrow \text{Pinv}\{\mathbf{M}_0 \mathbf{B}(\tilde{\beta}) \mathbf{M}_0' + \mathbf{H}_0^{-1}(\tilde{\mathbf{z}})/\lambda_2\} \mathbf{M}_0 \mathbf{B}(\tilde{\beta}) \mathbf{X}' \mathbf{y}$ ;
11   $k \leftarrow k + 1$ ;
12   $\hat{\beta}^{(k)} \leftarrow \mathbf{B}(\tilde{\beta})(\mathbf{X}' \mathbf{y} - \mathbf{M}_0' \hat{\mathbf{u}}^\circ)$ ;
13 end

```

---

which is adopted as the initial value in the adaptive lasso (Huang *et al.*, 2008b) to handle the high-dimensional problem. Huang *et al.* (2008b) also showed that, under certain conditions, the adaptive lasso estimator is consistent in variable selection and estimation if the initial estimator is the marginal regression estimator. This is because the univariate regression estimator is zero-consistent, in the sense that the estimators of the zero coefficients converge to zero, whereas those of the nonzero coefficients do

not. Our simulation results show that the univariate estimator is a good initial value.

To select  $\lambda_1$  and  $\lambda_2$ , we adopt the  $k$ -fold CV method. Specifically, in signal approximation, we pick all odd coefficients as the training set, and all even coefficients as the validation set. We search a grid of  $\lambda_1$  and  $\lambda_2$  using the two-fold CV method. For example, we have 10 grids evenly distributed on the interval  $[0.1, 10]$  for  $\lambda_1$ , and 10 grids evenly distributed on  $[1, 20]$  for  $\lambda_2$ . Then, we select the optimal  $(\lambda_1, \lambda_2)$  with the minimum CV error by searching over values in the 2D grid. Note that if  $\mathbf{X}$  is a general matrix, we recommend using five-fold CV to find the optimal tuning parameters,  $\lambda_1$  and  $\lambda_2$ .

## 5. Simulation study

In this section, we carry out simulations on the fused BAR method, BAR fusion, fused lasso, and  $\ell_1$  fusion. Note that the difference between “fused” and “fusion” constraints is that the former encourages sparsity both in the coefficients and their differences, whereas the latter penalizes the flatness of coefficients only. For instance, the BAR fusion method is a special case of the BAR method, with penalties imposed on the differences between adjacent coefficients only. Comparisons of their performance in terms of variable selection, estimation, and prediction are presented. We

use the R package *genlasso* for  $\ell_1$  fusion and fused lasso. The response variable is generated from the regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}_0 + \sigma\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

The following Examples 4.4–4.8 are considered. Specifically, Example 4.4 is designed to check the ability to produce a piecewise constant estimate using fewer jumps, and Example 4.5 checks whether these methods can successfully detect the single nonzero coefficients. In Examples 4.6–4.7, we assess the performance of Algorithm 1 for a general design of  $\mathbf{X}$  when  $p > n$  and  $p < n$ , respectively. To examine the performance of these methods on detecting true smaller jumps, we conduct Examples 4.8–4.9. In Example 4.10, we simulate a toy image and compare the performance of the fused BAR method with that of the fused lasso method in image denoising.

**Example 4.4.** (Signal approximation.) Set  $\mathbf{X} = \mathbf{I}$ . Let  $\sigma = 0.8$ ,  $n = 200$ , and the true signal be

$$\boldsymbol{\beta}_0 = \underbrace{(0, \dots, 0)}_{20}, \underbrace{5, \dots, 5}_9, \underbrace{0, \dots, 0}_{41}, \underbrace{3.5, \dots, 3.5}_9, \underbrace{0, \dots, 0}_{21}, \underbrace{4.5, \dots, 4.5}_{19}, \underbrace{0, \dots, 0}_{81}.$$

The number of nonzero coefficients is 37.

**Example 4.5.** (Singular nonzero value.) Set  $\mathbf{X} = \mathbf{I}$ . Let  $\sigma = 0.8$ ,  $n = 250$ ,

and the true signal be

$$\beta_0 = (\underbrace{0, \dots, 0}_{24}, \underbrace{5, 0, \dots, 0}_{125}, \underbrace{4.5, \dots, 4.5}_{49}, \underbrace{0, \dots, 0}_{51})'.$$

The number of nonzero coefficients is 50 with a single nonzero coefficient

$$\beta_{25} = 5.$$

**Example 4.6.** (A general matrix  $\mathbf{X}$ , with  $p_n > n$ .) Let  $\sigma = 10$ ,  $p = 250$ ,  $n = 200$ , and the true coefficients be

$$\beta_0 = (\underbrace{0, \dots, 0}_{19}, \underbrace{5, \dots, 5}_{25}, \underbrace{0, \dots, 0}_{56}, \underbrace{3, \dots, 3}_{29}, \underbrace{0, \dots, 0}_{51}, \underbrace{-4, \dots, -4}_{29}, \underbrace{0, \dots, 0}_{41})'.$$

The number of nonzero coefficients is 83. We generate  $x_{ij} \sim \mathcal{N}(0, 1)$ , for all  $1 \leq i \leq n$  and  $1 \leq j \leq p_n$ .

**Example 4.7.** (A general matrix  $\mathbf{X}$  with  $p_n < n$ .) Let  $\sigma = 5$ ,  $p = 100$ ,  $n = 200$ , and the true coefficients be

$$\beta_0 = (\underbrace{0, \dots, 0}_{9}, \underbrace{-2, \dots, -2}_{15}, \underbrace{0, \dots, 0}_{26}, \underbrace{4, \dots, 4}_{19}, \underbrace{0, \dots, 0}_{31})'.$$

The number of nonzero coefficients is 34. We generate  $x_{ij} \sim \mathcal{N}(0, 1)$ , for all  $1 \leq i \leq n$  and  $1 \leq j \leq p_n$ .

**Example 4.8.** We use the same model as in Example 4.4, but with  $\beta_j = 0.85$ , for all  $\beta_j \neq 0$ .

**Example 4.9.** We use the same model as in Example 4.5, but with  $\beta_j = 0.85$ , for all  $\beta_j \neq 0$ .

**Example 4.10.** (Image denoising.) We design a  $20 \times 20$  pixel toy image. The noise, following a normal distribution with mean zero and variance  $(14/51)^2$ , is added to the original image. We calculate the reconstruction errors by the fused BAR and the fused lasso denoising, respectively.

Tables 1–2 summarize the results for Examples 4.4–4.7 and Examples 4.8–4.9, each with 20 replications. The two tables present the number of selected features (NOS), number of falsely selected variables (NOFS), percentage of true nonzero coefficients the model selected (TM), number of jumps (NOJ), mean absolute bias (MAB), fitted mean squared error (FMSE), single value selection (SVS), and minimum CV error. Figures 1–5 depict the estimated coefficients using the fused BAR, BAR fusion, fused lasso and  $\ell_1$  fusion for Examples 4.4–4.9. The image processing results for Example 4.10 are shown in Figure 6.

It can be seen from Tables 1–2 that the NOS, NOFS, and MAB of the fused BAR estimator are relatively smaller than those of the fused Lasso. This implies that the resulting fused BAR estimator exhibits better performance in terms of variable selection. Moreover, Figures 1–3 show that the fused BAR obtains coefficients that are piecewise constant, with

Table 1: Mean and standard deviation (in parentheses) of the results using the fused BAR and the fused Lasso for Examples 4.4–4.7.

	Example 4.4		Example 4.5	
	fused BAR	fused Lasso	fused BAR	fused Lasso
NOS	37.350 (0.933)	96.850 (34.973)	49.950 (0.394)	114.550 (45.658)
NOFS	0.350 (0.933)	59.850 (34.973)	0.050 (0.224)	64.700(45.542)
TM	100% (0.000)	100% (0.000)	99.8% (0.006)	99.7% (0.007)
NOJ	6.850 (1.496)	37.050 (13.839)	3.250 (1.070)	16.450 (7.749)
MAB	0.056 (0.018)	0.148 (0.039)	0.050 (0.022)	0.112 (0.027)
FMSE	0.620 (0.058)	0.545 (0.097)	0.679 (0.076)	0.688 (0.072)
CV error	1.200 (0.132)	1.211 (0.138)	0.922 (0.096)	0.892 (0.083)
SVS	–	–	0.950 (0.224)	0.850 (0.366)
	Example 4.6		Example 4.7	
	fused BAR	fused Lasso	fused BAR	fused Lasso
NOS	96.150 (16.000)	202.800 (42.010)	35.250 (2.197)	56.400 (13.697)
NOFS	13.300(15.885)	119.800 (42.010)	1.250 (2.197)	22.400 (13.697)
TM	99.8%(0.004)	100% (0.000)	100% (0.000)	100% (0.000)
NOJ	7.250 (2.124)	23.450 (5.094)	4.250 (0.639)	13.500 (1.318)
MAB	0.060 (0.033)	0.162 (0.041)	0.025 (0.005)	0.058 (0.006)
FMSE	102.699 (9.217)	91.191 (11.323)	8.218 (0.263)	8.124 (0.175)
CV error	116.314 (13.951)	125.785 (12.500)	8.371 (0.161)	9.548 (0.227)
Test error	113.799 (27.965)	113.629 (22.332)	8.796 (0.945)	10.699 (1.314)

Table 2: Mean and standard deviation (in parentheses) of the results using the fused BAR and the fused Lasso for Examples 4.8–4.9.

	Example 4.8			
	fused BAR	fused Lasso	BAR fusion	$\ell_1$ fusion
NOS	45.050 (17.760)	94.200 (31.629)	–	–
NOFS	18.350 (14.449)	60.550 (29.366)	–	–
TM	72.2% (0.170)	90.9 % (0.121)	–	–
NOJ	7.000 (3.598)	26.500 (13.873)	4.100 (3.611)	19.300 (10.458)
MAB	0.122 (0.040)	0.126 (0.030)	0.218 (0.060)	0.196 (0.038)
FMSE	0.584 (0.075)	0.564 (0.085)	0.654 (0.107)	0.581 (0.107)
CV error	0.732 (0.081)	0.712 (0.074)	0.739 (0.082)	0.726 (0.084)
	Example 4.9			
	fused BAR	fused Lasso	BAR fusion	$\ell_1$ fusion
NOS	56.050 (12.890)	118.100 (46.348)	–	–
NOFS	12.200 (11.901)	69.450 (46.025)	–	–
TM	87.7% (0.097)	97.3% (0.032)	–	–
NOJ	4.050 (2.438)	15.650 (9.637)	3.550 (1.905)	14.350 (6.175)
MAB	0.065 (0.029)	0.089 (0.027)	0.126 (0.064)	0.128 (0.041)
FMSE	0.607 (0.056)	0.602 (0.065)	0.622 (0.070)	0.591 (0.060)
CV error	0.684 (0.068)	0.675 (0.062)	0.698 (0.070)	0.685 (0.064)
SVS	0.000 (0.000)	0.500 (0.513)	–	–

fewer jumps. This is also shown in Table 1, where the NOJ is much smaller than the fused Lasso. In addition, Figure 2 indicates that the fused BAR

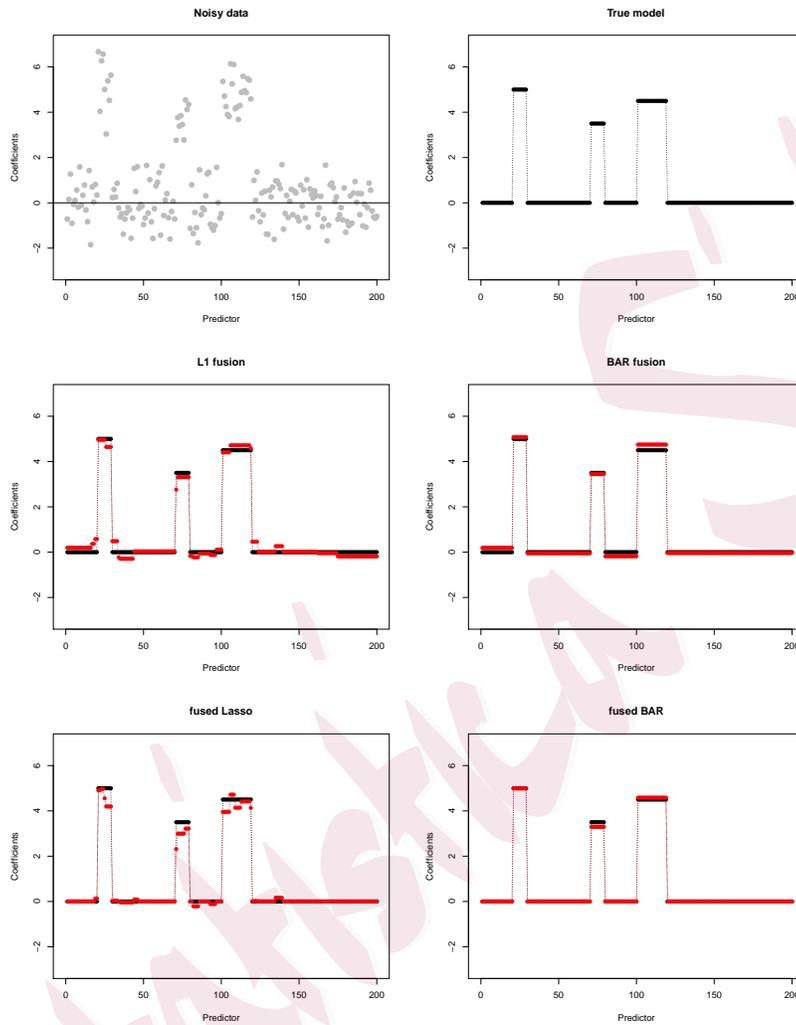


Figure 1: Estimated coefficients using  $\ell_1$  fusion, BAR fusion, fused lasso and fused BAR for Example 4.4.

method is sensitive to the single-value coefficient, and its SVS in Table 1 is larger than that of the fused lasso method. On the other hand, we see from Table 2 and Figures 4–5 that when the true jumps are relatively

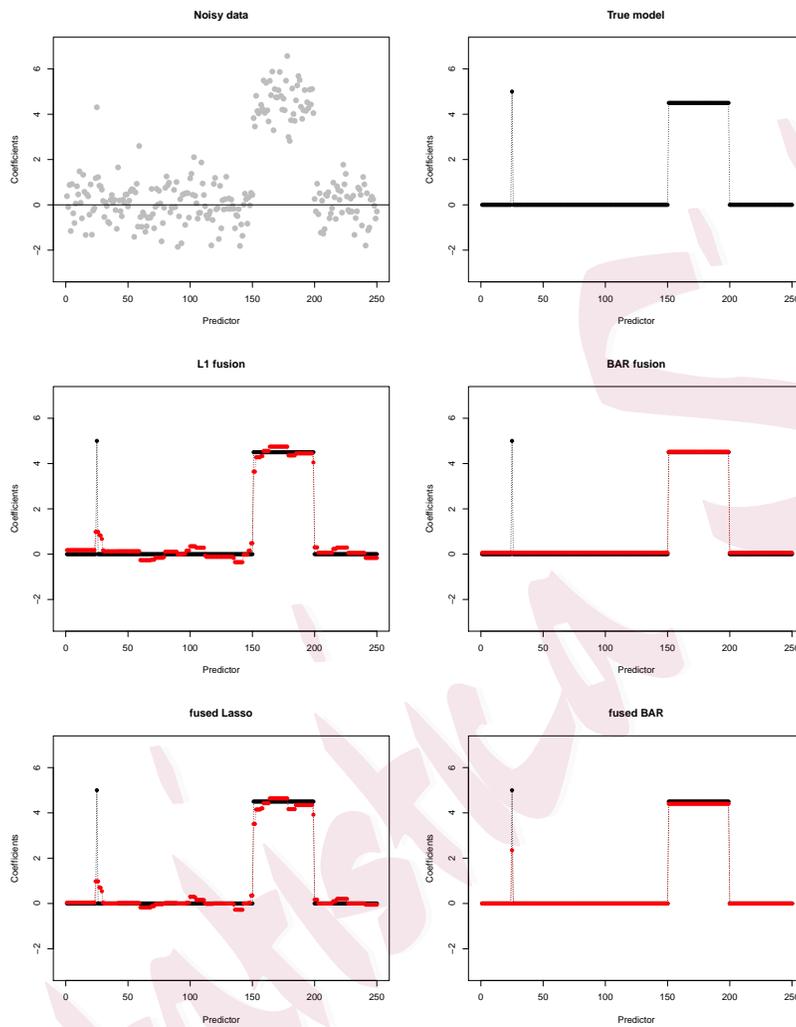


Figure 2: Estimated coefficients using  $\ell_1$  fusion, BAR fusion, fused lasso and fused BAR for Example 4.5.

smaller, the fused BAR and BAR fusion can still detect these smaller jumps, with flatter estimates than those of the fused lasso and the  $\ell_1$  fusion. In Example 4.9, the fused lasso seems to successfully detect a single nonzero

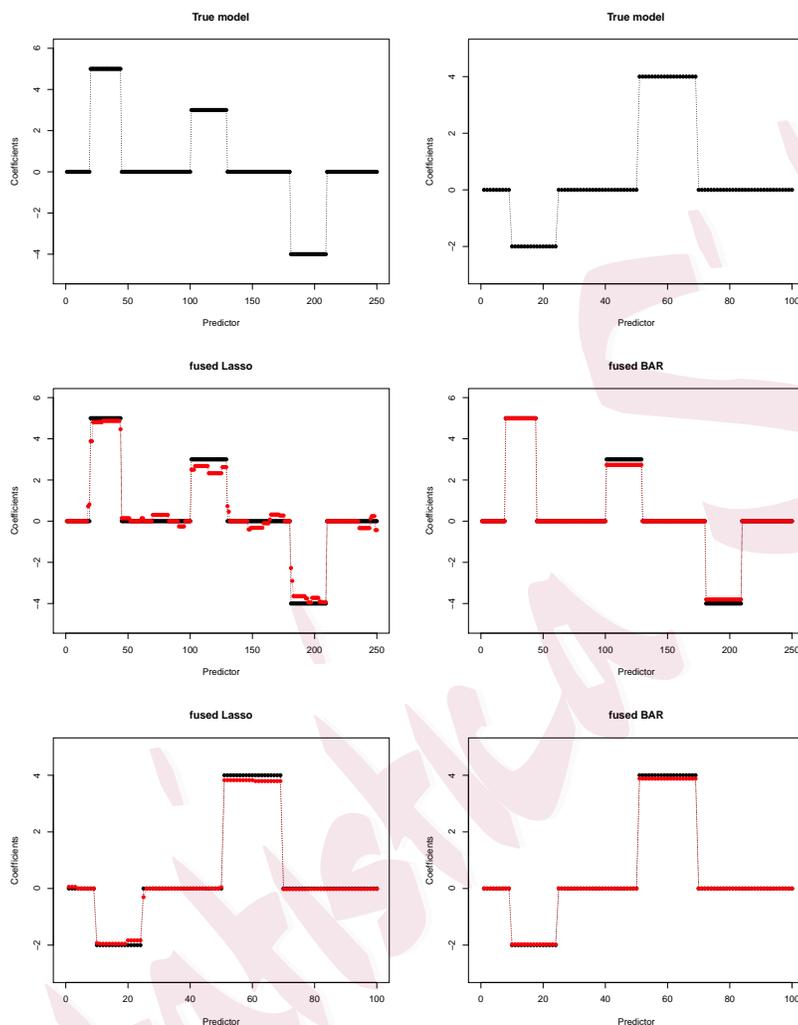


Figure 3: Estimated coefficients using fused lasso and fused BAR for Example 4.6 (the first row) and Example 4.7 (the second row).

value, because it has a larger SVS value, from Table 2. However, our limited experiments show that the fused lasso estimate does not essentially capture an up-and-down jumping structure, similar to Figure 5. This phenomenon

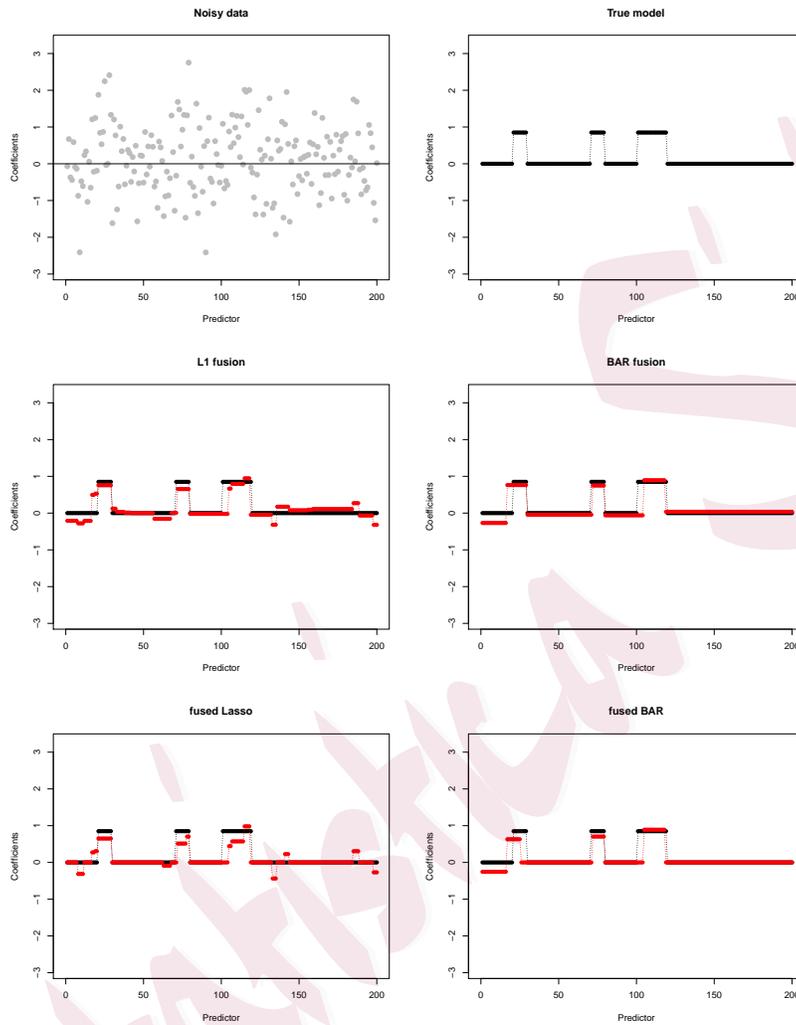


Figure 4: Estimated coefficients using  $\ell_1$  fusion, BAR fusion, fused lasso and fused BAR for Example 4.8.

is reasonable, because the single true nonzero is merged with a larger noise and, thus, is more difficult to detect. Lastly, as shown in Figure 4, the fused BAR method is comparable with the fused lasso for image processing. More

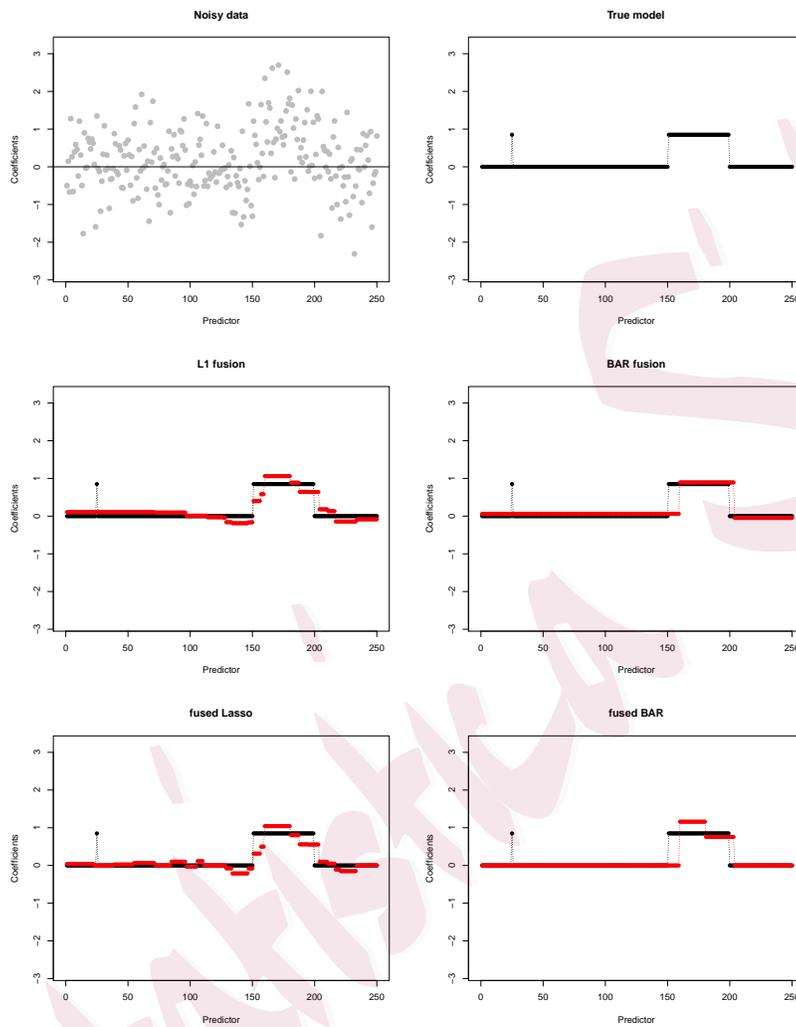


Figure 5: Estimated coefficients using  $\ell_1$  fusion, BAR fusion, fused lasso and fused BAR for Example 4.9.

precisely, the 2D fused BAR reduces the reconstruction error of the 2D fused lasso from 5.525 to 1.051. Overall, the simulation results contain supportive evidence that the fused BAR method works reasonably well in terms of

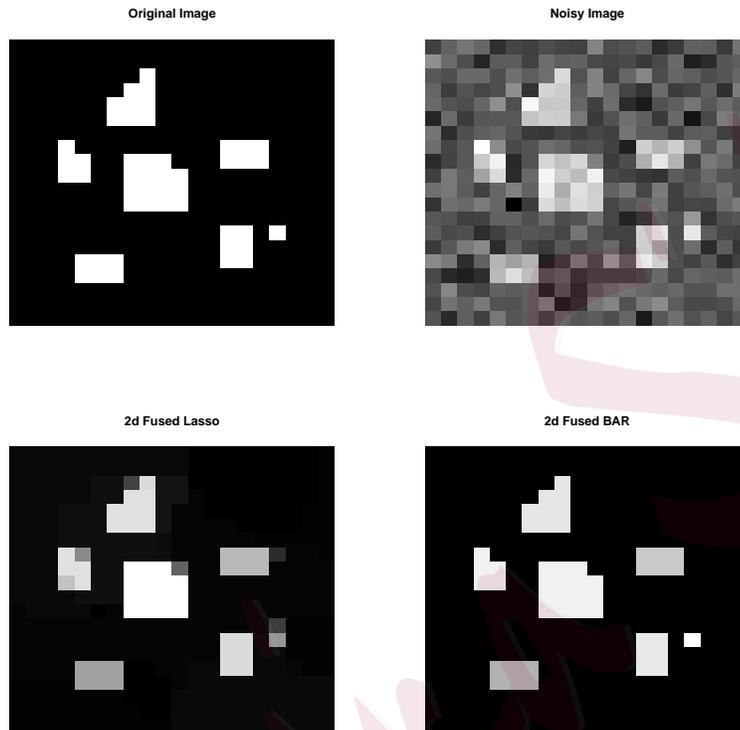


Figure 6: Results of Example 4.8 for image processing using the fused lasso and the fused BAR.

variable selection, estimation, and prediction compared with the fused lasso method. As one reviewer pointed out, this may be because the lasso is a biased estimation for large coefficients and differences of coefficients. On the other hand, this phenomenon may be due to the variable selection inconsistency of the lasso in some scenarios (Zou and Hastie, 2005).

## 6. Real examples

### 6.1. CGH array denoising

In cancer research, the copy number variations (CNV) data are important data sets that have an adjacent relationship. CNVs are typically in the form of segments of various lengths (Rippe *et al.*, 2012). The comparative genomic hybridization (CGH) array is a powerful tool used to detect genetic alterations, such as deletions and copy number increases, and regions of gains or losses in DNA copy numbers (Pinkel *et al.*, 1998; Wang *et al.*, 2005). To facilitate the detection of alterations, the array of CGH data is set to the  $\log_2$  ratio of the number of DNA copies in tumor cells divided by that in normal or reference cells. Therefore, a positive CGH value, called a gain, indicates an increase in the number of DNA copies, whereas a loss is shown by a negative value. CGH signals are usually approximated by a piecewise constant sequence or a function with segmented areas of zero values. In recent years, many approaches, such as the EM-based method (Myers *et al.*, 2004), hidden Markov models (Fridlyand *et al.*, 2004; Liu *et al.*, 2010), and a segmentation algorithm (Venkatraman and Olshen, 2007) have been developed for the visualization of CGH signals and inferences segmented values. The fused lasso method has been applied to identify the gains and losses in the CGH arrays (Tibshirani and Wang, 2007).

Table 3: Summary of the analysis results: CGH data.

	Tuning parameters	NOS	NOJ	FMSE	CV error
Fused BAR	$\lambda_1 = 2.154e - 05, \lambda_2 = 0.889$	732	11	0.166	0.370
BAR fusion	$\lambda = 0.910$	–	11	0.167	0.371
Fused Lasso	$\lambda_1 = 0.005, \lambda_2 = 2.081$	942	40	0.176	0.321
$L_1$ Fusion	$\lambda = 2.081$	–	40	0.176	0.321

We apply the fused BAR, BAR fusion, fused Lasso, and  $\ell_1$  fusion methods to the CGH arrays. The CGH data are obtained from the R package *cghFLasso*. The results are illustrated in Table 3 and Figure 7. Table 3 indicates that the fused BAR selects a smaller number of features than the fused lasso does, and the mean squared errors of the fused BAR fitting are smaller. Figure 7 shows that the fused BAR is sensitive to the outliers. The signals recovered using fused BAR and BAR fusion are flatter than those recovered using fused lasso and  $\ell_1$  fusion.

## 6.2. Lena image processing

We use the 2D fused lasso and fused BAR to denoise the Lena image in the R package *filling*. We added Gaussian noise with a standard deviation of 20 to the original image. Because zero does not represent a natural baseline in this image, we tried the  $\ell_1$  fusion model with  $\lambda_1 = 0$ , as well as the BAR fusion model. Then, we found the optimal value of  $\lambda_2$  for each method using two-fold cross-validation. The reconstruction errors from the

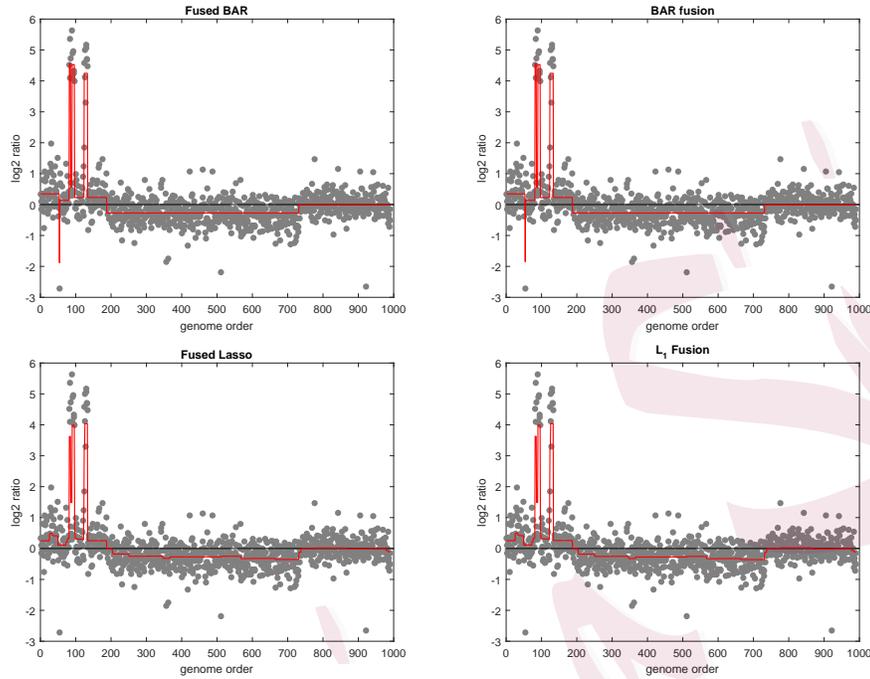


Figure 7: The 1D fused lasso and fused BAR applied to CGH data. The red lines are the estimated CGH signals. The black solid line is  $y = 0$ .

original noiseless image are 6.256 for the BAR fusion and 6.731 for the  $\ell_1$  fusion. Although the BAR fusion has a smaller reconstruction error than the  $\ell_1$  fusion, the two methods have their respective advantages on image processing. Specifically, the BAR fusion solution, shown in the bottom-right panel of Figure 8, gives a better approximation to the smoothness of the image, especially for the background, whereas the  $\ell_1$  fusion estimate shown in the bottom left panel recovers greater details in the characterization of Lena.



Figure 8: Top-left panel:  $128 \times 128$  pixels grayscale image of Lena. Top-right panel: Gaussian noise with standard deviation 20 has been added. Bottom-left panel: Solution of fused lasso with  $\lambda_1 = 0$  and  $\lambda_2$  chosen by CV. Bottom-right panel: Solution of fused BAR with  $\lambda_1 = 0$  and  $\lambda_2$  chosen by CV.

## 7. Discussion

In this paper, we have proposed a BAR method for variable selection and the pattern estimation of regression coefficients. Its oracle properties are demonstrated under proper conditions. As one of the special cases, the fused BAR is introduced and thoroughly discussed, with applications in signal approximation and image denoising. To make it easy to implement, the associated algorithms are established based on the Lagrange method. The simulation study and real-data analysis show that the fused BAR method is comparable with the fused lasso in terms of recovering a lower-dimensional piecewise constant structure and reconstructing an image. The BAR approach can be further connected with those methods that penalize the linear combinations of coefficients, and is expected to be applied in many other scientific fields.

## Supplementary Material

The technical proofs are provided in the online Supplementary Material.

## Acknowledgments

The authors are grateful to the Editor, Associate Editor and two anonymous reviewers for their insightful comments and suggestions that have substantially improved the presentation and the content of this paper.

Linlin Dai's research was supported by the Fundamental Research Funds for the Central Universities of China, approval numbers JBK140507 and JBK1806002. Kani Chen's research was supported by the Hong Kong Research Grant Council grants 16309816, 16300714, 600813, and 600612. The research of Gang Li was partly supported by National Institute of Health Grants P30 CA-16042, UL1TR000124-02, and P50 CA211015.

## References

- Chang, S. G., Yu, B. and Vetterli, M. (2000). Adaptive wavelet thresholding for image denoising and compression. *IEEE Trans. Image Process.* **9**, 1532–1546.
- Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *J. Amer. Statist. Assoc.* **74**, 829–836.
- Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika* **81**, 425–455.
- Eilers, P. H. (2003). A perfect smoother. *Analytical chemistry* **75**, 3631–3636.
- Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.* **96**, 1348–1360.
- Fridlyand, J., Snijders, A. M., Pinkel, D., Albertson, D. G. and Jain, A. N. (2004). Hidden Markov models approach to the analysis of array CGH data. *J. Multivariate Anal.* **90**, 132–153.

- Fu, W. J. (1998). Penalized regressions: the bridge versus the lasso. *J. Comput. Graph. Statist.* **7**, 397–416.
- Gasser, T., Müller, H.-G. and Mammitzsch, V. (1985). Kernels for nonparametric curve estimation. *J. Roy. Statist. Soc. Ser. B* **47**, 238–252.
- Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell* **6**, 721–741.
- Huang, J., Horowitz, J. L. and Ma, S. (2008). Asymptotic properties of bridge estimators in sparse high-dimensional regression models. *Ann. Statist.* **36**, 587–613.
- Huang, J., Ma, S. and Zhang, C.-H. (2008). Adaptive Lasso for sparse high-dimensional regression models. *Statist. Sinica* **18**, 1603–1618.
- Huang, J., Ma, S., Xie, H. and Zhang, C.-H. (2009). A group bridge approach for variable selection. *Biometrika* **96**, 339–355.
- Lam, K. Y., Westrick, Z. M., Müller, C. L., Christiaen, L. and Bonneau, R. (2016). Fused regression for multi-source gene regulatory network inference. *PLoS computational biology* **12**, e1005157.
- Liu, Z., Li, A., Schulz, V., Chen, M. and D. Tuck. (2010). Mixhmm: inferring copy number variation and allelic imbalance using snp arrays and tumor samples mixed with stromal cells. *PLoS one* **5**, e10909.
- Müller, H.-G. and Stadtmüller, U. (1987). Variable bandwidth kernel estimators of regression

- curves. *Ann. Statist.* **15**, 182–201.
- Myers, C. L., Dunham, M. J., Kung, S.-Y. and Troyanskaya, O. G. (2004). Accurate detection of aneuploidies in array cgh and gene expression microarray data. *Bioinformatics* **20**, 3533–3543.
- Pinkel, D., Seagraves, R., Sudar, D., Clark, S., Poole, I., Kowbel, D., Collins, C., Kuo, W.-L., Chen, C., Zhai, Y. *et al.* (1998). High resolution analysis of dna copy number variation using comparative genomic hybridization to microarrays. *Nature genet.* **20**, 207.
- Price, B. S., Geyer, C. J. and Rothman, A. J. (2015). Ridge fusion in statistical learning. *J. Comput. Graph. Statist.* **24**, 439–454.
- Rinaldo, A. (2009). Properties and refinements of the fused lasso. *Ann. Statist.* **37**, 2922–2952.
- Rippe, R. C., Meulman, J. J. and Eilers, P. H. (2012). Visualization of genomic changes by segmented smoothing using an  $\ell_0$  penalty. *PLoS one* **7**, e38230.
- Rudin, L. I., Osher, S. and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Phys. D* **60**, 259–268.
- Ruppert, D., Wand, M. P. and Carroll, R. J. (2009). Semiparametric regression during 2003–2007. *Electron. J. Stat.* **3**, 1193–1256.
- Shen, X. and Huang, H.-C. (2010). Grouping pursuit through a regularization solution surface. *J. Amer. Statist. Assoc.* **105**, 727–739.
- Shin, S., Fine, J. and Liu, Y. (2016). Adaptive estimation with partially overlapping models.

- Statist. Sinica* **26**, 235–253.
- Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2013). A sparse-group lasso. *J. Comput. Graph. Statist.* **22**, 231–245.
- Tang, L. and Song, P. X.-K. (2016). Fused lasso approach in regression coefficients clustering—learning parameter heterogeneity in data integration. *J. Mach. Learn. Res.* **17**, 3815–3937.
- Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **73**, 273–282.
- Tibshirani, R. and Wang, P. (2007). Spatial smoothing and hot spot detection for cgh data using the fused lasso. *Biostatistics* **9**, 18–29.
- Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **67**, 91–108.
- Tibshirani, R. J. and Taylor, J. (2011). The solution path of the generalized lasso. *Ann. Statist.* **39**, 1335–1371.
- Venkatraman, E. and Olshen, A. B. (2007). A faster circular binary segmentation algorithm for the analysis of array cgh data. *Bioinformatics* **23**, 657–663.
- Wang, F., Wang, L. and Song, P. X.-K. (2016). Fused lasso with the adaptation of parameter ordering in combining multiple studies with repeated measurements. *Biometrics* **72**, 1184–1193.
- Wang, P., Kim, Y., Pollack, J., Narasimhan, B. and Tibshirani, R. (2005). A method for calling

---

REFERENCES41

gains and losses in array cgh data. *Biostatistics* **6**, 45–58.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **68**, 47–67.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *Ann. Statist.* **38**, 894–942.

Zhu, Y., Shen, X. and Pan, W. (2013). Simultaneous grouping pursuit and feature selection over an undirected graph. *J. Amer. Statist. Assoc.* **108**, 713–725.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **67**, 301–320.

Zou, H. (2006). The adaptive lasso and its oracle properties. *J. Amer. Statist. Assoc.* **101**, 1418–1429.

Linlin Dai

Center of Statistical Research, School of Statistics, Southwestern University of Finance and Economics

E-mail: ldaiab@swufe.edu.cn

Kani Chen

Department of Mathematics, Hong Kong University of Science and Technology

E-mail: makchen@ust.hk

Gang Li

---

REFERENCES<sup>42</sup>

Department of Biostatistics and Biomathematics, University of California, Los Angeles

E-mail: vli@ucla.edu

Statistica Sinica