Statistica Sinica Preprint No: SS-2018-0075	
Title	The Broken Adaptive Ridge Procedure and Its Applications
Manuscript ID	SS-2018-0075
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202018.0075
Complete List of Authors	Linlin Dai Kani Chen and Gang Li
Corresponding Author	Linlin Dai
E-mail	ldaiab@connect.ust.hk
Notice: Accepted version subject to English editing.	

Statistica Sinica

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 paper, we study the *broken adaptive ridge method* to estimate lower-dimensional patterns of coefficients in regression models. Based on the re-weighted ℓ_2-penalization, the new method can simultaneously recover both the true sparsity and the inherent structures of the features, which makes 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 in signal approximation and image processing. The associated algorithms are numerically easy to implement. Various simulation studies and real data analyses illustrate its advantages over the fused lasso method.

Key words and phrases: Oracle estimator, Linear regression, Re-weighted ℓ_2-

penalization.

1. Introduction

Identifying the underlying dynamics of the dataset of interest is an important task in many applications, including, for instance, denoising, forecasting, filtering and even more sophisticated analysis in machine learning research. In a high dimensional setting, the underlying patterns of 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 paper, we mainly focus on variables that naturally have some local structures, such as piecewise constancy, linear trend or being grouped. An overarching goal is to effectively reduce the dimension of covariates as well as estimate their underlying structures.

The ℓ_0-penalized regression is one of the most natural methods for variable selection, which directly penalizes the cardinality of a model. Rinaldo

(2009) considered the application of the ℓ_0-penalization in finding a piecewise constant function to approximate a signal. Due to the lack of convexity, the ℓ_0-penalization procedure is computationally difficult to implement, especially for high dimensional datasets. A body of effort has been devoted to approaches based on different penalties including ℓ_1-norm penalties and ℓ_2-norm penalties. The fused lasso method (Tibshirani *et al.*, 2005) based on the ℓ_1-penalization simultaneously captures the sudden jumps and infers those nonzero segments in a noisy signal or gene sequence. This novel approach uses the ℓ_1 as well as 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 the algorithmic viewpoint, the idea of fused lasso penalization has its roots in the well-known total variation method (Rudin *et al.*, 1992) that impacted enormously on the modern imaging science. For more recent developments of fused lasso and its variants in network inference, see Shen and Huang (2010); Zhu *et al.* (2013); Wang *et al.* (2016); Shin *et al.* (2016); 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 ℓ_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 individual feature levels. Other advancements using the ℓ_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, theoretical justification of the oracle property (Fan and Li, 2001) of many of them remains quite challenging.

In this paper, instead of finding a desirable solution in one single step, we propose an iterative re-weighted ℓ_2-penalization procedure, referred to as *broken adaptive ridge* (BAR) method. It 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 in signal processing, gene detection, trend filtering or image denoising. Second, it 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, i.e., it performs as well as if the correct underlying model were given in advance. Since the adaptive objective func-

tion 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 by using the Lagrange multiplier technique. The results of numerical studies demonstrate that, compared with the fused lasso, the fused BAR method has a good performance on variable selection and structure estimation.

The rest of the paper is organized as follows. The BAR method is detailed 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. 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 σ^2. Suppose that the response variable $\mathbf{y} = (y_1, \ldots, y_n)$ is centered and the covariates matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_{p_n})$ is standardized by column vectors. We are concerned with the problem of

the recovery of the sparsity and the underlying patterns of feature vectors. Throughout the paper, $\|\cdot\|$ represents the Euclidean norm for a vector and the spectral norm for a matrix.

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

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

where

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

It results from the convexity and differentiability of the objective function in (2.1) that

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

In principle, the ridge estimator

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

is chosen to be the initial value, where $\xi > 0$ is a tuning parameter. The proposed estimator is, thereby, referred to as the *broken adaptive ridge* (BAR) estimator, which is defined as the limit of the iterative algorithm $\hat{\boldsymbol{\beta}}^{(j)} = \mathbf{g}\{\hat{\boldsymbol{\beta}}^{(j-1)}\}$, namely,

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

Since 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. It is worth emphasizing that the data-dependent weight $\{c_k(\tilde{\boldsymbol{\beta}})\}^{-2}$ is more clever than a constant weight c^{-2}. As the sample size grows, the weights for zero $\mathbf{d}_k'\boldsymbol{\beta}$ get to infinity, whereas those for 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 is able to produce a sparse solution as well as estimate local structures of predictors through an iterative procedure.

Noticing that the term $\mathbf{d}_k'\boldsymbol{\beta}$ represents any linear combination of $\boldsymbol{\beta}$, which 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 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 jth component being 1. 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 in this instance is appropriate

to do variable 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 including 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 $j = 1, 2, \ldots, p_n$ and the remaining $\mathbf{d}_j = (0, \ldots, -1, 1, \ldots, 0)'$ with the $(j - p_n)$th element being -1 and $(j - p_n + 1)$th element being 1. In such a way, the expression (2.1) can be written as

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

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 1D fused BAR estimator, since the penalties are imposed both on 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

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

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 to cope 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

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

where $\lambda_1 > 0$ and $\lambda_2 > 0$ are both 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 slops between two consecutive coefficients and thus results 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

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

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

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

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

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

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 the 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 $n \to \infty$,

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

(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 ℓ_2-norm of covariance matrix $\boldsymbol{\Sigma}_n$ is bounded away from zero and infinity. Condition (A2) restricts the number of covariates, the number of nonzero linear combinations of covariates, the tuning parameter, and the smallest nonzero linear combination. It is also made to ensure that those nonzero $\mathbf{d}_k'\boldsymbol{\beta}_0$ are identifiable. Condition (A3) is made for the simplicity of proof and is satisfied for many commonly-used penalties, such as fusion penalty and trend filter penalty. For high-dimensional data, $\|\mathbf{d}_k\|$ would be allowed to diverge to infinity at some rate as $n \to \infty$. Such relaxation would not necessarily affect the asymptotic properties of the BAR estimate, since 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 $\delta_n \to \infty$ such that $\lambda_n/(\delta_n p_n) \to \infty$, define $\mathfrak{B} \equiv \{\boldsymbol{\beta} \in \mathbb{R}^{p_n} : \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\| \le \delta_n\sqrt{p_n/n}\}$. Then, with probability tending to 1:*

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

(b)

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

Remark 1. The statement (2.7) reveals that $\lim_{k\to\infty} \mathbf{T}_2'\hat{\boldsymbol{\beta}}^{(k)} \equiv \mathbf{T}_2'\hat{\boldsymbol{\beta}} = \mathbf{0}$

with probability tending to 1. In other words, the BAR estimator is zero-consistent in the sense that those zero linear combinations of coefficients would be exactly zero as $n \to \infty$. 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, since $\mathbf{T}_2'\boldsymbol{\beta}_0 = \mathbf{0}$, the regression model is reduced to

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

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

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

Similarly, by a gradient rule, we can obtain

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

The asymptotic normality of those nonzero linear combinations $\mathbf{d}_k'\hat{\boldsymbol{\beta}}$, $k = 1, \ldots, q_n$ 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 $\|\mathbf{a}_n\| \le 1$, let $s_n^2 = \sigma^2\mathbf{a}_n'\tilde{\boldsymbol{\Sigma}}_{n1}^{-1}\mathbf{a}_n$. 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 1, there exists a unique fixed point of* $\mathbf{f}(\cdot)$ *denoted by* $\hat{\boldsymbol{\theta}}^\circ$. *Furthermore, as* $n \to \infty$,

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

with probability tending to 1.

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 $\hat{\boldsymbol{\theta}}^\circ$ implies 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 $\mathrm{P}(\mathbf{T}_1'\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\theta}}^\circ) \to 1$ as $n \to \infty$.

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 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 1:*

(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 Lemma 1;*

(ii) $\mathbf{T}_2'\hat{\boldsymbol{\beta}} = \mathbf{0}$;

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

Remark 3. For more insight into the BAR procedure and its oracle properties, we 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 $(q_n+1) \le k \le K_n$. The oracle properties of the BAR estimator is essentially due to 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 $(q_n+1) \le k \le K_n$, the weight $\{\mathbf{d}_k'\mathbf{T}_2\mathbf{T}_2'\hat{\boldsymbol{\beta}}^{(0)}\}^{-2}$ would be 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 $1 \le k \le q_n$, 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) 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. As a result, Algorithm 1 is proposed for the BAR procedure and can be used to do 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 $\mathbf{H}_1(\tilde{\boldsymbol{\beta}}) = \mathrm{diag}(\tilde{\beta}_i^{-2})$, $\mathbf{H}_2(\tilde{\boldsymbol{\beta}}) = \mathrm{diag}\{(\mathbf{M}\tilde{\boldsymbol{\beta}})_i^{-2}\}$ with $(\mathbf{M}\tilde{\boldsymbol{\beta}})_i$ being the ith component of $\mathbf{M}\tilde{\boldsymbol{\beta}}$. Since 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 would stop at a suboptimal value. To avoid those 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 $\mathbf{H}_1(\tilde{\mathbf{z}}) = \mathrm{diag}(\tilde{z}_i^{-2})$ 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

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

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 $\hat{\boldsymbol{\beta}}^\circ = \{\mathbf{X}'\mathbf{X} + \lambda_1 \mathbf{H}_1(\tilde{\boldsymbol{\beta}})\}^{-1}(\mathbf{X}'\mathbf{y} - \mathbf{M}'\mathbf{u}/2)$. In a similar way, 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 get 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 Pinv(). One 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, set $\mathbf{X} = \mathbf{I}$, when we do signal approximation.

4.2. 2D fused BAR implementation

We now investigate the implementation of the 2D fused BAR method with application in image denoising. Distinguishing from 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 $\mathbf{M}_2$ respectively capture the vertical and horizontal neighbors 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 numerically 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

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

It is observed 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 consequence, we set $\mathbf{M}_0 = \mathbf{M}^*$ in the Algorithm 1 for image denoising.

It is worth mentioning that the Algorithm 1 is a general algorithm for the BAR method and its application is not restricted to signal approximation and image processing. Generally speaking, any design of $\mathbf{d}_k$ can be incorporated in $\mathbf{M}$. The Algorithm 1 is flexible in the sense that it allows different penalties imposed on different 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 λ_1 and λ_2 need to be chosen carefully. The BAR method recommends the ridge estimator as the initial $\hat{\boldsymbol{\beta}}^{(0)}$ with the tuning parameter ξ being carefully chosen by the 5-fold cross-validation (CV). However, when sample size n is small, the value of ξ chosen by CV may vary due 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 $p \gg n$, namely

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

Algorithm 1: Fused BAR Algorithm

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

Input $\mathbf{y}$, $\mathbf{X}$, $\mathbf{M}_0$, $\hat{\boldsymbol{\beta}}^{(0)}$, λ_1, λ_2 and ϵ;

$k \leftarrow 0$;

$\tilde{\boldsymbol{\beta}} \leftarrow \hat{\boldsymbol{\beta}}^{(0)}$;

while $(\|\tilde{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}^{(k)}\| > \epsilon \ or \ k = 0)$ **do**

  $\tilde{\boldsymbol{\beta}} \leftarrow \hat{\boldsymbol{\beta}}^{(k)}$;

  $\tilde{\mathbf{z}} \leftarrow \mathbf{M}_0\tilde{\boldsymbol{\beta}}$;

  $\mathbf{H}_1^{-1}(\tilde{\boldsymbol{\beta}}) \leftarrow \mathrm{diag}(\tilde{\beta}_j^2)$;

  $\mathbf{H}_0^{-1}(\tilde{\mathbf{z}}) \leftarrow \mathrm{diag}(\tilde{z}_j^2)$;

  $\mathbf{B}(\tilde{\boldsymbol{\beta}}) \leftarrow \mathbf{H}_1^{-1}(\tilde{\boldsymbol{\beta}})\{\mathbf{X}'\mathbf{X}\mathbf{H}_1^{-1}(\tilde{\boldsymbol{\beta}}) + \lambda_1\mathbf{I}_n\}^{-1}$;

  $\hat{\mathbf{u}}^\circ \leftarrow \mathrm{Pinv}\{\mathbf{M}_0\mathbf{B}(\tilde{\boldsymbol{\beta}})\mathbf{M}_0' + \mathbf{H}_0^{-1}(\tilde{\mathbf{z}})/\lambda_2\}\mathbf{M}_0\mathbf{B}(\tilde{\boldsymbol{\beta}})\mathbf{X}'\mathbf{y}$;

  $k \leftarrow k + 1$;

  $\hat{\boldsymbol{\beta}}^{(k)} \leftarrow \mathbf{B}(\tilde{\boldsymbol{\beta}})(\mathbf{X}'\mathbf{y} - \mathbf{M}_0'\hat{\mathbf{u}}^\circ)$;

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 due to the fact that the univariate regression estimator is zero-consistent in the sense that the estimators of zero coefficients converge to zero, while the estimators of non-zero coeffi-

cients do not converge to zero. Our simulation results evidenced that the univariate estimator is a good initial value.

For the selections of λ_1 and λ_2, we adopt k-fold CV method. Specifically, in signal approximation, we pick all of the odd coefficients as the training set and the even coefficients as the validation set. And we search a grid of λ_1 and λ_2 by the two-fold CV method. For example, 10 grids evenly distributed on the interval $[0.1, 10]$ for λ_1 and 10 grids evenly distributed on $[1, 20]$ for λ_2. Then we select the optimal (λ_1, λ_2) with the minimum CV error by searching over values in the 2D grid. Further, if $\mathbf{X}$ is a general matrix, we recommend to use five-fold CV to find the optimal tuning parameters λ_1 and λ_2.

5. Simulation study

In this section, we carry out simulations on the fused BAR method, the BAR fusion, the fused lasso and the ℓ_1 fusion. Notice that the difference between "fused" and "fusion" constraints is that the former encourages sparsity both in the coefficients and their differences, while the latter only penalizes the flatness of coefficients. For instance, the BAR fusion method is a special case of the BAR method with penalties imposed only on the differences between adjacent coefficients. The comparisons of their performance in terms of variable selection, estimation and prediction are presented. We

use the R package *genlasso* for ℓ_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 of producing a piecewise constant estimate with fewer number of jumps, while Example 4.5 focuses on checking that 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 illustrate the performance of fused BAR method compared with 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

$$\boldsymbol{\beta}_0 = (\underbrace{0, \ldots, 0}_{20}, \underbrace{5, \ldots, 5}_{9}, \underbrace{0, \ldots, 0}_{41}, \underbrace{3.5, \ldots, 3.5}_{9}, \underbrace{0, \ldots, 0}_{21}, \underbrace{4.5, \ldots, 4.5}_{19}, \underbrace{0, \ldots, 0}_{81})'.$$

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

$$\boldsymbol{\beta}_0 = (\underbrace{0,\ldots,0}_{24}, 5, \underbrace{0,\ldots,0}_{125}, \underbrace{4.5,\ldots,4.5}_{49}, \underbrace{0,\ldots,0}_{51})'.$$

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

$$\boldsymbol{\beta}_0 = (\underbrace{0,\ldots,0}_{19}, \underbrace{5,\ldots,5}_{25}, \underbrace{0,\ldots,0}_{56}, \underbrace{3,\ldots,3}_{29}, \underbrace{0,\ldots,0}_{51}, \underbrace{-4,\ldots,-4}_{29}, \underbrace{0,\ldots,0}_{41})'.$$

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

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

$$\boldsymbol{\beta}_0 = (\underbrace{0,\ldots,0}_{9}, \underbrace{-2,\ldots,-2}_{15}, \underbrace{0,\ldots,0}_{26}, \underbrace{4,\ldots,4}_{19}, \underbrace{0,\ldots,0}_{31})'.$$

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

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 20×20 pixel toy image. The noise following 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 of Examples 4.4–4.7 and Examples 4.8–4.9 each with 20 replications, respectively. The two tables present the number of selected features (NOS), the number of falsely selected variables (NOFS), the percent of true nonzero coefficients model selected (TM), the number of jumps (NOJ), the mean absolute bias (MAB), the fitted mean squared error (FMSE), the single value selection (SVS) and the minimum CV error. Figures 1–5 depict the estimated coefficients using the fused BAR, the BAR fusion, the fused lasso and the ℓ_1 fusion for Examples 4.4–4.9. The image processing results of 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 that of fused Lasso. This implies that the resulting fused BAR estimator has a better performance on variable selection. Moreover, Figures 1–3 show that the fused BAR

Table 1: Mean and the standard deviation (in parentheses) of 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 the standard deviation (in parentheses) of results using the fused BAR and the fused Lasso for Examples 4.8–4.9.

	Example 4.8			
	fused BAR	fused Lasso	BAR fusion	ℓ_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	ℓ_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)	–	–

can obtain coefficients that are piecewise constant with fewer jumps. This fact is also evidenced in Table 1, where the NOJ is much smaller than the

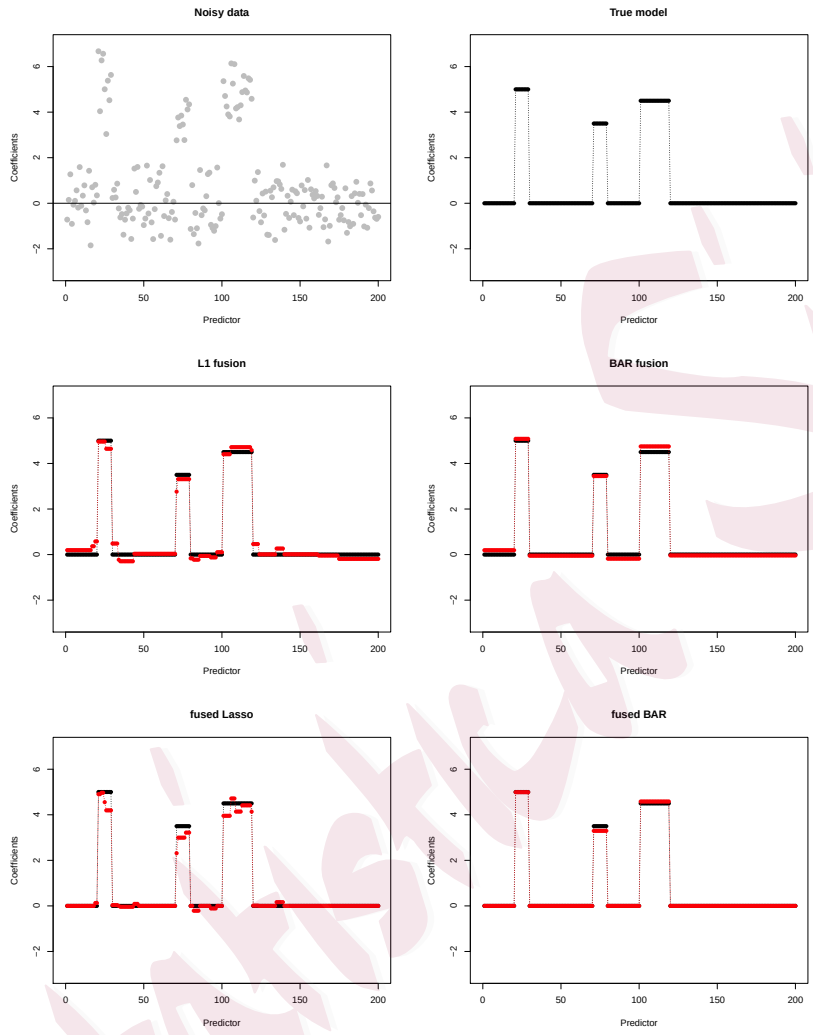

Figure 1: Estimated coefficients using ℓ_1 fusion, BAR fusion, fused lasso and fused BAR for Example 4.4.

fused Lasso. In addition, Figure 2 indicates that the fused BAR method is sensitive to the single value coefficient and its SVS in Table 1 is larger than the fused lasso method. On the other hand, we see from Table 2 and

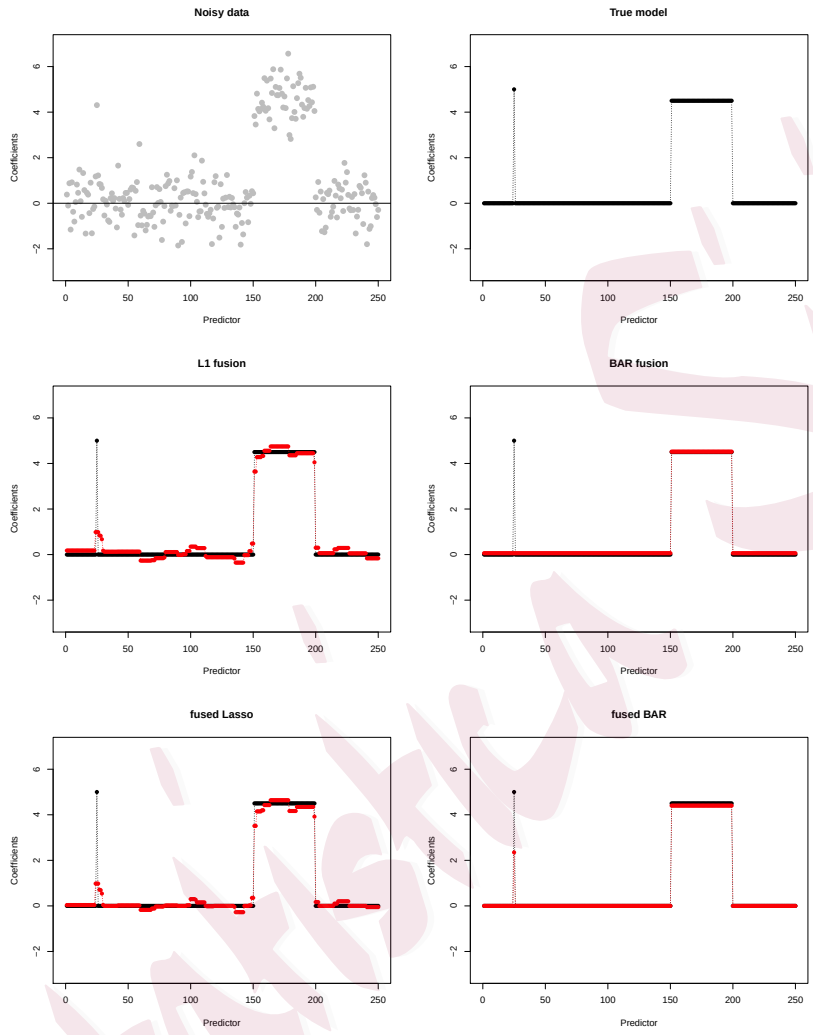

Figure 2: Estimated coefficients using ℓ_1 fusion, BAR fusion, fused lasso and fused BAR for Example 4.5.

Figures 4–5 that when the true jumps are relatively smaller, the fused BAR and BAR fusion can still detect these smaller jumps with a flatter estimate, compared with the fused lasso and the ℓ_1 fusion. In Example 4.9, the

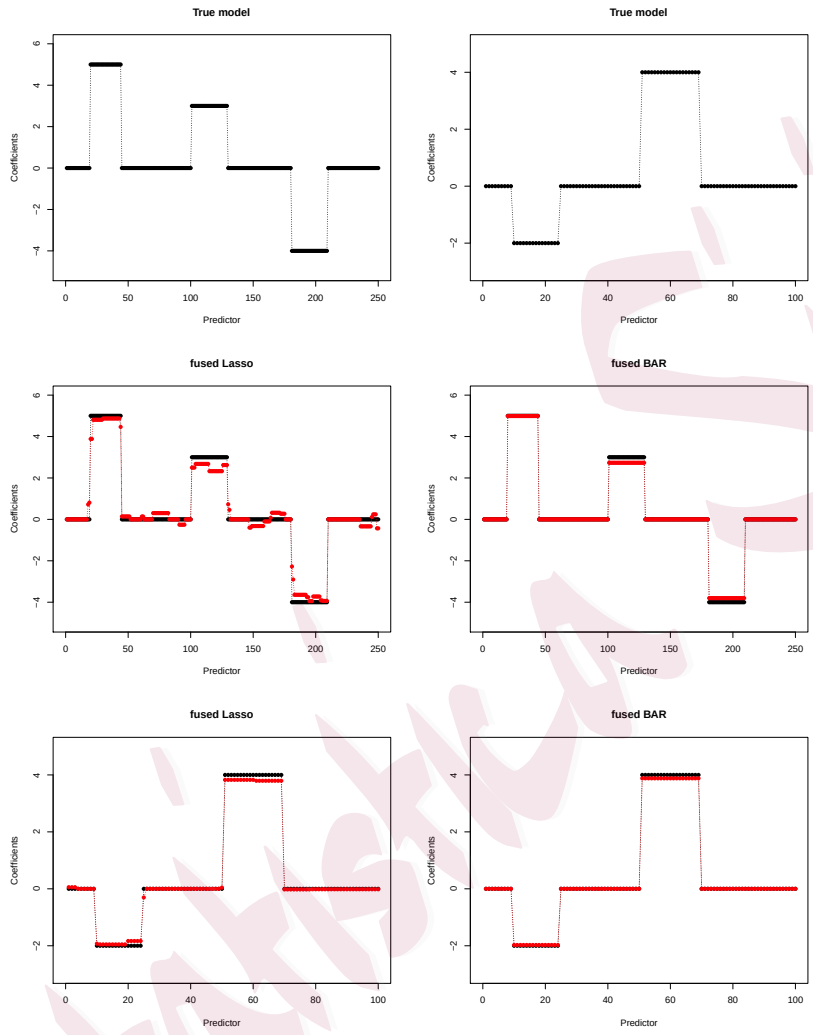

Figure 3: Estimated coefficients using fused lasso and fused BAR for Example 4.6 (the first row) and Example 4.7 (the second row).

fused lasso seems to successfully detect a single nonzero value since it has a larger SVS value seeing from Table 2. However, we observed by our limited experiments that the fused lasso estimate does not essentially capture an

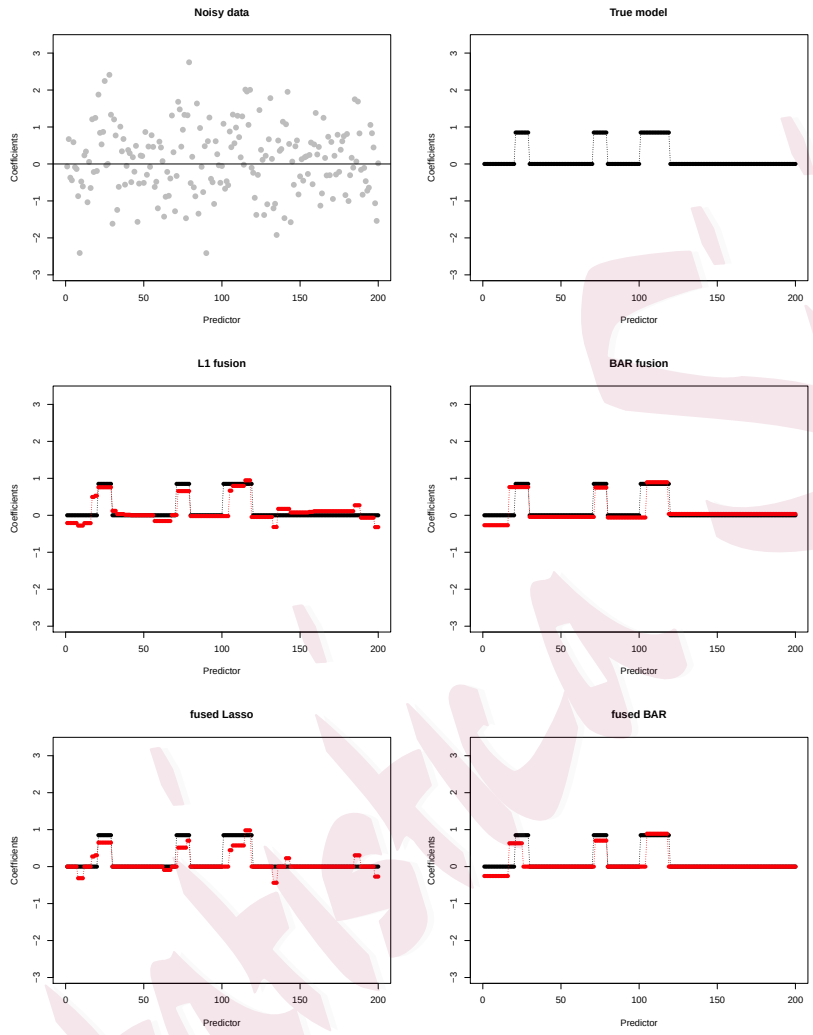

Figure 4: Estimated coefficients using ℓ_1 fusion, BAR fusion, fused lasso and fused BAR for Example 4.8.

up-and-down jumping structure, similar to Figure 5. This phenomenon is reasonable because the single true nonzero is merged with larger noise and thus hard to detect. At last, it is seen from Figure 4 that the fused

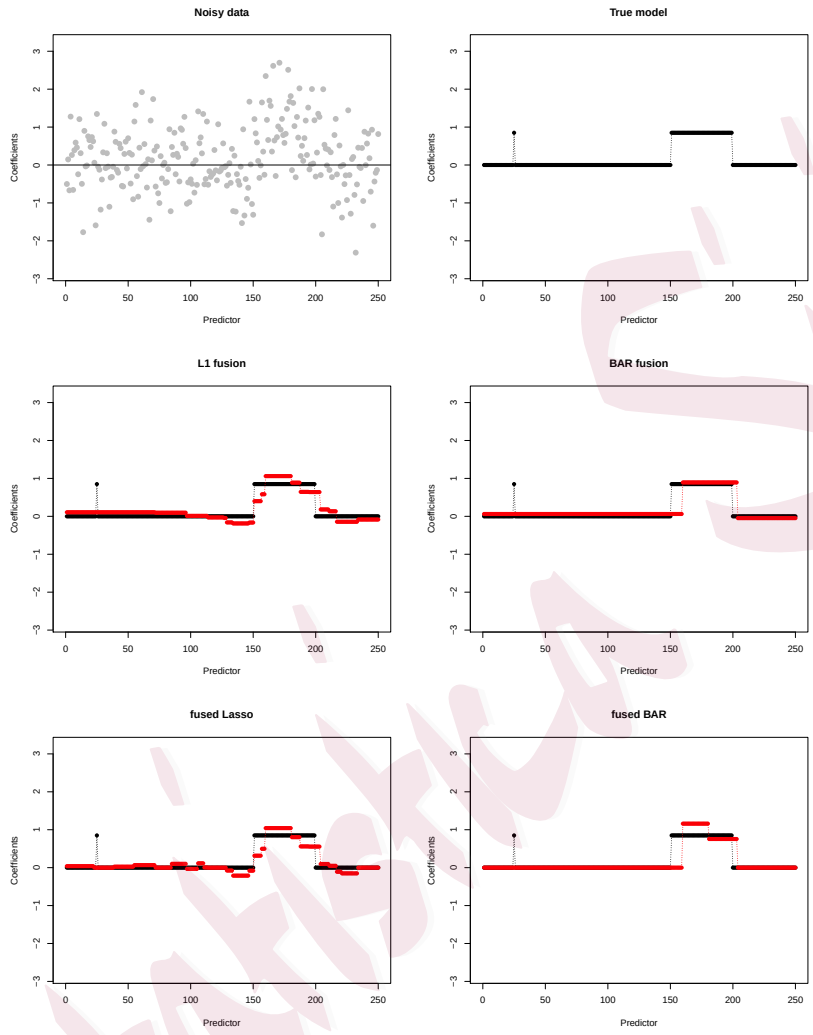

Figure 5: Estimated coefficients using ℓ_1 fusion, BAR fusion, fused lasso and fused BAR for Example 4.9.

BAR method is comparable to the fused lasso on image processing. More 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

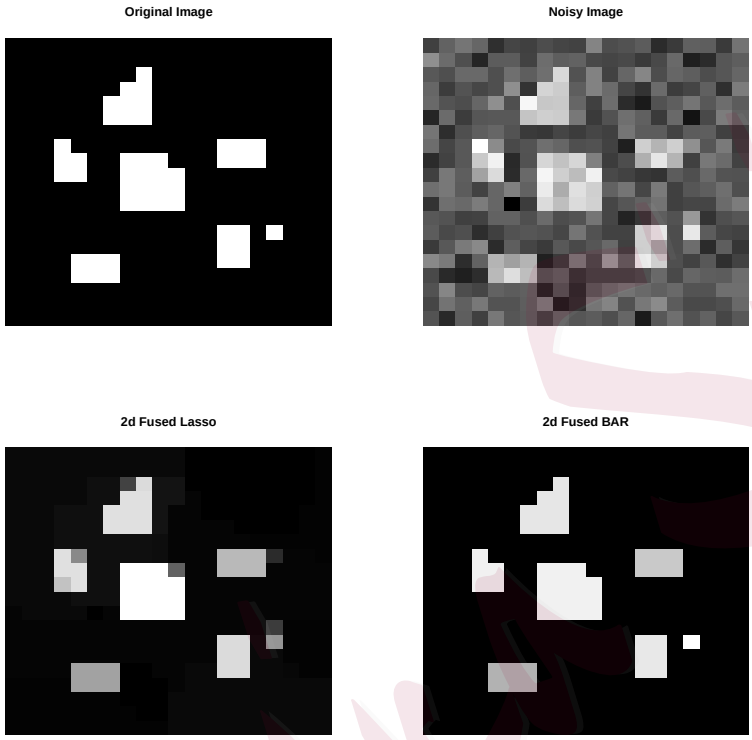

Figure 6: Results of Example 4.8 for image processing using the fused lasso and the fused BAR.

supportive evidence that the fused BAR method works reasonably well in variable selection, estimation and prediction compared with the fused lasso. As one reviewer pointed out, one reason is likely to be that lasso is a biased estimation for large coefficients and differences of coefficients. On the other hand, this phenomenon is possibly due to the variable selection inconsistency of lasso in some scenarios (Zou and Hastie, 2005).

6. Real examples

6.1. CGH array denoising

In cancer research, the copy number variations data (CNV) is one of the important datasets that have an adjacent relationship. CNVs is typically in the form of segments of various lengths (Rippe *et al.*, 2012). The comparative genomic hybridization (CGH) array is a powerful tool 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 be the log2 ratio of the number of DNA copies in tumor cells over that in normal or reference cells. Therefore, a positive CGH value called a gain indicates an increase in the number of DNA copies, while a loss would occur when a negative value is given. CGH signals are usually approximated by a piecewise constant sequence or 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 signal and the inference of segmented values. The fused lasso method has been applied to identify those gains and losses in the CGH arrays (Tibshirani and Wang,

Table 3: Summary of the results of analysing 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

2007).

We apply the fused BAR, BAR fusion, fused Lasso and ℓ_1 fusion methods to the CGH arrays. The CGH data is 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 fused lasso, and the mean squared errors of fused BAR fitting is 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 ℓ_1 fusion.

6.2. Lena image processing

We explore the use of the two-dimensional fused lasso and fused BAR for denoising the Lena image in the R package *filling*. We added Gaussian noise with a standard deviation of 20 to the original image. As zero does not represent a natural baseline in this image, we tried the ℓ_1 fusion model with $\lambda_1 = 0$ and similarly the BAR fusion model. We found the optimal value

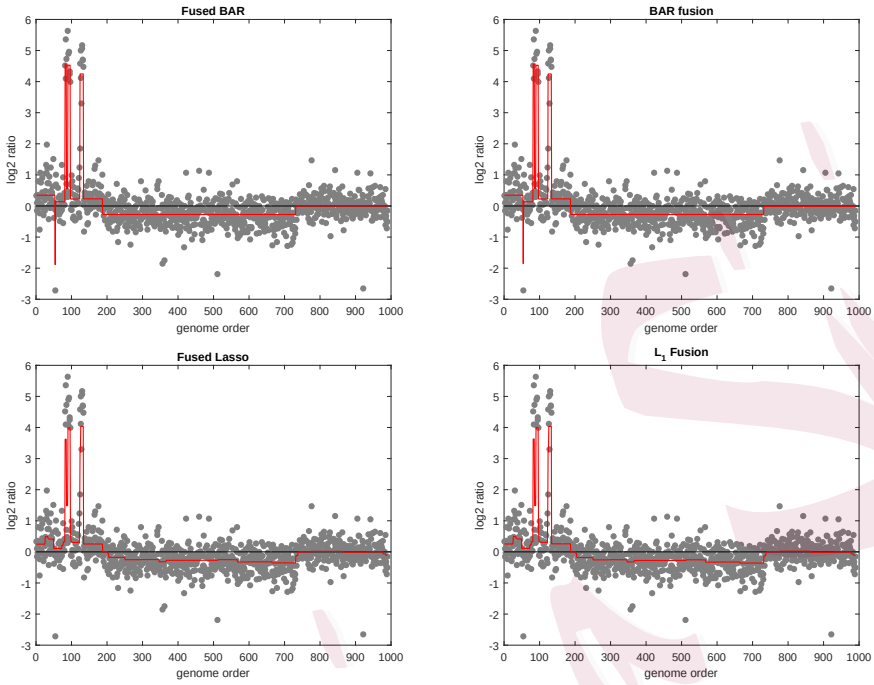

Figure 7: The 1D fused lasso and fused BAR applied to CGH data. The red lines are the estimated CGH signals. The black solid line is $y = 0$.

of λ_2 for each method using two-fold cross-validation. The reconstruction errors from the original noiseless image are 6.256 for the BAR fusion and 6.731 for the ℓ_1 fusion, respectively. Although the BAR fusion has a smaller reconstruction error than the ℓ_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, while the ℓ_1 fusion estimate shown in the bottom left panel recovers more details in the

characterization of Lena.

7. Discussion

In this paper, we have proposed the BAR method to do variable selection and 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 fused BAR method is comparable to 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 Materials

The technical proofs are provided in the supplementary material.

Acknowledgements

The authors are grateful to the Editor, the Associate Editor and two anonymous reviewers for their insightful comments and suggestions that

Figure 8: Top-left panel: 128×128 pixels gray-scale 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 λ_2 chosen by CV. Bottom-right panel: solution of fused BAR with $\lambda_1 = 0$ and λ_2 chosen by CV.

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 with 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., Segraves, 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 ℓ_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 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 Finances and Economics and

Department of Mathematics, Hong Kong University of Science and Technology

E-mail: ldaiab@connect.ust.hk

Kani Chen

Department of Mathematics, Hong Kong University of Science and Technology

E-mail: makchen@ust.hk

Gang Li

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

E-mail: vli@ucla.edu