Journal of Machine Learning Research 18 (2018) 1-36 Submitted 10/16; Revised 8/17; Published 4/18

The DFS Fused Lasso: Linear-Time Denoising over General Graphs

Oscar Hernan Madrid Padilla OMADRID@BERKELEY.EDU
Department of Statistics
University of California
Berkeley, CA 94720, USA

James Sharpnack JSHARPNA@UCDAVIES.EDU
Department of Statistics
University of California
Davis, CA 95616, USA

James G. Scott JAMES.SCOTT@MCCOMBS.UTEXAS.EDU
Department of Information, Risk,
and Operations Management;
Department of Statistics and Data Sciences
University of Texas
Austin, TX 78712, USA

Ryan J. Tibshirani RYANTIBS@CMU.EDU
Machine Learning Department; Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Editor: Massimiliano Pontil

Abstract

The fused lasso, also known as (anisotropic) total variation denoising, is widely used for piecewise constant signal estimation with respect to a given undirected graph. The fused lasso estimate is highly nontrivial to compute when the underlying graph is large and has an arbitrary structure. But for a special graph structure, namely, the chain graph, the fused lasso—or simply, 1d fused lasso—can be computed in linear time. In this paper, we revisit a result recently established in the online classification literature (Herbster et al., 2009; Cesa-Bianchi et al., 2013) and show that it has important implications for signal denoising on graphs. The result can be translated to our setting as follows. Given a general graph, if we run the standard depth-first search (DFS) traversal algorithm, then the total variation of any signal over the chain graph induced by DFS is no more than twice its total variation over the original graph.

This result leads to several interesting theoretical and computational conclusions. Letting m and n denote the number of edges and nodes, respectively, of the graph in consideration, it implies that for an underlying signal with total variation t over the graph, the fused lasso (properly tuned) achieves a mean squared error rate of $t^{2/3}n^{-2/3}$. Moreover, precisely the same mean squared error rate is achieved by running the 1d fused lasso on the DFS-induced chain graph. Importantly, the latter estimator is simple and computationally cheap, requiring $O(m)$ operations to construct the DFS-induced chain and $O(n)$ operations to compute the 1d fused lasso solution over this chain. Further, for trees that have bounded maximum degree, the error rate of $t^{2/3}n^{-2/3}$ cannot be improved, in the sense that it is the

minimax rate for signals that have total variation t over the tree. Finally, several related results also hold—for example, the analogous result holds for a roughness measure defined by the ℓ_0 norm of differences across edges in place of the total variation metric.
Keywords: fused lasso, total variation denoising, graph denoising, depth-first search

1. Introduction

We study the graph denoising problem, i.e., estimation of a signal $\theta_0 \in \mathbb{R}^n$ from noisy data

$$y_i = \theta_{0,i} + \epsilon_i, \quad i = 1, \dots, n, \tag{1}$$

when the components of θ_0 are associated with the vertices of an undirected, connected graph $G = (V, E)$. Without a loss of generality, we denote $V = \{1, \dots, n\}$. Versions of this problem arise in diverse areas of science and engineering, such as gene expression analysis, protein mass spectrometry, and image denoising. The problem is also archetypal of numerous internet-scale machine learning tasks that involve propagating labels or information across edges in a network (e.g., a network of users, web pages, or YouTube videos).

Methods for graph denoising have been studied extensively in machine learning and signal processing. In machine learning, graph kernels have been proposed for classification and regression, in both supervised and semi-supervised data settings (e.g., Belkin and Niyogi (2002); Smola and Kondor (2003); Zhu et al. (2003); Zhou et al. (2005)). In signal processing, a considerable focus has been placed on the construction of wavelets over graphs (e.g., Crovella and Kolaczyk (2003); Coifman and Maggioni (2006); Gavish et al. (2010); Hammond et al. (2011); Sharpnack et al. (2013); Shuman et al. (2013)). We will focus our study on the *fused lasso* over graphs, also known as (anisotropic) *total variation* denoising over graphs. Proposed by Rudin et al. (1992) in the signal processing literature, and Tibshirani et al. (2005) in the statistics literature, the fused lasso estimate is defined by the solution of a convex optimization problem,

$$\hat{\theta}_G = \underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} \ \frac{1}{2}\|y - \theta\|_2^2 + \lambda \|\nabla_G \theta\|_1, \tag{2}$$

where $y = (y_1, \dots, y_n) \in \mathbb{R}^n$ the vector of observed data, $\lambda \geq 0$ is a tuning parameter, and $\nabla_G \in \mathbb{R}^{m \times n}$ is the edge incidence matrix of the graph G. Note that the subscript on the incidence matrix ∇_G and the fused lasso solution $\hat{\theta}_G$ in (2) emphasize that these quantities are defined with respect to the graph G. The edge incidence matrix ∇_G can be defined as follows, using some notation and terminology from algebraic graph theory (e.g., Godsil and Royle (2001)). First, we assign an arbitrary orientation to edges in the graph, i.e., for each edge $e \in E$, we arbitrarily select one of the two joined vertices to be the head, denoted e^+, and the other to be the tail, denoted e^-. Then, we define a row $(\nabla_G)_e$ of ∇_G, corresponding to the edge e, by

$$(\nabla_G)_{e,e^+} = 1, \ \ (\nabla_G)_{e,e^-} = -1, \ \ (\nabla_G)_{e,v} = 0 \ \text{ for all } v \neq e^+, e^-,$$

for each $e \in E$. Hence, for an arbitrary $\theta \in \mathbb{R}^n$, we have

$$\|\nabla_G \theta\|_1 = \sum_{e \in E} |\theta_{e^+} - \theta_{e^-}|.$$

We can see that the particular choice of orientation does not affect the value $\|\nabla_G \theta\|_1$, which we refer to as the *total variation* of θ over the graph G.

1.1 Summary of results

We will wait until Section 1.3 to give a detailed review of literature, both computational and theoretical, on the fused lasso. Here we simply highlight a key computational aspect of the fused lasso to motivate the main results in our paper. The fused lasso solution in (2), for a graph G of arbitrary structure, is highly nontrivial to compute. For a chain graph, however, the fused lasso solution can be computed in linear time (e.g., using dynamic programming or specialized taut-string methods).

The question we address is: how can we use this fact to our advantage, when seeking to solve (2) over an arbitrary graph? Given a generic graph structure G that has m edges and n nodes, it is obvious that we can define a chain graph based on the ordering of nodes produced by depth-first search (DFS). Far less obvious is that for any signal, its total variation along the DFS-induced chain graph never exceeds twice its total variation over the original graph. This fact follows closely from a similar result found in the online graph-structured classification literature (Herbster et al., 2009; Cesa-Bianchi et al., 2013), and for our purposes, it has three notable consequences, described below.

1. No matter the structure of G, we can denoise any signal defined over this graph in $O(m+n)$ operations: $O(m)$ operations for DFS and $O(n)$ operations for the 1d fused lasso on the induced chain. We call the corresponding estimator—the 1d fused lasso run on the DFS-induced chain—the *DFS fused lasso*.

2. For an underlying signal θ_0 that generates the data, as in (1), such that $\theta_0 \in \mathrm{BV}_G(t)$, where $\mathrm{BV}_G(t)$ is the class of signals with total variation at most t, defined in (4), the DFS fused lasso estimator has mean squared error (MSE) on the order of $t^{2/3}n^{-2/3}$.

3. For an underlying signal $\theta_0 \in \mathrm{BV}_G(t)$, the fused lasso estimator over the original graph, in (2), also has MSE on the order of $t^{2/3}n^{-2/3}$.

The fact that such a fast rate, $t^{2/3}n^{-2/3}$, applies for the fused lasso estimator over *any* connected graph structure is somewhat surprising. It implies that the chain graph represents the hardest graph structure for denoising signals of bounded variation—at least, hardest for the fused lasso, since as we have shown, error rates on general connected graphs can be no worse than the chain rate of $t^{2/3}n^{-2/3}$.

We also complement these MSE upper bounds with the following minimax lower bound over trees.

4. When G is a tree of bounded max degree, the minimax MSE over the class $\mathrm{BV}_G(t)$ scales at the rate $t^{2/3}n^{-2/3}$. Hence, in this setting, the DFS fused lasso estimator attains the optimal rate, as does the fused lasso estimator over G.

Lastly, we prove the following for signals with a bounded number of nonzero edge differences.

5. For an underlying signal $\theta_0 \in \mathrm{BD}_G(s)$, where $\mathrm{BD}_G(s)$ is the class of signals with at most s nonzero edge differences, defined in (5), the DFS fused lasso (under a condition on the spacing of nonzero differences over the DFS-induced chain) has MSE on the

order of $s(\log s + \log\log n)\log n/n + s^{3/2}/n$. When G is a tree, the minimax MSE over the class $\mathrm{BD}_G(s)$ scales as $s\log(n/s)/n$. Thus, in this setting, the DFS fused lasso estimator is only off by a $\log\log n$ factor provided that s is small.

Thus DFS fused lasso gives us an $O(n)$ time algorithm for nearly minimax rate-optimal denoising over trees. On paper, this only saves a factor of $O(\log n)$ operations, as recent work (to be described in Section 1.3) has produced an $O(n\log n)$ time algorithm for the fused lasso over trees, by extending earlier dynamic programming ideas over chains. However, dynamic programming on a tree is (a) much more complex than dynamic programming on a chain (since it relies on sophisticated data structures), and (b) noticeably slower in practice than dynamic programming over a chain, especially for large problem sizes. Hence there is still a meaningful difference—both in terms of simplicity and practical computational efficiency—between the DFS fused lasso estimator and the fused lasso over a generic tree.

For a general graph structure, we cannot claim that the statistical rates attained by the DFS fused lasso estimator are optimal, nor can we claim that they match those of fused lasso over the original graph. As an example, recent work (to be discussed in Section 1.3) studying the fused lasso over grid graphs shows that estimation error rates for this problem can be much faster than those attained by the DFS fused lasso (and thus the minimax rates over trees). What should be emphasized, however, is that the DFS fused lasso can still be a practically useful method for any graph, running in linear time (in the number of edges) no matter the graph structure, a scaling that is beneficial for truly large problem sizes.

1.2 Assumptions and notation

Our theory will be primarily phrased in terms of the mean squared error (MSE) for an estimator $\hat{\theta}$ of the mean parameter θ_0 in (1), assuming that $\epsilon = (\epsilon_1, \dots, \epsilon_n)$ has i.i.d. mean zero sub-Gaussian components, i.e.,

$$\mathbb{E}(\epsilon_i) = 0, \quad \text{and} \quad \mathbb{P}(|\epsilon_i| > t) \leq M\exp\big(-t^2/(2\sigma^2)\big), \ \text{all } t \geq 0, \quad \text{for } i = 1, \dots, n, \tag{3}$$

for constants $M, \sigma > 0$. The MSE of $\hat{\theta}$ will be denoted, with a slight abuse of notation, by

$$\|\hat{\theta} - \theta_0\|_n^2 = \frac{1}{n}\|\hat{\theta} - \theta_0\|_2^2.$$

(In general, for a vector $x \in \mathbb{R}^n$, we denote its scaled ℓ_2 norm by $\|x\|_n = \|x\|_2/\sqrt{n}$.) Of course, the MSE will depend not only on the estimator $\hat{\theta}$ in question but also on the assumptions that we make about θ_0. We will focus our study on two classes of signals. The first is the *bounded variation class*, defined with respect to the graph G, and a radius parameter $t > 0$, as

$$\mathrm{BV}_G(t) = \{\theta \in \mathbb{R}^n : \|\nabla_G\theta\|_1 \leq t\}. \tag{4}$$

The second is the *bounded differences class*, defined again with respect to the graph G, and a now a sparsity parameter $s > 0$, as

$$\mathrm{BD}_G(s) = \{\theta \in \mathbb{R}^n : \|\nabla_G\theta\|_0 \leq s\}. \tag{5}$$

We call measure of roughness used in the bounded differences class the *cut metric*, given by replacing the ℓ_1 norm used to define the total variation metric by the ℓ_0 norm, i.e.,

$$\|\nabla_G \theta\|_0 = \sum_{e \in E} 1\{\theta_{e^+} \neq \theta_{e^-}\},$$

which counts the number of nonzero edge differences that appear in θ. Hence, we may think of the former class in (4) as representing a type of weak sparsity across these edge differences, and the latter class in (5) as representing a type of strong sparsity in edge differences.

When dealing with the chain graph, on n vertices, we will use the following modifications to our notation. We write $\nabla_{1\mathrm{d}} \in \mathbb{R}^{(n-1)\times n}$ for the edge incidence matrix of the chain, i.e.,

$$\nabla_{1\mathrm{d}} = \begin{bmatrix} -1 & 1 & 0 & \ldots & 0 \\ 0 & -1 & 1 & \ldots & 0 \\ \vdots & & \ddots & \ddots & \\ 0 & 0 & \ldots & -1 & 1 \end{bmatrix}. \tag{6}$$

We also write $\hat{\theta}_{1\mathrm{d}}$ for the solution of the fused lasso problem in (2) over the chain, also called the *1d fused lasso* solution, i.e., to be explicit,

$$\hat{\theta}_{1\mathrm{d}} = \operatorname*{argmin}_{\theta \in \mathbb{R}^n} \frac{1}{2}\|y - \theta\|_2^2 + \lambda \sum_{i=1}^{n-1} |\theta_{i+1} - \theta_i|. \tag{7}$$

We write $\mathrm{BV}_{1\mathrm{d}}(t)$ and $\mathrm{BD}_{1\mathrm{d}}(s)$ for the bounded variation and bounded differences classes with respect to the chain, i.e., to be explicit,

$$\mathrm{BV}_{1\mathrm{d}}(t) = \{\theta \in \mathbb{R}^n : \|\nabla_{1\mathrm{d}}\theta\|_1 \leq t\},$$
$$\mathrm{BD}_{1\mathrm{d}}(s) = \{\theta \in \mathbb{R}^n : \|\nabla_{1\mathrm{d}}\theta\|_0 \leq s\}.$$

Lastly, in addition to the standard notation $a_n = O(b_n)$, for sequences a_n, b_n such that a_n/b_n is upper bounded for n large enough, we use $a_n \asymp b_n$ to denote that both $a_n = O(b_n)$ and $a_n^{-1} = O(b_n^{-1})$. Also, for random sequences A_n, B_n, we use $A_n = O_{\mathbb{P}}(B_n)$ to denote that A_n/B_n is bounded in probability.

1.3 Related work

Since its inception in the signal processing and statistics communities in Rudin et al. (1992) and Tibshirani et al. (2005), respectively, there has been an impressive amount of work on total variation penalization and the fused lasso. We do not attempt to give a complete coverage, but point out some relevant computational and theoretical advances, covering the two categories separately.

Computational. On the computational side, it is first worth pointing out that there are multiple efficient algorithms for solving the fused lasso problem over a chain graph, i.e., the 1d fused lasso problem. Davies and Kovac (2001) derived an algorithm based on a "taut string" perspective that solves the 1d fused lasso problem in $O(n)$ time (but, the fact

that their taut string method solves the 1d fused lasso problem was not explicitly stated in the work). This was later extended by Condat (2012); Barbero and Sra (2014) to allow for arbitrary weights in both of the individual penalty and loss terms. Johnson (2013) proposed an entirely different $O(n)$ time algorithm for the fused lasso based on dynamic programming. The taut string and dynamic programming algorithms are extremely fast in practice (e.g., they can solve a 1d fused lasso problem with n in the tens of millions in just a few seconds on a standard laptop).

Kolmogorov et al. (2016) extended the dynamic programming approach of Johnson (2013) to solve the fused lasso problem on a tree. Their algorithm is theoretically very efficient, with $O(n \log n)$ running time, but the implementation that achieves this running time (we have found) can be practically slow for large problem sizes, compared to dynamic programming on a chain graph. Alternative implementations are possible, and may well improve practical efficiency, but as far as we see it, they will all involve somewhat sophisticated data structures in the "merge" steps in the forward pass of dynamic programming.

Barbero and Sra (2014) extended (though not in the same direct manner) fast 1d fused lasso optimizers to work over grid graphs, using operator splitting techniques like Douglas-Rachford splitting. Their techniques appear to be quite efficient in practice, and the authors provide thorough comparisons and a thorough literature review of related methods. Over general graphs structures, many algorithms have been proposed, e.g., to highlight a few: Chambolle and Darbon (2009) described a direct algorithm based on a reduction to parametric max flow programming; Hoefling (2010); Tibshirani and Taylor (2011) gave solution path algorithms (tracing out the solution in (2) over all $\lambda \in [0, \infty]$); Chambolle and Pock (2011) described what can be seen as a kind of preconditioned ADMM-style algorithm; Kovac and Smith (2011) described an active set approach; Tansey and Scott (2015) leveraged fast 1d fused lasso solvers in an ADMM decomposition over trails of the graph; most recently, Landrieu and Obozinski (2015) derived a new method based on graph cuts. We emphasize that, even with the advent of these numerous clever computational techniques for the fused lasso over general graphs, it is still far slower to solve the fused lasso over an arbitrary graph than it is to solve the fused lasso over a chain.

Theoretical. On the theoretical side, it seems that the majority of statistical theory on the fused lasso can be placed into two categories: analysis of changepoint recovery, and analysis of MSE. Some examples of works focusing on changepoint recovery are Rinaldo (2009); Harchaoui and Levy-Leduc (2010); Qian and Jia (2012); Rojas and Wahlberg (2014). The statistical theory will concern MSE rates, and hence we give a more detailed review of related literature for this topic.

We begin with results for chain graphs. Mammen and van de Geer (1997) proved, when $\theta_0 \in \mathrm{BV}_{1\mathrm{d}}(t)$, that the 1d fused lasso estimator estimator $\hat{\theta}_{1\mathrm{d}}$ with $\lambda \asymp t^{-1/3} n^{1/3}$ satisfies

$$\|\hat{\theta}_{1\mathrm{d}} - \theta_0\|_n^2 = O_{\mathbb{P}}(t^{2/3} n^{-2/3}). \tag{8}$$

This is indeed the minimax MSE rate for the class $\mathrm{BV}_{1\mathrm{d}}(t)$, as implied by the minimax results in Donoho and Johnstone (1998). (For descriptions of the above upper bound and this minimax rate in a language more in line with that of the current paper, see Tibshirani (2014).) Recently, Lin et al. (2016) improved on earlier results for the bounded differences class in Dalalyan et al. (2014), and proved that when $\theta_0 \in \mathrm{BD}_{1\mathrm{d}}(s)$, the 1d fused lasso

estimator $\hat{\theta}_{1\mathrm{d}}$ with $\lambda \asymp (nW_n)^{1/4}$ satisfies

$$\|\hat{\theta}_{1\mathrm{d}} - \theta_0\|_n^2 = O_{\mathbb{P}}\bigg(\frac{s}{n}\Big((\log s + \log\log n)\log n + \sqrt{n/W_n}\Big)\bigg), \tag{9}$$

where W_n denotes the minimum distance between positions at which nonzero differences occur in θ_0, more precisely, $W_n = \min\{|i-j| : (\nabla_{1\mathrm{d}}\theta_0)_i \neq 0, (\nabla_{1\mathrm{d}}\theta_0)_j \neq 0\}$. When these nonzero differences or "jumps" in θ_0 are evenly spaced apart, we have $W_n \asymp n/s$, and the above becomes, for $\lambda \asymp \sqrt{n}s^{-1/4}$,

$$\|\hat{\theta}_{1\mathrm{d}} - \theta_0\|_n^2 = O_{\mathbb{P}}\bigg(\frac{s(\log s + \log\log n)\log n}{n} + \frac{s^{3/2}}{n}\bigg). \tag{10}$$

This is quite close to the minimax lower bound, whose rate is $s\log(n/s)/n$, that we establish for the class $\mathrm{BD}_{1\mathrm{d}}(s)$, in Theorem 8. (The minimax lower bound that we prove in this theorem actually holds beyond the chain graph, and applies to tree graphs). We can see that the 1d fused lasso rate in (10) is only off by a factor of $\log\log n$, provided that s does not grow too fast (specifically, $s = O((\log n \log\log n)^2)$).

Beyond chain graphs, the story is in general much less clear, however, interesting results are known in special cases. For a d-dimensional grid graph, with $d \geq 2$, Hutter and Rigollet (2016) recently improved on results of Wang et al. (2016), showing that for $\theta_0 \in \mathrm{BV}_G(t) \cap \mathrm{BD}_G(s)$, the fused lasso estimator $\hat{\theta}_G$ over G satisfies

$$\|\hat{\theta}_G - \theta_0\|_n^2 = O_{\mathbb{P}}\bigg(\min\{t, s\}\frac{\log^a n}{n}\bigg). \tag{11}$$

when $\lambda \asymp \log^{a/2} n$, where $a = 2$ if $d = 2$, and $a = 1$ if $d \geq 3$. A minimax lower bound on the MSE rate for the $\mathrm{BV}_G(t)$ class over a grid G of dimension $d \geq 2$ was established to be $t\sqrt{\log(n/t)}/n$, by Sadhanala et al. (2016). This makes the rate achieved by the fused lasso in (11) nearly optimal for bounded variation signals, off by at most a $\log^{3/2} n$ factor when $d = 2$, and a $\log n$ factor when $d \geq 3$.

Wang et al. (2016); Hutter and Rigollet (2016) also derived MSE rates for the fused lasso over several other graph structures, such as Erdos-Renyi random graphs, Ramanujan d-regular graphs, star graphs, and complete graphs. As it is perhaps the most relevant to our goals in this paper, we highlight the MSE bound from Wang et al. (2016) that applies to arbitrary connected graphs. Their Theorem 3 implies, for a generic connected graph G, $\theta_0 \in \mathrm{BV}_G(t)$, that the fused lasso estimator $\hat{\theta}_G$ over G with $\lambda \asymp \sqrt{n\log n}$ satisfies

$$\|\hat{\theta}_G - \theta_0\|_n^2 = O_{\mathbb{P}}\bigg(t\sqrt{\frac{\log n}{n}}\bigg). \tag{12}$$

(See Appendix A.1 for details.) In Theorem 4, we show that the universal $tn^{-1/2}$ rate (ignoring log terms) in (12) for the fused lasso over an arbitrary connected graph can be improved to $t^{2/3}n^{-2/3}$. In Theorem 3, we show that the same rate can indeed be achieved by a simple, linear-time algorithm: the DFS fused lasso.

1.4 Outline

In Section 2, we review a simple but key lemma relating the ℓ_1 norm (and ℓ_0) norm of differences on a tree and a chain induced by running DFS. (This follows closely from an analogous result on binary signals, in Herbster et al. (2009).) We then define the DFS fused lasso estimator. In Section 3, we derive MSE rates for the DFS fused lasso, and the fused lasso over the original graph G under consideration, for signals of bounded variation. We also derive lower bounds for the minimax MSE over trees. In Section 4, we proceed similarly but for signals in the bounded differences class. In Section 5, we present numerical experiments. In Section 6, we summarize our work and also describe some potential extensions.

2. The DFS fused lasso

In this section, we define the DFS-induced chain graph and the DFS fused lasso.

2.1 Tree and chain embeddings

We start by studying some of the fundamental properties associated with total variation on general graphs, and embedded trees and chains. Given a graph $G = (V, E)$, let $T = (V, E_T)$ be an arbitrary spanning tree of G. It is clear that for any signal, its total variation over T is no larger than its total variation over G,

$$\|\nabla_T \theta\|_1 = \sum_{e \in E_T} |\theta_{e^+} - \theta_{e^-}| \leq \sum_{e \in E} |\theta_{e^+} - \theta_{e^-}| = \|\nabla_G \theta\|_1, \quad \text{for all } \theta \in \mathbb{R}^n. \tag{13}$$

The above inequality, albeit very simple, reveals to us the following important fact: if the underlying mean θ_0 in (1) is assumed to be smooth with respect to the graph G, inasmuch as $\|\nabla_G \theta_0\|_1 \leq t$, then it must also be smooth with respect to any spanning tree T of G, since $\|\nabla_T \theta_0\|_1 \leq t$. Roughly speaking, computing the fused lasso solution in (2) over a spanning tree T, instead of G, would therefore still be reasonable for the denoising purposes, as the mean θ_0 would still be smooth over T according to the total variation metric.

The same property as in (14) also holds if we replace total variation by the cut metric:

$$\|\nabla_T \theta\|_0 = \sum_{e \in E_T} 1\{\theta_{e^+} \neq \theta_{e^-}\} \leq \sum_{e \in E} 1\{\theta_{e^+} \neq \theta_{e^-}\} = \|\nabla_G \theta\|_0, \quad \text{for all } \theta \in \mathbb{R}^n. \tag{14}$$

Thus for the mean θ_0, the property $\|\nabla_G \theta_0\|_0 \leq s$ again implies $\|\nabla_T \theta_0\|_0 \leq s$ for any spanning tree T of G, and this would again justify solving the fused lasso over T, in place of G, assuming smoothness of θ_0 with respect to the cut metric in the first place.

Here we go one step further than (13), (14), and assert that analogous properties actually hold for specially embedded chain graphs. The next lemma gives the key result.

Lemma 1 *Let $G = (V, E)$ be a connected graph, where recall we write $V = \{1, \ldots, n\}$. Consider depth-first search (DFS) run on G, and denote by $v_1, \ldots, v_n$ the nodes in the order in which they are reached by DFS. Hence, DFS first visits v_1, then v_2, then v_3, etc. This induces a bijection $\tau : \{1, \ldots, n\} \to \{1, \ldots, n\}$, such that*

$$\tau(i) = v_i, \quad \textit{for all } i = 1, \ldots, n.$$

Let $P \in \mathbb{R}^{n \times n}$ denote the permutation associated with τ. Then it holds that

$$\|\nabla_{1\mathrm{d}} P\theta\|_1 \leq 2\|\nabla_G \theta\|_1, \quad \textit{for all } \theta \in \mathbb{R}^n, \tag{15}$$

as well as

$$\|\nabla_{1\mathrm{d}} P\theta\|_0 \leq 2\|\nabla_G \theta\|_0, \quad \textit{for all } \theta \in \mathbb{R}^n. \tag{16}$$

Proof The proof is simple. Observe that

$$\|\nabla_{1\mathrm{d}} P\theta\|_1 = \sum_{i=1,\ldots,n-1} |\theta_{\tau(i+1)} - \theta_{\tau(i)}|, \tag{17}$$

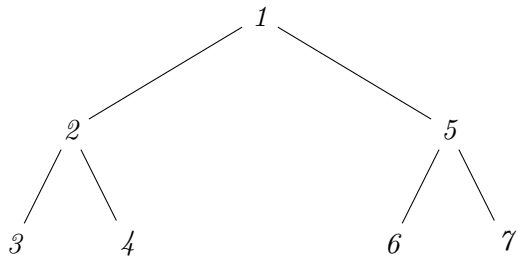
and consider an arbitrary summand $|\theta_{\tau(i+1)} - \theta_{\tau(i)}|$. There are now two cases to examine. First, suppose $\tau(i)$ is not a leaf in the spanning tree constructed by DFS; then there is an edge $e \in E$ such that $\{e^-, e^+\} = \{\tau(i), \tau(i+1)\}$, and $|\theta_{\tau(i+1)} - \theta_{\tau(i)}| = |\theta_{e^+} - \theta_{e^-}|$. Second, suppose that $\tau(i)$ is a leaf node in the DFS tree; then there is a path $p = \{p_1, \ldots, p_r\}$ in the graph such that $p_1 = \tau(i)$, $p_r = \tau(i+1)$, and each $\{p_j, p_{j+1}\} \in E$, $j = 1, \ldots, r-1$, so that by the triangle inequality

$$|\theta_{\tau(i+1)} - \theta_{\tau(i)}| \leq \sum_{j=1}^{r-1} |\theta_{p_{j+1}} - \theta_{p_j}|.$$

Applying this logic over all terms in the sum in (17), and invoking the fundamental property that DFS visits each edge exactly twice (e.g., Chapter 22 of Cormen et al. (2001)), we have established (15). The proof for (16) follows from precisely the same arguments. ■

Remark 2 *Lemma 1 has essentially already appeared in the literature, in a form for binary signals and a roughness metric given by the quadratic form in the graph Laplacian (in place the total variation metric or cut metric), see Herbster et al. (2009). The key idea behind this result and ours is the same, and the proof of Lemma 1 only requires minor modifications. For completeness, we have presented Lemma 1 and its proof anyway. More generally, we should note that graph embeddings—in particular, using chains and trees—are well-known in the online graph-based classification setting. See, e.g., Herbster et al. (2009); Cesa-Bianchi et al. (2013), and references therein.*

Example 1 *The idea behind Lemma 1 can also be clearly demonstrated through an example. We consider G to be a binary tree graph with $n = 7$ nodes, shown below, where we have labeled the nodes according to the order in which they are visited by DFS (i.e., so that here P is the identity).*



In this case,

$$\begin{aligned}\|\Delta_{1\mathrm{d}}\theta\|_1 &= \sum_{i=1}^{6} |\theta_{i+1} - \theta_i| \\ &\leq |\theta_2 - \theta_1| + |\theta_3 - \theta_2| + \big(|\theta_3 - \theta_2| + |\theta_4 - \theta_2|\big) + \big(|\theta_4 - \theta_2| + |\theta_2 - \theta_1| + |\theta_5 - \theta_1|\big) \\ &\quad + |\theta_6 - \theta_5| + \big(|\theta_6 - \theta_5| + |\theta_7 - \theta_5|\big) \\ &\leq 2 \sum_{e \in G} |\theta_{e^+} - \theta_{e^-}| = 2\|\nabla_G \theta\|_1,\end{aligned}$$

where in the inequality above, we have used triangle inequality for each term in parentheses individually.

2.2 The DFS fused lasso

We define the DFS fused lasso estimator, $\hat{\theta}_{\mathrm{DFS}}$, to be the fused lasso estimator over the chain graph induced by running DFS on G. Formally, if τ denotes the bijection associated with the DFS ordering (as described in Lemma 1), then the DFS-induced chain graph can be expressed as $C = (V, E_C)$ where $V = \{1, \ldots, n\}$ and $E_C = \{\{\tau(1), \tau(2)\}, \ldots, \{\tau(n-1), \tau(n)\}\}$. Denoting by P the permutation matrix associated with τ, the edge incidence matrix of C is simply $\nabla_C = \nabla_{1\mathrm{d}} P$, and the DFS fused lasso estimator is given by

$$\begin{aligned}\hat{\theta}_{\mathrm{DFS}} &= \operatorname*{argmin}_{\theta \in \mathbb{R}^n} \frac{1}{2}\|y - \theta\|_2^2 + \lambda \|\nabla_{1\mathrm{d}} P \theta\|_1 \\ &= P^\top \left(\operatorname*{argmin}_{\theta \in \mathbb{R}^n} \frac{1}{2}\|Py - \theta\|_2^2 + \lambda \sum_{i=1}^{n-1} |\theta_{i+1} - \theta_i| \right). \end{aligned} \tag{18}$$

Therefore, we only need to compute the 1d fused lasso estimator on a permuted data vector Py, and apply the inverse permutation operator $P^\top$, in order to compute $\hat{\theta}_{\mathrm{DFS}}$.

Given the permutation matrix P, the computational cost of (18) is $O(n)$, since, to recall the discussion in Section 1.3, the 1d fused lasso problem (7) can be solved in $O(n)$ operations with dynamic programming or taut string algorithms. The permutation P is obtained by running DFS, which requires $O(m)$ operations, and makes the total computation cost of the DFS fused lasso estimator $O(m + n)$.

It should be noted that, when multiple estimates are desired over the same graph G, we must only run DFS once, and all subsequent estimates on the induced chain require just $O(n)$ operations.

The bounds in (15), (16) for the DFS chain are like those in (13), (14) for spanning trees, and carry the same motivation as that discussed above for spanning trees, beneath (13), (14): if the mean θ_0 is assumed to be smooth with respect to t, insofar as its total variation satisfies $\|\nabla_G \theta_0\|_1 \leq t$, then denoising with respect to C would also be reasonable, in that $\|\nabla_{1\mathrm{d}} P \theta_0\|_1 \leq 2t$; the same can be said for the cut metric. However, it is the rapid $O(m+n)$ computational cost of the DFS fused lasso, and also the simplicity of the dynamic programming and taut string algorithms for the 1d fused lasso problem (7), that make (15), (16) particularly appealing compared to (13), (14). To recall the discussion in Section 1.3, the fused lasso can in principle be computed efficiently over a tree, in $O(n \log n)$ operations

using dynamic programming, but this requires a much more cumbersome implementation and in practice we have found it to be noticeably slower.

2.3 Running DFS on a spanning tree

We can think of the induced chain graph, as described in the last section, as being computed in two steps:

(i) run DFS to compute a spanning tree T of G;

(ii) run DFS on the spanning tree T to define the chain C.

Clearly, this is the same as running DFS on G to define the induced chain C, so decomposing this process into two steps as we have done above may seem odd. But this decomposition provides a useful perspective because it leads to the idea that we could compute the spanning tree T in Step (i) in any fashion, and then proceed with DFS on T in Step (ii) in order to define the chain C. Indeed, any spanning tree in Step (i) will lead to a chain C that has the properties (15), (16) as guaranteed by Lemma 1. This may be of interest if we could compute a spanning tree T that better represents the topology of the original graph G, so that the differences over the eventual chain C better mimicks those over G.

An example of a spanning tree whose topology is designed to reflect that of the original graph is a low-stretch spanning tree. Current interest on low-stretch spanning trees began with the breakthrough results in Elkin et al. (2008); most recently, Abraham and Neiman (2012) showed that a spanning tree with average stretch $O(\log n \log\log n)$ can be computed in $O(m \log n \log\log n)$ operations.

In Section 6.4, we discuss a setting in which the fused lasso problem (2) has arbitrary penalty weights, which gives rise to a weighted graph G. In this setting, an example of a spanning tree that can be crafted so that its edges represent important differences in the original graph is a maximum spanning tree. Prim's and Kruskal's minimum spanning tree algorithms, each of which take $O(m \log n)$ time (Cormen et al., 2001), can be used to compute a maximum spanning tree after we negate all edge weights.

2.4 Averaging multiple DFS estimators

Notice that several DFS-induced chains can be formed from a single seed graph G, by running DFS itself on G with different random starts (or random decisions about which edge to follow at each step in DFS), or by computing different spanning trees T of G (possibly themselves randomized) on which we run DFS, or by some combination, etc. Denoting by $\hat{\theta}^{(1)}_{\mathrm{DFS}}, \hat{\theta}^{(2)}_{\mathrm{DFS}}, \ldots, \hat{\theta}^{(K)}_{\mathrm{DFS}}$ the DFS fused lasso estimators fit to K different induced chains, we might believe that the average estimator, $(1/K)\sum_{k=1}^{K} \hat{\theta}^{(k)}_{\mathrm{DFS}}$, will have good denoising performance, as it incorporates fusion at each node in multiple directions. In Section 5, we demonstrate that this intuition holds true (at least, across the set of experiments we consider).

3. Analysis for signals of bounded variation

Throughout this section, we assume that the underlying mean θ_0 in (1) satisfies $\theta_0 \in \mathrm{BV}_G(t)$ for a generic connected graph G. We derive upper bounds on the MSE rates of the DFS

fused lasso and the fused lasso over G. We also derive a tight lower bound on the minimax MSE when G is a tree that has bounded degree.

3.1 The DFS fused lasso

The analysis for the DFS fused lasso estimator is rather straightforward. By assumption, $\|\nabla_G \theta_0\|_1 \leq t$, and thus $\|\nabla_{1\mathrm{d}} P\theta_0\|_1 \leq 2t$ by (15) in Lemma 1. Hence, we may think of our model (1) as giving us i.i.d. data Py around $P\theta_0 \in \mathrm{BV}_{1\mathrm{d}}(2t)$, and we may apply existing results from Mammen and van de Geer (1997) on the 1d fused lasso for bounded variation signals, as described in (8) in Section 1.3. This establishes the following.

Theorem 3 *Consider a data model* (1)*, with i.i.d. sub-Gaussian errors as in* (3)*, and* $\theta_0 \in \mathrm{BV}_G(t)$*, where* G *is a generic connected graph. Then for any DFS ordering of* G *yielding a permutation matrix* P*, the DFS fused lasso estimator* $\hat{\theta}_{\mathrm{DFS}}$ *in* (18)*, with a choice of tuning parameter* $\lambda \asymp t^{-1/3} n^{1/3}$*, has MSE converging in probability at the rate*

$$\|\hat{\theta}_{\mathrm{DFS}} - \theta_0\|_n^2 = O_{\mathbb{P}}(t^{2/3} n^{-2/3}). \tag{19}$$

We note that, if multiple DFS fused lasso estimators $\hat{\theta}^{(1)}_{\mathrm{DFS}}, \hat{\theta}^{(2)}_{\mathrm{DFS}}, \ldots, \hat{\theta}^{(K)}_{\mathrm{DFS}}$ are computed across multiple different DFS-induced chains on G, then the average estimator clearly satisfies the same bound as in (19),

$$\left\| \frac{1}{K} \sum_{k=1}^{K} \hat{\theta}^{(k)}_{\mathrm{DFS}} - \theta_0 \right\|_n^2 = O_{\mathbb{P}}(t^{2/3} n^{-2/3}),$$

provided that K is held constant, by the triangle inequality.

3.2 The graph fused lasso

Interestingly, the chain embedding result (15) in Lemma 1 is not only helpful for establishing the MSE rate for the DFS fused lasso estimator in Theorem 3, but it can also be used to improve the best known rate for the original fused lasso estimator over the graph G. In Section 1.3, we described a result (12) that follows from Wang et al. (2016), establishing an MSE rate of $tn^{-1/2}$ rate (ignoring log terms) for the fused lasso estimator over a connected graph G, when $\|\nabla_G \theta_0\|_1 \leq t$. In fact, as we will now show, this can be improved to a rate of $t^{2/3} n^{-2/3}$, just as in (19) for the DFS fused lasso.

Wang et al. (2016) present a framework for deriving fast MSE rates for fused lasso estimators based on entropy. They show in their Lemma 9 that a bound in probability on the sub-Gaussian complexity

$$\max_{x \in \mathcal{S}_G(1)} \frac{\epsilon^\top x}{\|x\|_2^{1-w/2}}, \tag{20}$$

for some $0 < w < 2$, where $\mathcal{S}_G(1) = \{x \in \mathrm{row}(\nabla_G) : \|\nabla_G x\|_1 \leq 1\}$, leads to a bound in probability on the MSE of the fused lasso estimator $\hat{\theta}_G$ over G. (Wang et al. (2016) actually assume Gaussian errors, but their Lemma 9, Theorem 10, Lemma 11, and Corollary 12 still hold for sub-Gaussian errors as in (3)). The sub-Gaussian complexity in (20) is typically controlled via an entropy bound on the class $\mathcal{S}_G(1)$. Typically, one thinks of controlling

entropy by focusing on specific classes of graph structures G. Perhaps surprisingly, Lemma 1 shows we can uniformly control the sub-Gaussian complexity (20) over all connected graphs.

For any DFS-induced chain C constructed from G, note first that

$$\mathrm{row}(\nabla_G) = \mathrm{span}\{\mathbb{1}\}^\perp = \mathrm{row}(\nabla_C),$$

where $\mathbb{1} = (1, \ldots, 1) \in \mathbb{R}^n$ is the vector of all 1s. This, and (15) in Lemma 1, imply that

$$\max_{x \in \mathcal{S}_G(1)} \frac{\epsilon^\top x}{\|x\|_2^{1-w/2}} \leq \max_{x \in \mathcal{S}_C(2)} \frac{\epsilon^\top x}{\|x\|_2^{1-w/2}}.$$

Now, taking $w = 1$,

$$\max_{x \in \mathcal{S}_C(2)} \frac{\epsilon^\top x}{\|x\|_2^{1/2}} = \max_{\substack{x \,:\, \mathbb{1}^\top x = 0, \\ \|\nabla_{1\mathrm{d}} P x\|_1 \leq 2}} \frac{\epsilon^\top x}{\|x\|_2^{1/2}} = \max_{\substack{x \,:\, \mathbb{1}^\top x = 0, \\ \|\nabla_{1\mathrm{d}} x\|_1 \leq 1}} \frac{2^{-1/2} (P\epsilon)^\top x}{\|x\|_2^{1/2}} = O_{\mathbb{P}}(n^{1/4}).$$

The last step (asserting that the penultimate term is $O_{\mathbb{P}}(n^{1/4})$) holds by first noting that $P\epsilon$ is equal in law to ϵ (as we have assumed i.i.d. components of the error vector), and then applying results on the chain graph in Theorem 10, Lemma 11, and Corollary 12 of Wang et al. (2016). Applying Lemma 9 of Wang et al. (2016), we have now established the following result.

Theorem 4 *Consider a data model* (1)*, with i.i.d. sub-Gaussian errors as in* (3)*, and* $\theta_0 \in \mathrm{BV}_G(t)$*, where* G *is a generic connected graph. Then the fused lasso estimator* $\hat{\theta}_G$ *over* G*, in* (2)*, under a choice of tuning parameter* $\lambda \asymp t^{-1/3} n^{1/3}$*, has MSE converging in probability at the rate*

$$\|\hat{\theta}_G - \theta_0\|_n^2 = O_{\mathbb{P}}(t^{2/3} n^{-2/3}). \tag{21}$$

In a sense, the above theorem suggests that the chain graph is among the hardest graphs for denoising bounded variation signals, since the fused lasso estimator on any connected graph G will achieve an MSE rate in that is at least as good as in the chain rate, if not better. In this vein, it is worth emphasizing that the MSE bound in (21) is not tight for certain graph structures; a good example is the 2d grid, where we must compare (21) from the theorem to the known MSE bound in (11) from Hutter and Rigollet (2016), the latter being only log factors from optimal, as shown in Sadhanala et al. (2016). It is natural for the 2d grid graph to consider the scaling $t \asymp \sqrt{n}$ (as argued in Sadhanala et al. (2016)), in which case the rates for the fused lasso estimator are $n^{-1/3}$ from Theorem 4 versus $(\log^2 n) n^{-1/2}$ from Hutter and Rigollet (2016).

3.3 Minimax lower bound over trees

We derive a lower bound for the MSE over the class $\mathrm{BV}_G(t)$ when G is a tree graph. The proof applies Assouad's Lemma (Yu, 1997), over a discrete set of probability measures constructed by a careful partitioning of the vertices of G, that balances both the sizes of each partition element and the number of edges crossing in between partition elements. It is deferred until Appendix A.2.

Theorem 5 *Consider a data model* (1)*, with i.i.d. Gaussian errors* $\epsilon_i \sim N(0, \sigma^2)$*,* $i = 1, \ldots, n$*, and with* $\theta_0 \in \mathrm{BV}_G(t)$*, where* G *is a tree graph, having maximum degree* $d_{\max}$*. Then there exists absolute constants* $N, C > 0$*, such that for* $n/(td_{\max}) > N$*,*

$$\inf_{\hat{\theta}} \sup_{\theta_0 \in \mathrm{BV}_G(t)} \mathbb{E}\|\hat{\theta} - \theta_0\|_n^2 \geq C\bigg(\frac{t}{\sigma d_{\max}^2 n}\bigg)^{2/3}. \tag{22}$$

The theorem demonstrates that, for trees of bounded degree, such as the chain and balanced d-ary trees, the fused lasso estimator over the tree achieves achieves the minimax rate, as does the DFS fused lasso.

4. Analysis for signals with bounded differences

We assume that the underlying mean θ_0 in (1) satisfies $\theta_0 \in \mathrm{BD}_G(s)$ for a generic connected graph G. We analyze the MSE of the DFS fused lasso, as well as (a particular formulation of) wavelet denoising over G. We again establish a lower bound on the minimax MSE when G is a tree.

4.1 The DFS fused lasso

As it was for the bounded variation case, the analysis for the DFS fused lasso estimator is straightforward. By assumption, $\|\nabla_G \theta_0\|_0 \leq s$, thus $\|\nabla_{1\mathrm{d}} P\theta_0\|_0 \leq 2s$ by (16) in Lemma 1, and we may think of our model (1) as having i.i.d. data Py around $P\theta_0 \in \mathrm{BD}_{1\mathrm{d}}(2s)$. Applying an existing result on the 1d fused lasso for bounded differences signals, as described in (9), from Lin et al. (2016), gives the following result.

Theorem 6 *Consider a data model* (1)*, with i.i.d. sub-Gaussian errors as in* (3)*, and* $\theta_0 \in \mathrm{BD}_G(s)$*, for a connected graph* G*. Consider an arbitrary DFS ordering of* G*, that defines a permutation matrix* P *and the DFS fused lasso estimator* $\hat{\theta}_{\mathrm{DFS}}$ *in* (18)*. Denote by* $W_n = \min\{|i - j| : (\nabla_{1\mathrm{d}} P\theta_0)_i \neq 0, (\nabla_{1\mathrm{d}} P\theta_0)_j \neq 0\}$ *the minimum distance between positions, measured along the DFS-induced chain, at which nonzero differences or jumps occur in* θ_0*. Then, under a choice of tuning parameter* $\lambda \asymp (nW_n)^{1/4}$*, the DFS fused lasso estimator has MSE converging in probability at the rate*

$$\|\hat{\theta}_{\mathrm{DFS}} - \theta_0\|_n^2 = O_{\mathbb{P}}\bigg(\frac{s}{n}\Big((\log s + \log\log n)\log n + \sqrt{n/W_n}\Big)\bigg). \tag{23}$$

Hence, if the s *jumps along the DFS chain are evenly spaced apart, i.e.,* $W_n \asymp n/s$*, then for* $\lambda \asymp \sqrt{n} s^{-1/4}$*,*

$$\|\hat{\theta}_{\mathrm{DFS}} - \theta_0\|_n^2 = O_{\mathbb{P}}\bigg(\frac{s(\log s + \log\log n)\log n}{n} + \frac{s^{3/2}}{n}\bigg). \tag{24}$$

An undesirable feature of applying existing 1d fused lasso results for signals with bounded differences, in the above result, is the dependence on W_n in the DFS fused lasso error bound (23) (we applied the result (9) from Lin et al. (2016), but the bounds from Dalalyan et al. (2014) also depend on W_n, and as far as we can tell, so should any analysis of the 1d fused

lasso for signals with bounded differences). In the 1d setting, assuming that $W_n \asymp n/s$, which says that jumps in θ_0 occur at roughly equally spaced positions, is fairly reasonable; but to assume the same when the jumps are measured with respect to the DFS-induced chain, as we must in order to establish (24), is perhaps not. Even if the differences apparent in θ_0 over edges in G are somehow (loosely speaking) spaced far apart, running DFS could well produce an ordering such that jumps in $P\theta_0$ occur at positions very close together. We reiterate that the MSE bounds for the DFS fused lasso for bounded variation signals, in Theorem 3, do not suffer from any such complications.

4.2 Graph wavelet denoising

We compare the performances of the DFS fused lasso and wavelet denoising using spanning tree wavelets, for signals with bounded differences. For spanning tree wavelets, the construction starts with a spanning tree and carefully defines a hierarchical decomposition by recursively finding and splitting around a balancing vertex, which is a vertex whose adjacent subtrees are of size at most half of the original tree; this decomposition is used to construct an unbalanced Haar wavelet basis, as in Singh et al. (2010). In Sharpnack et al. (2013), it was shown that for any connected graph G, the constructed wavelet basis $W \in \mathbb{R}^{n \times n}$ satisfies

$$\|W\theta\|_0 \leq \lceil \log d_{\max} \rceil \lceil \log n \rceil \|\nabla_G \theta\|_0, \quad \text{for all } \theta \in \mathbb{R}^n, \tag{25}$$

where $d_{\max}$ is the maximum degree of G, and the above holds regardless of choice of spanning tree in the wavelet construction. Now consider the wavelet denoising estimator

$$\hat{\theta}_W = \operatorname*{argmin}_{\theta \in \mathbb{R}^n} \ \frac{1}{2}\|y - \theta\|_2^2 + \lambda \|W\theta\|_1. \tag{26}$$

The following is an immediate consequence of (25), the fact that the wavelet basis W is orthonormal, and standard results about soft-thresholding (e.g., Lemma 2.8 in (Johnstone, 2011)).

Theorem 7 *Consider a data model (1), with i.i.d. Gaussian errors $\epsilon_i \sim N(0, \sigma^2)$, $i = 1, \ldots, n$, and with $\theta_0 \in \mathrm{BV}_G(t)$, where G is a connected graph, having maximum degree $d_{\max}$. Then the spanning tree wavelet estimator $\hat{\theta}_W$ in (26), with a choice $\lambda \asymp \sqrt{\log n}$, has MSE converging in expectation at the rate*

$$\mathbb{E}\|\hat{\theta}_W - \theta_0\|_n^2 = O\bigg(\frac{s \log d_{\max} \log^2 n}{n}\bigg). \tag{27}$$

The result in (27) has the advantage over the DFS fused lasso result in (23) that it does not depend on a hard-to-interpret quantity like W_n, the minimum spacing between jumps along the DFS-induced chain. But when (say) $d_{\max} \asymp 1$, $s \asymp 1$, and we are willing to assume that $W_n \asymp n$ (meaning the jumps of θ_0 occur at positions evenly spaced apart on the DFS chain), we can see that the spanning tree wavelet rate in (27) is just slightly slower than the DFS fused lasso rate in (24), by a factor of $\log n / \log\log n$.

While the comparison between the DFS fused lasso and wavelet rates, (23) and (27), show an advantage to spanning tree wavelet denoising, as it does not require assumptions about the spacings between nonzero differences in θ_0, we have found nonetheless that the

DFS fused lasso to performs well in practice compared to spanning tree wavelets, and indeed often outperforms the latter in terms of MSE. Experiments comparing the two methods are presented Section 5.

4.3 Minimax lower bound for trees

We now derive a lower bound for the MSE over the class $\mathrm{BD}_G(s)$ when G is a tree graph. The proof relates the current denoising problem to one of estimating sparse normal means, with a careful construction of the sparsity set using degree properties of trees. It is deferred until Appendix A.3.

Theorem 8 *Consider a data model* (1)*, with i.i.d. Gaussian errors* $\epsilon_i \sim N(0, \sigma^2)$*,* $i = 1, \ldots, n$*, and with* $\theta_0 \in \mathrm{BD}_G(s)$*, where* G *is a tree. Then there are absolute constants* $N, C > 0$*, such that for* $n/s > N$*,*

$$\inf_{\hat{\theta}} \sup_{\theta_0 \in \mathrm{BD}_G(s)} \mathbb{E}\|\hat{\theta} - \theta_0\|_n^2 \geq C\sigma^2 \frac{s}{n} \log\Big(\frac{n}{s}\Big). \tag{28}$$

The MSE lower bound in (28) shows that, when we are willing to assume that $W_n \asymp n/s$ in the DFS-induced chain, the DFS fused lasso estimator is a $\log\log n$ factor away from the optimal rate, provided that s is not too large, namely $s = O((\log n \log\log n)^2)$. The spanning tree wavelet estimator, on the other hand, is a $\log n$ factor from optimal, without any real restrictions on s, i.e., it suffices to have $s = O(n^a)$ for some $a > 0$. It is worth remarking that, for large enough s, the lower bound in (28) is perhaps not very interesting, as in such a case, we may as well consider the bounded variation lower bound in (22), which will likely be tighter (faster).

5. Experiments

In this section we compare experimentally the speed and accuracy of two approaches for denoising signals on graphs: the graph fused lasso, and the fused lasso along the chain graph induced by a DFS ordering. In our experiments, we see that the DFS-based denoiser sacrifices a modest amount in terms of mean squared error, while providing gains (sometimes considerable) in computational speed. This shows that our main theorem, in addition to providing new insights on MSE rates for the graph fused lasso, also has important practice consequences. For truly massive problems, where the full graph denoising problem is impractical to solve, we may use the linear-time DFS fused lasso denoiser, and obtain a favorable tradeoff of accuracy for speed.

5.1 Generic graphs

We begin by considering three examples of large graphs of (more or less) generic structure, derived from road networks in three states: California, Pennsylvania, and Texas. Data on these road networks are freely available at `https://snap.stanford.edu`. In these networks, intersections and endpoints are represented by nodes, and roads connecting these intersections or endpoints are represented by undirected edges; see Leskovec et al. (2009) for more details. For each network, we use the biggest connected component as our graph structure to run comparisons. The graph corresponding to California has $n = 1957027$ nodes

and $m = 2760388$ edges, the one for Pennsylvania has $n = 1088092$ nodes and $m = 1541898$ edges, and the graph for Texas has $n = 1351137$ nodes and $m = 1879201$ edges. We compare Laplacian smoothing versus the fused lasso over a DFS-induced chain, on the graphs from the three states. We do not compare with the fused lasso over the original graphs, due to its prohibitive computational cost at such large scales.

We used the following procedure to construct a synthetic signal $\theta_0 \in \mathbb{R}^n$ on each of the road network graphs, of piecewise constant nature:

- an initial seed node v_1 is selected uniformly at random from the nodes $V = \{1, \dots, n\}$ in the graph;
- a component C_1 is formed based on the $\lfloor n/10 \rfloor$ nodes closest to v_1 (where the distance between two nodes in the graph is given by the length of the shortest path between them);
- a second seed node v_2 is selected uniformly at random from $G \setminus C_1$;
- a component C_2 is formed based on the $\lfloor n/10 \rfloor$ nodes closest to v_2 (again in shortest path distance);
- this process is repeated[1] until we have a partition $C_1, \dots, C_{10}$ of the node set V into components of (roughly) equal size, and $\theta_0 \in \mathbb{R}^n$ is defined to take constant values on each of these components.

In our experiments, we considered 20 values of the total variation for the underlying signal. For each, the signal θ_0 was scaled appropriately to achieve the given total variation value, and data $y \in \mathbb{R}^n$ was generated by adding i.i.d. $N(0, 0.2^2)$ noise to the components of θ_0. For each data instance y, the DFS fused lasso and Laplacian smoothing estimators, the former defined by (18) and the latter by

$$\hat{\theta}_{\text{Lap}} = \underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} \ \frac{1}{2}\|y - \theta\|_2^2 + \lambda \theta^\top L_G \theta, \tag{29}$$

where $L_G = \nabla_G^\top \nabla_G$ is the Laplacian matrix of the given graph G, and each estimator is computed over 20 values of its own tuning parameter. Then, the value of the tuning parameter minimizing the average MSE, over 50 draws of data y around θ_0, was selected for each method. Finally, this optimized MSE, averaged over the 50 draws of data y, and further, over 10 repetitions of the procedure for constructing the signal θ_0 explained above, was recorded. Figure 1 displays the optimized MSE for the DFS fused lasso and Laplacian smoothing, as the total variation of the underlying signal varies, for the three road network graphs.

As we can see from the figure, for low values of the underlying total variation, i.e., low signal-to-noise ratio (SNR) levels, Laplacian smoothing and the DFS fused lasso, each tuned to optimality, perform about the same. This is because at low enough SNR levels, each will be approximating θ_0 by something like $\bar{y}\mathbb{1}$, with $\bar{y}$ being the sample average of the data vector y. But as the SNR increases, we see that the DFS fused lasso outpeforms

1. Here, whenever isolated small components are unintentionally created, we add them to the big components

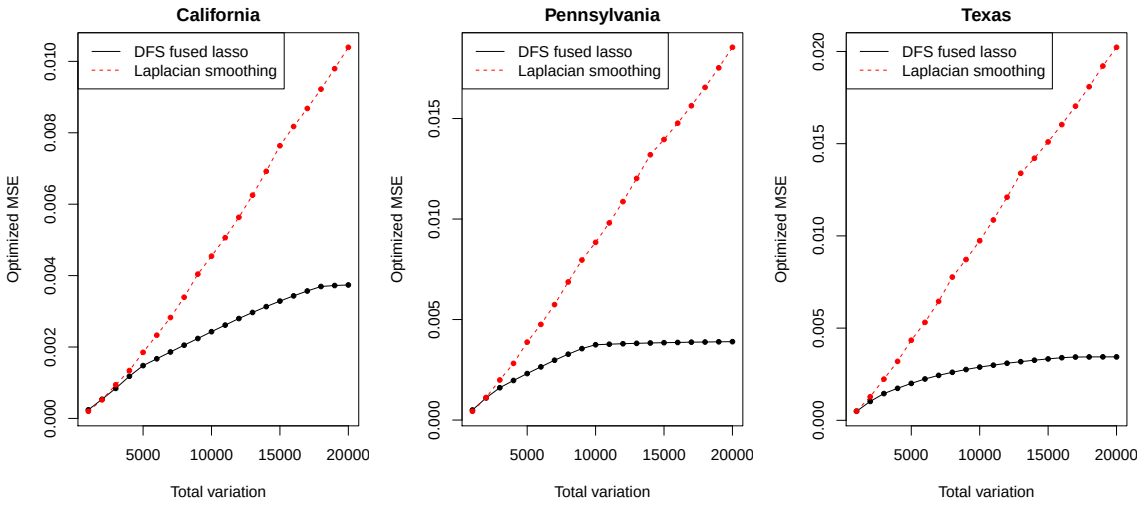


Figure 1: *The optimized MSE for the DFS fused lasso and Laplacian smoothing (i.e., MSE achieved by these methods under optimal tuning) is plotted as a function of the total variation of the underlying signal, for each of the three road network graphs. This has been averaged over 50 draws of data y for each construction of the underlying signal θ_0, and 10 repetitions in constructing θ_0 itself. For low values of the underlying total variation, i.e., low SNR levels, the two methods perform about the same, but as the SNR increases, the DFS fused lasso outperforms Laplacian smoothing by a considerable margin.*

Laplacian smoothing by a considerable amount. This might seem surprising, as Laplacian smoothing uses information from the entire graph, whereas the DFS fused lasso reduces the rich structure of the road network graph in each case to that of an embedded chain. However, Laplacian smoothing is a linear smoother (meaning that $\hat{\theta}_{\text{Lap}}$ in (29) is a linear function of the data y), and therefore it comes with certain limitations when estimating signals of bounded variation (e.g., see the seminal work of Donoho and Johnstone (1998), and the more recent graph-based work of Sadhanala et al. (2016)). In contrast, the DFS fused lasso is a nonlinear estimator, and while it discards some information in the original graph structure, it retains enough of the strong adaptivity properties of the fused lasso over the original graph to statistically dominate a linear estimator like Laplacian smoothing.

Lastly, in terms of computational time, it took an average of 82.67 seconds, 44.02 seconds, and 54.49 seconds to compute the 20 DFS fused lasso solutions (i.e., over the 20 tuning parameter values) for the road network graphs from California, Pennsylvania, and Texas, respectively (the averages are taken over the 50 draws of data y around each signal θ_0, and the 10 repetitions in constructing θ_0). By comparison, it took an average of 2748.26 seconds, 1891.97 seconds, and 1487.36 seconds to compute the 20 Laplacian smoothing solutions for the same graphs. The computations and timings were performed on a standard laptop computer (with a 2.80GHz Intel Core i7-2640M processor). For the DFS fused lasso, in each problem instance, we first computed a DFS ordering using the `dfs` function from the R package `igraph`, which is an R wrapper for a C++ implementation of DFS, and initialized the algorithm at a random node for the root. We then computed the appropriate 1d fused lasso solutions using the `trendfilter` function from the R package `glmgen`, which

is an R wrapper for a C++ implementation of the fast (linear-time) dynamic programming algorithm in Johnson (2013). For Laplacian smoothing, we used the `solve` function from the R package `Matrix`, which is an R wrapper for a C++ implementation of the sparse Cholesky-based solver in Davis and Hager (2009). For such large graphs, alternative algorithms, such as (preconditioned) conjugate gradient methods, could certainly be more efficient in computing Laplacian smoothing solutions; our reported timings are only meant to indicate that the DFS fused lasso is efficiently computable at problem sizes that are large enough that even a simple linear method like Laplacian smoothing becomes nontrivial.

5.2 2d grid graphs

Next we consider a denoising example on a 2d grid graph of dimension 1000×1000, so that the number of nodes is $n = 1000000$ and the number of edges is $m = 1998000$. We generated a synthetic piecewise constant signal $\theta_0 \in \mathbb{R}^{1000 \times 1000}$ over the 2d grid, shown in the top left corner of Figure 2, where a color scale (displayed in the accompanying color legend) is used, with red denoting the smallest possible value and yellow the largest possible value. Data $y \in \mathbb{R}^{1000 \times 1000}$ was generated by adding i.i.d. $N(0, 1)$ noise to the components of θ_0, displayed in the top middle panel of Figure 2. We then computed the 2d fused lasso solution, i.e., the fused lasso solution over the full 2d grid[2] graph, as well as three DFS-based variations: the DFS fused lasso solution using a random DFS ordering (given by running DFS beginning at a random node), labeled as "1 random DFS" in the figure; the average of DFS fused lasso solutions over 5 random DFS orderings, labeled "5 random DFS" in the figure; and the average of DFS fused lasso solutions over 2 "snake" DFS orderings (one given by collecting and joining all horizontal edges and the other all vertical edges) labeled "2 snake DFS" in the figure. The tuning parameter for each method displayed in the figure was chosen to minimize the average MSE over 100 draws of the data y from the specified model. Visually, we can see that the full 2d fused lasso solution is the most accurate, however, the 1 random DFS, 5 random DFS, and 2 snake DFS solutions all still clearly capture the structure inherent in the underlying signal. Of the three DFS variations, the 5 random DFS estimator is visually most accurate; the 1 random DFS estimator is comparably "blotchy", and the 2 snake DFS estimator is comparably "stripey".

The left panel of 3 shows the optimized MSE for each method, i.e., the minimum of the average MSE over 100 draws of the data y, when we consider 20 choices for the tuning parameter. This optimized MSE is plotted as a function of the sample size, which runs from $n = 2500$ (a 50×50 grid) to $n = 1000000$ (a 1000×1000 grid), and in each case the underlying signal is formed by taking an appropriate (sub)resolution of the image in the top left panel of Figure 2. The 2d fused lasso provides the fastest decrease in MSE as n grows, followed by the 5 random DFS estimator, then the 1 random DFS estimator, and the 2 snake DFS estimator. This is not a surprise, since the 2d fused lasso uses the information from the full 2d grid. Indeed, comparing (11) and (19), we recall that the 2d fused lasso enjoys an MSE rate of $t \log^2 n/n$ when θ_0 has 2d total variation t, whereas the

2. In the previous subsection, we remarked that the fused lasso was prohibitive to compute over the road network graphs, which each have in between 1 or 2 million nodes and 1.5 and 3 million edges. Here, we are able to compute the fused lasso over a graph with 1 million nodes and nearly 2 million edges. The difference is the specialized 2d grid structure in the present example, which is leveraged by the proximal stacking technique in Barbero and Sra (2011) that we use to compute the 2d fused lasso solutions

DFS fused lasso has an MSE rate of only $(t/n)^{2/3}$ in this setting. When $t \asymp \sqrt{n}$, which is a natural scaling for the underlying total variation in 2d and also the scaling considered in the experimental setup for the figure, these rates are $(\log^2 n)n^{-1/2}$ for the 2d fused lasso, and $n^{-1/3}$ for the DFS fused lasso. The figure uses a log-log plot, so the MSE curves all appear to have linear trends, and the fitted slopes roughly match these theoretical MSE rates (-0.58 for the 2d fused lasso, and -0.39, -0.40, and -0.36 for the three DFS variations).

The right panel of Figure 3 shows the runtimes for each method (averaged over 100 draws of the data y), as a function of the sample size n. The runtime for each method counts the total time taken to compute solutions across 20 tuning parameter values. The computations and timings were carried out on a standard desktop computer (with a 3.40GHz Intel Core i7-4770 processor). To compute 2d fused lasso solutions, we used the `TVgen` function in the Matlab package `proxTV`, which is a Matlab wrapper for a C++ implementation of the proximal stacking technique described in Barbero and Sra (2014). For the DFS fused lasso, we computed initial DFS orderings using the `dfs` function from the Matlab package `MathBGL`, and then, as before, used the C++ implementation available through `glmgen` to compute the appropriate 1d fused lasso solutions. The figure uses a log-log plot, and hence we can see that all DFS-based estimators are quite a bit more efficient than the 2d fused lasso estimator.

5.3 Tree graphs

We finish with denoising comparisons on tree graphs, for sample sizes varying from $n = 100$ to $n = 5300$. For each sample size n, a random tree is constructed via a sequential process in which each node is assigned a number of children between 2 and 10 (uniformly at random). Given a tree, an underlying signal $\theta_0 \in \mathbb{R}^n$ is constructed to be piecewise constant with total variation $5\sqrt{n}$ (the piecewise constant construction here is made easy because the oriented incidence matrix of a tree is invertible). Data $y \in \mathbb{R}^n$ was generated by adding i.i.d. $N(0, 1)$ noise to θ_0. We compared the fused lasso estimator over the full tree, 1 random DFS and 5 random DFS estimators (using the terminology from the last subsection), and the wavelet smoothing estimator defined in (26). For each estimator, we computed the entire solution path using the path algorithm of Tibshirani and Taylor (2011) implemented in the R package `genlasso`, and selected the step along the path to minimize the average MSE over 50 draws of data y around θ_0, and 10 repetitions in constructing θ_0. (The full solution path can be computed here because each estimator can be cast as a generalized lasso problem, and because the problem sizes considered here are not enormous.)

The left panel of Figure 4 plots this optimized MSE as a function of the sample size n. We see that the fused lasso estimator over the full tree and the 5 random DFS estimator perform more or less equivalently over all sample sizes. The 1 random DFS estimator is slightly worse, and the wavelet smoothing estimator is considerably worse. The right panel shows the the MSE as a function of the effective degrees of freedom of each estimator, for a particular data instance with $n = 5300$. We see that both the tree fused lasso and 1 random DFS estimators achieve their optimum MSEs at solutions of low complexity (degrees of freedom), whereas wavelet smoothing does not come close to achieving this MSE across its entire path of solutions.

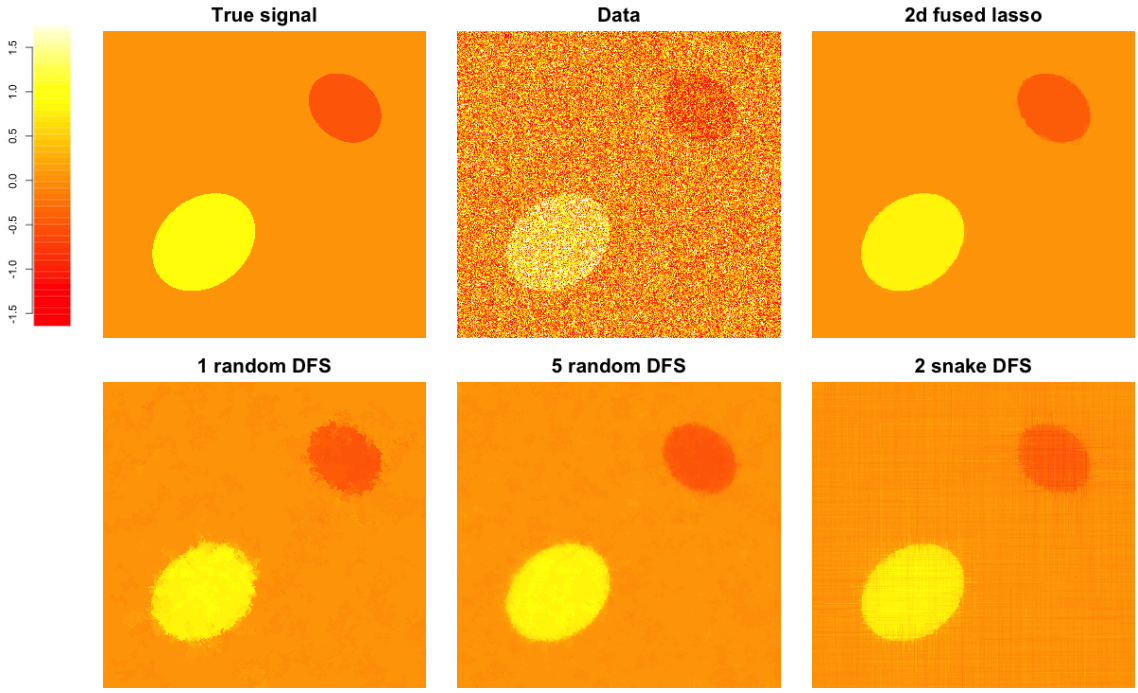


Figure 2: *Underlying signal, data, and solutions from the 2d fused lasso and different variations on the DFS fused lasso fit over a* 1000×1000 *grid.*

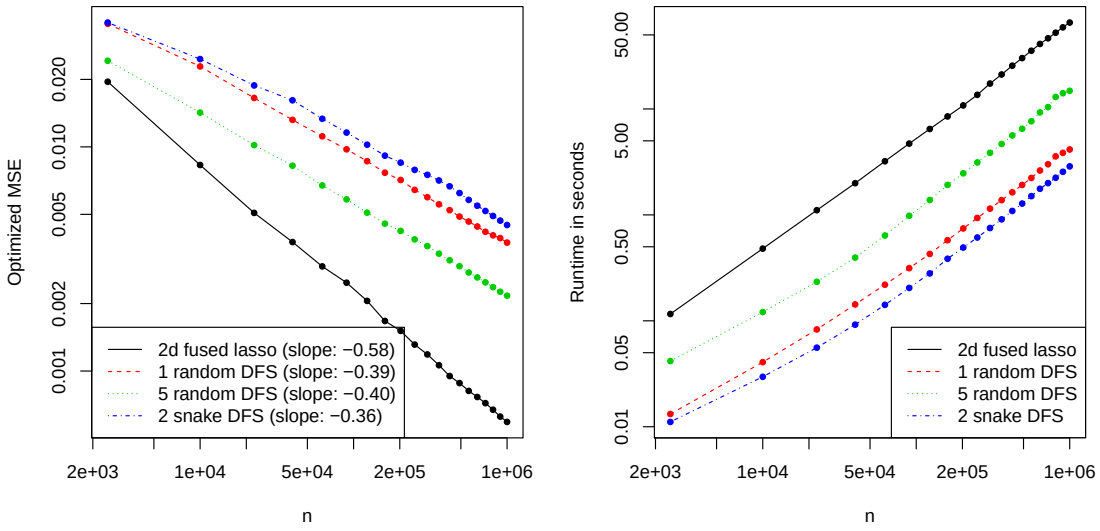


Figure 3: *Optimized MSE and runtime for the 2d fused lasso and DFS fused lasso estimators over a 2d grid, as the grid size n (total number of nodes) varies.*

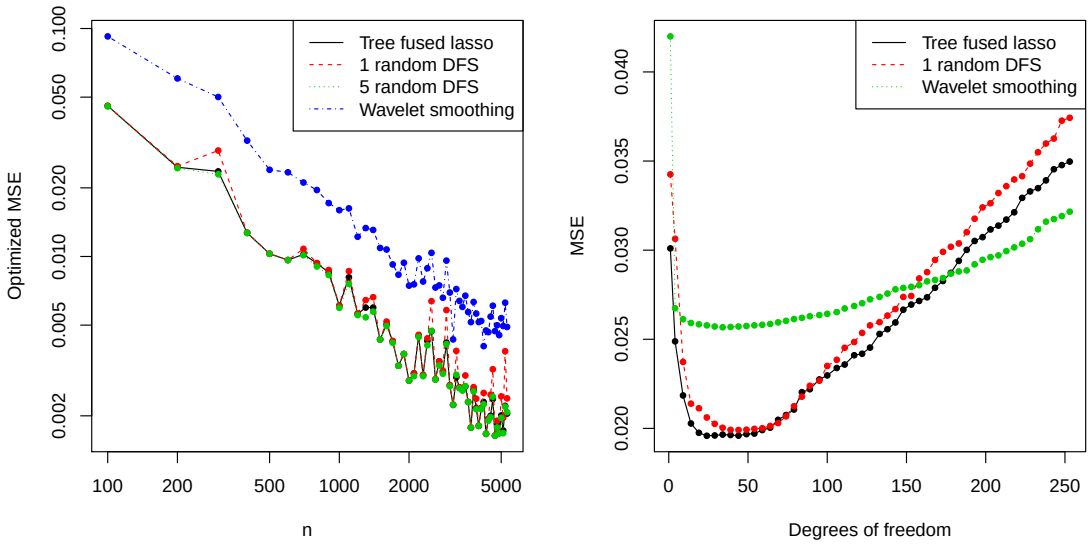


Figure 4: *The left panel shows the optimized MSE as a function of the sample size n for the fused lasso over a tree graph, as well as the 1 random DFS and 5 random DFS estimators, and wavelet smoothing. The right panel shows the MSE as a function of the degrees of freedom of each estimator, for a particular data example with $n = 5300$.*

6. Discussion

Recently, there has been a significant amount on interest on graph-structured denoising. Much of this work has focused on the construction of graph kernels or wavelet bases. We have proposed and studied a simple method, defined by computing the 1d fused lasso over a particular DFS-induced ordering of the nodes of a general graph. This linear-time algorithm comes with strong theoretical guarantees for signals of bounded variation (achieving optimal MSE rates for trees of bounded degree), as well as guarantees for signals with a bounded number of nonzero differences (achieving nearly optimal rates under a condition on the spacings of jumps along the DFS-induced chain). We summarize our theoretical results in Table 1.

Practically, we have seen that the DFS fused lasso can often represent a useful trade-off between computational efficiency and statistical accuracy, versus competing methods that offer better statistical denoising power but are more computationally expensive, especially for large problems. A simple trick like averaging multiple DFS fused lasso fits, over multiple random DFS-induced chains, often improves statistical accuracy at little increased computational cost. Several extensions along these lines, and other lines, are possible. To study any of them in detail is beyond the scope of this paper. We discuss them briefly below, leaving detailed follow-up to future work.

6.1 Beyond simple averaging

Given multiple DFS fused lasso estimators, $\hat{\theta}^{(1)}_{\mathrm{DFS}}, \ldots, \hat{\theta}^{(K)}_{\mathrm{DFS}}$, obtained using multiple DFS-induced chains computed on the same graph G, there are several possibilities for intelligently combining these estimators beyond the simple average, denoted (say) $\bar{\theta}^{(K)}_{\mathrm{DFS}} = (1/K) \sum_{k=1}^{K} \hat{\theta}^{(k)}_{\mathrm{DFS}}$.

	$\mathrm{BV}_G(t)$, $t \asymp 1$	$\mathrm{BD}_G(s)$, $s \asymp 1$
Fused lasso, $\hat{\theta}_G$	$n^{-2/3}$	unknown
Spanning tree wavelets, $\hat{\theta}_W$	unknown	$(\log^2 n \log d_{\max})/n$
DFS fused lasso, $\hat{\theta}_{\mathrm{DFS}}$	$n^{-2/3}$	$(\log n \log\log n)/n^*$
Tree lower bound	$n^{-2/3} d_{\max}^{-4/3}$	$\log n/n$

Table 1: *A summary of the theoretical results derived in this paper. All rates are on the mean squared error (MSE) scale ($\mathbb{E}\|\hat{\theta} - \theta_0\|_n^2$ for an estimator $\hat{\theta}$), and for simplicity, are presented under a constant scaling for t, s, the radii in the $\mathrm{BV}_G(t), \mathrm{BD}_G(s)$ classes, respectively. The superscript "$*$" in the $\mathrm{BD}_G(s)$ rate for the DFS fused lasso is used to emphasize that this rate only holds under the assumption that $W_n \asymp n$. Also, we write $d_{\max}$ to denote the max degree of the graph in question.*

To better preserve edges in the combined estimator, we could run a simple nonlinear filter—for example, a median filter, over $\hat{\theta}^{(1)}_{\mathrm{DFS}}, \ldots, \hat{\theta}^{(K)}_{\mathrm{DFS}}$ (meaning that the combined estimator is defined by taking medians over local neighborhoods of all of the individual estimators). A more sophisticated approach would be to compute the DFS fused lasso estimators sequentially, using the $(k-1)$st estimator to modify the response in some way in the 1d fused lasso problem that defines the kth DFS fused lasso estimator. We are intentionally vague here with the specifics, because such a modification could be implemented in various ways; for example, it could be useful to borrow ideas from the boosting literature, which would have us treat each DFS fused lasso estimator as a weak learner.

6.2 Distributed algorithm

For large graphs, we should be able to both compute a DFS ordering over G, and solve the DFS fused lasso problem in (18), in a distributed fashion. There are many algorithms for distributed DFS, offering a variety of communication and time complexities; see, e.g., Tsin (2002) for a survey. Distributed algorithms for the 1d fused lasso are not as common, though we can appeal to the now well-studied framework for distributed optimization via the alternating direction method of multipliers (ADMM) from Boyd et al. (2011). Different formulations for the auxiliary variables present us with different options for communication costs. We have found that, for a formulation that requires $O(1)$-length messages to be communicated between processors, the algorithm typically converges in a reasonably small number of iterations.

6.3 Theory for piecewise constant signals

The bounded differences class $\mathrm{BD}_G(s)$ in (5) is defined in terms of the cut metric $\|\nabla_G \theta\|_0$ of a parameter θ, which recall, counts the number of nonzero differences occurring in θ over edges in the graph G. The cut metric measures a notion of strong sparsity (compared to the weaker notion measured by the total variation metric) in a signal θ, over edge differences; but, it may not be measuring sparsity on the "right" scale for certain graphs G. Specifically,

the cut metric $\|\nabla_G \theta\|_0$ can actually be quite large for a parameter θ that is piecewise constant over G, with a small number of pieces—these are groups of connected nodes that are assigned the same constant value in θ. Over the 2d grid graph, e.g., one can easily define a parameter θ that has only (say) two constant pieces but on the order of $\sqrt{n}$ nonzero edge differences. Therefore, for such a "simple" configuration of the parameter θ, the cut metric $\|\nabla_G \theta\|_0$ is deceivingly large.

To formally define a metric that measures the number of constant pieces in a parameter θ, with respect to a graph $G = (V, E)$, we introduce a bit of notation. Denote by $Z(\theta) \subseteq E$ the subset of edges over which θ exhibits differences of zero, i.e., $Z(\theta) = \{e \in E : \theta_{e^+} = \theta_{e^-}\}$. Also write $(\nabla_G)_{Z(\theta)}$ for the submatrix of the edge incidence matrix ∇_G with rows indexed by $Z(\theta)$. We consider a metric defined by

$$\rho_G(\theta) = \text{nullity}\big((\nabla_G)_{Z(\theta)}\big),$$

where nullity$(\cdot)$ denotes the dimension of the null space of its argument. An equivalent definition is

$$\rho_G(\theta) = \text{the number of connected components in } (V, E \setminus Z(\theta)).$$

We may now define the *piecewise constant class*, with respect to G, and a parameter $s > 0$,

$$\text{PC}_G(s) = \{\theta \in \mathbb{R}^n : \rho_G(\theta) \leq s\}.$$

It is not hard to to see that $\text{BV}_G(s) \subseteq \text{PC}_G(s)$ (assuming only that G is connected), but for certain graph topologies, the latter class $\text{PC}_G(s)$ will be much larger. Indeed, to repeat what we conveyed above, for the 2d grid one can naturally define a parameter θ such that $\theta \in \text{BD}_G(\sqrt{n})$ and $\theta \in \text{PC}_G(2)$.

We conjecture that the fused lasso estimator over G can achieve a fast MSE rate when the mean θ_0 in (1) exhibits a small number of constant pieces, i.e., $\theta_0 \in \text{PC}_G(s)$, provided that these pieces are of roughly equal size.

6.4 Weighted graphs

The key result in Lemma 1 can be extended to the setting of a weighted graph $G = (V, E, w)$, with $w_e \geq 0$ denoting the edge weight associated to an edge $e \in E$.

Lemma 9 *Let $G = (V, E, w)$ be a connected weighted graph, where recall we write $V = \{1, \ldots, n\}$, and we assume all edge weights are nonnegative. Consider running DFS on G, and denote by $\tau : \{1, \ldots, n\} \to \{1, \ldots, n\}$ the induced permutation, so that if $v_1, \ldots, v_n$ are the nodes in the order that they are traversed by DFS, then*

$$\tau(i) = v_i, \quad \text{for all } i = 1, \ldots, n.$$

Denote $w_{\min} = \min_{e \in E} w_e$, the minimum edge weight present in the graph, and define

$$\tilde{w}_{\tau(i),\tau(i+1)} = \begin{cases} w_e & \text{if } e = \{\tau(i), \tau(i+1)\} \in E, \\ w_{\min} & \text{otherwise,} \end{cases} \quad \text{for all } i = 1, \ldots, n-1. \tag{30}$$

It holds that

$$\sum_{i=1}^{n-1} \tilde{w}_{\tau(i),\tau(i+1)} \big|\theta_{\tau(i+1)} - \theta_{\tau(i)}\big| \leq 2 \sum_{e \in E} w_e |\theta_{e^+} - \theta_{e^-}|, \quad \textit{for all } \theta \in \mathbb{R}^n, \tag{31}$$

as well as

$$\sum_{i=1}^{n-1} \tilde{w}_{\tau(i),\tau(i+1)} 1\big\{\theta_{\tau(i+1)} \neq \theta_{\tau(i)}\big\} \leq 2 \sum_{e \in E} w_e 1\{\theta_{e^+} \neq \theta_{e^-}\}, \quad \textit{for all } \theta \in \mathbb{R}^n. \tag{32}$$

Remark 10 *As with Lemma 1, Lemma 9 has also essentially appeared in the literature, in a form for binary signals and the roughness metric being the quadratic form in the graph Laplacian, see Cesa-Bianchi et al. (2013). The main idea behind this result and Lemma 9 is the same, and the latter follows from the former with only minor modifications.*

For simplicity, when edges in the DFS chain do not appear in the original graph, we have defined their corresponding weights to be the minimum of the weights in the original graph. Instead, we could have taken the approach in Cesa-Bianchi et al. (2013), and defined the weight for such an edge in the DFS chain to be the minimum of weights from original edges in its backtracking path.

The bounds in (31), (32) are the analogies of (15), (16) but for a weighted graph G; indeed we see that we can still embed a DFS chain into G, but this chain itself comes with edge weights, as in (30). These new edge weights in the chain do not cause any computational issues; the 1d fused lasso problem with arbitrary penalty weights can still be solved in $O(n)$ time using the taut string algorithm in Barbero and Sra (2014). Thus, in principle, all of the results in this paper should carry over in some form to weighted graphs.

6.5 Robustness to signal perturbation

Here we briefly explore some robustness properties of the DFS fused lasso estimator, under perbutations of the mean parameter θ_0 in (1) with respect to a small number of nodes. This is based on the approach taken by Cesa-Bianchi et al. (2013) to study robustness in their online graph-structured classification setting. Let $G = (V, E, w)$ a weighted graph, and write $\|\nabla_G \theta\|_1$ for the weighted total variation of a signal θ, i.e.,

$$\|\nabla_G \theta\|_1 = \sum_{e \in E} w_e |\theta_{e^+} - \theta_{e^-}|.$$

Denote by θ^δ a perturbed version of θ, where any number of entries are given by adding an amount $\delta > 0$ to the corresponding entries of θ. (More general forms of perturbations can also be investigated but slightly complicate the discussion that follows, so for simplicity, we will limit ourselves to considering this simple model for perturbations.) Denote by

$$I(\theta, \theta^\delta) = \Big\{ i \in \{1, \ldots, n\} : \theta_i \neq \theta_i^\delta \Big\}$$

the subset of nodes at which the components of θ and θ^δ differ. Observe

$$\begin{aligned}
\big|\|\nabla_G \theta\|_1 \,-\, \|\nabla_G \theta^\delta\|_1\big| &= \Bigg|\sum_{\{e^+,e^-\}\cap I(\theta,\theta^\delta)\neq\emptyset} w_e \left(|\theta_{e^+} - \theta_{e^-}| \,-\, |\theta^\delta_{e^+} - \theta^\delta_{e^-}|\right)\Bigg| \\
&= \Bigg|\sum_{|\{e^+,e^-\}\cap I(\theta,\theta^\delta)|=1} w_e \left(|\theta_{e^+} - \theta_{e^-}| \,-\, |\theta^\delta_{e^+} - \theta^\delta_{e^-}|\right)\Bigg| \\
&\leq \sum_{|\{e^+,e^-\}\cap I(\theta,\theta^\delta)|=1} w_e \left(|\theta_{e^+} - \theta^\delta_{e^+}| \,+\, |\theta_{e^-} - \theta^\delta_{e^-}|\right) \\
&= \delta\, \mathrm{cut}_G(I(\theta,\theta^\delta)),
\end{aligned}$$

where, for a subset $S \subset V$, we use $\mathrm{cut}_G(S)$ to denote the cost of the cut in between S and S^c in the graph G, i.e., the sum of edge weights among edges with one endpoint in S and the other in S^c.

Thus we have shown that the (weighted) total variation metric associated with G has modulus of continuity $\delta\, \mathrm{cut}_G(I(\theta,\theta^\delta))$ with respect to perturbations θ^δ of θ. This scales linearly in δ, but how does it scale in $|I(\theta,\theta^\delta)|$ (the number of discrepancies)? The answer to this depends on the geometry of G, in particular, it depends on whether perturbations occur at nodes having high (weighted) degree. Even when $|I(\theta,\theta^\delta)|$ is small, $\mathrm{cut}_G(I(\theta,\theta^\delta))$ can be very large—an example is given below with G being a star graph. Further, the modulus of continuity $\delta\, \mathrm{cut}_G(I(\theta,\theta^\delta))$ is tight, i.e., it can always be achieved[3], and thus for graphs G with nodes of high (weighted) degrees, in a worst-case sense, small perturbations of θ can lead to big differences in $\|\nabla_G \theta\|_1$.

For the DFS-induced chain graph derived from G, the worst-case is not nearly as bad. Letting C denote this chain and P the corresponding permutation matrix, by the same argument as above,

$$\Big|\|\nabla_C P\theta\|_1 \,-\, \|\nabla_C P\theta^\delta\|_1\Big| \,\leq\, \delta\, \mathrm{cut}_C(I(\theta,\theta^\delta)).$$

But the key difference now is that $\mathrm{cut}_C(I(\theta,\theta^\delta))$ sums the edge weights of at most $2\,|I(\theta,\theta^\delta)|$ edges in C, due to the special structure of the chain. Denoting by

$$v_G(k) \,=\, \max_{F\subset E, |F|=k} \sum_{e\in F} w_e$$

the sum of the k largest edge weights in G, and similarly for C, we can proceed to upper bound the right-hand side above,

$$\Big|\|\nabla_C P\theta\|_1 \,-\, \|\nabla_C P\theta^\delta\|_1\Big| \,\leq\, \delta\, v_C(2\,|I(\theta,\theta^\delta)|) \,\leq\, \delta\, v_C(2\,|I(\theta,\theta^\delta)|),$$

the second inequality following from the definition of the weights in the chain graph, as in (30). When all weights in the original graph G are unity, e.g., this says that perturbing k

3. By this we mean that, for any graph G, any subset $S \subset V$ of nodes at which perturbations are to occur, there exist θ, θ^δ such that $I(\theta,\theta^\delta) = S$ and $|\|\nabla_G \theta\|_1 \,-\, \|\nabla_G \theta^\delta\|_1| = \delta\, \mathrm{cut}_G(I(\theta,\theta^\delta))$.

of the entries of any input signal by an amount δ results in a total variation difference over the DFS-induced chain of at most $2\delta k$.

Consider the following instructive example, borrowed from Cesa-Bianchi et al. (2013). Let G be a star graph, having one center node that is connected to each of the remaining $n-1$ nodes, and no other edges in the graph. Assume that all weights of G are unity, and define θ to take a value 0 on the center node, and 1 on all others. Then by flipping the value of the center node from 0 to 1, the total variation over G changes from $n-1$ to 0. However, for any DFS-induced chain, we see from the above analysis that the total variation can change by at most 2, making it more robust to such a perturbation. We can rephrase this robustness property in an interesting way. Suppose that the mean θ_0 in (1) is defined over the star graph G to take a value 0 on the center node, and 1 on all others. Then θ_0 is "close" to a signal of bounded variation, since changing only its value at the center node makes it have zero total variation, and yet in its current configuration it has total variation $n-1$. Hence, we would not expect the fused lasso estimator over G to be consistent, when fit to data drawn around θ_0. But the total variation of θ_0 as measured over the DFS-induced chain C is at most 2 (as the total variation over C is zero once we perturb the value of center node from 0 to 1). With respect to C then, the parameter θ_0 is certainly of bounded variation and using the DFS fused lasso estimator, we will achieve an MSE rate of $n^{-2/3}$. The DFS fused lasso thus exhibits a robustness property, in that it allows us to accurately estimate signals that are "close to" the set of bounded variation signals over G.

6.6 Potts and energy minimization

Replacing the total variation metric by the cut metric in the fused lasso problem (2) gives us

$$\tilde{\theta}_G = \underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} \ \frac{1}{2}\|y-\theta\|_2^2 + \lambda \|\nabla_G \theta\|_0, \tag{33}$$

often called the *Potts minimization* problem. Because the 1d Potts minimization problem

$$\tilde{\theta}_{1\mathrm{d}} = \underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} \ \frac{1}{2}\|y-\theta\|_2^2 + \lambda \|\nabla_{1\mathrm{d}} \theta\|_0 \tag{34}$$

can be solved efficiently, e.g., in worst-case $O(n^2)$ time with dynamic programming Bellman (1961); Johnson (2013), the same strategy that we have proposed in this paper can be applied to reduce the graph Potts problem (33) to a 1d Potts problem (34), via a DFS ordering of the nodes. This may be especially interesting as the original Potts problem (33) is non–convex and generally intractable (i.e., intractable to solve to global optimality) for an arbitrary graph structure, so a reduction to a worst-case quadratic-time denoiser is perhaps very valuable.

When the optimization domain in (33) is a discrete set, the problem is often called an *energy minimization* problem, as in Boykov et al. (2001). It has not escaped our notice that our technique of denoising over DFS-induced chains could be useful for this setting, as well.

Acknowledgments

The authors thank the Associate Editor and Referees for their helpful comments, especially for pointing out the connections to the previous papers of Herbster et al. (2009); Cesa-Bianchi et al. (2013), and the signal perturbation angle covered in Section 6.5. The last author thanks Veeranjaneyulu Sadhanala and Yu-Xiang Wang for early stimulating discussions. Ryan J. Tibshirani was supported by NSF grant DMS-1554123.

Appendix A. Proofs

A.1 Derivation of (12) from Theorem 3 in Wang et al. (2016)

We first establish a result on the exact form for the inverse of (an augmented version of) the edge incidence matrix of a generic tree $T = (V, E_T)$, where, recall $V = \{1, \dots, n\}$. Without a loss of generality, we may assume that the root of T is at node 1. For $m \leq n$, we define a path in T, of length m, to be a sequence $p_1, \dots, p_m$ such that $\{p_r, p_{r+1}\} \in E_T$ for each $r = 1, \dots, m-1$. We allow for the possibility that $m = 1$, in which case the path has just one node. For any $j, k, \ell = 1, \dots, n$, we say that j is on the path from k to ℓ if there exists a path $p_1, \dots, p_m$ such that $p_1 = k$, $p_m = \ell$ and $p_r = j$ for some $r = 1, \dots, m$. For each node $i = 2, \dots, n$ (each node other than the root), we define its parent $p(i)$ to be the node connected to i which is on the path from the root to i.

We can also assume without a loss of generality that for each $i = 2, \dots, n$, the $(i-1)$st row of ∇_T corresponds to the edge $\{p(i), i\}$, and thus we can write

$$(\nabla_T)_{i-1,j} = \begin{cases} -1 & \text{if } j = p(i), \\ 1 & \text{if } j = i, \\ 0 & \text{if } j \in \{1, \dots, n\} \setminus \{i, p(i)\}. \end{cases}$$

for each $j = 1, \dots, n$. The next lemma describes the inverse of ∇_T, in the appropriate sense.

Lemma 11 *Let $e_1 = (1, 0, \dots, 0) \in \mathbb{R}^n$, and define the matrix $A_T \in \mathbb{R}^{n \times n}$ by*

$$(A_T)_{i,j} = \begin{cases} 1 & \textit{if } j \textit{ is on the path from the root to } i, \\ 0 & \textit{otherwise}, \end{cases} \tag{35}$$

for each $i, j = 1, \dots, n$. Then

$$A_T = \begin{pmatrix} e_1^\top \\ \nabla_T \end{pmatrix}^{-1}.$$

Proof We will prove that the product

$$B = \begin{pmatrix} e_1^\top \\ \nabla_T \end{pmatrix} A_T$$

is the identity. As the root of T corresponds to node 1, we have that by definition of A_T that its first column is

$$(A_T)_{\cdot,1} = (1, \dots, 1),$$

which implies that the first column of B is

$$B_{\cdot,1} = e_1.$$

Moreover, by definition of A_T, its first row is

$$(A_T)_{1,\cdot} = e_1^\top,$$

which implies that the first row of B is

$$B_{1,\cdot} = e_1^\top.$$

Let us now assume that i, j are each not the root. We proceed to consider three cases.

Case 1. Let $j \neq i$, and j be on the path from the root to i. Then j is also on the path from the root to $p(i)$. This implies that

$$B_{ij} = \begin{pmatrix} e_1^\top \\ \nabla_T \end{pmatrix}_{i,\cdot} (A_T)_{\cdot,j} = (\nabla_T)_{i-1,\cdot}(A_T)_{\cdot,j} = 1 - 1 = 0.$$

Case 2. Let $j \neq i$, and j not be on the path from the root to i. Then j is not on the path from the root to $p(i)$, which implies that

$$B_{ij} = \begin{pmatrix} e_1^\top \\ \nabla_T \end{pmatrix}_{i,\cdot} (A_T)_{\cdot,j} = (\nabla_T)_{i-1,\cdot}(A_T)_{\cdot,j} = 0 - 0 = 0.$$

Case 3. Let $j = i$. Then j is on the path from the root to i, and j is not on the path from the root to $p(i)$. Hence,

$$B_{ij} = \begin{pmatrix} e_1^\top \\ \nabla_T \end{pmatrix}_{i,\cdot} (A_T)_{\cdot,j} = (\nabla_T)_{i-1,\cdot}(A_T)_{\cdot,j} = -1 \cdot 0 + 1 \cdot 1 = 1.$$

Assembling these three cases, we have shown that $B = I$, completing the proof. ■

We now establish (12).

Proof [Proof of (12)] The proof of Theorem 3 in Wang et al. (2016) proceeds as in standard basic inequality arguments for the lasso, and arrives at the step

$$\|\Pi^\perp(\hat{\theta}_G - \theta_0)\|_2^2 \leq 2\epsilon^\top \Pi^\perp(\hat{\theta}_G - \theta_0) + 2\lambda\|\nabla_G \theta_0\|_1 - 2\lambda\|\nabla_G \hat{\theta}_G\|_1,$$

where $\Pi^\perp$ is the projection matrix onto the space $\mathrm{span}\{\mathbb{1}\}^\perp$, i.e., the linear space of all vectors orthogonal to the vector $\mathbb{1} = (1, \ldots, 1) \in \mathbb{R}^n$ of all 1s. The proof in Wang et al. (2016) uses the identity $\Pi^\perp = \nabla_G^\dagger \nabla_G$, where $\nabla_G^\dagger$ denotes the pseudoinverse of ∇_G. However, notice that we may also write $\Pi^\perp = \nabla_T^\dagger \nabla_T$ for any spanning tree T of G. Then, exactly the same arguments as in Wang et al. (2016) produce the MSE bound

$$\|\hat{\theta}_G - \theta_0\|_n^2 = O_{\mathbb{P}}\bigg(\frac{M(\nabla_T)\sqrt{\log n}}{n}\|\nabla_G \theta_0\|_1\bigg),$$

where $M(\nabla_T)$ is the maximum ℓ_2 norm among the columns of $\nabla_T^\dagger$. We show below, using Lemma 11, that $M(\nabla_T) \leq \sqrt{n}$, and this gives the desired MSE rate.

For any $b \in \mathbb{R}^{n-1}$, we may characterize $\nabla_T^\dagger b$ as the unique solution $x \in \mathbb{R}^n$ to the linear system

$$\nabla_T x = b,$$

such that $\mathbb{1}^\top x = 0$, i.e., the unique solution to the linear system

$$\begin{pmatrix} e_1^\top \\ \nabla_T \end{pmatrix} x = \begin{pmatrix} a \\ b \end{pmatrix},$$

for a value of $a \in \mathbb{R}$ such that $\mathbb{1}^\top x = 0$. By Lemma 11, we may write

$$x = A_T \begin{pmatrix} a \\ b \end{pmatrix},$$

so that the constraint $0 = \mathbb{1}^\top x = na + \mathbb{1}^\top (A_T)_{\cdot,2:n} b$ gives $a = -(\mathbb{1}/n)^\top (A_T)_{\cdot,2:n} b$, and

$$x = (I - \mathbb{1}\mathbb{1}^\top/n)(A_T)_{\cdot,2:n} b.$$

Evaluating this across $b = e_1, \ldots, e_n$, we find that the maximum ℓ_2 norm of columns of $\nabla_T^\dagger$ is bounded by the maximum ℓ_2 norm of columns of $(A_T)_{\cdot,2:n}$, which, from the definition in (35), is at most $\sqrt{n}$. ■

A.2 Proof of Theorem 5

We first present two preliminary lemmas.

Lemma 12 *Let $S_1, \ldots, S_m$ be a partition of the nodes of G such that the total number of edges with ends in distinct elements of the partition is at most s. Let $k \leq \min_{i=1,\ldots,m} |S_i|$. Then*

$$\inf_{\hat{\theta}} \sup_{\theta_0 \in \mathrm{BV}_G(t)} \mathbb{E}\|\hat{\theta} - \theta_0\|_2^2 \geq \frac{kmt^2}{4\sigma^2 s^2} \exp\bigg(- \frac{kt^2}{\sigma^2 s^2} \bigg).$$

Proof For each $\eta \in \{-1, 1\}^m$, define

$$\theta_\eta = \frac{\delta}{2} \sum_{i=1}^m \eta_i \frac{1_{S_i}}{\sqrt{|S_i|}},$$

where $\delta > 0$ will be specified shortly. Also define the class $\mathcal{P} = \{N(\theta_\eta, \sigma^2 I) : \eta \in \{-1, 1\}^m\}$. Note that $\|\nabla_G \theta_\eta\|_1 \leq \delta s/\sqrt{k}$, so to embed $\mathcal{P}$ into the class $\{N(\theta, \sigma^2 I) : \theta \in \mathrm{BV}_G(t)\}$, we set $\delta = t\sqrt{k}/s$.

Let $\eta, \eta' \in \{-1, 1\}^m$ differ in only one coordinate. Then the KL divergence between the corresponding induced measures in $\mathcal{P}$ is $\|\theta_\eta - \theta_{\eta'}\|_2^2/\sigma^2 \leq \delta^2/\sigma^2$. Hence by Assouad's Lemma (Yu, 1997), and a well-known lower bound on the affinity between probability measures in terms of KL divergence,

$$\inf_{\hat{\theta}} \sup_{\theta_0 \in \mathrm{BV}_G(t)} \mathbb{E}\|\hat{\theta} - \theta_0\|_2^2 \geq \frac{\delta^2 m}{4\sigma^2} \exp\bigg(- \frac{\delta^2}{\sigma^2} \bigg).$$

The result follows by plugging in the specified value for δ. ■

Lemma 13 *Let G be a tree with maximum degree $d_{\max}$, and $k \in \{1, \dots, n\}$ be arbitrary. Then there exists a partition as in Lemma 12, $s = m - 1$, and*

$$k \leq \min_{i=1,\dots,m} |S_i| \leq k(d_{\max} + 1).$$

Proof Our proof proceeds inductively. We begin by constructing S_1', the smallest subtree among all those having size at least k, and generated by a cut of size 1 (i.e., separated from the graph by the removal of 1 edge). Note that $|S_1'| \leq kd_{\max}$, because if not then S_1' has at least k internal nodes, and we can remove its root to produce another subtree whose size is smaller but still at least k.

For the inductive step, assume $S_1', \dots, S_\ell'$ have been constructed. We consider two cases. (For a subgraph G' of G, we denote by $G - G'$ the complement subgraph, given by removing all nodes in G', and all edges incident to a node in G'.)

Case 1. If $|G - \cup_{i=1}^{\ell} S_i'| > k$, then we construct $S_{\ell+1}'$, the smallest subtree of $G - \cup_{i=1}^{l} S_i'$ among all those having size at least k, and generated by a cut of size 1. As before, we obtain that $|S_{\ell+1}'| \leq kd_{\max}$.

Case 2. If $|G - \cup_{i=1}^{\ell} S_i'| \leq k$, then the process is stopped. We define $S_i = S_i'$, $i = 1, \dots, \ell - 1$, as well as $S_\ell = S_\ell' \cup (G - \cup_{i=1}^{\ell} S_i')$. With $m = \ell$, the result follows. ■

We now demonstrate a more precise characterization of the lower bound in Theorem 5, from which the result in the theorem can be derived.

Theorem 14 *Let G be a tree with maximum degree $d_{\max}$. Then*

$$\inf_{\hat{\theta}} \sup_{\theta_0 \in \mathrm{BV}_G(t)} \mathbb{E}\|\hat{\theta} - \theta_0\|_2^2 \geq \frac{t^2}{4e\sigma^2 n} \left(\left(\frac{\sigma n}{2t(d_{\max} + 1)} \right)^{2/3} - 1 \right)^2.$$

Proof Set $s = m - 1$ and

$$k = \left\lfloor \left(\frac{\sigma n}{2t(d_{\max} + 1)} \right)^{2/3} \right\rfloor.$$

By Lemmas 12 and 13,

$$\begin{aligned}
\inf_{\hat{\theta}} \sup_{\theta_0 \in \mathrm{BV}_G(t)} \mathbb{E}\|\hat{\theta} - \theta_0\|_2^2 &\geq \frac{kmt^2}{4\sigma^2 s^2} \exp\left(- \frac{kt^2}{\sigma^2 s^2} \right) \\
&\geq \frac{kt^2}{4\sigma^2 m} \exp\left(- \frac{kt^2}{\sigma^2 (m-1)^2} \right) \\
&\geq \frac{kt^2}{4\sigma^2 m} \exp\left(- \frac{t^2 k^3 (d_{\max} + 1)^2}{\sigma^2 n^2} \frac{m^2}{(m-1)^2} \right) \\
&\geq \frac{kt^2}{4\sigma^2 n} \exp\left(- \frac{4t^2 k^3 (d_{\max} + 1)^2}{\sigma^2 n^2} \right) \\
&\geq \frac{k^2 t^2}{4\sigma^2 n} \exp(-1).
\end{aligned}$$

In the above, the third line uses $n/m \leq k d_{\max}$ as given by Lemma 13, the fourth line simply uses $m \leq n$ and $m^2/(m-1)^2 \leq 4$ (as $m \geq 2$), and the last line uses the definition of k. Thus, because

$$k \geq \left(\frac{\sigma n}{2t(d_{\max}+1)} \right)^{2/3} - 1,$$

we have established the desired result. ■

A.3 Proof of Theorem 8

First we establish that, as G is a tree, the number of nodes of degree at most 2 is at least $n/2$. Denote by d_i be the degree of the node i, for each $i = 1, \ldots, n$. Then

$$2(n-1) = \sum_{i=1}^n d_i = \sum_{i\,:\,d_i \leq 2} d_i + \sum_{i\,:\,d_i \geq 3} d_i \geq |\{i : d_i \leq 2\}| + 3|\{i : d_i \geq 3\}| = 3n - 2|\{i : d_i \leq 2\}|.$$

Hence, rearranging, we find that $|\{i : d_i \leq 2\}| \geq n/2 + 1$.

Let $\mathcal{I} = \{i : d_i \leq 2\}$ so that $|\mathcal{I}| \geq \lceil n/2 \rceil$ and stipulate that $|\mathcal{I}|$ is even without loss of generality. Let k be the largest even number such that $k \leq s/2$. Define

$$\mathcal{B} = \{z \in \mathbb{R}^n : z_{\mathcal{I}} \in \{-1, 0, +1\}^{|\mathcal{I}|},\ z_{\mathcal{I}^c} = 0,\ \|z\|_0 = k\}.$$

Note that by construction $\mathcal{B} \subseteq \mathrm{BD}_G(s)$.

Assume $s \leq n/6$. Then this implies $k/2 \leq n/6 \leq |\mathcal{I}|/3$. By Lemma 4 in Raskutti et al. (2011), there exists $\tilde{\mathcal{B}} \subseteq \mathcal{B}$ such that

$$\log |\tilde{\mathcal{B}}| \geq \frac{k}{2} \log \left(\frac{|\mathcal{I}| - k}{k/2} \right),$$

and $\|z - z'\|_2^2 \geq k/2$ for all $z, z' \in \tilde{\mathcal{B}}$. Defining $\mathcal{B}_0 = 2\delta\tilde{\mathcal{B}}$, for $\delta > 0$ to be specified shortly, we now have $\|z - z'\|_2^2 \geq 2\delta^2 k$ for all $z, z' \in \mathcal{B}_0$.

For $\theta \in \mathcal{B}_0$, let us consider comparing the measure $P_\theta = N(\theta, \sigma^2 I)$ against $P_0 = N(0, \sigma^2 I)$: the KL divergence between these two satisfies $K(P_\theta || P_0) = \|\theta\|_2^2/\sigma^2 = 2\delta^2 k/\sigma^2$. Let $\delta = \sqrt{\alpha\sigma^2/(2k)\log|\mathcal{B}_0|}$, for a parameter $\alpha < 1/8$ that we will specify later. We have

$$\frac{1}{|\mathcal{B}_0|} \sum_{\theta \in \mathcal{B}_0} K(P_\theta || P_0) \leq \alpha \log |\mathcal{B}_0|.$$

Hence by Theorem 2.5 in Tsybakov (2009),

$$\inf_{\hat{\theta}} \sup_{\theta_0 \in \mathrm{BD}_G(s)} \mathbb{P}(\|\hat{\theta} - \theta_0\|_2^2 \geq \delta^2 k) \geq \frac{\sqrt{|\mathcal{B}_0|}}{1 + \sqrt{|\mathcal{B}_0|}} \left(1 - 2\alpha - \sqrt{\frac{2\alpha}{\log|\mathcal{B}_0|}} \right). \tag{36}$$

It holds that

$$\delta^2 k = \frac{\alpha\sigma^2}{2} \log|\mathcal{B}_0| \geq \frac{\alpha\sigma^2 k}{4} \log \frac{|\mathcal{I}| - k}{k/2} \geq C\sigma^2 s \log\left(\frac{n}{s}\right),$$

for some constant $C > 0$ depending on α alone. Moreover, the right-hand side in (36) can be lower bounded by (say) $1/4$ by taking α to be small enough and assuming n/s is large enough. Thus we have established

$$\inf_{\hat{\theta}} \sup_{\theta_0 \in \mathrm{BD}_G(s)} \mathbb{P}\left(\|\hat{\theta} - \theta_0\|_2^2 \geq C\sigma^2 s \log\left(\frac{n}{s}\right) \right) \geq \frac{1}{4},$$

and the result follows by Markov's inequality.

References

Ittai Abraham and Ofer Neiman. Using petal-decompositions to build a low stretch spanning tree. *ACM Symposium on Theory of Computing*, 44:395–406, 2012.

Alvaro Barbero and Suvrit Sra. Fast Newton-type methods for total variation regularization. *International Conference on Machine Learning*, 28:313–320, 2011.

Álvaro Barbero and Suvrit Sra. Modular proximal optimization for multidimensional total-variation regularization. *arXiv preprint arXiv:1411.0589*, 2014.

Mikhail Belkin and Partha Niyogi. Using manifold structure for partially labelled classification. *Advances in Neural Information Processing Systems*, 15, 2002.

Richard Bellman. On the approximation of curves by line segments using dynamic programming. *Communications of the ACM*, 4(6):284, 1961.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11): 1–18, 2001.

Nicolo Cesa-Bianchi, Claudio Gentile, Fabio Vitale, and Giovanni Zappella. Random spanning trees and the prediction of weighted graphs. *Journal of Machine Learning Research*, 14(1):1251–1284, 2013.

Antonin Chambolle and Jérôme Darbon. On total variation minimization and surface evolution using parametric maximum flows. *International Journal of Computer Vision*, 84(3):288–307, 2009.

Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40: 120–145, 2011.

Ronald Coifman and Mauro Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(2):53–94, 2006.

Laurent Condat. A direct algorithm for 1d total variation denoising. *HAL preprint hal-00675043*, 2012.

Thomas Cormen, Clifford Stein, Ronald Rivest, and Charles Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

Mark Crovella and Eric Kolaczyk. Graph wavelets for spatial traffic analysis. *Annual Joint Conference of the IEEE Computer and Communications IEEE Societies*, 3:1848–1857, 2003.

Arnak Dalalyan, Mohamed Hebiri, and Johannes Lederer. On the prediction performance of the lasso. *To appear, Bernoulli*, 2014.

P. Laurie Davies and Arne Kovac. Local extremes, runs, strings and multiresolution. *Annals of Statistics*, 29(1):1–65, 2001.

Timothy Davis and William Hager. Dynamic supernodes in sparse Cholesky update/downdate and triangular solves. *ACM Transactions on Mathematical Software*, 35 (4):1–23, 2009.

David L Donoho and Iain M Johnstone. Minimax estimation via wavelet shrinkage. *Annals of Statistics*, 26(8):879–921, 1998.

Michael Elkin, Yuval Emek, Daniel Spielman, and Shang-Hua Teng. Lower-stretch spanning trees. *SIAM Journal on Computing*, 38(2):608–628, 2008.

Matan Gavish, Boaz Nadler, and Ronald Coifman. Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning. *International Conference on Machine Learning*, 27, 2010.

Chris Godsil and Gordon Royle. *Algebraic Graph Theory*. Springer, 2001.

David Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

Zaid Harchaoui and Celine Levy-Leduc. Multiple change-point estimation with a total variation penalty. *Journal of the American Statistical Association*, 105(492):1480–1493, 2010.

Mark Herbster, Guy Lever, and Massimiliano Pontil. Online prediction on large diameter graphs. In *Advances in Neural Information Processing Systems*, pages 649–656, 2009.

Holger Hoefling. A path algorithm for the fused lasso signal approximator. *Journal of Computational and Graphical Statistics*, 19(4):984–1006, 2010.

Jan-Christian Hutter and Philippe Rigollet. Optimal rates for total variation denoising. *Annual Conference on Learning Theory*, 29:1115–1146, 2016.

Nicholas Johnson. A dynamic programming algorithm for the fused lasso and l_0-segmentation. *Journal of Computational and Graphical Statistics*, 22(2):246–260, 2013.

Iain Johnstone. Gaussian estimation: sequence and wavelet models. *Unpublished manuscript*, 2011.

Vladimir Kolmogorov, Thomas Pock, and Michal Rolinek. Total variation on a tree. *SIAM Journal of Imaging Sciences*, 9(2):605–636, 2016.

Arne Kovac and Andrew Smith. Nonparametric regression on a graph. *Journal of Computational and Graphical Statistics*, 20(2):432–447, 2011.

Loic Landrieu and Guillaume Obozinski. Cut pursuit: fast algorithms to learn piecewise constant functions on general weighted graphs. *HAL preprint hal-01306779*, 2015.

Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

Kevin Lin, James Sharpnack, Alessandro Rinaldo, and Ryan J Tibshirani. Approximate recovery in changepoint problems, from ℓ_2 estimation error rates. *arXiv preprint arXiv:1606.06746*, 2016.

Enno Mammen and Sara van de Geer. Locally apadtive regression splines. *Annals of Statistics*, 25(1):387–413, 1997.

Junyang Qian and Jinzhu Jia. On pattern recovery of the fused lasso. *arXiv preprint arXiv:1211.5194*, 2012.

Garvesh Raskutti, Martin Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE Transactions on Information Theory*, 57(10):6976–6994, 2011.

Alessandro Rinaldo. Properties and refinements of the fused lasso. *The Annals of Statistics*, 37(5):2922–2952, 2009.

Cristian R Rojas and Bo Wahlberg. On change point detection using the fused lasso method. *arXiv preprint arXiv:1401.5408*, 2014.

Leonid Rudin, Stanley Osher, and Emad Faterni. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

Veeranjaneyulu Sadhanala, Yu-Xiang Wang, and Ryan J. Tibshirani. Total variation classes beyond 1d: Minimax rates, and the limitations of linear smoothers. *To appear, Neural Information Processing Systems*, 2016.

James Sharpnack, Akshay Krishnamurthy, and Aarti Singh. Detecting activations over graphs using spanning tree wavelet bases. *International Conference on Artificial Intelligence and Statistics*, 16:536–544, 2013.

David Shuman, Sunil Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30 (3):83–98, 2013.

Aarti Singh, Robert Nowak, and Robert Calderbank. Detecting weak but hierarchically-structured patterns in networks. *International Conference on Artificial Intelligence and Statistics*, 13:749–756, 2010.

Alexander Smola and Risi Kondor. Kernels and regularization on graphs. *Annual Conference on Learning Theory*, 16, 2003.

Wesley Tansey and James Scott. A fast and flexible algorithm for the graph-fused lasso. *arXiv preprint arXiv:1505.06475*, 2015.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B*, 67 (1):91–108, 2005.

Ryan J. Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323, 2014.

Ryan J. Tibshirani and Jonathan Taylor. The solution path of the generalized lasso. *Annals of Statistics*, 39(3):1335–1371, 2011.

Y. H. Tsin. Some remarks on distributed depth-first search. *Information Processing Letters*, 82:173–178, 2002.

Alexander Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

Yu-Xiang Wang, James Sharpnack, Alex Smola, and Ryan J Tibshirani. Trend filtering on graphs. *Journal of Machine Learning Research*, 17(105):1–41, 2016.

Bin Yu. Assouad, Fano, and Le Cam. In *Festschrift for Lucien Le Cam*, pages 423–435. Springer, 1997.

Dengyong Zhou, Jiayuan Huang, and Bernhard Scholkopf. Learning from labeled and unlabeled data on a directed graph. *International Conference on Machine Learning*, 22: 1036–1043, 2005.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. *International Conference on Machine Learning*, 20: 912–919, 2003.