Trend Filtering on Graphs

Yu-Xiang Wang[1]
yuxiangw@cs.cmu.edu

James Sharpnack[3]
jsharpna@gmail.com

Alex Smola[1,4]
alex@smola.org

Ryan J. Tibshirani[1,2]
ryantibs@stat.cmu.edu

[1] Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213
[2] Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213
[3] Mathematics Department, University of California at San Diego, La Jolla, CA 10280
[4] Marianas Labs, Pittsburgh, PA 15213

Abstract

We introduce a family of adaptive estimators on graphs, based on penalizing the ℓ_1 norm of discrete graph differences. This generalizes the idea of trend filtering [20, 40], used for univariate nonparametric regression, to graphs. Analogous to the univariate case, graph trend filtering exhibits a level of local adaptivity unmatched by the usual ℓ_2-based graph smoothers. It is also defined by a convex minimization problem that is readily solved (e.g., by fast ADMM or Newton algorithms). We demonstrate the merits of graph trend filtering through both examples and theory.

Keywords: *trend filtering, graph smoothing, total variation denoising, fused lasso, local adaptivity*

1 Introduction

Nonparametric regression has a rich history in statistics, carrying well over 50 years of associated literature. The goal of this paper is to port a successful idea in univariate nonparametric regression, trend filtering [36, 20, 40, 46], to the setting of estimation on graphs. The proposed estimator, graph trend filtering, shares three key properties of trend filtering in the univariate setting.

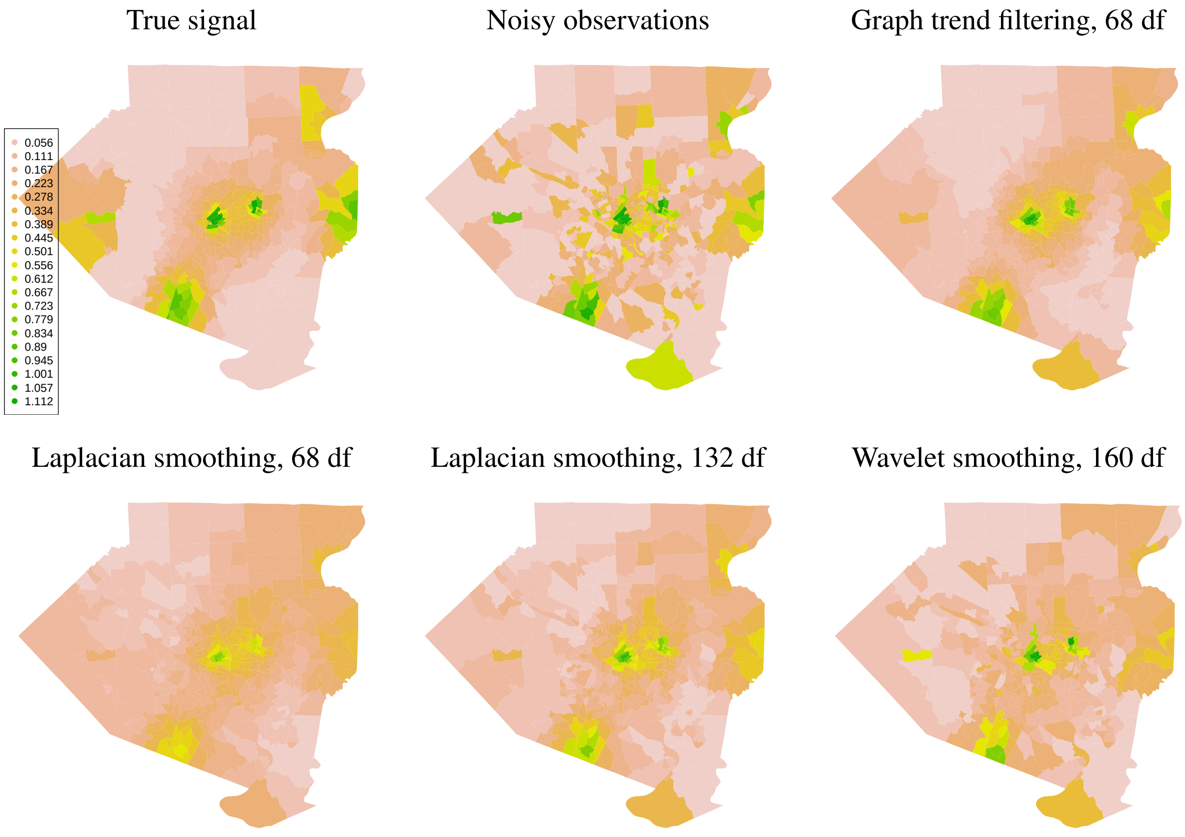
1. **Local adaptivity:** graph trend filtering can adapt to inhomogeneity in the level of smoothness of an observed signal across nodes. This stands in constrast to the usual ℓ_2-based methods, e.g., Laplacian regularization [34], which enforce smoothness globally with a much heavier hand, and tends to yield estimates that are either smooth or else wiggly throughout.
2. **Computational efficiency:** graph trend filtering is defined by a regularized least squares problem, in which the penalty term is nonsmooth, but convex and structured enough to permit efficient large-scale computation.
3. **Analysis regularization:** the graph trend filtering problem directly penalizes (possibly higher order) differences in the fitted signal across nodes. Therefore graph trend filtering falls into what is called the *analysis* framework for defining estimators. Alternatively, in the *synthesis* framework, we would first construct a suitable basis over the graph, and then regress the observed signal over this basis; e.g., [33] survey a number of such approaches using wavelets; likewise, kernel methods regularize in terms of the eigenfunctions of the graph Laplacian [21]. An advantage of analysis regularization is that it easily yields complex extensions of the basic estimator by mixing penalties.

As a motivating example, consider a denoising problem on 402 census tracts of Allegheny County, PA, arranged into a graph with 402 vertices and 2382 edges obtained by connecting spatially adjacent tracts. To illustrate the adaptive property of graph trend filtering we generated an artificial signal with inhomogeneous smoothness across the nodes, and two sharp peaks near the center of the graph, as can be seen in the top left panel of Figure 1. (The signal was formed using a mixture of five Gaussians, in the underlying spatial coordinates.) We drew noisy observations around this signal, shown in the top right panel of the figure, and we fit graph trend filtering, graph Laplacian smoothing, and wavelet smoothing to these observations. Graph trend filtering is to be defined in Section 2 (here we used $k = 2$, quadratic order); the latter two, recall, are defined by the optimization problems

$$\min_{\beta \in \mathbb{R}^n} \|y - \beta\|_2^2 + \lambda \beta^\top L \beta \quad \text{(Laplacian smoothing)},$$

$$\min_{\theta \in \mathbb{R}^n} \frac{1}{2}\|y - W\theta\|_2^2 + \lambda \|\theta\|_1 \quad \text{(wavelet smoothing)}.$$

where $y \in \mathbb{R}^n$ the vector of observations measured over the $n = 402$ nodes in the graph, $L \in \mathbb{R}^{n \times n}$ is the graph Laplacian matrix, and $W \in \mathbb{R}^{n \times n}$ is a wavelet basis built over the graph. The wavelet smoothing problem displayed above is really an oversimplified representation of the class of wavelets methods, since it only encapsulates estimators that employ an orthogonal wavelet basis W (and soft-threshold the wavelet coefficients). For the present experiment, we constructed W according to the spanning tree wavelet design of [31]; we found this construction performed best among the graph wavelet designs we considered for the data at hand. For completeness, the results from alternative wavelet designs are given in the Appendix.


True signal

Noisy observations

Graph trend filtering, 68 df

Laplacian smoothing, 68 df

Laplacian smoothing, 132 df

Wavelet smoothing, 160 df

Figure 1: Color maps for the Allegheny County example.

Graph trend filtering, Laplacian smoothing, and wavelet smoothing each have their own regularization parameters λ, and these parameters are not generally on the same scale. Therefore, in our comparisons we use effective degrees of freedom (df) as a common measure for the complexities of the fitted models. The top right panel of Figure 1 shows the graph trend filtering estimate with 68 df. We see that it adaptively fits the sharp peaks in the center of the graph, and smooths out the surrounding regions appropriately. The graph Laplacian estimate with 68 df (bottom left), substantially oversmooths the high peaks in the center, while at 132 df (bottom middle), it begins to detect the high peaks in the center, but undersmooths neighboring regions. Wavelet smoothing performs quite poorly across all df values—it appears to be most affected by the level of noise in the observations.

As a more quantitative assessment, Figure 2 shows the mean squared errors between the estimates and the true underlying signal. The differences in performance here are analogous to the univariate case, when comparing trend filtering to smoothing splines [40]. At smaller df values, Laplacian smoothing, due to its global considerations, fails to adapt to local differences across nodes. Trend filtering performs much better at low df values, and yet it matches Laplacian smoothing when both are sufficiently complex, i.e., in the overfitting regime. This demonstrates that the local flexibility of trend filtering estimates is a key attribute.

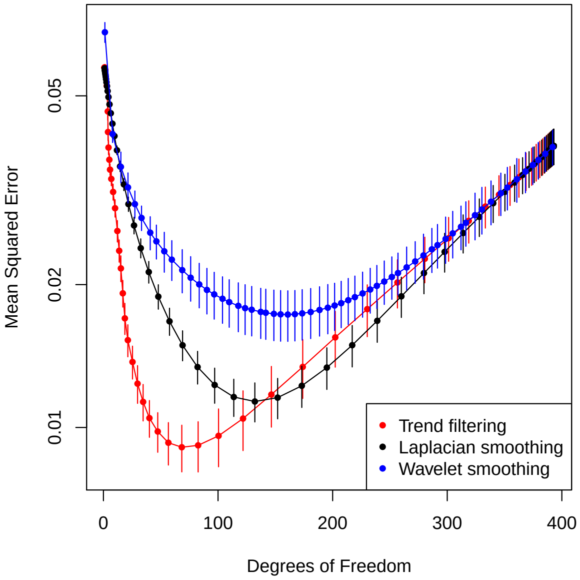


Figure 2: Mean squared errors for the Allegheny County example. Results were averaged over 10 simulations; the bars denote ±1 standard errors.

Here is an outline for the rest of this article. Section 2 defines graph trend filtering and gives underlying motivation and intuition. Section 3 covers basic properties and extensions of the graph trend filtering estimator. Section 4 examines computational approaches, and Section 5 looks at a number of both real and simulated data examples. Section 6 presents asymptotic error bounds for graph trend filtering. Section 7 concludes with a discussion. As for notation, we write X_A to extract the rows of a matrix $X \in \mathbb{R}^{m \times n}$ that correspond to a subset $A \subseteq \{1, \dots m\}$, and X_{-A} to extract the complementary rows. We use a similar convention for vectors: x_A and x_{-A} denote the components of a vector $x \in \mathbb{R}^m$ that correspond to the set A and its complement, respectively. We write $\mathrm{row}(X)$ and $\mathrm{null}(X)$ for the row and null spaces of X, respectively, and $X^\dagger$ for the pseudoinverse of X, with $X^\dagger = (X^\top X)^\dagger X^\top$ when X is rectangular.

2 Trend Filtering on Graphs

2.1 Review: Univariate Trend Filtering

We begin by reviewing trend filtering in the univariate setting, where discrete difference operators play a central role. Suppose that we observe $y = (y_1, \dots y_n) \in \mathbb{R}^n$ across input locations $x = (x_1, \dots x_n) \in \mathbb{R}^n$; for simplicity, suppose that the inputs are evenly spaced, say, $x = (1, \dots n)$. Given an integer $k \geq 0$, the kth order trend filtering estimate $\hat{\beta} = (\hat{\beta}_1, \dots \hat{\beta}_n)$ is defined as

$$\hat{\beta} = \operatorname*{argmin}_{\beta \in \mathbb{R}^n} \frac{1}{2}\|y - \beta\|_2^2 + \lambda \|D^{(k+1)}\beta\|_1, \tag{1}$$

where $\lambda \geq 0$ is a tuning parameter, and $D^{(k+1)}$ is the discrete difference operator of order $k+1$. When $k = 0$, problem (1) employs the first difference operator,

$$D^{(1)} = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \\ 0 & 0 & \dots & -1 & 1 \end{bmatrix}. \tag{2}$$

Therefore $\|D^{(1)}\beta\|_1 = \sum_{i=1}^{n-1} |\beta_{i+1} - \beta_i|$, and the 0th order trend filtering estimate in (1) reduces to the 1-dimensional fused lasso estimator [39], also called 1-dimensional total variation denoising [28]. For $k \geq 1$ the operator $D^{(k+1)}$ is defined recursively by

$$D^{(k+1)} = D^{(1)} D^{(k)}, \tag{3}$$

with $D^{(1)}$ above denoting the $(n-k-1) \times (n-k)$ version of the first difference operator in (2). In words, $D^{(k+1)}$ is given by taking first differences of kth differences. The interpretation is hence that problem (1) penalizes the changes in the kth discrete differences of the fitted trend. The estimated components $\hat{\beta}_1, \dots \hat{\beta}_n$ exhibit the form of a kth order piecewise polynomial function, evaluated over the input locations $x_1, \dots x_n$. This can be formally verified [40, 46] by examining a continuous-time analog of (1).

2.2 Trend Filtering over Graphs

Let $G = (V, E)$ be an graph, with vertices $V = \{1, \dots n\}$ and undirected edges $E = \{e_1, \dots e_m\}$, and suppose that we observe $y = (y_1, \dots y_n) \in \mathbb{R}^n$ over the nodes. Following the univariate definition in (1), we define the kth order *graph trend filtering* (GTF) estimate $\hat{\beta} = (\hat{\beta}_1, \dots \hat{\beta}_n)$ by

$$\hat{\beta} = \operatorname*{argmin}_{\beta \in \mathbb{R}^n} \frac{1}{2}\|y - \beta\|_2^2 + \lambda \|\Delta^{(k+1)}\beta\|_1. \tag{4}$$

In broad terms, this problem (like univariate trend filtering) is a type of generalized lasso problem [41], in which the penalty matrix $\Delta^{(k+1)}$ is a suitably defined *graph difference operator*, of order $k+1$. In fact, the novelty in our proposal lies entirely within the definition of this operator.

When $k = 0$, we define first order graph difference operator $\Delta^{(1)}$ in such a way it yields the graph-equivalent of a penalty on local differences:

$$\|\Delta^{(1)}\beta\|_1 = \sum_{(i,j) \in E} |\beta_i - \beta_j|.$$

so that the penalty term in (4) sums the absolute differences across connected nodes in G. To achieve this, we let $\Delta^{(1)} \in \{-1, 0, 1\}^{m \times n}$ be the oriented incidence matrix of the graph G, containing one row for each edge in the graph; specifically, if $e_\ell = (i, j)$, then $\Delta^{(1)}$ has ℓth row

$$\Delta^{(1)}_\ell = (0, \ldots \underset{\overset{\uparrow}{i}}{-1}, \ldots \underset{\overset{\uparrow}{j}}{1}, \ldots 0), \tag{5}$$

where the orientations of signs are arbitrary. Like trend filtering in the 1d setting, the 0th order graph trend filtering estimate coincides with the fused lasso (total variation regularized) estimate over G [17, 41, 30].

For $k \geq 1$, we use a recursion to define the higher order graph difference operators, in a manner similar to the univariate case. The recursion alternates in multiplying by the first difference operator $\Delta^{(1)}$ and its transpose (taking into account that this matrix not square):

$$\Delta^{(k+1)} = \begin{cases} (\Delta^{(1)})^\top \Delta^{(k)} = L^{\frac{k+1}{2}} & \text{for odd } k \\ \Delta^{(1)} \Delta^{(k)} = DL^{\frac{k}{2}} & \text{for even } k. \end{cases} \tag{6}$$

Above, we abbreviated the oriented incidence matrix $\Delta^{(1)}$ by D of G, and exploited the fact that $L = D^\top D$ is one representation for the graph Laplacian matrix. Note that $\Delta^{(k+1)} \in \mathbb{R}^{n \times n}$ for odd k, and $\Delta^{(k+1)} \in \mathbb{R}^{m \times n}$ for even k.

An important point is that our defined graph difference operators (5), (6) reduce to the univariate ones (2), (3) in the case of a chain graph (in which $V = \{1, \ldots n\}$ and $E = \{(i, i{+}1) : i = 1, \ldots n{-}1\}$), modulo boundary terms. That is, when k is even, if one removes the first $k/2$ rows and last $k/2$ rows of $\Delta^{(k+1)}$ for the chain graph, then one recovers $D^{(k+1)}$; when k is odd, if one removes the first and last $(k+1)/2$ rows of $\Delta^{(k+1)}$ for the chain graph, then one recovers $D^{(k+1)}$. Further intuition for our graph difference operators is given next.

2.3 Piecewise Polynomials over Graphs

We give some insight for our definition of graph difference operators (5), (6), based on the idea of piecewise polynomials over graphs. In the univariate case, as described in Section 2.1, sparsity of β under the difference operator $D^{(k+1)}$ implies a specific kth order piecewise polynomial structure for the components of β [40, 46]. Since the components of β correspond to (real-valued) input locations $x = (x_1, \ldots x_n)$, the interpretation of a piecewise polynomial here is unambiguous. But for a graph, one might ask: does sparsity of $\Delta^{(k+1)}\beta$ mean that the components of β are piecewise polynomial? And what does the latter even mean, as the components of β are defined over the nodes? To address these questions, we intuitively *define* a piecewise polynomial over a graph, and show that it implies sparsity under our constructed graph difference operators.

- **Piecewise constant ($k = 0$):** we say that a signal β is piecewise constant over a graph G if many of the differences $\beta_i - \beta_j$ are zero across edges $(i, j) \in E$ in G. Note that this is exactly the property associated with sparsity of $\Delta^{(1)}\beta$, since $\Delta^{(1)} = D$, the oriented incidence matrix of G.

- **Piecewise linear ($k = 1$):** we say that a signal β has a piecewise linear structure over G if β satisfies

$$\beta_i - \frac{1}{n_i} \sum_{(i,j) \in E} \beta_j = 0,$$

for many nodes $i \in V$, where n_i is the number of nodes adjacent to i. In words, we are requiring that the signal components can be linearly interpolated from its neighboring values at many nodes in the graph. This is quite a natural notion of (piecewise) linearity: requiring that β_i be equal to the average of its neighboring values would enforce linearity at β_i under an appropriate embedding of the points in Euclidean space. Again, this is precisely the same as requiring $\Delta^{(2)}\beta$ to be sparse, since $\Delta^{(2)} = L$, the graph Laplacian.

- **Piecewise polynomial ($k \geq 2$):** We say that β has a piecewise quadratic structure over G if the first differences $\alpha_i - \alpha_j$ of the second differences $\alpha = \Delta^{(2)}\beta$ are mostly zero, over edges $(i, j) \in E$. Likewise, β has a piecewise cubic structure over G if the second differences $\alpha_i - \frac{1}{n_i}\sum_{(i,j)\in E} \alpha_j$ of the second differences $\alpha = \Delta^{(2)}\beta$ are mostly zero, over nodes $i \in V$. This argument extends, alternating between leading first and second differences for even and odd k. Sparsity of $\Delta^{(k+1)}\beta$ in either case exactly corresponds to many of these differences being zero, by construction.

In Figure 3, we illustrate the graph trend filtering estimator on a 2d grid graph of dimension 20 × 20, i.e., a grid graph with 400 nodes and 740 edges. For each of the cases $k = 0, 1, 2$, we generated synthetic measurements over the grid, and computed a GTF estimate of the corresponding order. We chose the 2d grid setting so that the piecewise polynomial nature of GTF estimates could be visualized. Below each plot, the utilized graph trend filtering penalty is displayed in more explicit detail.

2.4 ℓ_1 versus ℓ_2 Regularization

It is instructive to compare the graph trend filtering estimator, as defined in (4), (5), (6) to Laplacian smoothing [34]. Standard Laplacian smoothing uses the same least squares loss as in (4), but replaces the penalty term with $\beta^\top L\beta$. A natural generalization would be to allow for a power of the Laplacian matrix L, and define kth order graph Laplacian smoothing according to

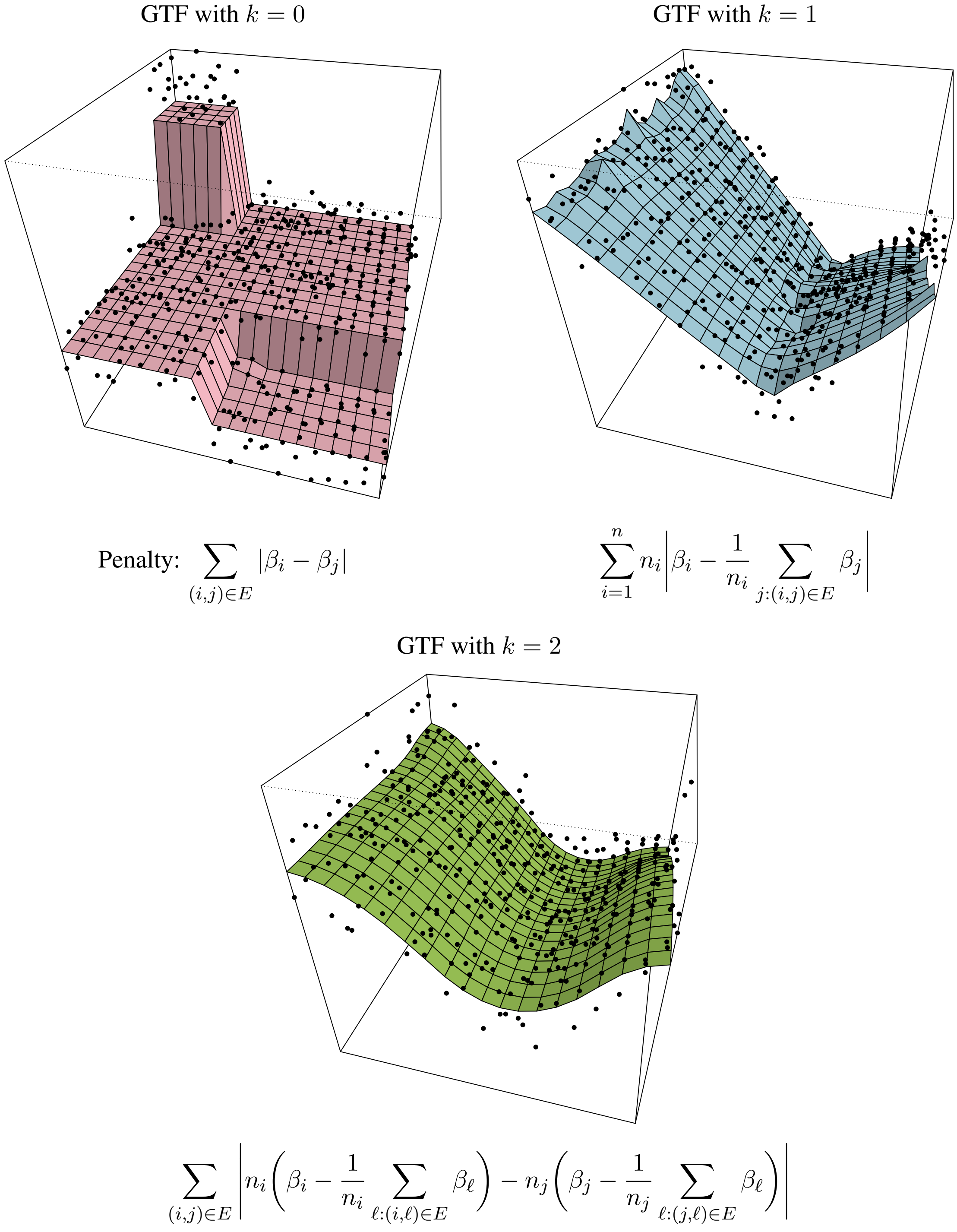
$$\hat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \; \|y - \beta\|_2^2 + \lambda \beta^\top L^{k+1} \beta. \tag{7}$$

The above penalty term can be written as $\|L^{(k+1)/2}\beta\|_2^2$ for odd k, and $\|DL^{k/2}\beta\|_2^2$ for even k; i.e., this penalty is exactly $\|\Delta^{(k+1)}\beta\|_2^2$ for the graph difference operator $\Delta^{(k+1)}$ defined previously.

As we can see, the critical difference between graph Laplacian smoothing (7) and graph trend filtering (4) lies in the choice of penalty norm: ℓ_2 in the former, and ℓ_1 in the latter. The effect of the ℓ_1 penalty is that the GTF program can set many (higher order) graph differences to zero exactly, and leave others at large nonzero values; i.e., the GTF estimate can simultaneously be smooth in some parts of the graph and wiggly in others. On the other hand, due to the (squared) ℓ_2 penalty, the graph Laplacian smoother cannot set any graph differences to zero exactly, and roughly speaking, must choose between making all graph differences small or large. The relevant analogy here is the comparison between the lasso and ridge regression, or univariate trend filtering and smoothing splines [40], and the suggestion is that GTF can adapt to the proper local degree of smoothness, while Laplacian smoothing cannot. This is strongly supported by the examples given throughout this paper.

2.5 Related Work

Some authors from the signal processing community, e.g., [5, 29], have studied total generalized variation (TGV), a higher order variant of total variation regularization. Moreover, several discrete versions of these operators have been proposed. They are often similar to the construction that we have. However, the focus of these works is mostly on how well a discrete functional approximates its continuous counterpart. This is quite different from our concern, as a signal on a graph (say a social network) may not have any meaningful continuous-space embedding at all. In addition, we are not aware of any study on the statistical properties of these regularizers. In fact, our theoretical analysis in Section 6 may be extended to these methods too.



GTF with $k = 0$

GTF with $k = 1$

Penalty: $\sum_{(i,j)\in E} |\beta_i - \beta_j|$

$$\sum_{i=1}^{n} n_i \bigg| \beta_i - \frac{1}{n_i} \sum_{j:(i,j)\in E} \beta_j \bigg|$$

GTF with $k = 2$

$$\sum_{(i,j)\in E} \bigg| n_i \Big(\beta_i - \frac{1}{n_i} \sum_{\ell:(i,\ell)\in E} \beta_\ell \Big) - n_j \Big(\beta_j - \frac{1}{n_j} \sum_{\ell:(j,\ell)\in E} \beta_\ell \Big) \bigg|$$

Figure 3: Graph trend filtering estimates of orders $k = 0, 1, 2$ on a 2d grid. The utilized ℓ_1 graph difference penalties are shown in elementwise detail below each plot (first, second, and third order graph differences).

3 Properties and Extensions

3.1 Basic Structure and Degrees of Freedom

We now describe the basic structure of graph trend filtering estimates and present an unbiased estimate for their degrees of freedom. Let the tuning parameter λ be arbitrary but fixed. By virtue of the ℓ_1 penalty in (4), the solution $\hat{\beta}$ satisfies $\mathrm{supp}(\Delta^{(k+1)}\hat{\beta}) = A$ for some active set A (typically A is smaller when λ is larger). Trivially, we can reexpress this as $\Delta^{(k+1)}_{-A}\hat{\beta} = 0$, or $\hat{\beta} \in \mathrm{null}(\Delta^{(k+1)}_{-A})$. Therefore, the basic structure of GTF estimates is revealed by analyzing the null space of the suboperator $\Delta^{(k+1)}_{-A}$.

Lemma 1. *Assume without a loss of generality that G is connected (otherwise the results apply to each connected component of G). Let D, L be the oriented incidence matrix and Laplacian matrix of G. For even k, let $A \subseteq \{1, \ldots m\}$, and let G_{-A} denote the subgraph induced by removing the edges indexed by A (i.e., removing edges e_ℓ, $\ell \in A$). Let $C_1, \ldots C_s$ be the connected components of G_{-A}. Then*

$$\mathrm{null}(\Delta^{(k+1)}_{-A}) = \mathrm{span}\{\mathbb{1}\} + (L^\dagger)^{\frac{k}{2}}\mathrm{span}\{\mathbb{1}_{C_1}, \ldots \mathbb{1}_{C_s}\},$$

where $\mathbb{1} = (1, \ldots 1) \in \mathbb{R}^n$, and $\mathbb{1}_{C_1}, \ldots \mathbb{1}_{C_s} \in \mathbb{R}^n$ are the indicator vectors over connected components. For odd k, let $A \subseteq \{1, \ldots n\}$. Then

$$\mathrm{null}(\Delta^{(k+1)}_{-A}) = \mathrm{span}\{\mathbb{1}\} + \{(L^\dagger)^{\frac{k+1}{2}}v : v_{-A} = 0\}.$$

The proof of Lemma 1 appears in the Appendix. The lemma is useful for a few reasons. First, as motivated above, it describes the coarse structure of GTF solutions. When $k = 0$, we can see (as $(L^\dagger)^{0/2} = I$) that $\hat{\beta}$ will indeed be piecewise constant over groups of nodes $C_1, \ldots C_s$ of G. For $k = 2, 4, \ldots$, this structure is smoothed by multiplying such piecewise constant levels by $(L^\dagger)^{k/2}$. Meanwhile, for $k = 1, 3 \ldots$, the structure of the GTF estimate is based on assigning nonzero values to a subset A of nodes, and then smoothing through multiplication by $(L^\dagger)^{(k+1)/2}$. Both of these smoothing operations, which depend on $L^\dagger$, have interesting interpretations in terms of to the electrical network perspective for graphs. This is developed in the next subsection.

A second use of Lemma 1 is that it leads to a simple expression for the degrees of freedom, i.e., the effective number of parameters, of the GTF estimate $\hat{\beta}$. From results on generalized lasso problems [41, 42], we have $\mathrm{df}(\hat{\beta}) = \mathbb{E}[\mathrm{nullity}(\Delta^{(k+1)}_{-A})]$, with A denoting the support of $\Delta^{(k+1)}\hat{\beta}$, and $\mathrm{nullity}(X)$ the dimension of the null space of a matrix X. Applying Lemma 1 then gives the following.

Lemma 2. *Assume that G is connected. Let $\hat{\beta}$ denote the GTF estimate at a fixed but arbitrary value of λ. Under the normal error model $y \sim \mathcal{N}(\beta_0, \sigma^2 I)$, the GTF estimate $\hat{\beta}$ has degrees of freedom given by*

$$\mathrm{df}(\hat{\beta}) = \begin{cases} \mathbb{E}\left[\max\left\{|A|, 1\right\}\right] & \textit{odd } k, \\ \mathbb{E}\left[\textit{number of connected components of } G_{-A}\right] & \textit{even } k. \end{cases}$$

Here $A = \mathrm{supp}(\Delta^{(k+1)}\hat{\beta})$ denotes the active set of $\hat{\beta}$.

As a result of Lemma 2, we can form simple unbiased estimate for $\mathrm{df}(\hat{\beta})$; for k odd, this is $\max\{|A|, 1\}$, and for k even, this is the number of connected components of G_{-A}, where A is the support of $\Delta^{(k+1)}\hat{\beta}$. When reporting degrees of freedom for graph trend filtering (as in the example in the introduction), we use these unbiased estimates.

3.2 Electrical Network Interpretation

Lemma 1 reveals a mathematical structure for GTF estimates $\hat{\beta}$, which satisfy $\hat{\beta} \in \mathrm{null}(\Delta^{(k+1)}_{-A})$ for some set A. It is interesting to interpret the results using the electrical network perspective for graphs [45]. In this perspective, we imagine replacing each edge in the graph with a resistor of value 1. If $u \in \mathbb{R}^n$ describes how much current is going in at each node in the graph, then $v = Lu$ describes the induced voltage at each node. Provided that $\mathbb{1}^\top c = 0$, which means that the total accumulation of current in the network is 0, we can solve for the current values from the voltage values: $u = L^\dagger v$.

The odd case in Lemma 1 asserts that

$$\mathrm{null}(\Delta^{(k+1)}_{-A}) = \mathrm{span}\{\mathbb{1}\} + \{(L^\dagger)^{\frac{k+1}{2}} v : v_{-A} = 0\}.$$

For $k = 1$, this says that GTF estimates are formed by assigning a sparse number of nodes in the graph a nonzero voltage v, then solving for the induced current $L^\dagger v$ (and shifting this entire current vector by a constant amount). For $k = 3$, we assign a sparse number of nodes a nonzero voltage, solve for the induced current, and then *repeat this*: we relabel the induced current as input voltages to the nodes, and compute the new induced current. This process is again iterated for $k = 5, 7, \ldots$.

The even case in Lemma 1 asserts that

$$\mathrm{null}(\Delta^{(k+1)}_{-A}) = \mathrm{span}\{\mathbb{1}\} + (L^\dagger)^{\frac{k}{2}} \mathrm{span}\{\mathbb{1}_{C_1}, \ldots \mathbb{1}_{C_s}\}.$$

For $k = 2$, this result says that GTF estimates are given by choosing a partition $C_1, \ldots C_s$ of the nodes, and assigning a constant input voltage to each element of the partition. We then solve for the induced current (and potentially shift this by an overall constant amount). The process is iterated for $k = 4, 6, \ldots$ by relabeling the induced current as input voltage.

The comparison between the structure of estimates for $k = 2$ and $k = 3$ is informative: in a sense, the above tells us that 2nd order GTF estimates will be *smoother* than 3rd order estimates, as a sparse input voltage vector need not induce a current that is piecewise constant over nodes in the graph. For example, an input voltage vector that has only a few nodes with very large nonzero values will induce a current that is peaked around these nodes, but not piecewise constant.

3.3 Extensions

Several extensions of the proposed graph trend filtering model are possible. Trend filtering over a weighted graph, for example, could be performed by using a properly weighted version of the edge incidence matrix in (5), and carrying forward the same recursion in (6) for the higher order difference operators. As another example, the Gaussian regression loss in (4) could be changed to another suitable likelihood-derived losses in order to accommodate a different data type for y, say, logistic regression loss for binary data, or Poisson regression loss for count data.

In Section 5.2, we explore a modest extension of GTF, where we add a strongly convex prior term to the criterion in (4) to assist in performing graph-based imputation from partially observed data over the nodes. In Section 5.3, we investigate a modification of the proposed regularization scheme, where we add a pure ℓ_1 penalty on β in (4), hence forming a sparse variant of GTF. Other potentially interesting penalty extensions include: mixing graph difference penalties of various orders, and tying together several denoising tasks with a group penalty. Extensions such as these are easily built, recall, as a result of the analysis framework used by the GTF program, wherein the estimate defined through direct regularization via an analyzing operator, the ℓ_1-based graph difference penalty $\|\Delta^{(k+1)}\beta\|_1$.

4 Computation

Graph trend filtering is defined by a convex optimization problem (4). In principle this means that, at least for small or moderately sized problems, its solutions can be reliably computed using a variety of standard algorithms. In order to handle larger scale problems, we describe two specialized algorithms that improve on generic procedures by taking advantage of the structure of $\Delta^{(k+1)}$.

4.1 A Fast ADMM Algorithm

We reparametrize (4) by introducing auxiliary variables, so that we can apply ADMM. For even k, we use a special transformation that is critical for fast computation (following [27] in univariate trend filtering); for odd k, this is not possible. The reparametrizations for even and odd k are

$$\min_{\beta, z \in \mathbb{R}^n} \frac{1}{2}\|y-\beta\|_2^2 + \lambda\|Dz\|_1 \;\text{ s.t. }\; z = L^{\frac{k}{2}}x,$$
$$\min_{\beta, z \in \mathbb{R}^n} \frac{1}{2}\|y-\beta\|_2^2 + \lambda\|z\|_1 \;\text{ s.t. }\; z = L^{\frac{k+1}{2}}x,$$

respectively. Recall that D is the oriented incidence matrix and L is the graph Laplacian. The augmented Lagrangian is

$$\frac{1}{2}\|y-\beta\|_2^2 + \lambda\|Sz\|_1 + \frac{\rho}{2}\|z - L^q\beta + u\|_2^2 - \frac{\rho}{2}\|u\|_2^2,$$

where $S = D$ or $S = I$ depending on whether k is even or odd, and likewise $q = k/2$ or $q = (k+1)/2$. ADMM then proceeds by iteratively minimizing the augmented Lagrangian over β, minimizing over z, and performing a dual update over u. The β and z updates are of the form, for some b,

$$\beta \leftarrow (I + \rho L^{2q})^{-1} b, \tag{8}$$
$$z \leftarrow \operatorname*{argmin}_{x \in \mathbb{R}^n} \frac{1}{2}\|b - x\|_2^2 + \frac{\lambda}{\rho}\|Sx\|_1, \tag{9}$$

The linear system in (8) is well-conditioned, sparse, and can be solved efficiently using the preconditioned conjugate gradient method. This involves only multiplication with Laplacian matrices. For a small enough choices of $\rho > 0$ (the augmented Lagrangian parameter), the system in (8) is diagonally dominant, special Laplacian/SDD solvers can be applied, which run in almost linear time [35, 22, 19].

For $S = I$, the update in (9) is simply given by soft-thresholding, and for $S = D$, it is given by graph TV denoising, i.e., given by solving a graph fused lasso problem. Note that this subproblem has the exact structure of the graph trend filtering problem (4) with $k = 0$. A direct approach for graph TV denoising is available based on parametric max-flow [8], and this algorithm is empirically much faster than its worst-case complexity [4]. In the special case that the underlying graph is a grid, a promising alternative method employs proximal stacking techniques [2].

4.2 A Fast Newton Method

As an alternative to ADMM, a projected Newton-type method [3, 1] can be used to solve (4) via its dual problem:

$$\hat{v} = \operatorname*{argmin}_{v \in \mathbb{R}^r} \|y - (\Delta^{(k+1)})^\top v\|_2^2 \;\text{ s.t. }\; \|v\|_\infty \le \lambda.$$

The solution of (4) is then given by $\hat{\beta} = y - (\Delta^{(k+1)})^\top \hat{v}$. (For univariate trend filtering, [20] adopt a similar strategy, but instead use an interior point method.) The projected Newton method performs updates using a reduced Hessian, so abbreviating $\Delta = \Delta^{(k+1)}$, each iteration boils down to

$$v \leftarrow a + (\Delta_I^\top)^\dagger b, \tag{10}$$

for some a, b and set of indices I. The linear system in (10) is always sparse, but conditioning becomes an issue as k grows (note that the same problem does not occur in (8) because of the addition of the identity matrix I). We have found empirically that a preconditioned conjugate gradient method works quite well for (10) for $k = 1$, but struggles for larger k.

4.3 Computation Summary

In our experience, the following algorithms work well for the various order k of graph trend filtering. We remark that orders $k = 0, 1, 2$ are of most practical interest (and solutions of polynomial order $k \geq 3$ are less likely to be sought in practice).[1]

Order	Algorithm
$k = 0$	Parametric max-flow
$k = 1$	Projected Newton method
$k = 2, 4, \ldots$	ADMM with parametric max-flow
$k = 3, 5, \ldots$	ADMM with soft-thresholding

Figure 4 compares performances of the described algorithms on a moderately large simulated example, using a 2d grid graph. We see that when $k = 1$, the projected Newton method converges faster than ADMM (superlinear versus at best linear convergence). When $k = 2$, the story is reversed, as the projected Newton iterations quickly become stagnant, and the ADMM enjoys better convergence.

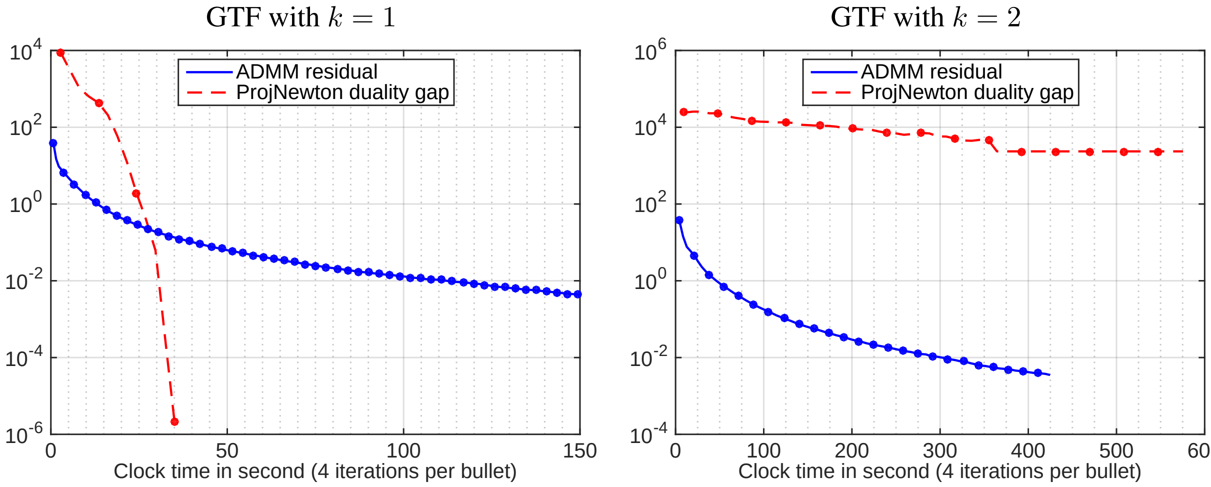


Figure 4: Convergence plots for projected Newton method and ADMM for solving GTF with $k = 1$ and $k = 2$. The algorithms are all run on a 2d grid graph (an 512×512 image) with 262,144 nodes and 523,264 edges. For projected Newton, we plot the duality gap across iterations; for ADMM, we plot the average of the primal and dual residuals (which also serves as a valid suboptimality bound in the ADMM framework).

[1] Loosely speaking, each order $k = 0, 1, 2$ provides solutions that exhibit a different class of structure: $k = 0$ gives piecewise constant solutions, $k = 1$ gives piecewise linear, and $k = 2$ gives piecewise smooth. All orders $k \geq 3$ continue to give piecewise smooth fits, with less and less transparent differences (the practical differences between piecewise quadratic versus piecewise linear fits is greater than piecewise cubic versus piecewise quadratic, etc.). Since the conditioning of the graph trend filtering operator $\Delta^{(k+1)}$ worsens as k increases, which makes computation more difficult, it makes most practical sense to simply choose $k = 2$ whenever a piecewise smooth fit is desired.

5 Examples

5.1 Trend Filtering over the Facebook Graph

In the Introduction, we examined the denoising power of graph trend filtering in a spatial setting. Here we examine the behavior of graph trend filtering on a nonplanar graph: the Facebook graph from the Stanford Network Analysis Project (http://snap.stanford.edu). This is composed of 4039 nodes representing Facebook users, and 88,234 edges representing friendships, collected from real survey participants; the graph has one connected component, but the observed degree sequence is very mixed, ranging from 1 to 1045 (see [26] for more details).

We generated synthetic measurements over the Facebook nodes (users) based on three different ground truth models, so that we can precisely evaluate and compare the estimation accuracy of GTF, Laplacian smoothing, and wavelet smoothing. For the latter, we again used the spanning tree wavelet design of [31], because it performed among the best of wavelets designs in all data settings considered here. Results from other wavelet designs are presented in the Appendix. The three ground truth models represent very different scenarios for the underlying signal x, each one favorable to different estimation methods. These are:

1. **Dense Poisson equation:** we solved the Poisson equation $Lx = b$ for x, where b is arbitrary and dense (its entries were i.i.d. normal draws).
2. **Sparse Poisson equation:** we solved the Poisson equation $Lx = b$ for x, where b is sparse and has 30 nonzero entries (again i.i.d. normal draws).
3. **Inhomogeneous random walk:** we ran a set of decaying random walks at different starter nodes in the graph, and recorded in x the total number of visits at each node. Specifically, we chose 10 nodes as starter nodes, and assigned each starter node a decay probability uniformly at random between 0 and 1 (this is the probability that the walk terminates at each step instead of travelling to a neighboring node). At each starter node, we also sent out a varying number of random walks, chosen uniformly between 0 and 1000.

In each case, the synthetic measurements were formed by adding noise to x. We note that model 1 is designed to be favorable for Laplace smoothing; model 2 is designed to be favorable for GTF; and in the inhomogeneity in model 3 is designed to be challenging for Laplacian smoothing, and favorable for the more adaptive GTF and wavelet methods.

Figure 5 shows the performance of the three estimation methods, over a wide range of noise levels in the synthetic measurements; performance here is measured by the best achieved mean squared error, allowing each method to be tuned optimally at each noise level. The summary: GTF estimates are (expectedly) superior when the Laplacian-based sparsity pattern is in effect (model 2), but are nonetheless highly competitive in both other settings—the dense case, in which Laplacian smoothing thrives, and the inhomogeneous random walk case, in which wavelets thrive.

5.2 Graph-Based Transductive Learning over UCI Data

Graph trend filtering can used for graph-based transductive learning, as motivated by the work of [37, 38], who rely on Laplacian regularization. Consider a semi-supervised learning setting, where we are given only a small number of seed labels over nodes of a graph, and the goal is to impute the labels on the remaining nodes. Write $O \subseteq \{1, \dots n\}$ for the set of observed nodes, and assume that each observed label falls into $\{1, \dots K\}$. Then we can define the modified absorption problem under graph trend filtering regularization

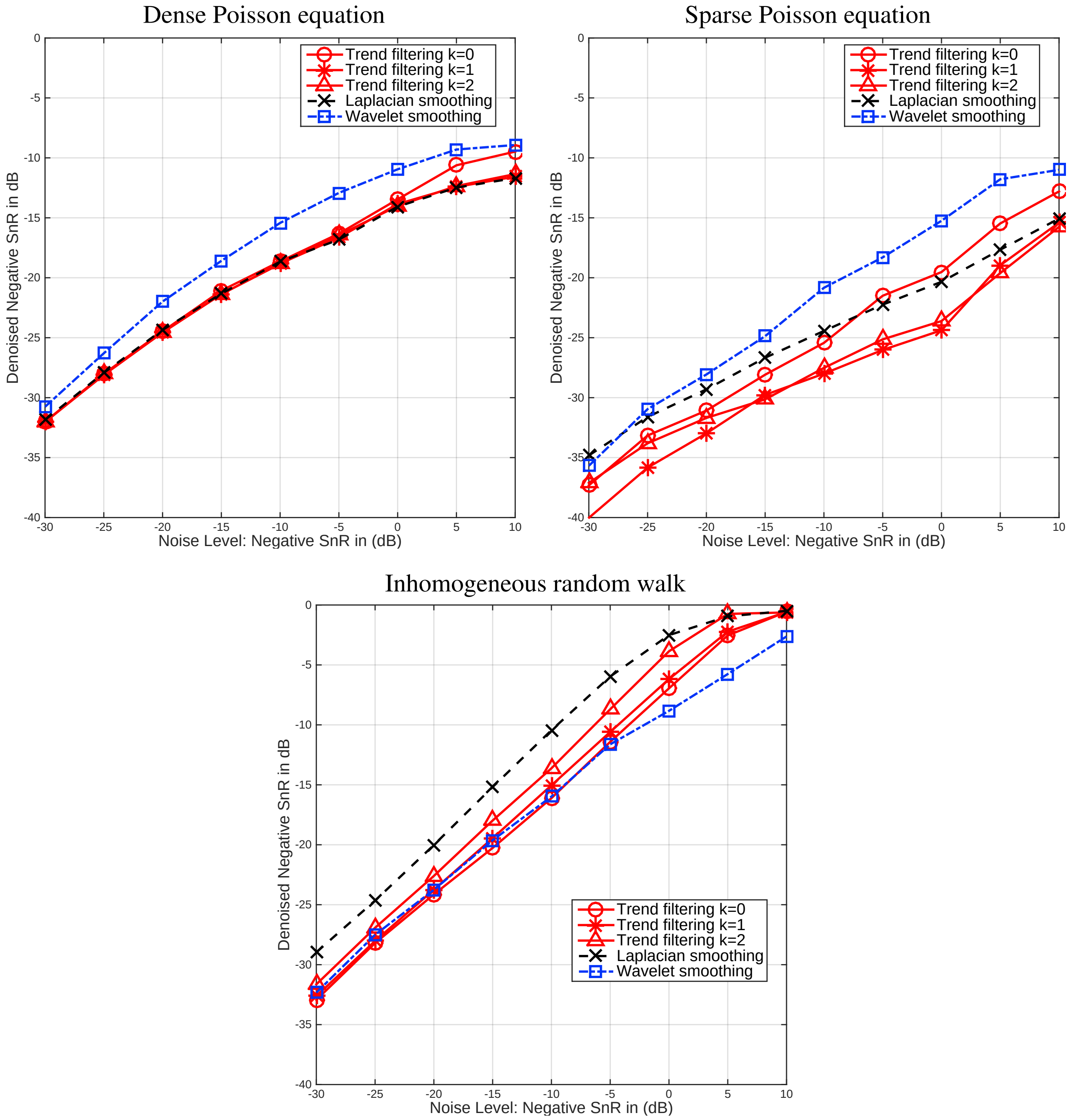


Figure 5: Performance of GTF and others for three generative models on the Facebook graph. The x-axis shows the negative SnR: $10\log_{10}(n\sigma^2/\|x\|_2^2)$, where $n = 4039$, x is the underlying signal, and σ^2 is the noise variance. Hence the noise level is increasing from left to right. The y-axis shows the denoised negative SnR: $10\log_{10}(\mathrm{MSE}/\|x\|_2^2)$, where MSE denotes mean squared error, so the achieved MSE is increasing from bottom to top.

(MAD-GTF) by

$$\hat{B} = \underset{B \in \mathbb{R}^{n\times K}}{\mathrm{argmin}} \; \sum_{j=1}^K \sum_{i \in O} (Y_{ij} - B_{ij})^2 + \lambda \sum_{j=1}^K \|\Delta^{(k+1)} B_j\|_1 + \epsilon \sum_{j=1}^K \|R_j - B_j\|_2^2. \tag{11}$$

The matrix $Y \in \mathbb{R}^{n\times K}$ is an indicator matrix: each observed row $i \in O$ is described by $Y_{ij} = 1$ if class j is observed and $Y_{ij} = 0$ otherwise. The matrix $B \in \mathbb{R}^{n\times K}$ contains fitted probabilities, with B_{ij} giving the probability that node i is of class j. We write B_j for its jth column, and hence the middle term in the above criterion encourages each set of class probabilities to behave smoothly over the graph. The last term in the above criterion ties the fitted probabilities to some given prior weights $R \in \mathbb{R}^{n\times K}$. In principle ϵ could act as a second tuning parameter, but for simplicity we take ϵ to be small and fixed, with any $\epsilon > 0$ guaranteeing that the criterion in (11) is strictly convex, and thus has a unique solution $\hat{B}$. The entries of $\hat{B}$ need not be probabilites in any strict sense, but we can still interpret them as relative probabilities, and imputation can be performed by assigning each unobserved node $i \notin O$ a class label j such that $\hat{B}_{ij}$ is largest.

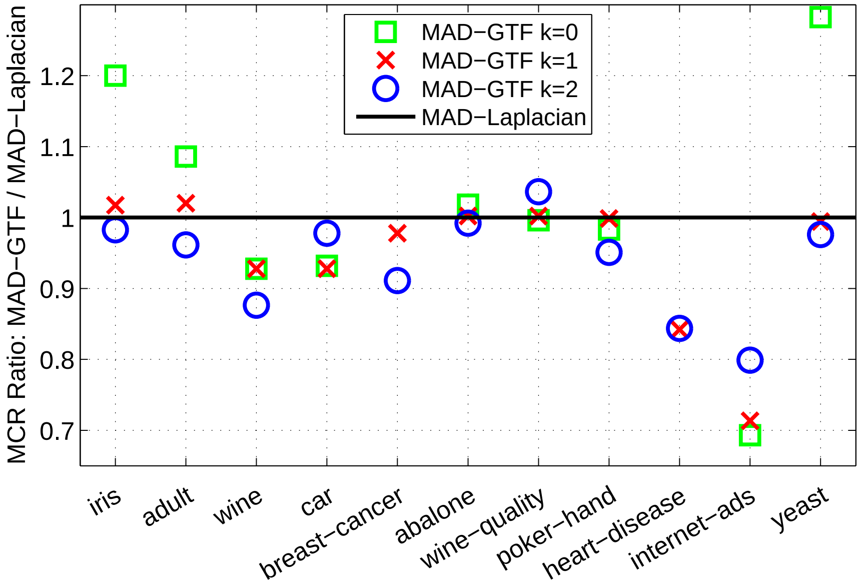


Figure 6: Ratio of the misclassification rate of MAD-GTF to MAD-Laplacian, for graph-based imputation, on the 11 most popular UCI classification data sets.

Our specification of MAD-GTF only deviates from the MAD proposal of [37] in that these authors used the Laplacian regularization term $\sum_{j=1}^K B_j^\top L B_j$, in place of ℓ_1-based graph difference regularizer in (11). If the underlying class probabilities are thought to have heterogeneous smoothness over the graph, then replacing the Laplacian regularizer with the GTF-designed one might lead to better performance. As a broad comparison of the two methods, we ran them on the 11 most popular classification data sets from the UCI Machine Learning repository (http://archive.ics.uci.edu/ml/).[2] For each data set, we constructed a 5-nearest-neighbor graph based on the Euclidean distance between provided features, and randomly selected 5 seeds per class to serve as the observed class labels. Then we set $\epsilon = 0.01$, used prior weights $R_{ij} = 1/K$ for all i and j, and chose the tuning parameter λ over a wide grid of values to represent the best achievable performance by each method, on each experiment. Figure 6 and Table 1 summarize the

[2] We used all data sets here, except the “forest-fires” data set, which is a regression problem. Also, we zero-filled the missing data in “internet-ads” data set and used a random one third of the data in the “poker” data set.

	iris	adult	wine	car	breast	abalone	wine-qual.	poker	heart	ads	yeast
# of classes	3	2	3	4	2	29	6	10	2	2	10
# of samples	150	32,561	178	1,728	569	4,177	1,599	3,000	303	3,279	1,484
Laplacian	0.085	0.270	0.060	0.316	0.064	0.872	0.712	0.814	0.208	0.306	0.566
GTF, $k = 0$	0.102	0.293	0.055	0.294	0.500	0.888	0.709	0.801	0.472	0.212	0.726
p-value	0.254	0.648	0.406	0.091	0.000	0.090	0.953	0.732	0.000	0.006	0.000
GTF, $k = 1$	0.087	0.275	0.055	0.293	0.063	0.874	0.713	0.813	0.175	0.218	0.563
p-value	0.443	0.413	0.025	0.012	0.498	0.699	0.920	0.801	0.134	0.054	0.636
GTF, $k = 2$	0.084	0.259	0.052	0.309	0.059	0.865	0.738	0.774	0.175	0.244	0.552
p-value	0.798	0.482	0.024	0.523	0.073	0.144	0.479	0.138	0.301	0.212	0.100

Table 1: Misclassification rates of MAD-Laplacian and MAD-GTF for imputation in the UCI data sets. We also compute p-values over the 10 repetitions for each data set (10 draws of nodes to serve as seed labels) via paired t-tests. Cases where MAD-GTF achieves significantly better misclassification rate, at the 0.1 level, are highlighted in green; cases where MAD-GTF achieves a significantly worse miclassification rate, at the 0.1 level, are highlighted in red.

misclassification rates from imputation using MAD-Laplacian and MAD-GTF, averaged over 10 repetitions of the randomly selected seed labels. We see that MAD-GTF with $k = 0$ (basically a graph partition akin to MRF-based graph cut, using an Ising model) does not seem to work as well as the smoother alternatives. Importantly, MAD-GTF with $k = 1$ and $k = 2$ both perform at least as well, and sometimes better, than MAD-Laplacian on each one of the UCI data sets. Recall that these data sets were selected entirely based on their popularity, and not at all on the belief that they might represent favorable scenarios for GTF (i.e., not on the prospect that they might exhibit some heterogeneity in the distribution of class labels over their respective graphs). Therefore, the fact that MAD-GTF nonetheless performs competitively in such a broad range of experiments is convincing evidence for the utility of the GTF regularizer.

5.3 Event Detection with NYC Taxi Trips Data

We illustrate a sparse variant of our proposed regularizers, given by adding a pure ℓ_1 penalty to the coefficients in (4), as in

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \ \frac{1}{2}\|y - \beta\|_2^2 + \lambda_1 \|\Delta^{(k+1)}\beta\|_1 + \lambda_2\|\beta\|_1. \tag{12}$$

We call this *sparse graph trend filtering*, now with two tuning parameters $\lambda_1, \lambda_2 \geq 0$. Under the proper tuning, the sparse GTF estimate will be zero at many nodes in the graph, and will otherwise deviate smoothly from zero. This can be useful in situations where the observed signal represents a difference between two smooth processes that are mostly similar, but exhibit (perhaps significant) differences over a few regions of the graph. Here we apply it to the problem of detecting events based on abnormalities in the number of taxi trips at different locations of New York city. This data set was kindly provided by authors of [13], who obtained the data from NYC Taxi & Limosine Commission.[3] Specifically, we consider the graph to be the road network of Manhattan, which contains 3874 nodes (junctions) and 7070 edges (sections of roads that connect two junctions). For measurements over the nodes, we used the number of taxi pickups and dropoffs over a particular time period of interest: 12:00–2:00 pm on June 26, 2011, corresponding to the Gay Pride parade. As pickups and dropoffs do not generically occur at road junctions, we used interpolation to form counts over the graph nodes. A baseline seasonal average was calculated by considering data from the same time block 12:00–2:00 pm on the same day of the week across the nearest eight weeks. Thus the

[3] These authors also considered event detection, but their topological definition of an "event" is very different from what we considered here, and hence our results not directly comparable.

measurements y were then taken to be the difference between the counts observed during the Gay Pride parade and the seasonal averages.

Note that the nonzero node estimates from sparse GTF applied to y, after proper tuning, mark events of interest, because they convey substantial differences between the observed and expected taxi counts. According to descriptions in the news, we know that the Gay Pride parade was a giant march down at noon from 36th St. and Fifth Ave. all the way to Christopher St. in Greenwich Village, and traffic was blocked over the entire route for two hours (meaning no pickups and dropoffs could occur). We therefore hand-labeled this route as a crude "ground truth" for the event of interest, illustrated in the left panel of Figure 7.

In the bottom two panels of Figure 7, we compare sparse GTF with $k = 0$ (i.e., the sparse graph fused lasso) and a sparse variant of Laplacian smoothing, obtained by replacing the first regularization term in (12) by $\beta^\top L\beta$. For a qualitative visual comparison, the smoothing parameter λ_1 was chosen so that both methods have 200 degrees of freedom (without any sparsity imposed). The sparsity parameter was then set as $\lambda_2 = 0.2$. Similar to what we have seen already, GTF is able to better localize its estimates around strong inhomogenous spikes in the measurements, and is able to better capture the event of interest. The result of sparse Laplacian smoothing is far from localized around the ground truth event, and displays many nonzero node estimates throughout distant regions of the graph. If we were to decrease its flexibility (increase the smoothing parameter λ_1 in its problem formulation), then the sparse Laplacian output would display more smoothness over the graph, but the node estimates around the ground truth region would also be grossly shrunken.

6 Estimation Error Bounds

In this section, we assume that $y \sim \mathcal{N}(\beta_0, \sigma^2 I)$, and study asymptotic error rates for graph trend filtering. (The assumption of a normal error model could be relaxed, but is used for simplicity). Our analysis actually focuses more broadly on the generalized lasso problem

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \; \frac{1}{2}\|y - \beta\|_2^2 + \lambda\|\Delta\beta\|_1, \tag{13}$$

where $\Delta \in \mathbb{R}^{r\times n}$ is an arbitrary linear operator, and r denotes its number of rows. Throughout, we specialize the derived results to the graph difference operator $\Delta = \Delta^{(k+1)}$, to obtain concrete statements about GTF over particular graphs. All proofs are deferred to the Appendix.

6.1 Basic Error Bounds

Using similar arguments to the basic inequality for the lasso [6], we have the following preliminary bound.

Theorem 3. *Let M denote the maximum ℓ_2 norm of the columns of $\Delta^\dagger$. Then for a tuning parameter value $\lambda = \Theta(M\sqrt{\log r})$, the generalized lasso estimate $\hat{\beta}$ in (13) has average squared error*

$$\frac{\|\hat{\beta} - \beta_0\|_2^2}{n} = O_{\mathbb{P}}\left(\frac{\operatorname{nullity}(\Delta)}{n} + \frac{M\sqrt{\log r}}{n} \cdot \|\Delta\beta_0\|_1\right).$$

Recall that $\operatorname{nullity}(\Delta)$ denotes the dimension of the null space of Δ. For the GTF operator $\Delta^{(k+1)}$ of any order k, note that $\operatorname{nullity}(\Delta^{(k+1)})$ is the number of connected components in the underlying graph.

When both $\|\Delta\beta_0\|_1 = O(1)$ and $\operatorname{nullity}(\Delta) = O(1)$, Theorem 3 says that the estimate $\hat{\beta}$ converges in average squared error at the rate $M\sqrt{\log r}/n$, in probability. This theorem is quite general, as it applies to any linear operator Δ, and one might therefore think that it cannot yield fast rates. Still, as we show next, it does imply consistency for graph trend filtering in certain cases.

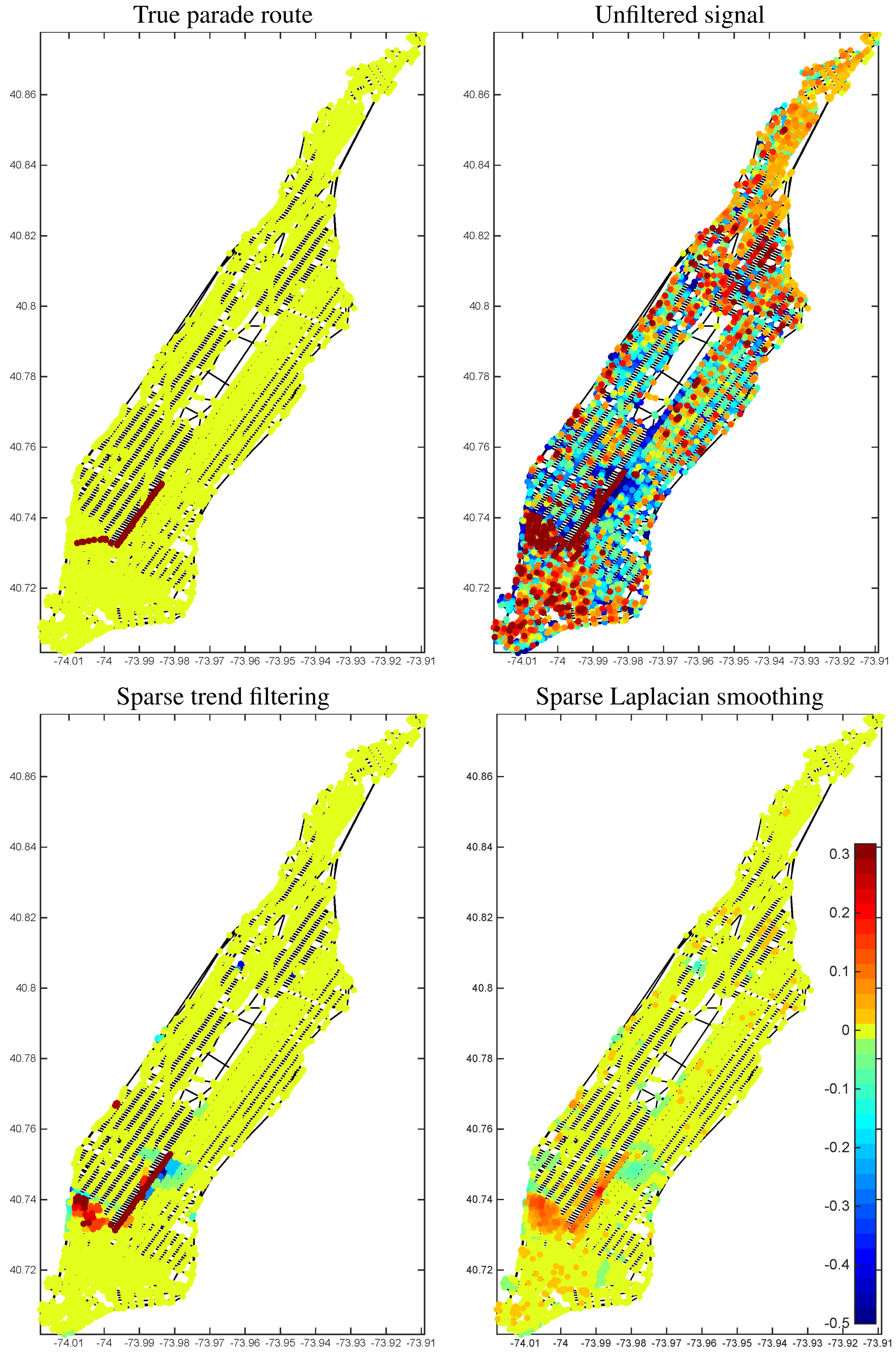


Figure 7: Comparison of sparse GTF and sparse Laplacian smoothing. We can see qualitatively that sparse GTF delivers better event detection with fewer false positives (zoomed-in, the sparse Laplacian plot shows a scattering of many non-zero colors).

Corollary 4. *Consider the trend filtering estimator $\hat{\beta}$ of order k, and the choice of the tuning parameter λ as in Theorem 3. Then:*

1. *for univariate trend filtering (i.e., essentially GTF on a chain graph),*

$$\frac{\|\hat{\beta}-\beta_0\|_2^2}{n} = O_{\mathbb{P}}\left(\sqrt{\frac{\log n}{n}} \cdot n^k \|D^{(k+1)}\beta_0\|_1\right);$$

2. *for GTF on an Erdos-Renyi random graph, with edge probability p, and expected degree $d = np \geq 1$,*

$$\frac{\|\hat{\beta}-\beta_0\|_2^2}{n} = O_{\mathbb{P}}\left(\frac{\sqrt{\log(nd)}}{nd^{\frac{k+1}{2}}} \cdot \|\Delta^{(k+1)}\beta_0\|_1\right);$$

3. *for GTF on a Ramanujan d-regular graph, and $d \geq 1$,*

$$\frac{\|\hat{\beta}-\beta_0\|_2^2}{n} = O_{\mathbb{P}}\left(\frac{\sqrt{\log(nd)}}{nd^{\frac{k+1}{2}}} \cdot \|\Delta^{(k+1)}\beta_0\|_1\right).$$

Cases 2 and 3 of Corollary 4 stem from the simple inequality $M \leq \|\Delta^\dagger\|_2$, the largest singular value of $\Delta^\dagger$. When $\Delta = \Delta^{(k+1)}$, the GTF operator of order $k+1$, we have $\|(\Delta^{(k+1)})^\dagger\|_2 \leq 1/\lambda_{\min}(L)^{(k+1)/2}$, where $\lambda_{\min}(L)$ is the smallest nonzero eigenvalue of the Laplacian L (also known as the Fiedler eigenvalue [15]). In general, $\lambda_{\min}(L)$ can be very small, leading to loose error bounds, but for the particular graphs in question, it is well-controlled. When $\|\Delta^{(k+1)}\beta_0\|_1$ is bounded, cases 2 and 3 of the corollary show that the average squared error of GTF converges at the rate $\sqrt{\log(nd)}/(nd^{(k+1)/2})$. As k increases, this rate is stronger, but so is the assumption that $\|\Delta^{(k+1)}\beta_0\|_1$ is bounded.

Case 1 in Corollary 4 covers univariate trend filtering (which, recall, is basically the same as GTF over a chain graph; the only differences between the two are boundary terms in the construction of the difference operators). The result in case 1 is based on direct calculation of M, using specific facts that are known about the univariate difference operators. It is natural in the univariate setting to assume that $n^k\|D^{(k+1)}\beta_0\|_1$ is bounded (this is the scaling that would link β_0 to the evaluations of a piecewise polynomial function f_0 over $[0,1]$, with $\mathrm{TV}(f_0^{(k)})$ bounded). Under such an assumption, the above corollary yields a convergence rate of $\sqrt{\log n/n}$ for univariate trend filtering, which is not tight. A more refined analysis shows the univariate trend filtering estimator to converge at the minimax optimal rate $n^{-(2k+2)/(2k+3)}$, proved in [40] by using a connection between univariate trend filtering and locally adaptive regression splines, and relying on sharp entropy-based rates for locally adaptive regression splines from [24]. We note that in a pure graph-centric setting, the latter strategy is not generally applicable, as the notion of a spline function does not obviously extend to the nodes of an arbitrary graph structure.

In the next subsections, we develop more advanced strategies for deriving fast GTF error rates, based on incoherence, and entropy. These can provide substantial improvements over the basic error bound established in this subsection, but are only applicable to certain graph models. Fortunately, this includes common graphs of interest, such as regular grids. To verify the sharpness of these alternative strategies, we will show that they can be used to recover optimal rates of convergence for trend filtering in the 1d setting.

6.2 Strong Error Bounds Based on Incoherence

A key step in the proof of Theorem 3 argues, roughly speaking, that

$$\epsilon^\top \Delta^\dagger \Delta x \leq \|(\Delta^\dagger)^\top \epsilon\|_\infty \|\Delta x\|_1 = O_{\mathbb{P}}(M\sqrt{\log r}\|\Delta x\|_1), \tag{14}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. The second bound holds by a standard result on maxima of Gaussians (recall that M is largest ℓ_2 norm of the columns of $\Delta^\dagger$). The first bound above uses Holder's inequality; note that this applies to any ϵ, Δ, i.e., it does not use any information about the distribution of ϵ, or the properties of Δ. The next lemma reveals a potential advantage that can be gained from replacing the bound (14), stemming from Holder's inequality, with a "linearized" bound.

Lemma 5. *Denote $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, and assume that*

$$\max_{x \in \mathcal{S}_\Delta(1)} \frac{\epsilon^\top x - A}{\|x\|_2} = O_{\mathbb{P}}(B), \tag{15}$$

where $\mathcal{S}_\Delta(1) = \{x \in \mathrm{row}(\Delta) : \|\Delta x\|_1 \leq 1\}$. With $\lambda = \Theta(A)$, the generalized lasso estimate $\hat\beta$ satisfies

$$\frac{\|\hat\beta - \beta_0\|_2^2}{n} = O_{\mathbb{P}}\left(\frac{\mathrm{nullity}(\Delta)}{n} + \frac{B^2}{n} + \frac{A}{n} \cdot \|\Delta\beta_0\|_1\right).$$

The inequality in (15) is referred to as a "linearized" bound because it implies that for $x \in \mathcal{S}_\Delta(1)$,

$$\epsilon^\top x = O_{\mathbb{P}}(A + B\|x\|_2),$$

and the right-hand side is a linear function of $\|x\|_2$. Indeed, for $A = M\sqrt{2\log r}$ and $B = 0$, this encompasses the bound in (14) as a special case, and the result of Lemma 5 reduces to that of Theorem 3. But the result in Lemma 5 can be much stronger, if A, B can be adjusted so that A is smaller than $M\sqrt{2\log r}$, and B is also small. Such an arrangement is possible for certain operators Δ; e.g., it is possible under an incoherence-type assumption on Δ.

Theorem 6. *Let $q = \mathrm{rank}(\Delta)$, and let $\xi_1 \leq \ldots \leq \xi_q$ denote the singular values of Δ, in increasing order. Also let $u_1, \ldots u_q$ be the corresponding left singular vectors. Assume that these vectors are incoherent:*

$$\|u_i\|_\infty \leq \mu/\sqrt{n}, \quad i = 1, \ldots q,$$

for some constant $\mu \geq 1$. For $i_0 \in \{1, \ldots q\}$, let

$$\lambda = \Theta\left(\mu\sqrt{\frac{\log r}{n}\sum_{i=i_0+1}^{q}\frac{1}{\xi_i^2}}\right).$$

Then the generalized lasso estimate $\hat\beta$ has average squared error

$$\frac{\|\hat\beta - \beta_0\|_2^2}{n} = O_{\mathbb{P}}\left(\frac{\mathrm{nullity}(\Delta)}{n} + \frac{i_0}{n} + \frac{\mu}{n}\sqrt{\frac{\log r}{n}\sum_{i=i_0+1}^{q}\frac{1}{\xi_i^2}} \cdot \|\Delta\beta_0\|_1\right).$$

Theorem 6 is proved by leveraging the linearized bound (15), which holds under the incoherence condition on the singular vectors of Δ. Compared to the basic result in Theorem 3, the result in Theorem 6 is clearly stronger as it allows us to replace M—which can grow like the reciprocal of the minimum nonzero singular value of Δ—with something akin to the average reciprocal of larger singular values. But it does, of course, also make stronger assumptions (incoherence). It is interesting to note that the functional in the theorem, $\sum_{i=i_0+1}^{q} \xi_i^{-2}$, was also determined to play a leading role in error bounds for a graph Fourier based scan statistic in the hypothesis testing framework [32].

Applying the above theorem to the GTF estimator requires knowledge of the singular vectors of $\Delta = \Delta^{(k+1)}$, the $(k+1)$st order graph difference operator. The validity of an incoherence assumption on these singular vectors depend on the graph G in question. When k is odd, these singular vectors are eigenvectors of the Laplacian L; when k is even, they are left singular vectors of the edge incidence matrix D. Loosely speaking, these vectors will be incoherent when neighborhoods of different vertices look roughly the same. Most social networks will have this property for the bulk of their vertices (i.e., with the exception of a small number of high degree vertices). Grid graphs also have this property. First, we consider trend filtering over a 1d grid, i.e., a chain (which, recall, is essentially equivalent to univariate trend filtering).

Corollary 7. *Consider the GTF estimator $\hat{\beta}$ of order k, over a chain graph, i.e., a 1d grid graph. Letting*

$$\lambda = \Theta\left(n^{\frac{2k+1}{2k+3}} (\log n)^{\frac{1}{2k+3}} \|\Delta^{(k+1)}\beta_0\|_1^{-\frac{2k+1}{2k+3}} \right),$$

the estimator $\hat{\beta}$ (here, essentially, the univariate trend filtering estimator) satisfies

$$\frac{\|\hat{\beta} - \beta_0\|_2^2}{n} = O_{\mathbb{P}}\left(n^{-\frac{2k+2}{2k+3}} (\log n)^{\frac{1}{2k+3}} \cdot \left(n^k \|\Delta^{(k+1)}\beta_0\|_1 \right)^{\frac{2}{2k+3}} \right).$$

We note that the above corollary essentially recovers the optimal rate of convergence for the univariate trend filtering estimator, for all orders k. (To be precise, it studies GTF on a chain graph instead, but this is basically the same problem.) When $n^k \|\Delta^{(k+1)}\beta_0\|_1$ is assumed to be bounded, a natural assumption in the univariate setting, the corollary shows the estimator to converge at the rate $n^{-(2k+2)/(2k+3)}(\log n)^{1/(2k+3)}$. Ignoring the log factor, this matches the minimax optimal rate as established in [40, 46]. Importantly, the proof of Corollary 7, unlike that used in previous works, is free from any dependence on univariate spline functions; it is completely graph-theoretic, and only uses on the incoherence properties of the 1d grid graph. The strength of this approach is its wider applicability, which we demonstrate by moving up to 2d grids.

Corollary 8. *Consider the GTF estimator $\hat{\beta}$ of order k, over a 2d grid graph, of size $\sqrt{n} \times \sqrt{n}$. Letting*

$$\lambda = \Theta\left(n^{\frac{2k+1}{2k+5}} (\log n)^{\frac{1}{2k+5}} \|\Delta^{(k+1)}\beta_0\|_1^{-\frac{2k+1}{2k+5}} \right),$$

the estimator $\hat{\beta}$ satisfies

$$\frac{\|\hat{\beta} - \beta_0\|_2^2}{n} = O_{\mathbb{P}}\left(n^{-\frac{2k+4}{2k+5}} (\log n)^{\frac{1}{2k+5}} \cdot \left(n^{\frac{k}{2}} \|\Delta^{(k+1)}\beta_0\|_1 \right)^{\frac{4}{2k+5}} \right).$$

The 2d result in Corollary 8 is written in a form that mimics the 1d result in Corollary 7, as we claim that the analog of boundedness of $n^k \|\Delta^{(k+1)}\beta_0\|_1$ in 1d is boundedness of $n^{k/2} \|\Delta^{(k+1)}\beta_0\|_1$ in 2d.[4] Thus, under the appropriate boundedness condition, the 2d rate shows improvement over the 1d rate, which makes sense, since regularization here is being enforced in a richer manner. It is worthwhile highlighting the result for $k = 0$ in particular: this says that, when the sum of absolute discrete differences $\|\Delta^{(1)}\beta_0\|_1$ is bounded over a 2d grid, the 2d fused lasso (i.e., 2d total variation denoising) has error rate $n^{-4/5}$. This is faster than the $n^{-2/3}$ rate for the 1d fused lasso, when the sum of absolute differences $\|D^{(1)}\beta_0\|_1$ is bounded. Rates for higher dimensional grid graphs (for all k) follow from analogous arguments, but we omit the details.

6.3 Strong Error Bounds Based on Entropy

A different "fractional" bound on the Gaussian contrast $\epsilon^\top x$, over $x \in \mathcal{S}_\Delta(1)$, provides an alternate route to deriving sharper rates. This style of bound is inspired by the seminal work of [43].

[4] This is because $1/\sqrt{n}$ is the distance between adjacent 2d grid points, when viewed as a 2d lattice over $[0, 1]^2$.

Lemma 9. *Denote $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, and assume that for a constant $w < 2$,*

$$\max_{x \in \mathcal{S}_\Delta(1)} \frac{\epsilon^\top x}{\|x\|_2^{1-w/2}} = O_{\mathbb{P}}(K), \tag{16}$$

where recall $\mathcal{S}_\Delta(1) = \{x \in \mathrm{row}(\Delta) : \|\Delta x\|_1 \leq 1\}$. Then with

$$\lambda = \Theta\left(K^{\frac{2}{1+w/2}} \cdot \|\Delta\beta_0\|_1^{-\frac{1-w/2}{1+w/2}}\right),$$

the generalized lasso estimate $\hat{\beta}$ satisfies

$$\frac{\|\hat{\beta} - \beta_0\|_2^2}{n} = O_{\mathbb{P}}\left(\frac{\mathrm{nullity}(\Delta)}{n} + \frac{K^{\frac{2}{1+w/2}}}{n} \cdot \|\Delta\beta_0\|_1^{\frac{w}{1+w/2}}\right).$$

The main motivation for bounds of the form (16) is that they follow from entropy bounds on the set $\mathcal{S}_\Delta(1)$. Recall that for a set S, the covering number $N(\delta, S, \|\cdot\|)$ is the fewest number of balls of radius δ that cover S, with respect to the norm $\|\cdot\|$. The log covering or entropy number is $\log N(\delta, S, \|\cdot\|)$. In the next result, we make the connection between between entropy and fractional bounds precise; this follows closely from Lemma 3.5 of [43].

Theorem 10. *Suppose that there exist a constant $w < 2$ such that for n large enough,*

$$\log N(\delta, \mathcal{S}_\Delta(1), \|\cdot\|_2) \leq E\left(\frac{\sqrt{n}}{\delta}\right)^w, \tag{17}$$

for $\delta > 0$, where E can depend on n. Then the fractional bound in (16) holds with $K = \sqrt{E} n^{w/4}$, and as a result, for

$$\lambda = \Theta\left(E^{\frac{1}{1+w/2}} n^{\frac{w/2}{1+w/2}} \|\Delta\beta_0\|_1^{-\frac{1-w/2}{1+w/2}}\right),$$

the generalized lasso estimate $\hat{\beta}$ has average squared error

$$\frac{\|\hat{\beta} - \beta_0\|_2^2}{n} = O_{\mathbb{P}}\left(\frac{\mathrm{nullity}(\Delta)}{n} + E^{\frac{1}{1+w/2}} n^{-\frac{1}{1+w/2}} \cdot \|\Delta\beta_0\|_1^{\frac{w}{1+w/2}}\right).$$

To make use of the result in Theorem 10, we must obtain an entropy bound as in (17), on the set $\mathcal{S}_\Delta(1)$. The literature on entropy numbers is rich, and there are various methods for computing entropy bounds, any of which can be used for these purposes as long as the bounds fit the form of (17), as required by the theorem. For bounding the entropy of a set like $\mathcal{S}_\Delta(1)$, two common techniques are to use a characterization of the spectral decay of $\Delta^\dagger$, or an analysis of the correlations between columns of $\Delta^\dagger$. For a nice review of such strategies and their applications, we refer the reader to Section 6 of [44] and Section 14.12 of [6]. We do not pursue either of these two strategies in the current paper. We instead consider a third, somewhat more transparent strategy, based on a covering number bound of the columns of $\Delta^\dagger$.

Lemma 11. *Let $g_1, \ldots g_r$ denote the "atoms" associated with the operator Δ, i.e., the columns of $\Delta^\dagger$, and let $\mathcal{G} = \{\pm g_i : i = 1, \ldots r\}$ denote the symmetrized set of atoms. Suppose that there exists constants ζ, C_0 with the following property: for each $j = 1, \ldots 2r$, there is an arrangement of j balls having radius at most*

$$C_0 \sqrt{n} j^{-1/\zeta},$$

with respect to the norm $\|\cdot\|_2$, that covers $\mathcal{G}$. Then the entropy bound in (17) is met with $w = 2\zeta/(2+\zeta)$ and $E = O(1)$. Therefore, the generalized lasso estimate $\hat{\beta}$, with

$$\lambda = \Theta\left(n^{\frac{\zeta}{2+2\zeta}} \|\Delta\beta_0\|_1^{-\frac{1}{1+\zeta}}\right),$$

satisfies

$$\frac{\|\hat{\beta}-\beta_0\|_2^2}{n} = O_{\mathbb{P}}\left(\frac{\mathrm{nullity}(\Delta)}{n} + n^{-\frac{2+\zeta}{2+2\zeta}} \cdot \|\Delta\beta_0\|_1^{\frac{\zeta}{1+\zeta}}\right).$$

The entropy-based results in this subsection (Lemma 9, Theorem 10, and Lemma 11) may appear more complex than those involving incoherence in the previous subsection (Lemma 5 and Theorem 6). Indeed, the same can be said of their proofs, which can be found in the Appendix. But after all this entropy machinery has all been established, it can actually be remarkably easy to use, say, Lemma 11 to produce sharp results. We conclude by giving an example.

Corollary 12. *Consider the 1d fused lasso, i.e., the GTF estimator with $k = 0$ over a chain graph. In this case, we have $\Delta = D^{(1)}$, the univariate difference operator, and the symmetrized set $\mathcal{G}$ of atoms can be covered by j balls with radius at most $\sqrt{2n/j}$, for $j = 1, \dots 2(n-1)$. Hence, with $\lambda = \Theta(n^{1/3}\|D^{(1)}\beta_0\|_1^{-1/3})$, the 1d fused lasso estimate $\hat{\beta}$ satisfies*

$$\frac{\|\hat{\beta}-\beta_0\|_2^2}{n} = O_{\mathbb{P}}\left(n^{-2/3} \cdot \|D^{(1)}\beta_0\|_1^{2/3}\right).$$

This corollary rederives the optimal convergence rate of $n^{-2/3}$ for the univariate fused lasso, assuming boundedness of $\|D^{(1)}\beta_0\|_1$, as has been already shown in [24, 40]. Like Corollary 7 (but unlike previous works), its proof does not rely on any special facts about 1d functions of bounded variation. It only uses a covering number bound on the columns of the operator $(D^{(1)})^+$, a strategy that, in principle, extends to many other settings (graphs). It is worth emphasizing just how simple this covering number construction is, compared to the incoherence-based arguments that lead to the same result; we invite the curious reader to compare the proofs of Corollaries 7 and 12.

7 Discussion

In this work, we proposed graph trend filtering as a useful alternative to Laplacian and wavelet smoothers on graphs. This is analogous to the usefulness of univariate trend filtering in nonparametric regression, as an alternative to smoothing splines and wavelets [40]. We have documented empirical evidence for the superior local adaptivity of the ℓ_1-based GTF over the ℓ_2-based graph Laplacian smoother, and the superior robustness of GTF over wavelet smoothing in high-noise scenarios. Our theoretical analysis provides a basis for a deeper understanding of the estimation properties of GTF. More precise theoretical characterizations involving entropy will be the topic of future work, as will comparisons between the error rates achieved by GTF and other common estimators, such as Laplacian smoothing. These extensions, and many others, are well within reach.

Acknowledgments

YW was supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office. JS was supported by NSF Grant DMS-1223137. AS was supported by a Google Faculty Research Grant. RT was supported by NSF Grant DMS-1309174.

A Additional Analysis from Alternative Wavelet Designs

A.1 Allegheny County Example

In addition to considering the wavelet design of [31] for the Allegheny County example, we also considered designs of [10]—a classic method that builds diffusion wavelets on a graph, and [18]—a more recent graph wavelet construction. In contrast to [31], which produces a single signal-independent orthogonal basis for a graph, both [10, 18] build wavelet packets from a given graph structure. A wavelet packet is an overcomplete basis indexed by a hierarchical data structure that can be used to generate an exponential number of orthogonal bases. This construction is computationally expensive as it typically involves computing eigendecompositions of large matrices. Once the wavelet packet has been constructed, for each input signal that one observes over the graph in question, one runs a “best basis” algorithm to choose a particular orthogonal basis from the wavelet packet by optimizing a particular cost function of the eventual wavelet coefficients. This is based on a message-passing-like dynamic programming algorithm, and can be quite efficient. Lastly, the denoising procedure is defined as usual (e.g., as in [12]), namely, one performs the basis transformation, soft-thresholds (or hard-thresholds) the coefficients, and then reconstructs the denoised signal.

In our experiments, we used the wavelet implementations released by the authors of [10, 18] with their default settings. In particular, the former implementation of [10] builds wavelets from a diffusion operator constructed from the adjacency matrix of a graph, and the cost function for the best basis is defined by the ℓ_1 norm of the wavelet coefficients. The latter implementation of [18] uses a more exhaustive search, building wavelet packets through a hierarchical partitioning and eigentransform of three different Laplacian matrices and a fourth generalized Haar-Walsh transform (GHWT), then choosing the best basis from all four collections by optimizing a meta cost function of the ℓ_p norm of wavelet coefficients over $p \in \{0.1, 0.2, \dots 2\}$. This is the “cumulative relative error” defined in equation (7.5) of [18].

In the left panel of Figure 8, we plot the mean squared errors for these new wavelet methods over the same 10 simulations from the Allegheny County example in Figure 2 of Section A.1. The middle and right panels of the figure show the denoised signals from the new methods fit to the data in Figure 1, at their optimal degrees of freedom (df) values (in terms of the achieved MSE). We can see that the spanning tree wavelet design of [31] is the best performer among the three candidate wavelet designs. In a rough sense, the construction of [18] seems to perform similarly to that of [31], in that the MSE is best for larger df values (corresponding to more nonzero wavelet coefficients, i.e., complex fitted models), whereas the construction of [10] performs best for smaller df values (fewer nonzero wavelet coefficients, i.e., simpler fitted models).

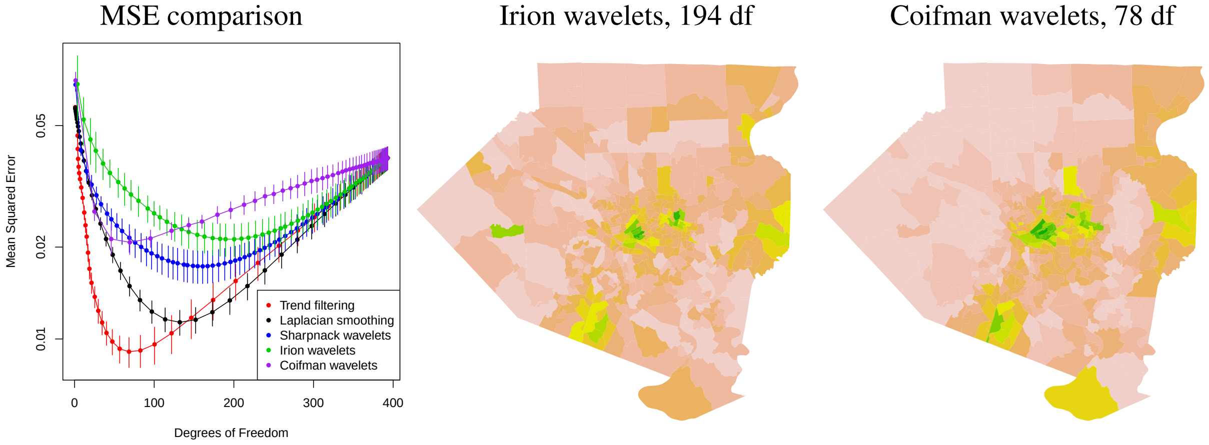


Figure 8: Additional wavelet analysis of the Allegheny County example.

A.2 Facebook Graph Example

Again, we consider the designs of [10, 18] for the Facebook graph example of Section 5.1. Due to practical reasons, we had to change some of the default settings in the implementations provided by the authors of these wavelet methods; in the wavelet implementation of [10], we took the power of the diffusion operator to be 1 instead of 4 (since the latter choice threw an error in the provided code); and in the wavelet implementation of [18], we used another "best basis" algorithm that only searches within the basis collection from the GHWT eigendecomposition, as the original algorithm was too slow due to the larger scale considered in this example. (In most examples in [18], the chosen bases come from the GHWT eigendecomposition.) We view these changes as minor, because when the same changes were applied to the methods of [10, 18] on the smaller Allegheny County example, there are no obvious differences in the results.

Figure 9 shows the results for the two new wavelet methods on the Facebook graph simulation, using the same setup as in Figure 5. Once again, we find that the spanning tree wavelets of [31] perform better or on par with the other two wavelet methods across essentially all scenarios.

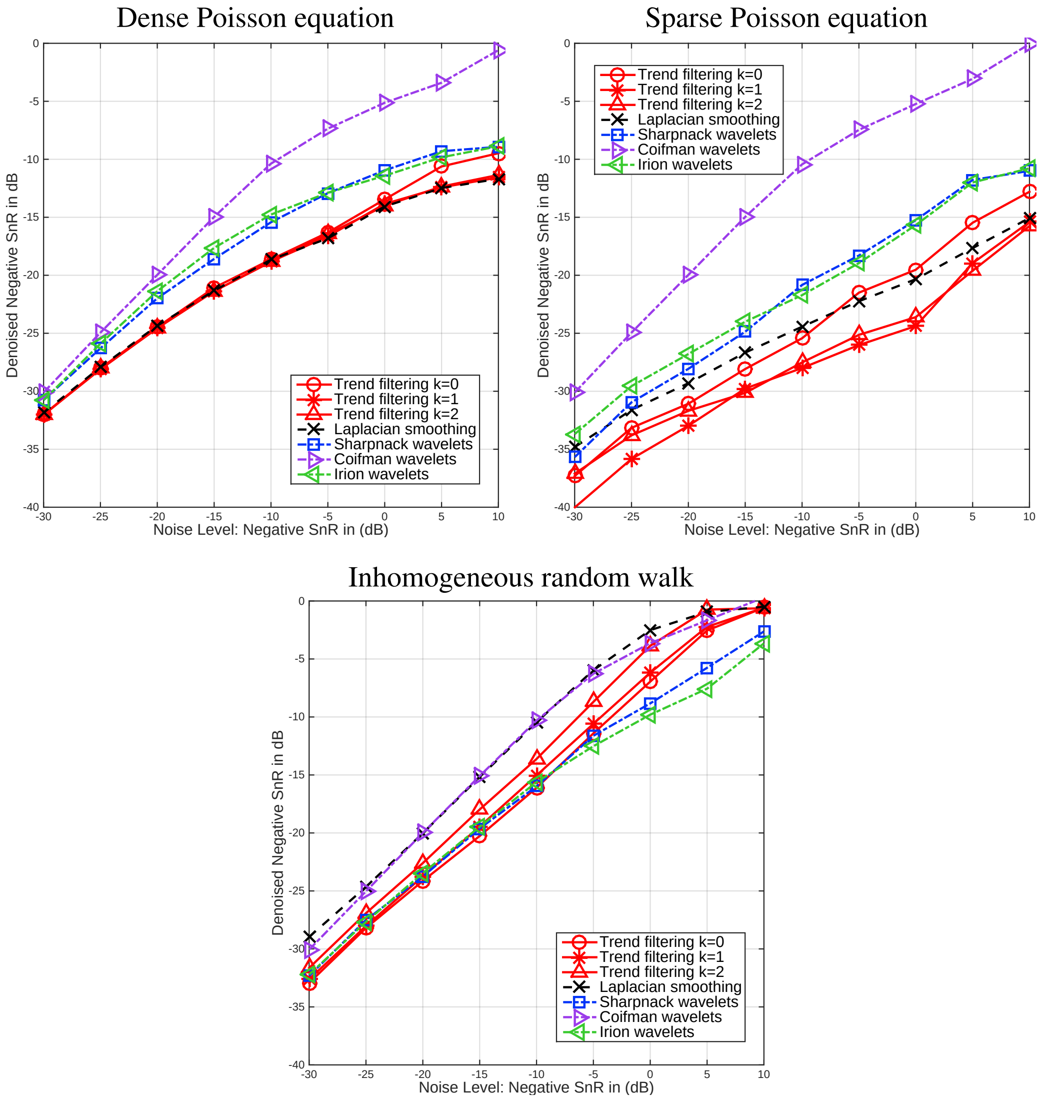


Figure 9: Additional wavelet analysis of the Facebook graph example.

B Proofs of Theoretical Results

B.1 Proof of Lemma 1

For even k, we have $\Delta^{(k+1)} = DL^{k/2}$, so if A denotes a subset of edges, then $\Delta^{(k+1)}_{-A} = D_{-A}L^{k/2}$. Recall that for a connected graph, $\mathrm{null}(L) = \mathrm{span}\{\mathbb{1}\}$, and the same is true for any power of L. This means that we can write

$$\mathrm{null}(\Delta^{(k+1)}) = \mathrm{span}\{\mathbb{1}\} + \mathrm{span}\{\mathbb{1}\}^\perp \cap \{u : DL^{\frac{k}{2}}u = 0\}.$$

Note that if $\mathbb{1}^\top u = 0$, then $v = L^{\frac{k}{2}}u \iff u = (L^\dagger)^{\frac{k}{2}}u$. Moreover, if G_{-A} has connected components $C_1, \dots C_s$, then $\mathrm{null}(D_{-A}) = \mathrm{span}\{\mathbb{1}_{C_1}, \dots \mathbb{1}_{C_s}\}$. Putting these statements together proves the result for even k. For k odd, the arguments are similar.

B.2 Proof of Theorem 3

By assumption we can write

$$y = \beta_0 + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I).$$

Denote $R = \mathrm{row}(\Delta)$, the row space of Δ, and $R^\perp = \mathrm{null}(\Delta)$, the null space of Δ. Also let P_R be the projection onto R, and $P_{R^\perp}$ the projection onto $R^\perp$. Consider

$$\hat{\beta} = \mathop{\mathrm{argmin}}_{\beta \in \mathbb{R}^n} \frac{1}{2}\|y - \beta\|_2^2 + \lambda\|\Delta\beta\|_1,$$

$$\tilde{\beta} = \mathop{\mathrm{argmin}}_{\beta \in \mathbb{R}^n} \frac{1}{2}\|P_R y - \beta\|_2^2 + \lambda\|\Delta\beta\|_1.$$

The first quantity $\hat{\beta} \in \mathbb{R}^n$ is the estimate of interest, the second one $\tilde{\beta} \in R$ is easier to analyze. Note that

$$\hat{\beta} = P_{R^\perp} y + \tilde{\beta},$$

and write $\|x\|_R = \|P_R x\|_2$, $\|x\|_{R^\perp} = \|P_{R^\perp} x\|_2$. Then

$$\|\hat{\beta} - \beta_0\|_2^2 = \|\epsilon\|_{R^\perp}^2 + \|\tilde{\beta} - \beta_0\|_R^2.$$

The first term is on the order $\dim(R^\perp) = \mathrm{nullity}(\Delta)$, and it suffices to bound the second term. Now we establish a basic inequality for $\tilde{\beta}$. By optimality of $\tilde{\beta}$, we have

$$\frac{1}{2}\|y - \tilde{\beta}\|_R^2 + \lambda\|\Delta\tilde{\beta}\|_1 \le \frac{1}{2}\|y - \beta_0\|_R^2 + \lambda\|\Delta\beta_0\|_1,$$

and after rearranging terms,

$$\|\tilde{\beta} - \beta_0\|_R^2 \le 2\epsilon^\top P_R(\tilde{\beta} - \beta_0) + 2\lambda\|\Delta\beta_0\|_1 - 2\lambda\|\Delta\tilde{\beta}\|_1. \tag{18}$$

This is our basic inequality. In the first term above, we use $P_R = \Delta^\dagger\Delta$, and apply Holder's inequality:

$$\epsilon^\top \Delta^\dagger\Delta(\tilde{\beta} - \beta_0) \le \|(\Delta^\dagger)^\top\epsilon\|_\infty \|\Delta(\tilde{\beta} - \beta_0)\|_1. \tag{19}$$

If $\lambda \ge \|(\Delta^\dagger)^\top\epsilon\|_\infty$, then from (18), (19), and the triangle inequality, we see that

$$\|\tilde{\beta} - \beta_0\|_R^2 \le 4\lambda\|\Delta\beta_0\|_1.$$

Well, $\|(\Delta^\dagger)^\top\epsilon\|_\infty = O_{\mathbb{P}}(M\sqrt{\log r})$ by a standard result on the maximum of Gaussians (derived using the union bound, and Mills' bound on the Gaussian tail), where recall M is the maximum ℓ_2 norm of the columns of $\Delta^\dagger$. Thus with $\lambda = \Theta(M\sqrt{\log r})$, we have from the above that

$$\|\tilde{\beta} - \beta_0\|_R^2 = O_{\mathbb{P}}\big(M\sqrt{\log r}\|\Delta\beta_0\|_1\big),$$

as desired.

B.3 Proof of Corollary 4

Case 1. When $\hat{\beta}$ is the univariate trend filtering estimator of order k, we are considering a penalty matrix $\Delta = D^{(k+1)}$, the univariate difference operator of order $k+1$. Note that $D^{(k+1)} \in \mathbb{R}^{(n-k-1)\times n}$, and its null space has constant dimension $k+1$. We show in Lemma 13 of Appendix B.4 that $(D^{(k+1)})^\dagger = P_R H_2^{(k)}/k!$, where $R = \mathrm{row}(D^{(k+1)})$, and $H_2^{(k)} \in \mathbb{R}^{n\times(n-k-1)}$ contains the last $n-k-1$ columns of the order k falling factorial basis matrix [46], evaluated over the input points $x_1 = 1, \dots x_n = n$. The largest column norm of $P_R H_2^{(k)}/k!$ is on the order of $n^{k+1/2}$, which proves the result.

Cases 2 and 3. When G is the Ramanujan d-regular graph, the number of edges in the graph is $O(nd)$. The operator $\Delta = \Delta^{(k+1)}$ has number of rows $r = n$ when k is odd and $r = O(nd)$ when k is even; overall this is $O(nd)$. The dimension of the null space of Δ is constant (it is in fact 1, since the graph is connected). When G is the Erdos-Renyi random graph, the same bounds apply to the number of rows and the dimension of the null space, except that the bounds become probabilistic ones.

Now we apply the crude inequality, with e_i, $i = 1, \dots r$ denoting the standard basis vectors,

$$M = \max_{i=1,\dots r} \Delta^\dagger e_i \le \max_{\|x\|_2 \le 1} \Delta^\dagger x = \|\Delta^\dagger\|_2,$$

the right-hand side being the maximum singular value of $\Delta^\dagger$. As $\Delta = \Delta^{(k+1)}$, the graph difference operator of order $k+1$, we claim that

$$\|\Delta^\dagger\|_2 \le 1/\lambda_{\min}(L)^{\frac{k+1}{2}}, \tag{20}$$

where $\lambda_{\min}(L)$ denotes the smallest nonzero eigenvalue of the graph Laplacian L. To see this, note first that $\|\Delta^\dagger\|_2 = 1/\sigma_{\min}(\Delta)$, where the denominator is the smallest nonzero singular value of Δ. Now for odd k, we have $\Delta^{(k+1)} = L^{(k+1)/2}$, and the claim follows as

$$\sigma_{\min}(L^{\frac{k+1}{2}}) = \min_{x\in R: \|x\|_2 \le 1} \|L^{\frac{k+1}{2}} x\|_2 \ge \big(\sigma_{\min}(L)\big)^{\frac{k+1}{2}},$$

and $\sigma_{\min}(L) = \lambda_{\min}(L)$, since L is symmetric. Above, R denotes the row space of L (the space orthogonal to the vector $\mathbb{1}$ of all 1s). For even k, we have $\Delta^{(k+1)} = DL^{k/2}$, and again

$$\sigma_{\min}(DL^{\frac{k}{2}}) = \min_{x\in R: \|x\|_2 \le 1} \|DL^{\frac{k+1}{2}} x\|_2 \ge \sigma_{\min}(D)\big(\sigma_{\min}(L)\big)^{\frac{k}{2}},$$

where $\sigma_{\min}(D) = \sqrt{\lambda_{\min}(L)}$, since $D^\top D = L$. This verifies the claim.

Having established (20), it suffices to lower bound $\lambda_{\min}(L)$ for the two graphs in question. Indeed, for both graphs, we have the lower bound

$$\lambda_{\min}(L) = \Omega(d - \sqrt{d}).$$

e.g., see [23, 25] for the Ramanujan graph and [14, 9] for the Erdos-Renyi graph. This completes the proof.

B.4 Calculation of $(D^{(k+1)})^\dagger$

Lemma 13. *The $(k+1)$st order discrete difference operator has pseudoinverse $(D^{(k+1)})^\dagger = P_R H_2^{(k)}/k!$, where we denote $R = \mathrm{row}(D^{(k+1)})$, and $H_2^{(k)} \in \mathbb{R}^{n\times(n-k-1)}$ the last $n-k-1$ columns of the kth order falling factorial basis matrix.*

Proof. We abbreviate $D = D^{(k+1)}$, and consider the linear system

$$DD^\top x = Db \tag{21}$$

in x, where $b \in \mathbb{R}^n$ is arbitrary. We seek an expression for $x = (DD^\top)^{-1}D^\top = (D^\dagger)^\top b$, and this will tell us the form of $D^\dagger$. Define

$$\tilde{D} = \begin{bmatrix} C \\ D \end{bmatrix} \in \mathbb{R}^{n \times n},$$

where $C \in \mathbb{R}^{(k+1)\times n}$ is the matrix that collects the first row of each lower order difference operator, defined in Lemma 2 of [46]. From this same lemma, we know that

$$\tilde{D}^{-1} = H/k!,$$

where $H = H^{(k)}$ is falling factorial basis matrix of order k, evaluated over $x_1, \ldots x_n$. With this in mind, consider the expanded linear system

$$\begin{bmatrix} CC^\top & CD^\top \\ DC^\top & DD^\top \end{bmatrix} \begin{bmatrix} w \\ x \end{bmatrix} = \begin{bmatrix} a \\ Db \end{bmatrix}. \tag{22}$$

The second equation reads

$$DC^\top w + DD^\top x = Db,$$

and so if we can choose a in (22) so that at the solution we have $w = 0$, then x is the solution in (21). The first equation in (22) reads

$$CC^\top w + CD^\top x = a,$$

i.e.,

$$w = (CC^\top)^{-1}(a - CD^\top x).$$

That is, we want to choose

$$a = CD^\top x = CD^\top (DD^\top)^{-1} Db = CP_R b,$$

where P_R is the projection onto row space of D. Thus we can reexpress (22) as

$$\tilde{D}\tilde{D}^\top \begin{bmatrix} w \\ x \end{bmatrix} = \begin{bmatrix} CP_R b \\ Db \end{bmatrix} = \tilde{D} P_R b$$

and, using $\tilde{D}^{-1} = H/k!$,

$$\begin{bmatrix} w \\ x \end{bmatrix} = H^\top P_R b / k!.$$

Finally, writing H_2 for the last $n-k-1$ columns of H, we have $x = H_2^\top P_R b/k!$, as desired. □

Remark. The above proof did not rely on the input points $x_1, \ldots x_n$; indeed, the result holds true for any sequence of inputs used to define the discrete difference matrix and falling factorial basis matrix.

B.5 Proof of Lemma 5

We follow the proof of Theorem 3, up until the application of Holder's inequality in (19). In place of this step, we use the linearized bound in (15), which we claim implies that

$$\epsilon^\top P_R(\tilde{\beta} - \beta_0) \le \tilde{B} \|\tilde{\beta} - \beta_0\|_R + A \|\Delta(\tilde{\beta} - \beta_0)\|_1,$$

where $\tilde{B} = O_{\mathbb{P}}(B)$. This simply follows from applying (15) to $x = P_R(\tilde{\beta} - \beta_0)/\|\Delta(\tilde{\beta} - \beta_0)\|_1$, which is easily seen to be an element of $\mathcal{S}_\Delta(1)$. Hence we can take take $\lambda = \Theta(A)$, and argue as in the proof of Theorem 3 to arrive at

$$\|\tilde{\beta} - \beta_0\|_R^2 \le \tilde{B} \|\tilde{\beta} - \beta_0\|_R + \tilde{A} \|\Delta\beta_0\|_1,$$

where $\tilde{A} = O_{\mathbb{P}}(A)$. The above is a quadratic inequality of the form $ax^2 - bx - c \leq 0$ in $x = \|\tilde{\beta} - \beta_0\|_R$. As $a > 0$, the larger of its two roots serves as a bound for x, i.e., $x \leq (b + \sqrt{b^2 + 4ac})/(2a) \leq b/a + \sqrt{c/a}$, or $x^2 \leq 2b^2/a^2 + 2c/a$, which means that

$$\|\tilde{\beta} - \beta_0\|_R^2 \leq 2\tilde{B}^2 + 2\tilde{A}\|\Delta\beta_0\|_1 = O_{\mathbb{P}}\big(B^2 + A\|\Delta\beta_0\|_1\big),$$

completing the proof.

B.6 Proof of Theorem 6

For an index $i_0 \in \{1, \dots q\}$, let

$$C = \mu\sqrt{\frac{2\log 2r}{n}\sum_{i=i_0+1}^{q}\frac{1}{\xi_i^2}}.$$

We will show that

$$\max_{x \in \mathcal{S}_\Delta(1)} \frac{\epsilon^\top x - 1.001\sigma C}{\|x\|_2} = O_{\mathbb{P}}(\sqrt{i_0}).$$

Invoking Lemma 5 with $A = 1.001\sigma C$ and $b = \sqrt{i_0}$ would then give the result.

Henceforth we denote $[i] = \{1, \dots i\}$. Recall that $q = \mathrm{rank}(\Delta)$. Let the singular value decomposition of Δ be

$$\Delta = U\Sigma V^\top,$$

where $U \in \mathbb{R}^{r \times q}$, $V \in \mathbb{R}^{n \times q}$ are orthogonal, and $\Sigma \in \mathbb{R}^{q \times q}$ has diagonal elements $(\Sigma)_{ii} = \xi_i > 0$, $i \in [q]$. First, let us establish that

$$\Delta^\dagger = V\Sigma^{-1}U^\top.$$

Consider an arbitrary point $x = P_R z \in \mathcal{S}_\Delta(1)$. Denote the projection $P_{[i_0]} = V_{[i_0]}V_{[i_0]}^\top$ where $V_{[i_0]}$ contains the first i_0 right singular vectors. We can decompose

$$\epsilon^\top P_R z = \epsilon^\top P_{[i_0]} P_R z + \epsilon^\top (I - P_{[i_0]}) P_R z.$$

The first term can be bounded by

$$\epsilon^\top P_{[i_0]} P_R z \leq \|P_{[i_0]}\epsilon\|_2 \|z\|_R = O_{\mathbb{P}}(\sqrt{i_0}\|z\|_R),$$

using the fact that $\|P_{[i_0]}\epsilon\|_2^2 \overset{d}{=} \sum_{i=1}^{i_0} \epsilon_i^2$. We can bound the second term by

$$\epsilon^\top (I - P_{[i_0]}) P_R z = \epsilon^\top (I - P_{[i_0]}) \Delta^\dagger \Delta z \leq \|(\Delta^\dagger)^\top (I - P_{[i_0]})\epsilon\|_\infty,$$

using $P_R = \Delta^\dagger\Delta$, Holder's inequality, and the fact that $\|\Delta z\|_1 \leq 1$. Define $g_j = (I - P_{[i_0]})\Delta^\dagger e_j$ for $j \in [r]$ with e_j the jth canonical basis vector. So,

$$\|g_j\|_2^2 = \|[\,0 \;\; V_{[n]\setminus[i_0]}\,] \cdot \Sigma^{-1} U^\top e_j\|_2^2 \leq \frac{\mu^2}{n}\sum_{i=i_0+1}^{q}\frac{1}{\xi_i^2},$$

by rotational invariance of $\|\cdot\|_2$ and the incoherence assumption on the columns of U. By a standard result on maxima of Gaussians,

$$\|(\Delta^\dagger)^\top (I - P_{[i_0]})\epsilon\|_\infty = \max_{j \in [r]} |g_j^\top \epsilon| \leq 1.001\sigma\sqrt{2\log(2r)\frac{\mu^2}{n}\sum_{i=i_0+1}^{q}\frac{1}{\xi_i^2}} = 1.001\sigma C,$$

with probability approaching 1. Putting these two terms together completes the proof, as we have shown that

$$\frac{\epsilon^\top P_R z - 1.001\sigma C}{\|z\|_R} = O_{\mathbb{P}}(\sqrt{i_0}),$$

with the probability bound on the right-hand side not depending on z.

B.7 Proof of Corollary 7

We focus on the k odd and k even cases separately.

Case for k odd. When k is odd, we have $\Delta = \Delta^{(k+1)} = L^{(k+1)/2}$, where L the graph Laplacian of a chain graph (i.e., 1d grid graph), to be perfectly explicit,

$$L = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ -1 & 2 & -1 & \dots & 0 & 0 \\ 0 & -1 & 2 & \dots & 0 & 0 \\ \vdots & & \ddots & \ddots & \ddots & \\ 0 & 0 & \dots & -1 & 2 & -1 \\ 0 & 0 & \dots & 0 & -1 & 1 \end{bmatrix}.$$

In numerical methods for differential equations, this matrix L is called the finite difference operator for the 1d Laplace equation with Neumann boundary conditions (e.g., [11, 16]), and is known to have eigenvalues and eigenvectors

$$\xi_i = 4\sin^2\left(\frac{\pi(i-1)}{2n}\right), \quad \text{for } i = 1, \dots n,$$

$$u_{ij} = \begin{cases} \frac{1}{\sqrt{n}} & \text{if } i = 1 \\ \sqrt{\frac{2}{n}}\cos\left(\frac{\pi(i-1)(j-1/2)}{n}\right) & \text{otherwise} \end{cases}, \quad \text{for } i, j = 1, \dots n.$$

Therefore, the eigenvectors of L are incoherent with constant $\mu = \sqrt{2}$. This of course implies the same of $L^{(k+1)/2}$, which shares the eigenvectors of L. Meanwhile, the eigenvalues of $L^{(k+1)/2}$ are just given by raising those of L to the power of $(k+1)/2$, and for $i_0 \in \{1, \dots n\}$, we compute the partial sum of their squared reciprocals, as in

$$\frac{1}{n}\sum_{i=i_0+1}^{n} \frac{1}{\xi_i^{k+1}} = \frac{1}{n}\sum_{i=i_0+1}^{n} \frac{1}{4^{k+1}\sin^{2k+2}(\pi(i-1)/(2n))} \le \int_{(i_0-1)/n}^{(n-2)/n} \frac{1}{4^{k+1}\sin^{2k+2}(\pi x/2)} dx,$$

where we have used the fact that the right-endpoint Riemann sum, for a monotone nonincreasing function, is an underestimate of its integral. Continuing on, the above integral can be bounded by

$$\frac{1}{4^{k+1}\sin^{2k}(\pi i_0/(2n))} \int_{(i_0-1)/n}^{1} \frac{1}{\sin^2(\pi x/2)} dx = \frac{2\cot(\pi i_0/(2n))}{4^{k+1}\pi\sin^{2k}(\pi i_0/(2n))} \le \frac{1}{4^{k+1}\pi}\left(\frac{2n}{\pi i_0}\right)^{2k+1},$$

the last step using a Taylor expansion around 0. Hence to choose a tight a bound as possible in Theorem 6, we seek to balance i_0 with $\sqrt{(n/i_0)^{2k+1}\log n} \cdot \|\Delta^{(k+1)}\beta_0\|_1$. This is accomplished by choosing

$$i_0 = n^{\frac{2k+1}{2k+3}} (\log n)^{\frac{1}{2k+3}} \|\Delta^{(k+1)}\beta_0\|_1^{\frac{2}{2k+3}},$$

and applying Theorem 6 gives the result for k odd.

Case for k even. When k is even, we instead have $\Delta = \Delta^{(k+1)} = DL^{k/2}$, where D is the edge incidence matrix of a 1d chain, and $L = D^\top D$. It is clear that the left singular vectors of $DL^{k/2}$ are simply the left singular vectors of D, or equivalently, the eigenvectors of $DD^\top$. To be explicit,

$$DD^\top = \begin{bmatrix} 2 & -1 & 0 & \dots & 0 & 0 \\ -1 & 2 & -1 & \dots & 0 & 0 \\ 0 & -1 & 2 & \dots & 0 & 0 \\ \vdots & & \ddots & \ddots & \ddots & \\ 0 & 0 & \dots & -1 & 2 & -1 \\ 0 & 0 & \dots & 0 & -1 & 2 \end{bmatrix},$$

which in numerical methods, is called the finite difference operator associated with the 1d Laplace equation under Dirichlet boundary conditions (e.g., [11, 16]), and is known to have eigenvectors

$$u_{ij} = \sqrt{\frac{2}{n}} \sin\left(\frac{\pi i j}{n}\right), \quad \text{for } i,j = 1, \dots n-1.$$

It is evident that these vectors are incoherent, with constant $\mu = \sqrt{2}$. Furthermore, the singular values of $DL^{k/2}$ are exactly the eigenvalues of L raised to the power of $(k+1)/2$, and the remainder of the proof goes through as in the k odd case.

B.8 Proof of Corollary 8

Again we treat the k odd and even cases separately.

Case for k odd. As k is odd, the GTF operator is $\Delta = \Delta^{(k+1)} = L^{(k+1)/2}$, where the L is the Laplacian matrix of a 2d grid graph. Writing $L_{\mathrm{1d}} \in \mathbb{R}^{\ell \times \ell}$ for the Laplacian matrix over a 1d grid of size $\ell = \sqrt{n}$ (and $I \in \mathbb{R}^{\ell \times \ell}$ for the identity matrix), we note that

$$L = I \otimes L_{\mathrm{1d}} + L_{\mathrm{1d}} \otimes I,$$

i.e., the 2d grid Laplacian L is the Kronecker sum of the 1d grid Laplacian L_{1d}, so its eigenvectors are given by all pairwise Kronecker products of eigenvectors of L_{1d}, of the form $u_i \otimes u_j$. Moreover, it is not hard to see that each $u_i \otimes u_j$ has unit norm (since u_i, u_j do) and $\|u_i \otimes u_j\|_\infty \le 2/\sqrt{n}$. This allows us to conclude that the eigenvectors of L obey the incoherence property with $\mu = 2$.

The eigenvalues of L are given by all pairwise sums of eigenvalues in the 1d case. Indexing by 2d grid coordinates, we may write these as

$$\xi_{j_1,j_2} = 4\sin^2\left(\frac{\pi(j_1-1)}{2\ell}\right) + 4\sin^2\left(\frac{\pi(j_2-1)}{2\ell}\right), \quad \text{for } j_1, j_2 = 1, \dots \ell.$$

Eigenvalues of $L^{(k+1)/2}$ are just given by raising the above to the power of $(k+1)/2$, and for $j_0 \in \{1, \dots \ell\}$, we let $i_0 = j_0^2$, and compute the sum

$$\frac{1}{n} \sum_{\max\{j_1 j_2\} > j_0} \frac{1}{\xi_{j_1,j_2}^{k+1}} \le \frac{2}{n} \sum_{j_1=j_0+1}^{\ell} \sum_{j_2=1}^{\ell} \frac{1}{\xi_{j_1,j_2}^{k+1}} \le \frac{2}{\ell} \sum_{j_1=j_0+1}^{\ell} \frac{1}{4^{k+1} \sin^{2k+2}(\pi(j_1-1)/(2\ell))}.$$

Just as we argued in the 1d case (for k odd), the above is bounded by

$$\frac{2}{4^{k+1}\pi} \left(\frac{2\ell}{\pi j_0}\right)^{2k+1},$$

and thus we seek to balance $i_0 = j_0^2$ with $\sqrt{(\ell/j_0)^{2k+1}\log n} \cdot \|\Delta^{(k+1)}\beta_0\|_1$. This yields

$$j_0 = \ell^{\frac{2k+1}{2k+5}} (\log n)^{\frac{1}{2k+5}} \|\Delta^{(k+1)}\beta_0\|_1^{\frac{2}{2k+5}},$$

i.e.,

$$i_0 = n^{\frac{2k+1}{2k+5}} (\log n)^{\frac{2}{2k+5}} \|\Delta^{(k+1)}\beta_0\|_1^{\frac{4}{2k+5}},$$

and applying Theorem 6 gives the result for k odd.

Case for k even. For k even, we have the GTF operator being $\Delta = \Delta^{(k+1)} = DL^{k/2}$, where D is the edge incidence matrix of a 2d grid, and $L = D^\top D$. It will be helpful to write

$$D = \begin{bmatrix} I \otimes D_{1d} \\ D_{1d} \otimes I \end{bmatrix},$$

where $D_{1d} \in \mathbb{R}^{(\ell-1)\times\ell}$ is the difference operator for a 1d grid of size $\ell = \sqrt{n}$ (and $I \in \mathbb{R}^{\ell\times\ell}$ is the identity matrix). It suffices to check the incoherence of the left singular vectors of $DL^{k/2}$, since the eigenvalues of $DL^{k/2}$ are those of L raised to the power of $(k+1)/2$, and so the rest of the proof then follows precisely as in the case when k is odd. The left singular vectors of $DL^{k/2}$ are the same as the left singular vectors of D, which are the eigenvectors of $DD^\top$. Observe that

$$DD^\top = \begin{bmatrix} I \otimes D_{1d}D_{1d}^\top & D_{1d}^\top \otimes D_{1d} \\ D_{1d} \otimes D_{1d}^\top & D_{1d}D_{1d}^\top \otimes I \end{bmatrix}.$$

Let u_i, $i = 1, \ldots \ell-1$ be the eigenvectors of $D_{1d}D_{1d}^\top$, corresponding to eigenvalues λ_i, $i = 1, \ldots \ell - 1$. Define $v_i = D_{1d}^\top u_i/\sqrt{\lambda_i}$, $i = 1, \ldots \ell-1$, and $e = \mathbb{1}/\sqrt{\ell}$, where $\mathbb{1} = (1, \ldots 1) \in \mathbb{R}^\ell$ is the vector of all 1s. A straightforward calculation verifies that

$$DD^\top \begin{bmatrix} v_i \otimes u_i \\ u_i \otimes v_i \end{bmatrix} = 2\lambda_i \begin{bmatrix} v_i \otimes u_i \\ u_i \otimes v_i \end{bmatrix}, \quad \text{for } i = 1, \ldots \ell-1,$$

$$DD^\top \begin{bmatrix} e \otimes u_i \\ 0 \end{bmatrix} = \lambda_i \begin{bmatrix} e \otimes u_i \\ 0 \end{bmatrix}, \quad \text{for } i = 1, \ldots \ell-1,$$

$$DD^\top \begin{bmatrix} 0 \\ u_i \otimes e \end{bmatrix} = \lambda_i \begin{bmatrix} 0 \\ u_i \otimes e \end{bmatrix}, \quad \text{for } i = 1, \ldots \ell-1.$$

Hence we have derived $3(\ell-1)$ eigenvectors of $DD^\top$. Note that the vectors v_i, $i = 1, \ldots \ell-1$ are actually the eigenvectors of $L_{1d} = D_{1d}^\top D_{1d}$ (corresponding to the $\ell-1$ nonzero eigenvalues), and from our work in the 1d case, recall, both v_i, $i = 1, \ldots \ell-1$ (studied for k odd) and u_i, $i = 1, \ldots \ell-1$ (studied for k even) are unit vectors satisfying the incoherence property with $\mu = \sqrt{2}$. This means that the above eigenvectors are all unit norm, and are also incoherent, with constant $\mu = 2$.

There are $(\ell-1)(\ell-2)$ more eigenvectors of $DD^\top$, as the rank of $DD^\top$ is $n-1 = \ell^2-1$. A somewhat longer but still straightforward calculation verifies that

$$DD^\top \begin{bmatrix} v_i \otimes u_j + v_j \otimes u_i \\ \sqrt{\frac{\lambda_i}{\lambda_j}} u_i \otimes v_j + \sqrt{\frac{\lambda_j}{\lambda_i}} u_j \otimes v_i \end{bmatrix} = (\lambda_i + \lambda_j) \begin{bmatrix} v_i \otimes u_j + v_j \otimes u_i \\ \sqrt{\frac{\lambda_i}{\lambda_j}} u_i \otimes v_j + \sqrt{\frac{\lambda_j}{\lambda_i}} u_j \otimes v_i \end{bmatrix}, \quad \text{for } i < j,$$

$$DD^\top \begin{bmatrix} \sqrt{\frac{\lambda_j}{\lambda_i}} v_i \otimes u_j + \sqrt{\frac{\lambda_i}{\lambda_j}} v_j \otimes u_i \\ u_i \otimes v_j + u_j \otimes v_i \end{bmatrix} = (\lambda_i + \lambda_j) \begin{bmatrix} \sqrt{\frac{\lambda_j}{\lambda_i}} v_i \otimes u_j + \sqrt{\frac{\lambda_i}{\lambda_j}} v_j \otimes u_i \\ u_i \otimes v_j + u_j \otimes v_i \end{bmatrix}, \quad \text{for } i < j.$$

Modulo the appropriate normalization, we have derived the remaining $(\ell-1)(\ell-2)$ eigenvectors of $DD^\top$. It remains to check their incoherence, once we have normalized them (to have unit norm). As the eigenvectors in the first and second expressions above are simply (block) rearrangements of each other, it does not matter which form we study; consider, say, those in the second expression, and fix $i < j$. The entrywise absolute maximum of the eigenvector in question is at most $\sqrt{\lambda_j/\lambda_i}(4/\sqrt{n})$. Thus it suffices show that the normalization constant for this eigenvector is on the order of $\sqrt{\lambda_j/\lambda_i}$. Observe that

$$\left\| \begin{bmatrix} \sqrt{\frac{\lambda_j}{\lambda_i}} v_i \otimes u_j + \sqrt{\frac{\lambda_i}{\lambda_j}} v_j \otimes u_i \\ u_i \otimes v_j + u_j \otimes v_i \end{bmatrix} \right\|_2^2 = \left\| \begin{bmatrix} \sqrt{\frac{\lambda_j}{\lambda_i}} v_i \otimes u_j \\ u_i \otimes v_j \end{bmatrix} \right\|_2^2 + \left\| \begin{bmatrix} \sqrt{\frac{\lambda_i}{\lambda_j}} v_j \otimes u_i \\ u_j \otimes v_i \end{bmatrix} \right\|_2^2 .$$

Here the cross-term is $(v_i^\top \otimes u_j^\top)(v_j \otimes u_i) = (v_i^\top v_j)(u_i^\top u_j) = 0$, as $v_i^\top v_j = 0$ and $u_i^\top u_j = 0$. This means that the normalization constant lies within the interval $[\sqrt{\lambda_j/\lambda_i + 2}, \sqrt{2\lambda_j/\lambda_i + 2}]$. In particular, the lower bound shows that the incoherence property holds with $\mu = 4$. This completes the proof.

B.9 Proof of Lemma 9

As before, we follow the proof of Theorem 3 up until the application of Holder's inequality in (19), but we use the fractional bound in (16) instead. We claim that this implies

$$\epsilon^\top P_R(\tilde{\beta} - \beta_0) \le \tilde{K} \|\tilde{\beta} - \beta_0\|_R^{1-w/2} (\|\Delta\tilde{\beta}\|_1 + \|\Delta\beta_0\|_1)^{w/2},$$

where $\tilde{K} = O_{\mathbb{P}}(K)$. This is verified by noting that $x = P_R(\tilde{\beta} - \beta_0)/(\|\Delta\tilde{\beta}\|_1 + \|\Delta\beta_0\|_1) \in \mathcal{S}_\Delta(1)$, applying (16) to x, and then rearranging. Therefore, as in the proof of Theorem 3, we have

$$\|\tilde{\beta} - \beta_0\|_R^2 \le 2\tilde{K} \|\tilde{\beta} - \beta_0\|_R^{1-w/2} (\|\Delta\tilde{\beta}\|_1 + \|\Delta\beta_0\|_1)^{w/2} + 2\lambda(\|\Delta\beta_0\|_1 - \|\Delta\tilde{\beta}\|_1), \tag{23}$$

We now set

$$\lambda = \Theta\left(K^{\frac{2}{1+w/2}} \|\Delta\beta_0\|_1^{-\frac{1-w/2}{1+w/2}} \right),$$

and in the spirit of [43, 24], we proceed to argue in cases.

Case 1. Suppose that $\frac{1}{2}\|\Delta\tilde{\beta}\|_1 \ge \|\Delta\beta_0\|_1$. Then we see that (23) implies

$$0 \le \|\tilde{\beta} - \beta_0\|_R^2 \le \tilde{K} \|\tilde{\beta} - \beta_0\|_R^{1-w/2} \left(\frac{3}{2}\right)^{w/2} \|\Delta\tilde{\beta}\|_1^{w/2} - \lambda \|\Delta\tilde{\beta}\|_1, \tag{24}$$

so that

$$\lambda \|\Delta\tilde{\beta}\|_1 \le \tilde{K} \|\tilde{\beta} - \beta_0\|_R^{1-w/2} \|\Delta\tilde{\beta}\|_1^{w/2},$$

where for simplicity have absorbed a constant factor $2(3/2)^{w/2}$ into $\tilde{K}$ (since this does not change the fact that $\tilde{K} = O_{\mathbb{P}}(K)$), and thus

$$\|\Delta\tilde{\beta}\|_1 \le \left(\frac{\tilde{K}}{\lambda}\right)^{\frac{1}{1-w/2}} \|\tilde{\beta} - \beta_0\|_R.$$

Plugging this back into (24) gives

$$\|\tilde{\beta} - \beta_0\|_R^2 \le \tilde{K} \|\tilde{\beta} - \beta_0\|_R^{1-w/2} \left(\frac{\tilde{K}}{\lambda}\right)^{\frac{w/2}{1-w/2}} \|\tilde{\beta} - \beta_0\|_R^{w/2},$$

or

$$\|\tilde{\beta} - \beta_0\|_R \le \tilde{K}^{\frac{1}{1+w/2}} \left(\frac{1}{\lambda}\right)^{\frac{w/2}{1-w/2}} = O_{\mathbb{P}}\left(K^{\frac{1}{1+w/2}} \|\Delta\beta_0\|_1^{\frac{w/2}{1+w/2}} \right),$$

as desired.

Case 2. Suppose that $\frac{1}{2}\|\Delta\tilde{\beta}\|_1 \leq \|\Delta\beta_0\|_1$. Then from (23),

$$\|\tilde{\beta}-\beta_0\|_R^2 \leq \underbrace{2\lambda\|\Delta\beta_0\|_1}_{a} + \underbrace{2\tilde{K}\|\tilde{\beta}-\beta_0\|_R^{1-w/2}3^{w/2}\|\Delta\beta_0\|_1^{w/2}}_{b},$$

and hence either $\|\tilde{\beta}-\beta_0\|_R^2 \leq 2a$, or $\|\tilde{\beta}-\beta_0\|_R^2 \leq 2b$, and $a \leq b$. The first subcase is straightforward and leads to

$$\|\tilde{\beta}-\beta_0\|_R \leq 2\sqrt{\lambda\|\Delta\beta_0\|_1} = O_{\mathbb{P}}\left(K^{\frac{1}{1+w/2}}\|\Delta\beta_0\|_1^{\frac{w/2}{1+w/2}}\right),$$

as desired. In the second subcase, we have by assumption

$$\|\tilde{\beta}-\beta_0\|_R^2 \leq 2\tilde{K}\|\tilde{\beta}-\beta_0\|_R^{1-w/2}\|\Delta\beta_0\|_1^{w/2}, \tag{25}$$

$$2\lambda\|\Delta\beta_0\|_1 \leq \tilde{K}\|\tilde{\beta}-\beta_0\|_R^{1-w/2}\|\Delta\beta_0\|_1^{w/2}, \tag{26}$$

where again we have absorbed a constant factor $2(3^{w/2})$ into $\tilde{K}$. Working from (26), we derive

$$\|\Delta\beta_0\|_1 \leq \left(\frac{\tilde{K}}{2\lambda}\right)^{\frac{1}{1-w/2}}\|\tilde{\beta}-\beta_0\|_R,$$

and plugging this back into (25), we see

$$\|\tilde{\beta}-\beta_0\|_R^2 \leq 2\tilde{K}\|\tilde{\beta}-\beta_0\|_R^{1-w/2}\left(\frac{\tilde{K}}{2\lambda}\right)^{\frac{w/2}{1-w/2}}\|\tilde{\beta}-\beta_0\|_R^{w/2},$$

and finally

$$\|\tilde{\beta}-\beta_0\|_R \leq 2\tilde{K}^{\frac{1}{1+w/2}}\left(\frac{1}{\lambda}\right)^{\frac{w/2}{1-w/2}} = O_{\mathbb{P}}\left(K^{\frac{1}{1+w/2}}\|\Delta\beta_0\|_1^{\frac{w/2}{1+w/2}}\right).$$

This completes the second case, and the proof.

B.10 Proof of Theorem 10

The proof follows closely from Lemma 3.5 of [43]. However, this author uses a different problem scaling than ours, so some care must be taken in applying the lemma. First we abbreviate $\mathcal{S} = \mathcal{S}_\Delta(1)$, and define $\tilde{\mathcal{S}} = \mathcal{S}\cdot\sqrt{n}/M$, where recall M is the maximum column norm of $\Delta^\dagger$. Now it is not hard to check that

$$\mathcal{S} = \{x \in \mathrm{row}(\Delta) : \|\Delta x\|_1 \leq 1\} = \Delta^\dagger\{\alpha : \|\alpha\|_1 \leq 1\},$$

so that $\max_{x\in\mathcal{S}}\|x\|_2 \leq M$, and $\max_{x\in\tilde{\mathcal{S}}}\|x\|_2 \leq \sqrt{n}$. This is important because Lemma 3.5 of [43] concerns a form of "local" entropy that allows for deviations on the order of $\sqrt{n}$ in the norm $\|\cdot\|_2$, or equivalently, constant order in the scaled metric $\|\cdot\|_n = \|\cdot\|_2/\sqrt{n}$. Hence, the entropy bound in (17) translates into

$$\log N(\delta, \tilde{\mathcal{S}}, \|\cdot\|_2) \leq E\left(\frac{\sqrt{n}}{M}\right)^w\left(\frac{\sqrt{n}}{\delta}\right)^w,$$

that is,

$$\log N(\delta, \tilde{\mathcal{S}}, \|\cdot\|_n) \leq E\left(\frac{\sqrt{n}}{M}\right)^w\delta^{-w}.$$

Now we apply Lemma 3.5 of [43]: in the scaled metric used by this author,

$$\max_{x\in\tilde{\mathcal{S}}}\frac{\epsilon^\top x}{\sqrt{n}\|x\|_n^{1-w/2}} = O_{\mathbb{P}}\left(\sqrt{E}\left(\frac{\sqrt{n}}{M}\right)^{w/2}\right),$$

that is,

$$\max_{x \in \tilde{\mathcal{S}}} \frac{\epsilon^\top x}{\|x\|_2^{1-w/2}} = O_{\mathbb{P}}\left(\sqrt{E}\big(\sqrt{n}\big)^{w/2}\Big(\frac{\sqrt{n}}{M}\Big)^{w/2}\right),$$

and finally,

$$\max_{x \in \mathcal{S}} \frac{\epsilon^\top x}{\|x\|_2^{1-w/2}} = O_{\mathbb{P}}\left(\sqrt{E}\big(\sqrt{n}\big)^{w/2}\right),$$

as desired.

B.11 Proof of Corollary 11

For each $j = 1, \ldots 2r$, if $\mathcal{G}$ is covered by j balls having radius at most $C_0\sqrt{n}j^{-1/\zeta}$, with respect to the norm $\|\cdot\|_2$, then it is covered by j balls having radius at most $C_0 j^{-1/\zeta}$, with respect to the scaled norm $\|\cdot\|_n = \|\cdot\|_2/\sqrt{n}$. By Theorem 1 of [7], this implies that for each $j = 1, 2, 3, \ldots$, the convex hull $\mathrm{conv}(\mathcal{G})$ is covered by 2^j balls having radius at most $C_0' j^{-(1/2+1/\zeta)}$, with respect to $\|\cdot\|_n$, for another constant C_0'. Converting this back to an entropy bound in our original metric, and noting that $\mathrm{conv}(\mathcal{G}) = \mathcal{S}_\Delta(1)$, we have

$$\log(\delta, \mathcal{S}_\Delta(1), \|\cdot\|_2) \le C_0''\Big(\frac{\sqrt{n}}{\delta}\Big)^{\frac{1}{1/2+1/\zeta}},$$

for a constant C_0'', as needed. This proves the lemma.

B.12 Proof of Corollary 12

According to Lemma 13, we know that $(D^{(1)})^\dagger = P_{\mathbb{1}}^\perp H$, where H is an $n \times (n-1)$ lower triangular matrix with $H_{ij} = 1$ if $i > j$ and 0 otherwise, and $P_{\mathbb{1}}^\perp$ is the projection map orthogonal to the all 1s vector. Thus $g_i = P_{\mathbb{1}}^\perp h_i$, $i = 1, \ldots n-1$, with $h_1, \ldots h_{n-1}$ denoting the columns of H. It is immediately apparent that

$$\|g_i - g_\ell\|_2 \le \|h_i - h_\ell\|_2 \le \sqrt{i - \ell},$$

for all $i > \ell$. Now, given $2j$ balls at our disposal, consider centering the first j balls at

$$g_d, g_{2d}, \ldots g_{jd},$$

where $d = \lfloor n/j \rfloor$. Also let these balls have radius $\sqrt{n/j}$. By construction, then, we see that

$$\|g_1 - g_d\|_2 \le \sqrt{n/j},\ \|g_d - g_{2d}\|_2 \le \sqrt{n/j},\ \ldots \|g_{jd} - g_{n-1}\|_2 \le \sqrt{n/j},$$

which means that we have covered $g_1, \ldots g_{n-1}$ with j balls of radius $\sqrt{n/j}$. We can cover $-g_1, \ldots - g_{n-1}$ with the remaining j balls analogously. Therefore, we have shown that $2j$ balls require a radius of $\sqrt{n/j}$, or in other words, j balls require a radius of $\sqrt{2n/j}$.

References

[1] A. Barbero and S. Sra. Fast Newton-type methods for total variation regularization. In *International Conference on Machine Learning (ICML-11)*, volume 28, pages 313–320, 2011.

[2] A. Barbero and S. Sra. Modular proximal optimization for multidimensional total-variation regularization. arXiv: 1411.0589, 2014.

[3] D. P. Bertsekas. Projected Newton methods for optimization problems with simple constraints. *SIAM Journal on Control and Optimization*, 20(2):221–246, 1982.

[4] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(9): 1124–1137, 2004.

[5] K. Bredies, K. Kunisch, and T. Pock. Total generalized variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526, 2010.

[6] P. Buhlmann and S. van de Geer. *Statistics for High-Dimensional Data*. Springer, Berlin, 2011.

[7] B. Carl. Metric entropy of convex hulls in Hilbert spaces. *Bulletin of the London Mathematical Society*, 29(04):452–458, 1997.

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

[9] F. Chung and M. Radcliffe. On the spectra of general random graphs. *The Electronic Journal of Combinatorics*, 18(1), 2011.

[10] R. Coiman and M. Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21 (2):53–94, 2006.

[11] S. Conte and C. de Boor. *Elementary Numerical Analysis: An Algorithmic Approach*. McGraw-Hill, New York, 1980. International Series in Pure and Applied Mathematics.

[12] D. L. Donoho and I. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, 1995.

[13] H. Doraiswamy, N. Ferreira, T. Damoulas, J. Freire, and C. Silva. Using topological analysis to support event-guided exploration in urban data. *Visualization and Computer Graphics, IEEE Transactions on*, PP(99), 2014.

[14] U. Feige and E. Ofek. Spectral techniques applied to sparse random graphs. *Random Structures & Algorithms*, 27(2):251–275, 2005.

[15] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.

[16] S. K. Godunov and V. S. Ryabenkii. *Difference Schemes: An Introduction to the Underlying Theory*. Elsevier, Amsterdam, 1987. Number 19 in Studies in Mathematics and Its Applications.

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

[18] J. Irion. *Multiscale Transformations for Signals on Graphs: Methods and Applications*. PhD thesis, Department of Mathematical Sciences, University of California at Davis, 2015.

[19] J. Kelner, L. Orecchia, A. Sidford, and Z. A. Zhu. A simple, combinatorial algorithm for solving SDD systems in nearly-linear time. *Proceedings of the ACM Annual Symposium on Theory of Computing (STOC-13)*, 45:911–920, 2013.

[20] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky. ℓ_1 trend filtering. *SIAM Review*, 51(2):339–360, 2009.

[21] R. Kondor and J. D. Lafferty. Diffusion kernels on graphs and other discrete structures. In *International Conference on Machine Learning (ICML-02)*, pages 315–322, San Francisco, CA, 2002. Morgan Kaufmann.

[22] I. Koutis, G. Miller, and R. Peng. A nearly-$m \log n$ time solver for SDD linear systems. *Proceedings of the IEEE Annual Symposium on Foundations of Computer Science (FOCS-11)*, 52:590–598, 2011.

[23] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[24] E. Mammen and S. van de Geer. Locally adaptive regression splines. *Annals of Statistics*, 25(1): 387–413, 1997.

[25] A. W. Marcus, D. A. Spielman, and N. Srivastava. Ramanujan graphs and the solution of the Kadison-Singer problem. arXiv: 1408.4421, 2014.

[26] J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[27] A. Ramdas and R. J. Tibshirani. Fast and flexible admm algorithms for trend filtering. *Journal of*

Computational and Graphical Statistics, 2015.

[28] L. I. Rudin, S. Osher, and E. Faterni. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60:259–268, 1992.

[29] S. Setzer, G. Steidl, and T. Teuber. Infimal convolution regularizations with discrete l1-type functionals. *Communications in Mathematical Science*, 9(3):797–827, 2011.

[30] J. Sharpnack, A. Rinaldo, and A. Singh. Sparsistency via the edge lasso. *International Conference on Artificial Intelligence and Statistics (AISTATS-12)*, 15:1028–1036, 2012.

[31] J. Sharpnack, A. Singh, and A. Krishnamurthy. Detecting activations over graphs using spanning tree wavelet bases. In *International Conference on Artificial Intelligence and Statistics (AISTATS-13)*, volume 16, pages 536–544, 2013.

[32] J. Sharpnack, A. Singh, and A. Rinaldo. Changepoint detection over graphs with the spectral scan statistic. In *International Conference on Artificial Intelligence and Statistics (AISTATS-13)*, volume 16, pages 545–553, 2013.

[33] D. Shuman, S. Narang, P. Frossard, A. Ortega, and P. 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.

[34] A. J. Smola and R. Kondor. Kernels and regularization on graphs. In *Conf. Computational Learning Theory*, pages 144–158, 2003.

[35] D. Spielman and S.-H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. *Proceedings of the ACM Annual Symposium on Theory of Computing (STOC-04)*, 36:81–90, 2004.

[36] G. Steidl, S. Didas, and J. Neumann. Splines in higher order TV regularization. *International Journal of Computer Vision*, 70(3):214–255, 2006.

[37] P. P. Talukdar and K. Crammer. New regularized algorithms for transductive learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 442–457. Springer, 2009.

[38] P. P. Talukdar and F. Pereira. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1473–1481. Association for Computational Linguistics, 2010.

[39] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B*, 67(1):91–108, 2005.

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

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

[42] R. J. Tibshirani and J. Taylor. Degrees of freedom in lasso problems. *Annals of Statistics*, 40(2): 1198–1232, 2012.

[43] S. van de Geer. Estimating a regression function. *Annals of Statistics*, 18(2):907—924, 1990.

[44] S. van de Geer and J. Lederer. The lasso, correlated design, and improved oracle inequalities. *IMS Collecions*, 9:303–316, 2013.

[45] N. Vishnoi. $Lx = b$: Laplacian solvers and their algorithmic applications. *Foundations and Trends in Theoretical Computer Science*, 8(1–2):1–141, 2012.

[46] Y.-X. Wang, A. Smola, and R. J. Tibshirani. The falling factorial basis and its statistical properties. In *International Conference on Machine Learning (ICML-14)*, volume 31, 2014.