Trend Filtering on Graphs

Yu-Xiang Wang[1] **James Sharpnack**[3] **Alex Smola**[1,4] **Ryan J. Tibshirani**[1,2]
[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] Google, Strategic Technologies, Mountain View, CA 94303

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 [11, 26], 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 examples and theory.

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 [24, 11, 26, 29], 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 [22], which enforce smoothness globally with a much heavier hand, and tend 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 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., [21] study such an approach using wavelets; likewise, kernel methods regularize in terms of the eigenfunctions of the graph Laplacian [12]. An advantage of analysis regularization is that it easily yields complex extensions of the basic estimator by mixing penalties.

A Motivating Example. Consider an estimation problem on 402 census tracts of Allegheny County, PA, arranged into a graph with 402 vertices and 2382 edges 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. (This was generated from a mixture of Gaussians in the underlying spatial coordinates.) We drew noisy observations around this signal, shown in the top right panel, 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

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

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

Above, $y \in \mathbb{R}^n$ is the vector of observations across nodes, $n = 402$, $L \in \mathbb{R}^{n \times n}$ is the unnormalized Laplacian matrix over the graph, and $W \in \mathbb{R}^{n \times n}$ is a wavelet basis built over the graph (we followed the prescription of [21]). The three estimators each have their own regularization parameters λ; hence as a common measure for the complexities of the fitted models, we use degrees of freedom (df).

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38.

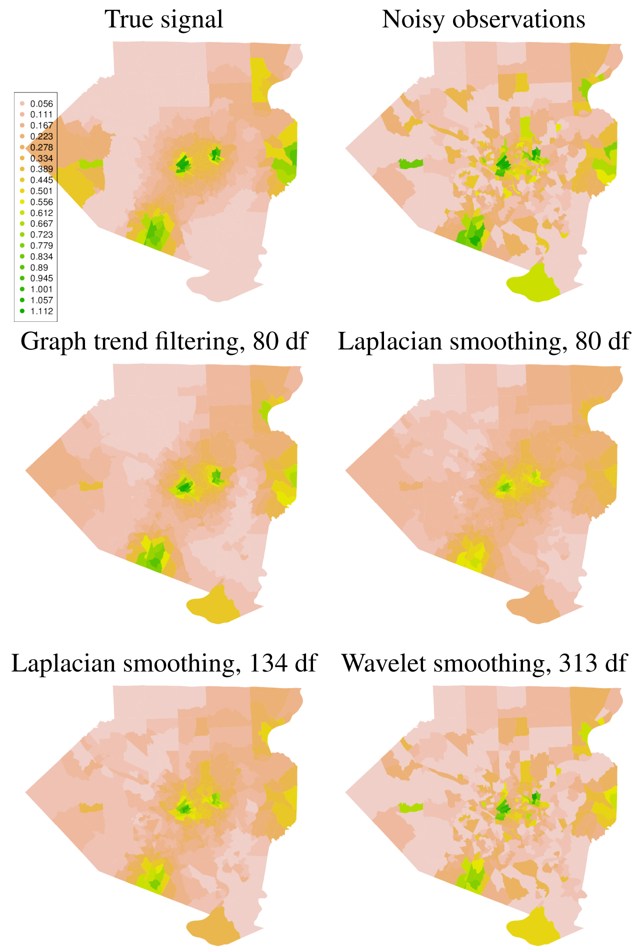


Figure 1: Color maps for the Allegheny County example.

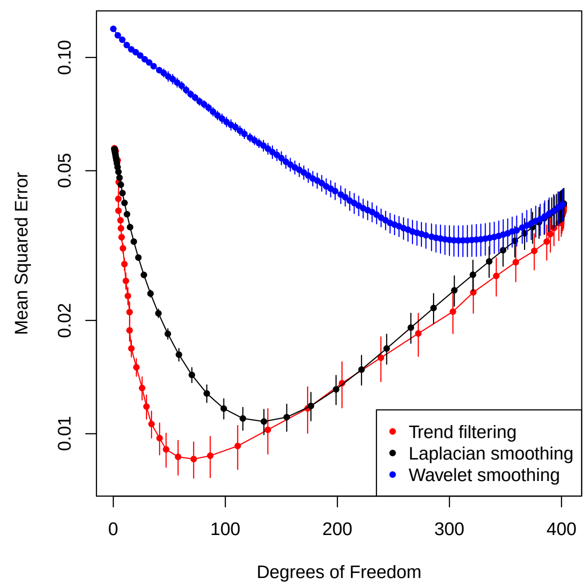


Figure 2: Mean squared errors for the Allegheny County example. Results were averaged over 10 simulations; the bars denote ± 1 standard errors.

The middle left panel of Figure 1 shows the graph trend filtering estimate with 80 df. We see that it adaptively fits to the sharp peaks in the center of the graph, and smooths out the surrounding regions appropriately. The graph Laplacian estimate with 80 df (middle right), substantially oversmooths the high peaks in the center, while at 134 df (bottom left), 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.

Furthermore, Figure 2 shows the mean squared errors between the estimates and the true signal. The differences in performance here are analogous to the univariate case, when comparing trend filtering to smoothing splines [26]. At the 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.

Outline. Section 2 defines graph trend filtering and covers basic properties. Section 3 examines computational approaches, Section 4 looks at more examples, and Section 5 presents theory. Section 6 concludes with a discussion.

Notation. For $X \in \mathbb{R}^{m \times n}$, we write X_A to extract the rows of X corresponding to a subset $A \subseteq \{1, \dots m\}$, and X_{-A} to extract the complementary rows. Similarly for vectors. 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. Here discrete difference operators play a central role. Suppose that we observe $y = (y_1, \dots y_n) \in \mathbb{R}^n$ across equally spaced input locations $x = (x_1, \dots x_n)$; for simplicity, 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,

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

hence $\|D^{(1)}\beta\|_1 = \sum_{i=1}^{n-1} |\beta_{i+1} - \beta_i|$, and the 0th order trend filtering estimate in (1) reduces to the 1-dimensional fused lasso estimator [25], also called 1-dimensional total variation denoising [17]. For $k \geq 1$ we define $D^{(k+1)}$ 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), i.e. $D^{(k+1)}$ 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 $\hat{\beta}_1, \ldots \hat{\beta}_n$ exhibit the form of a kth order piecewise polynomial function, evaluated over the input locations $x_1, \ldots x_n$. This can be formally verified [26, 29] by examining a continuous-space analog of (1).

2.2 Trend Filtering over Graphs

Let $G = (V, E)$ be an graph, with vertices $V = \{1, \ldots n\}$ and undirected edges $E = \{e_1, \ldots e_m\}$, and suppose that we observe $y = (y_1, \ldots y_n)$ over the nodes. Following the univariate definition in (1), we define the kth order *graph trend filtering* (GTF) estimate $\hat{\beta} = (\hat{\beta}_1, \ldots \hat{\beta}_n)$ 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 [27], 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:

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

In this case, 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

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

where the sign orientations are arbitrary. By construction, the 0th order graph trend filtering estimate is piecewise constant over nodes of G, and it is identical to the fused lasso estimate on G [9, 27, 19].

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 exploited the fact that $\Delta^{(2)} = (\Delta^{(1)})^\top \Delta^{(1)}$ is the unnormalized graph Laplacian L of G, and we abbreviated $\Delta^{(1)}$ by D. 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.

There may be multiple ways to generalize the univariate discrete difference operators (2), (3) to graphs, so why this particular definition? Intuition surrounding (5), (6) can be developed by considering piecewise polynomial signals over graphs; due to a lack of space, we defer this discussion to the supplementary document. Another important reasurrance is that our graph definitions (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.

2.3 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 [22]. 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

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

The above penalty term can be written as $\|L^{\frac{k+1}{2}}\beta\|_2^2$ for odd k, and $\|DL^{\frac{k}{2}}\beta\|_2^2$ for even k; i.e., the penalty in (7) 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 [26], and the high-level conclusion is that GTF can adapt to the proper local degree of smoothness, while Laplacian smoothing cannot.

2.4 Related Work

Some authors from the signal processing community, e.g., [4, 18], 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 this work 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 5 may extend to these methods too.

2.5 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 is straightforward and we omit it. 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 $(L^\dagger)^{\frac{k}{2}} = I$) that $\hat{\beta}$ will indeed be piecewise constant over groups of nodes $C_1, \ldots C_s$ of G. When $k = 2, 4, \ldots$, this structure is smoothed by multiplying such piecewise constant levels by $(L^\dagger)^{\frac{k}{2}}$. 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 smoothing through multiplication by $(L^\dagger)^{\frac{k+1}{2}}$. Both of these smoothing operations, which depend on $L^\dagger$, have interesting interpretations in terms of to the electrical network perspective for graphs; see the supplement.

Second, Lemma 1 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 [27, 28], 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 $\mathrm{nullity}(M)$ the dimension of the null space of a matrix M). 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*

$$\mathrm{df}(\hat{\beta}) = \begin{cases} \mathbb{E}\left[\max\{|A|, 1\}\right] & \textit{odd } k \\ \mathbb{E}\left[\textit{no. of connected components of } G_{-A}\right] & \textit{even } k. \end{cases}$$

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.

2.6 Extensions

The GTF problem in (4) lies in the analysis framework, wherein the estimate is defined through direct regularization via an analyzing operator (penalty term) $\|\Delta^{(k+1)}\beta\|_1$. A nice feature of this framework is that we can easily alter or extend the GTF estimator by adding other penalty terms. For example, by adding a pure ℓ_1 penalty on β itself, we arrive at *sparse graph trend filtering*,

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^n}{\mathrm{argmin}} \; \frac{1}{2}\|y - \beta\|_2^2 + \lambda_1\|\Delta^{(k+1)}\beta\|_1 + \lambda_2\|\beta\|_1, \quad (8)$$

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 scenarios 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. We give an example of sparse GTF in Section 4. Aside from this particular extension, many others are possible, e.g., by changing the loss to suit a classification problem, mixing graph difference penalties of various orders, or tying together several denoising tasks with a group penalty.

3 COMPUTATION

Graph trend filtering is defined by a convex optimization problem (4), and 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 large-scale problems, we describe two specialized algorithms that improve on generic procedures by taking advantage of the structure of $\Delta^{(k+1)}$.

3.1 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 [16] 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 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

$$\beta \leftarrow (I+\rho L^{2q})^{-1}b, \tag{9}$$
$$z \leftarrow \operatorname*{argmin}_{x\in\mathbb{R}^n} \frac{1}{2}\|b-x\|_2^2+\frac{\lambda}{\rho}\|Sx\|_1, \tag{10}$$

for some b. The linear system in (9) 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 ρ (augmented Lagrangian parameter), the system (9) is diagonally dominant, and thus we can solve it in almost linear time using a special Laplacian/SDD solver [23, 13, 10].

The update in (10) is soft-thresholding when $S=I$, and when $S=D$ it is given by graph TV denoising, i.e., the graph fused lasso. For the graph TV denoising problem, we rely on a direct solver based on parametric max-flow [6]. In fact, this algorithm solves (4) directly when $k=0$, and is much faster empirically than its worst case complexity [3].

3.2 A Fast Newton Method

As an alternative to ADMM, the projected Newton method [2, 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 via $\hat{\beta}=y-(\Delta^{(k+1)})^\top\hat{v}$. (For univariate trend filtering, [11] adopt a similar strategy, but instead use an interior point method.) Projected Newton method takes update steps using a reduced Hessian, so abbreviating $\Delta=\Delta^{(k+1)}$, each iteration boils down to

$$v \leftarrow a+(\Delta_I^\top)^\dagger b, \tag{11}$$

for some a,b and set of indices I. The linear system in (11) is always sparse, but conditioning becomes an issue as k grows (note: the same problem does not exist in (9) because of the addition of the identity matrix I). We have found empirically that a preconditioned conjugate gradient method works quite well for (11) for $k=1$, but for larger k it can struggle due to poor conditioning.

3.3 Computation Summary

In our experience with practical experiments, the following algorithms work best for the various graph trend orders k.

Order	Algorithm
$k=0$	Parametric max-flow [6]
$k=1$	Projected Newton method [2, 1]
$k=2,4,\dots$	ADMM with parametric max-flow
$k=3,5,\dots$	ADMM with soft-thresholding

Figure 3 demonstrates that the projected Newton method converges faster than ADMM (superlinear versus at best linear convergence), so when its updates can be performed efficiently ($k=1$), it is preferred. The figure also shows that the special ADMM algorithm (with max-flow) converges faster than the naive one (with soft-thresholding), so when applicable ($k=2$), it is preferred. We remark that orders the $k=0,1,2$ are of most practical interest, so we do not often run naive ADMM with soft-thresholding.

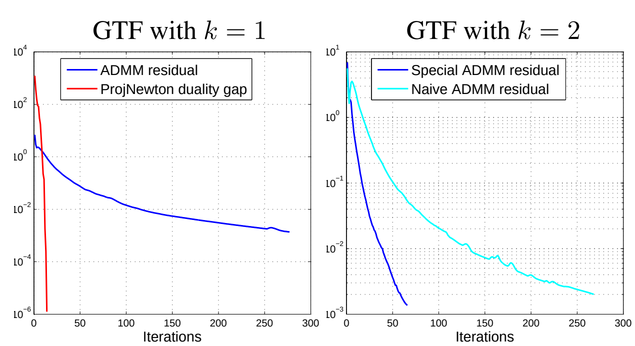


Figure 3: Convergence plots for projected Newton method and ADMM for solving GTF with $k=1$ and $k=2$. The algorithms are all run on a 2d grid graph (an image) with 16,384 nodes and 32,512 edges. For projected Newton, we plot the duality gap across iterations; for the ADMM routines, we plot the average of the primal and dual residuals in the ADMM framework (which also serves as a valid suboptimality bound).

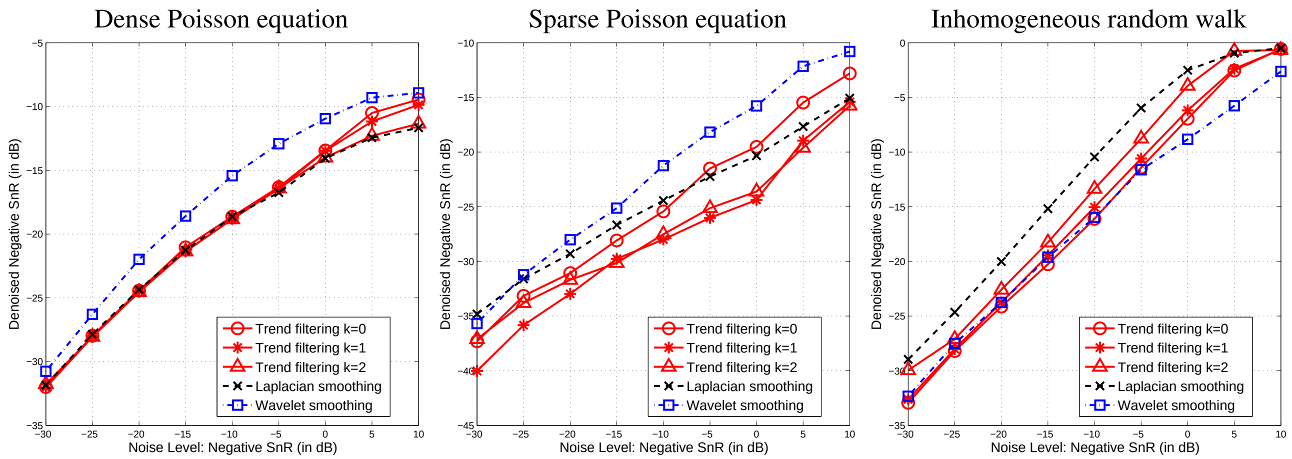


Figure 4: Performance of GTF and others for three generative models on the Facebook graph. The x-axis shows the negative SnR: $10\log_{10}(n\sigma^2/\|x\|_2^2)$, where $n = 4039$, x is the underlying signal, and σ^2 is the noise variance. Hence the noise level is increasing from left to right. The y-axis shows the denoised negative SnR: $10\log_{10}(\mathrm{MSE}/\|x\|_2^2)$, so the achieved MSE is increasing from bottom to top.

4 EXAMPLES

4.1 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 [15] 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. The three ground truth models represent very different scenarios for the underlying signal x, each one favorable to different estimation methods. These are:

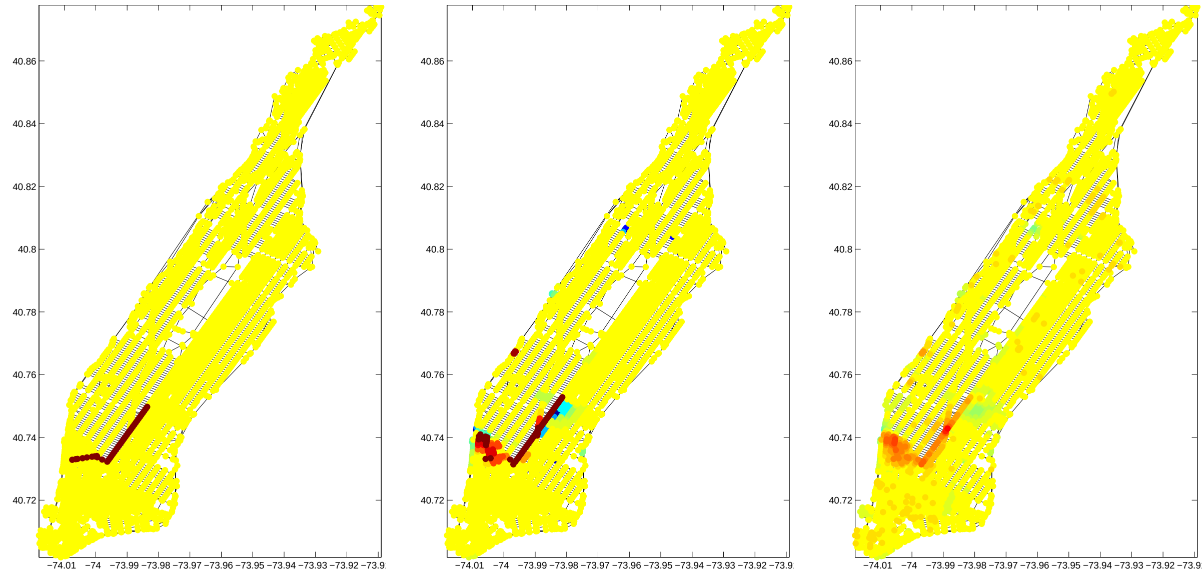
1. **Dense Poisson equation:** we solved the Poisson equation $Lx = b$ for x, where b is arbitrary and dense (its entries were i.i.d. normal draws).
2. **Sparse Poisson equation:** we solved the Poisson equation $Lx = b$ for x, where b is sparse and has 30 nonzero entries (again i.i.d. normal draws).
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 ends at any 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 4 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 is that GTF estimates are (expectedly) superior when the structured sparsity pattern exists (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.

4.2 Event Detection with NYC Taxi Trips Data

To illustrate the sparse graph trend filtering variant of our proposed regularizers, 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 Doraiswamy et al. [7], who obtained the data from NYC Taxi & Limosine Commission. 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.) Specifically, we consider the graph



The true parade route | Sparse trend filtering | Sparse Laplacian smoothing

Figure 5: Comparison of sparse GTF and sparse Laplacian smoothing. In the plots, yellow corresponds to a zero estimate. 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-yellow colors.

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. The 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 5.

In the middle and right panels of Figure 5, we compare sparse GTF (with $k = 0$) and a sparse variant of Laplacian smoothing ($k = 1$), defined by adding an ℓ_1 penalty to its criterion (7), as in (8). For a qualitative visual comparison, the smoothing parameter λ_1 was chosen so that both methods have 200 degrees of freedom (without any sparsity impoved). 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 in this setting, is able to better capture the event of interest.

5 THEORY

In this section we assume that $y \sim \mathcal{N}(\beta_0, \sigma^2 I)$ and derive asymptotic error guarantees for graph trend filtering. (The normal model could be relaxed but is used for simplicity.) Throughout we abbreviate $\Delta = \Delta^{(k+1)}$, and denote by r for the number of rows of Δ ($r = m$ for k even, and $r = n$ for k odd). All proofs are deferred to the supplement.

Using arguments in line with the basic inequality for the lasso [5], we can establish the following bound.

Theorem 3. *Assume that* $\mathrm{null}(\Delta)$ *has constant dimension, and let B denote the maximum ℓ_2 norm of columns of $\Delta^\dagger$. Then for $\lambda = \Theta(B\sqrt{\log r})$, the estimate $\hat\beta$ in* (4) *satisfies*

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

When the true signal is bounded under the GTF operator, $\|\Delta\beta_0\|_1 = O(1)$, the theorem says that the average squared error of GTF converges at the rate $B\sqrt{\log r}/n$, in probability. Theorem 3 is quite general, as it applies to trend filtering on any graph; indeed, it covers any generalized lasso problem, since Δ is treated as an arbitrary linear operator. One might therefore think that it cannot yield sharp rates. Still, as we show next, it does imply consistency in certain cases.

Corollary 4. *Consider the trend filtering estimate $\hat{\beta}$ of order k, with a choice of λ as in Theorem 3. Then:*

1. *for univariate trend filtering (essentially, GTF on a chain), $\|\hat{\beta}-\beta_0\|_2^2/n = O_{\mathbb{P}}(\sqrt{\log n/n}\cdot n^k\|\Delta\beta_0\|_1)$;*
2. *for GTF on an Erdos-Renyi random graph, with edge probability p, and expected degree $d = np \geq 1$, $\|\hat{\beta}-\beta_0\|_2^2/n = O_{\mathbb{P}}(\sqrt{\log(nd)}/(nd^{\frac{k+1}{2}})\cdot\|\Delta\beta_0\|_1)$;*
3. *for GTF on a Ramanujan d-regular graph, and $d \geq 1$, $\|\hat{\beta}-\beta_0\|_2^2/n = O_{\mathbb{P}}(\sqrt{\log(nd)}/(nd^{\frac{k+1}{2}})\cdot\|\Delta\beta_0\|_1)$.*

The results for cases 2 and 3 of Corollary 4 are based on the simple bound $B \leq \|\Delta^\dagger\|_2$, the largest singular value of $\Delta^\dagger$. When Δ is the $(k+1)$st order graph difference operator, it is not hard to see that $\|\Delta^\dagger\|_2 \leq 1/\lambda_{\min}(L)^{\frac{k+1}{2}}$, where $\lambda_{\min}(L)$ is the smallest nonzero eigenvalue of the Laplacian L (also known as the Fiedler value [8]). In general, $\lambda_{\min}(L)$ can be very small, leading to a loose error bound; but for the particular graphs in question, it is well-controlled. When $\|\Delta\beta_0\|_1$ is bounded, cases 2 and 3 of the corollary show the GTF estimate to be converging at the rate $\sqrt{\log(nd)}/(nd^{\frac{k+1}{2}})$; surely, as k increases, this rate grows stronger, but so does the assumption that $\|\Delta\beta_0\|_1 = \|\Delta^{(k+1)}\beta_0\|_1$ is bounded.

The rate for case 1 in Corollary 4, on univariate trend filtering, is based on direct calculation of B using specific facts about the univariate operator $\Delta^\dagger$. In the univariate setting, it is natural to assume that $n^k\|\Delta^{(k+1)}\beta_0\|_1$ is bounded; e.g., this happens when β_0 contains the evaluations of a kth order spline function f_0 over $[0,1]$, and $\mathrm{TV}(f_0^{(k)})$ is bounded. Under this assumption, the above corollary yields a convergence rate of $\sqrt{\log n/n}$ for univariate trend filtering. We note that this rate does not depend on k, and it is not tight and can be improved to $n^{-(2k+2)/(2k+3)}$ [26]. The latter rate is optimal for the univariate case, and is proved using more sophisticated metric entropy arguments [14]. Transferring over such entropy calculations to the general graph case is a topic for future work.

Even without metric entropy, the bound in Theorem 3 can be improved by assuming a type of incoherence condition.

Theorem 5. *Let $\xi_1 \leq \ldots \leq \xi_n$ denote the singular values of Δ, ordered to be increasing, and let $\psi_1, \ldots \psi_r$ be the left singular vectors (recall that r is the number of rows of Δ). Assume the incoherence condition:*

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

for some $\mu > 0$. Now let $i_0 \in \{1, \ldots n\}$ with $i_0 \to \infty$, and let $\lambda = \Theta(\mu[\log r/n \sum_{i=i_0+1}^n \xi_i^{-2}]^{1/2})$. Then $\hat{\beta}$ satisfies

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

Again we emphasize that this theorem is general in that it does not assume a priori that Δ is a graph difference operator, and only leverages the properties of Δ through its singular value decomposition. Compared to the basic bound in Theorem 3, the result in Theorem 5 is clearly stronger because it allows us to replace B—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 of the singular vectors of Δ).

It is interesting to note that the functional in Theorem 5, $\sum_{i=i_0+1}^n \xi_i^{-2}$, was also determined to play a leading role in an error bound for a graph Fourier based scan statistic in the hypothesis testing framework [20].

Graphs that are expected to exhibit the incoherence condition will be regular in the sense in that neighborhoods of different vertices look roughly the same. Social networks are likely to have this property for the bulk of their vertices (i.e., with the exception of a small number of high degree nodes). Another particular graph of this type is the regular torus in 2 dimensions with $\ell \times \ell$ vertices. We finish with a corollary regarding this graph.

Corollary 6. *Let G be a regular square $\ell \times \ell$ torus with $n = \ell^2$, and let $k = 1$. Then, with an appropriate choice of λ as in Theorem 5,*

$$\frac{\|\hat{\beta}-\beta_0\|_2^2}{n} = O_{\mathbb{P}}\left(\frac{(\log n)^{2/7}}{n^{4/7}}\cdot\|\Delta\beta_0\|_1\right).$$

6 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 utility of univariate trend filtering in nonparametric regression, as an alternative to smoothing splines and wavelets [26]. 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, and it is conjectured that metric entropy arguments will reveal an even sharper characterization for certain graph models. This and many other extensions, such as a compressed version of GTF, and a multitask version of GTF, 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.

References

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

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

[3] 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.

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

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

[6] 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.

[7] 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):1–1, 2014. ISSN 1077-2626. doi: 10.1109/TVCG.2014.2346449.

[8] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Math. J.*, 23(98):298–305, 1973.

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

[10] J. Kelner, L. Orecchia, A. Sidford, and Z. Zhu. A simple, combinatorial algorithm for solving sdd systems in nearly-linear time. In *Symposium on theory of computing*. ACM, 2013.

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

[12] R. Kondor and J. D. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proc. Intl. Conf. Machine Learning*, pages 315–322, San Francisco, CA, 2002. Morgan Kaufmann.

[13] I. Koutis, G. L. Miller, and R. Peng. A nearly-m log n time solver for sdd linear systems. In *Foundations of Computer Science (FOCS)*, pages 590–598. IEEE, 2011.

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

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

[16] A. Ramdas and R. J. Tibshirani. Fast and flexible admm algorithms for trend filtering. arXiv: 1406.2082, 2014.

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

[18] S. Setzer, G. Steidl, and T. Teuber. Infimal convolution regularizations with discrete l1-type functionals. *Comm. Math. Sci*, 9(3):797–872, 2011.

[19] J. Sharpnack, A. Rinaldo, and A. Singh. Sparsistency via the edge lasso. *International Conference on Artificial Intelligence and Statistics*, 15, 2012.

[20] J. Sharpnack, A. Rinaldo, and A. Singh. Changepoint detection over graphs with the spectral scan statistic. *arXiv preprint arXiv:1206.0773*, 2012.

[21] J. Sharpnack, A. Krishnamurthy, and A. Singh. Detecting activations over graphs using spanning tree wavelet bases. *International Conference on Artifical Intelligence and Statistics*, 16, 2013.

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

[23] D. A. Spielman and S.-H. Teng. Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *arXiv preprint cs/0607105*, 2006.

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

[25] 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.

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

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

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

[29] Y.-X. Wang, A. Smola, and R. J. Tibshirani. The falling factorial basis and its statistical properties. *International Conference on Machine Learning*, 31, 2014.