Bayesian Spectral Graph Denoising with Smoothness Prior

Sam Leone[1†], Xingzhi Sun[2†], Michael Perlmutter[3] and Smita Krishnaswamy[1,2,3,4,5,6,7]
[1]Program for Applied Mathematics, Yale University [2]Department of Computer Science, Yale University
[3]Department of Mathematics, Boise State University [4]Department of Genetics, Yale School of Medicine [5]Wu Tsai Institute, Yale University
[6]FAIR, Meta AI [7]Computational Biology and Bioinformatics Program, Yale University [†]Equal Contribution

2024 58th Annual Conference on Information Sciences and Systems (CISS) | 979-8-3503-6929-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/CISS59072.2024.10480177

***Abstract*—Here we consider the problem of denoising features associated to complex data, modeled as signals on a graph, via a smoothness prior. This is motivated in part by settings such as single-cell RNA where the data is very high-dimensional, but its structure can be captured via an affinity graph. This allows us to utilize ideas from graph signal processing. In particular, we present algorithms for the cases where the signal is perturbed by Gaussian noise, dropout, and uniformly distributed noise. The signals are assumed to follow a prior distribution defined in the frequency domain which favors signals which are smooth across the edges of the graph. By pairing this prior distribution with our three models of noise generation, we propose *Maximum A Posteriori* (M.A.P.) estimates of the true signal in the presence of noisy data and provide algorithms for computing the M.A.P. Finally, we demonstrate the algorithms' ability to effectively restore signals from white noise on image data and from severe dropout in single-cell RNA sequence data.**

***Index Terms*—denoising, graph signal processing, estimation**

I. Introduction

Signals defined on modern large-scale, irregularly structured data sets are often corrupted by large amounts of noise such as measurement error or missing measurements. This motivates one to estimate the most likely *true, uncorrupted* values of the signal based on both the noisy observations and their prior beliefs about the signal, which often takes the form of a smoothness assumption. We shall present an approach for producing such *Maximum A Posteriori* (M.A.P.) estimates which utilizes tools from spectral graph theory.

Our method is motivated by the explosion in recent decades of complex high-dimensional data, and associated signals, with very high noise levels. Such data may explicitly reside on a graph, e.g., social, energy, transportation, sensor, or neuronal networks [1], or it may implicitly have relationships between entities from which a graph can be built, for example, physical and biological systems, text, image, time series [2]–[6]

With the graph (either existing or built from data), we can treat features as signals (functions) on the graph, and apply methods in graph signal processing, especially spectral graph theory. Typically, a well-behaved signal defined on the vertices will take similar values at vertices that are more connected. This leads us to the prior that many functions of interest will be *smooth* on the graph, where the concept of smoothness can be quantified using tools from spectral graph theory and the eigendecomposition of the graph Laplacian. This intuition motivates the following approach. First, we assume a priori that the signal of interest is likely "fairly smooth" on the graph. Then, we model the noise of the observations. Finally, we produce an estimate of the true signal with the highest likelihood based on our prior beliefs and the observed measurements. Importantly, we note that the assumption that the signal is smooth is very mild and we do not assume that the signal (or the data on which it is defined) has any specific form. We provide details on how to implement this approach under several noise models and then demonstrate the effectiveness of our method on real-world and synthetic data sets. We also note that our method fills the gap of theoretical guarantees in the popular method MAGIC [4], which it outperforms due to the specific modeling of noise types.

II. Background & Related Work

A. An example for high-dimensional data

We first motivate our method via an example of denoising features associated to complex high-dimensional data. Single-cell RNA sequence (scRNA-seq) provides high resolution information about gene expression and is of great interest in molecular biology, clinical studies, and biotechnology [7], [8]. scRNA-seq data is high-dimensional, as it measures the expression of tens of thousands of genes on up to millions of cells [9], and suffers from high noise levels due to multiple sources. Reducing this noise is a crucial step, which is needed prior to downstream analysis [10].

In single-cell RNA sequence data, one obtains the gene-expression counts for a variety of genes in each cell. Each cell can then be viewed as a high-dimensional vector (whose i-th coordinate corresponds to the amount of gene i expressed). It is a common practice to turn this data, consisting of high-dimensional vectors (cells), into a graph by placing edges between cells which are close together in high-dimensional space, and viewing the expression of each gene, as a signal (function) defined on the cellular graph [4], [11]–[13].

B. Graph Signal Processing with Bayesian inference

Spectral graph theory concerns itself with the distribution of eigenvalues and eigenvectors of matrices associated with graphs. The set of eigenvalue-eigenvector pairs is known to uncover the geometric and algebraic properties of a graph. This is the observation that drives algorithms like spectral clustering

979-8-3503-6929-8/24/$31.00 ©2024 IEEE

and diffusion maps [6], the main intuition being that low frequency eigenpairs capture low-resolution, key information about a graph's structure.

Graph Signal Processing (GSP) utilizes tools from spectral graph theory to extend the Fourier transform from classical signal processing and time series analysis to the graph setting [1]. In the classical methods, signals can be denoised by mapping the signal to the Fourier domain, reducing the high-frequency component of the function, and inverting the Fourier transform to achieve a "smoother" version of the signal. In much the same way, GSP operates by representing graph signals in a basis eigenvectors for the graph Laplacian (defined below), whose corresponding eigenvalues may be interpreted as (squared) frequencies, and then reducing the high-frequency components. Filtering in this manner has been applied to use cases such as point cloud data, biological imaging, sensor networks, and more [14].

Bayesian inference is a fundamental method of parameter estimation. The typical form of this problem is that there is a random variable x drawn from some prior distribution and another random variable y whose distribution depends on x. The ambition of Bayesian estimation is, given only y, to estimate the underlying value of x using both prior information on x and the interaction between x and y.

Notably, two important, nonstandard aspects of our method are: (1) we do not have any explicit prior on a data on which the signals is defined, but rather directly build the graph from the data (if it does not already exist), and treat it as a deterministic structure; (2) we do not assume the signal has any specific form, but rather use a mild prior of its smoothness on the graph. These distinctions free us from the limitations of Bayesian models caused by model misspecification [15], and make our method generally applicable to the vast range of data sets regardless of the data distributions.

C. MAGIC: Markov Affinity-based Graph Imputation of Cells

MAGIC [4] is a commonly used method for denoising single-cell data. It is based on the idea that the high-dimensional data lies on a low-dimensional manifold, represented by a nearest neighbor graph. After building the graph, it uses data diffusion, which is random-walk on the graph to denoise the model. Its has been tremendously successful; however, it lacks a solid theoretical model. Our method fills this gap with GSP and Bayesian inference. Furthermore, by specifying cases of common noise models, we are able to adjust our model accordingly, allowing us to outperform MAGIC in these cases.

D. Notation and Defininitions

Throughout, we shall let $G = (V, E, w)$ denote a weighted, connected, and undirected graph with $|V| = n$ and $|E| = m$. Without loss of generality, we will assume $V = \{1, \dots, n\}$. We shall refer to functions $\boldsymbol{f} : V \to \mathbb{R}$ as *graph signals*. In a slight abuse of notation, we will not distinguish between $\boldsymbol{f}$ and the vector in $\mathbb{R}^n$ whose a-th entry is $\boldsymbol{f}(a)$, $1 \leq a \leq n$. We shall let $\mathbf{A}$ denote the *weighted* adjacency matrix and let $\boldsymbol{D} = \text{diag}(\mathbf{A1})$ denote the corresponding diagonal degree matrix.

Given $\mathbf{A}$ and $\boldsymbol{D}$, the *combinatorial Laplacian* is defined as $\boldsymbol{L} = \boldsymbol{D} - \mathbf{A}$. Is is well-known that $\boldsymbol{L}$ admits an orthonormal basis of eigenvectors, $\boldsymbol{L}\boldsymbol{\psi}_i = \lambda_i \boldsymbol{\psi}_i$, $1 \leq i \leq n$, where $\boldsymbol{\psi}_1 = \mathbf{1}$ and $0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_n$. It follows that $\boldsymbol{L}$ is a positive semi-definite matrix whose null space is equal to $\mathbf{span}\{1\}$. One may compute that the quadratic form corresponding to $\boldsymbol{L}$ is given by $\boldsymbol{f}^\top \boldsymbol{L}\boldsymbol{f} = \sum_{\{a,b\} \in E} w(a,b)(\boldsymbol{f}(a) - \boldsymbol{f}(b))^2$. Thus, setting $\boldsymbol{f} = \boldsymbol{\psi}_i$, the λ_i are interpreted as (squared) frequencies, representing the rate at which $\boldsymbol{\psi}_i$ oscillates over the graph, and the $\boldsymbol{\psi}_i$ are interpreted as *generalized Fourier modes*, where $\widehat{\boldsymbol{f}}(\lambda_i) = \langle \boldsymbol{f}, \boldsymbol{\psi}_i \rangle$ represents the portion of $\boldsymbol{f}$ at frequency λ_i. Since the $\boldsymbol{\psi}_i$ are an orthonormal basis, we have $\boldsymbol{f} = \sum_{i=1}^n \widehat{\boldsymbol{f}}(\lambda_i)\boldsymbol{\psi}_i$. Therefore, for a real-valued function h, we can define a corresponding filter by $h(\boldsymbol{f}) = \sum_{i=1}^n h(\lambda_i)\widehat{\boldsymbol{f}}(\lambda_i)\boldsymbol{\psi}_i$. We shall let $\boldsymbol{B}$ denote the weighted $m \times n$ incidence matrix, where rows correspond to edges and columns to vertices, whose entries are given by $\boldsymbol{B}(e,a) = -\boldsymbol{B}(e,b) = \sqrt{w(a,b)}$, if $e = (a,b)$ and $\boldsymbol{B}(e,v) = 0$ for all $v \notin \{a,b\}$ [16]. One may verify that the Laplacian can be factored as $\boldsymbol{L} = \boldsymbol{B}^\top \boldsymbol{B}$. (Here, we implicitly assume an arbitrary, but fixed, ordering of each edge (a,b). This arbitrary choice does not affect the identity $\boldsymbol{L} = \boldsymbol{B}^\top \boldsymbol{B}$ nor any of our analysis.)

We shall let $p(\boldsymbol{f})$ denote the probability distribution of a random variable $\boldsymbol{f}$ and shall let $p(\boldsymbol{f}|\boldsymbol{g})$ denote the conditional distribution of $\boldsymbol{f}$ given another random variable $\boldsymbol{g}$. We shall make use of the fact that by Bayes' theorem, $p(\boldsymbol{f}|\boldsymbol{g}) \propto p(\boldsymbol{f})p(\boldsymbol{g}|\boldsymbol{f})$, where $\propto$ denotes proportionality and the implied constant depends on $\boldsymbol{g}$.

III. Methods

Our goal is to estimate an unknown signal $\boldsymbol{f} \in \mathbb{R}^n$ based on an observation $\boldsymbol{g} \in \mathbb{R}^n$, which we interpret as a noisy version of $\boldsymbol{f}$ under various settings. In each case, we will assume that an observed $\boldsymbol{g}$ is obtained from a corruption of a true signal $\boldsymbol{f}$ which lies within a corresponding admissibility class $\Omega_{\boldsymbol{g}}$. We shall then define the *maximum a posteriori* estimate of $\boldsymbol{f}$ to be the most likely value of $\boldsymbol{f}$ based on (i) the fact that $\boldsymbol{g}$ was observed and (ii) our *a priori* beliefs on $\boldsymbol{f}$ discussed in the following subsection.

A. A Prior Distribution Based on Smoothness

We define prior distributions on $\widehat{\boldsymbol{f}}(\lambda_i)$ for $i = 2, \dots, n$, assuming that each $\widehat{\boldsymbol{f}}(\lambda_i)$ follows the probability distribution:

$$p_\kappa(\widehat{\boldsymbol{f}}(\lambda_i)) \propto \exp\left(-\kappa\lambda_i \widehat{\boldsymbol{f}}(\lambda_i)^2\right)$$

where κ is a fixed smoothing parameter. We further assume that the $\widehat{\boldsymbol{f}}(\lambda_i)$ are independent which implies that, for any fixed value of $\widehat{\boldsymbol{f}}(\lambda_1)$, the probability distribution of $\widehat{\boldsymbol{f}}$ satisfies

$$p_\kappa(\widehat{\boldsymbol{f}}) \propto \prod_{i=2}^n \exp\left(-\kappa\lambda_i \widehat{\boldsymbol{f}}(\lambda_i)^2\right) = \exp\left(-\kappa \sum_{i=2}^n \lambda_i \widehat{\boldsymbol{f}}(\lambda_i)^2\right).$$

We then give $\boldsymbol{f}$ the probability distribution defined by taking the inverse GFT of $\widehat{\boldsymbol{f}}$. We note that since the ψ_i are an orthonormal eigenbasis and $\boldsymbol{f} = \sum_{i=1}^{n} \widehat{\boldsymbol{f}}(\lambda_i)\psi_i$, we have

$$p_\kappa(\boldsymbol{f}) \propto \exp\Big(-\kappa \sum_{i=2}^{n} \lambda_i \widehat{\boldsymbol{f}}(\lambda_i)^2 \Big) = \exp\big(-\kappa \boldsymbol{f}^\top \boldsymbol{L} \boldsymbol{f}\big).$$

Therefore, we see that this probability distribution is defined so that the likelihood of $\boldsymbol{f}$ decreases with its variation across the graph and κ acts as a parameter controlling the tolerance towards fluctuation. Notably, we do not assume any prior distribution on $\widehat{\boldsymbol{f}}(\lambda_1)$ (although is some cases $\widehat{\boldsymbol{f}}(\lambda_1)$ will be implicitly constrained by the admissibility class $\Omega_{\boldsymbol{g}}$). Therefore, our maximum a posteriori estimate is simply the most likely value of $\boldsymbol{f}$ based on the $\boldsymbol{g}$ and our prior beliefs about $\widehat{\boldsymbol{f}}(\lambda_2), \ldots, \widehat{\boldsymbol{f}}(\lambda_n)$.

B. Gaussian Noise on the Graph

We first consider the setting where each of the Fourier coefficients is corrupted by Gaussian noise, i.e., $\widehat{\boldsymbol{g}}(\lambda_i) = \widehat{\boldsymbol{f}}(\lambda_i) + z_i$, $2 \le i \le n$, where each $z_i \sim \mathcal{N}(0, \sigma^2)$ is an independent normal random variable. We will further assume that the total noise $\boldsymbol{g} - \boldsymbol{f}$ has zero mean, which motivates us to define the admissibility class $\Omega_{\boldsymbol{g}} = \{\boldsymbol{f} : \widehat{\boldsymbol{f}}(\lambda_1) = \widehat{\boldsymbol{g}}(\lambda_1)\}$. By expanding the conditional and a priori densities and utilizing the fact that for a given $\boldsymbol{g}$, we have $p_\kappa(\boldsymbol{f}|\boldsymbol{g}) \propto p_\kappa(\boldsymbol{f}) p_\kappa(\boldsymbol{g}|\boldsymbol{f})$, one may derive a maximum a posteriori estimate of $\boldsymbol{f}$ given $\boldsymbol{g}$[1].

Theorem 1 (Gaussian Denoising). *Let $\boldsymbol{g}$ be given, and let $\Omega_{\boldsymbol{g}} = \{\boldsymbol{f} : \widehat{\boldsymbol{f}}(\lambda_1) = \widehat{\boldsymbol{g}}(\lambda_1)\}$. As above, assume that $\widehat{\boldsymbol{g}}(\lambda_i) = \widehat{\boldsymbol{f}}(\lambda_i) + z_i, 2 \le i \le n$, $z_i \sim \mathcal{N}(0, \sigma^2)$ and that our prior beliefs on $\boldsymbol{f}$ are as described in Section III-A. Then, the maximum a posteriori likelihood estimate of $\boldsymbol{f}$ given $\boldsymbol{g}$ is,*

$$\boldsymbol{f}_{map} = h(\boldsymbol{g}),$$

where $h(\boldsymbol{g})$ is a filter as described in Section II with $h(\lambda_i) = \frac{1}{1+2\kappa\sigma^2\lambda_i}$. Moreover, $\boldsymbol{f}_{map}$ can be computed, to within ϵ accuracy in the $\boldsymbol{L}$-norm ($\|\boldsymbol{f}\|_{\boldsymbol{L}}^2 = \boldsymbol{f}^\top \boldsymbol{L} \boldsymbol{f}$), in time $\tilde{\mathcal{O}}\big(m \log(\epsilon^{-1}) \min\big\{\sqrt{\log(n)}, \sqrt{\frac{2\kappa\sigma^2\lambda_{\max}+1}{2\kappa\sigma^2\lambda_{\min}+1}}\big\}\big)$.

We note that the minimum in the term describing the time complexity arises from the existence of two possible methods of computation, both of which are algorithms for solving linear systems in an implicit matrix $\boldsymbol{M}$ with a condition number of $\beta = \frac{2\kappa\sigma^2\lambda_{\max}+1}{2\kappa\sigma^2\lambda_{\min}+1}$. When $2\kappa\sigma^2$ is small, β is small and the conjugate gradient algorithm will terminate rapidly. Alternatively, when β is large, one may use the solver from [17] which requires $\tilde{\mathcal{O}}\big(m \log(\epsilon^{-1})\sqrt{\log(n)}\big)$ time.

In practice, σ^2 and κ are generally unknown. However, as the filter depends only on the product $2\kappa\sigma^2$, it suffices to estimate this quantity, which we denote by τ. We propose a method of moments estimator which calculates the expectation of $\boldsymbol{g}^\top \boldsymbol{L} \boldsymbol{g}, \boldsymbol{g}^\top \boldsymbol{L}^2 \boldsymbol{g}$ in terms of σ, κ and backsolves using the empirical values. Alternatively, we may regard τ as a smoothing parameter to be tuned, rather than a quantity needing estimation.

$$\tau \approx \frac{tr(\boldsymbol{L})\boldsymbol{g}^\top \boldsymbol{L} \boldsymbol{g} - (n-1)(\boldsymbol{L}\boldsymbol{g})^\top (\boldsymbol{L}\boldsymbol{g})}{tr(\boldsymbol{L})(\boldsymbol{L}\boldsymbol{g})^\top (\boldsymbol{L}\boldsymbol{g}) - tr(\boldsymbol{L}^2)\boldsymbol{g}^\top \boldsymbol{L} \boldsymbol{g}} \tag{1}$$

Note that, by the handshake lemma, $\text{tr}(\boldsymbol{L}) = \sum_{a \in V} \deg(a) = 2\sum_{(a,b)\in E} w(a,b)$. Furthermore, $\text{tr}(\boldsymbol{L}^2) = \sum_a \big(\deg(a)^2 + \sum_{(a,b)\in E} w(a,b)^2\big)$, and so both of these quantities can be calculated in $\mathcal{O}(m)$ time. Alternatively, we may regard τ as a smoothing parameter to be tuned, rather than a quantity needing estimation. We note that this filter may be viewed as a form of Tikhonov regularization [1].

C. Uniformly Distributed Noise

Next, we consider the case when the noise is a random uniform scaling in the vertex domain: $\boldsymbol{g}(a) = u(a)\boldsymbol{f}(a)$, where each $u(a) \sim \text{Unif}[0,1]$ is an independent uniform random variable. In this case, since $0 \le u(a) \le 1$, we set the admissibility class $\Omega_{\boldsymbol{g}} = \{\boldsymbol{f} : |\boldsymbol{f}(a)| \ge |\boldsymbol{g}(a)|, \text{sign}(\boldsymbol{f}) = \text{sign}(\boldsymbol{g}), \forall a \in V\}$. For such an $\boldsymbol{f} \in \Omega_{\boldsymbol{g}}$, one may compute that the a posteriori likelihood of $\boldsymbol{f} \in \Omega_{\boldsymbol{g}}$ given $\boldsymbol{g}$ is

$$p_\kappa(\boldsymbol{f}|\boldsymbol{g}) = p_\kappa(\boldsymbol{f}) \prod_{a \in V} \frac{1}{|\boldsymbol{f}(a)|}.$$

We will maximize the a posteriori likelihood by minimizing the negative log likelihood, which using basic properties of the logarithm leads us to the optimization problem

$$\min_{\boldsymbol{f} \in \Omega_{\boldsymbol{g}}} \mathcal{L}(\boldsymbol{f}), \quad \mathcal{L}(\boldsymbol{f}) = \kappa \boldsymbol{f}^\top \boldsymbol{L} \boldsymbol{f} + \sum_{a \in V} \log|\boldsymbol{f}(a)|.$$

In order to (approximately) solve this problem, we adopt a constrained Convex-Concave Procedure (CCP) [18] for the above. The CCP operates by splitting a function of the form $f(x) = f_{\text{concave}}(x) + f_{\text{convex}}(x)$ and approximating the concave portion linearly about the current solution; the relaxed problem is convex and can be solved more efficiently. The procedure is repeated until convergence, and it is known to be a descent algorithm. Applied this particular optimization, the CCP update of $\boldsymbol{f}^{t+1}$ from $\boldsymbol{f}^t$ is as follows:

$$\boldsymbol{f}^{t+1} = \arg\min_{\boldsymbol{f} \in \Omega_{\boldsymbol{g}}} \kappa \boldsymbol{f}^\top \boldsymbol{L} \boldsymbol{f} + \sum_{a \in V} \frac{\boldsymbol{f}(a)}{|\boldsymbol{f}^t(a)|}$$

We remark that $\boldsymbol{f}^{t+1}$ can be computed as a quadratic program and that the update provides a descent algorithm - $\mathcal{L}(\boldsymbol{f}^{t+1}) \le \mathcal{L}(\boldsymbol{f}^t)$. This is because the loss function is a quadratic function of $\boldsymbol{f}$ and the feasible region $\Omega_{\boldsymbol{g}}$ is a convex polyhedron.

D. Partial Observations & Bernoulli Dropout

In our final two models, we consider two settings where the noise behaves differently at different vertices. We assume that there is some (possibly unknown) set $S \subseteq V$ where $f(a)$ is exactly equal to $g(a)$ for all $a \in S$. We make no assumption regarding the relationship between $f(a)$ and $g(a)$ for $a \notin S$. This leads us to define the admissibility class

[1]Further details on the derivation of Theorem 1, and all of our other theoretical results, are available at https://arxiv.org/abs/2311.16378

$\Omega_{\boldsymbol{g}} = \{\boldsymbol{f} : \boldsymbol{f}(a) = \boldsymbol{g}(a) \text{ for all } a \in S\}$. We consider two practically useful variations of this problem:

1) Basic Interpolation: The set S is known.
2) Bernoulli Dropout: There is a "set of suspicion" ζ where we are unsure whether $a \in S$ or $a \in S^c$. There is also a (possibly empty) set ζ^c for which the observer is certain of their observations (i.e., we know $\zeta^c \subseteq S$). For each $a \in \zeta$, we assume that a is corrupted (i.e., $a \notin S$) with probability p and that $a \in S$ with probability $1-p$.

In this first scenario, the maximum a posteriori estimate of $\boldsymbol{f}$ is the most likely $\boldsymbol{f}$ that is equal to $\boldsymbol{g}$ over the observation set S: $\boldsymbol{f}_{\text{map}} = \arg\max_{f \in \Omega_{\boldsymbol{g}}} p_\kappa(\boldsymbol{f})$. Because of the monotonicity of the exponential function, this is equivalent to computing $\min_{f \in \Omega_{\boldsymbol{g}}} \boldsymbol{f}^\top \boldsymbol{L}\boldsymbol{f}$. This problem was studied in [19], which proved the following result. Notably, [19] predated the development of efficient solvers which could be used to compute $\boldsymbol{f}_{map}$ as in (2). However, now that such solvers exist [20], one may use them to compute the proposed estimate to accuracy ϵ in $\tilde{\mathcal{O}}(\widehat{m}\sqrt{\log \widehat{n}}\log(\epsilon^{-1}))$ time, where ∂S is the boundary of S, $\widehat{n} = |S^c \cup \partial S|$ and $\widehat{m} = |E(S^c, S^c) \cup E(S^c, S)|$, where $E(S_1, S_2)$ denotes the set of edges going from $S_1 \subseteq V$ to $S_2 \subseteq V$. We also denote $\boldsymbol{f}(A) := (\boldsymbol{f}(a_1), \boldsymbol{f}(a_2), \ldots, \boldsymbol{f}(a_k))$, where $\{a_1, a_2, \ldots, a_k\} = A \subseteq V$; $\boldsymbol{L}(S_1, S_2)$ and $\mathbf{A}(S_1, S_2)$ are the restrictions of $\boldsymbol{L}$ and $\mathbf{A}$ to rows in S_1 and columns in S_2, respectively; $\boldsymbol{B}(:, S_1)$ is the restriction of $\boldsymbol{B}$ to columns in S_1; $\forall S_1, S_2 \subseteq V$.

Theorem 2 (Restated from [19]). *Suppose S has at least one edge going to S^c. Then there exists a unique solution to $\min_{\boldsymbol{f} \in \Omega} \boldsymbol{f}^\top \boldsymbol{L}\boldsymbol{f}$. The interpolation of $\boldsymbol{f}$ to S^c is given by*

$$\boldsymbol{f}_{map}(S^c) = \boldsymbol{L}(S^c, S^c)^{-1}\mathbf{A}(S^c, S)\boldsymbol{g}(S). \tag{2}$$

Now, we consider the second, more challenging scenario where we observe a signal $\boldsymbol{g}$ which is equal to $\boldsymbol{f}$, except in a set of suspicion ζ where, with probability p, $g(a)$ is corrupted (i.e., not equal to $f(a)$). Based on the observation $\boldsymbol{g}$ alone, there is no obvious way to identify the set $S = \{a \in V : \boldsymbol{f}(a) = \boldsymbol{g}(a)\}$ (although we do know $\zeta^c \subseteq S$). However, we note that for $\boldsymbol{g}$ to take a given value, there must be $\|\boldsymbol{f}(\zeta) - \boldsymbol{g}(\zeta)\|_0$ corrupted entries and $|\zeta| - \|\boldsymbol{f}(\zeta) - \boldsymbol{g}(\zeta)\|_0$ uncorrupted entries. Since each entry is corrupted with probability p, we model $p_\kappa(\boldsymbol{g}|\boldsymbol{f}) \propto p^{\|\boldsymbol{f}(\zeta) - \boldsymbol{g}(\zeta)\|_0}(1-p)^{|\zeta| - \|\boldsymbol{f}(\zeta) - \boldsymbol{g}(\zeta)\|_0}$. Thus, for $\boldsymbol{f} \in \Omega_{\boldsymbol{g}}$, the negative log of the posterior likelihood of $\boldsymbol{f}$ can be estimated as:

$$\begin{aligned} -\log p_\kappa(\boldsymbol{f}|\boldsymbol{g}) = \kappa \boldsymbol{f}^\top \boldsymbol{L}\boldsymbol{f} \\ + \|\boldsymbol{f}(\zeta) - \boldsymbol{g}(\zeta)\|_0\big(\log(1-p) - \log(p)\big) \\ + \text{constant}. \end{aligned}$$

Therefore, if we define $\tau = \kappa^{-1}(\log(1-p) - \log(p))$, we observe the MAP is produced by the following minimization problem:

$$\boldsymbol{f}_{\text{map}} \in \min_{\boldsymbol{f} \in \Omega_{\boldsymbol{g}}} \boldsymbol{f}^\top \boldsymbol{L}\boldsymbol{f} + \tau \|\boldsymbol{f}(\zeta) - \boldsymbol{g}(\zeta)\|_0.$$

Note that the sign of τ is going to depend on $\log(1-p) - \log(p) = \log(\frac{1}{p} - 1)$. When $p < 1/2$, then the penalty term τ is positive; otherwise, $\tau < 0$ so we may assume all values have changed and estimate $\boldsymbol{f}$ using Theorem 2. When $p < 1/2$, the penalty term is positive. By breaking up $\boldsymbol{f}$ into $\boldsymbol{f}(\zeta)$ and $\boldsymbol{f}(\zeta^c)$, we may write the optimization as a regression problem:

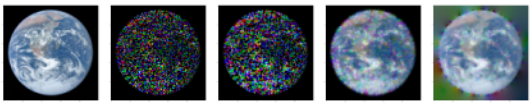

Fig. 1. The results of LASSO regularization along with different parameters of τ. In this case, the region of skepticism is the set of zeroes $\zeta = \{a \in V : \boldsymbol{g}(a) = 0\}$. The leftmost image is ground truth, the second image is the corrupted signal (i.i.d. across each pixel and channel with dropout probability $p = 0.7$). The last three images are Bernoulli-LASSO restorations with $\tau = 10^{-2}, 10^{-3}$, and $\tau = 0$.

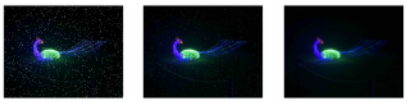

Fig. 2. A no trust algorithm applied to an image. Here, approximately 10% of pixels get corrupted by salt and pepper noise (left). Critically, the algorithm has no explicit knowledge of where. Parameters of $\tau = 10^{-5}, 10^{-6}$ (middle, right) are chosen and paired with LASSO regression.

Theorem 3. *When $p < 1/2$, the $\boldsymbol{f}_{map}$ is the* arg min *of the following sparse regression problem:*

$$\begin{aligned} \boldsymbol{f}_{map}(\zeta) \in \boldsymbol{g}(\zeta) + \arg\min_{\boldsymbol{x}} \big\{\tau\|\boldsymbol{x}\|_0 \\ + \|\boldsymbol{B}(:,\zeta)\boldsymbol{x} - \boldsymbol{B}(:,\zeta^c)\boldsymbol{g}(\zeta^c) + \boldsymbol{B}(:,\zeta)\boldsymbol{g}(\zeta)\|_2^2\big\}. \end{aligned}$$

And when $p \geq 1/2$, the solution is given by,

$$\boldsymbol{f}_{map}(\zeta) = \boldsymbol{L}(\zeta^c, \zeta^c)^{-1}\mathbf{A}(\zeta^c, \zeta)\boldsymbol{g}(\zeta^c).$$

In general, the problem of ℓ_0-regularized regression is NP-Hard [21]. However, numerous approximation methods exist including branch and bound [22] and an ℓ_2-based greedy algorithm [21]. Alternatively, we may consider a relaxed version of the minimization problem in which the ℓ_0 penalty term is replaced with an ℓ_1 penalty term. In this case, the relaxed $\boldsymbol{f}_{\text{map}}$ can be found via LASSO regression [23], for which many efficient algorithms exist.

Finally, we draw special attention to the "no-trust" case where $\zeta = V$, i.e. we are skeptical of all observations. Then the optimization can be written more simply:

$$\boldsymbol{f}_{\text{map}}(\zeta) \in \boldsymbol{g}(\zeta) + \arg\min_{\boldsymbol{x}} \|\boldsymbol{B}\boldsymbol{x} - \boldsymbol{B}\boldsymbol{g}\|_2^2 + \tau\|\boldsymbol{x}\|_0$$

The benefit of the no-trust estimate is that it makes few assumptions about the nature of the noise and does not require the user to come up with ζ.

IV. Experiments & Applications

A. Gaussian Noise on an Image

We first consider 1000 images belonging to the CIFAR-10 data set modeled as signals on 32×32 grid graph G; we use the convention of treating each pixel as a vertex connected to adjacent pixels. Importantly, we note that images *are not* our primary motivation. We include this example primarily

Fig. 3. Best restoration on "Noodle" for the spectral, local averaging, and nuclear norm based models when $\sigma = 50$.

to allow visualization of our method before proceeding to more complex graphs. For a fixed image, we add Gaussian noise with different variances σ^2. We then apply the filter proposed in Theorem 1, and consider the ℓ_2 norm between the restored signal and the ground truth. We compare to two other algorithms: local averaging, and weighted nuclear norm minimization. For the local averaging, we repeatedly set the value of a vertex to be the average of its neighbors for some number of iterations t. Note that this is equivalent to applying the powered diffusion operator [6] (or equivalently the random walk matrix) to the noisy signal g. The nuclear norm minimization based estimate is parameterized by τ, and is given by the solution to $\arg\min_{\boldsymbol{f}} \frac{1}{2}\|\boldsymbol{f} - \boldsymbol{g}\|^2 + \tau\|\boldsymbol{f}\|_\star$, where $\|\boldsymbol{f}\|_\star$ is the nuclear norm of $\boldsymbol{f}$ viewed as a matrix. The penalty τ corresponds to a convex relaxation of a low-rank penalty and is designed with the assumption that noise exists over excess left & right singular vectors of g. For the spectral estimate, we use the method of moments estimate for $2\kappa\sigma^2$ given by Equation 1. We then calculate, for every image, the restored signal using each possible t and τ. The average percent error is provided in the Table I. We see that in the high-noise setting, the spectral denoising algorithm outperforms both local averaging and the nuclear norm estimate as a consequence of our low frequency prior.

TABLE I
TOTAL PERCENT ERROR ($\|\boldsymbol{f}_{true} - \boldsymbol{f}_{map}\|/\|\boldsymbol{f}_{true}\|^2$) FOR EACH COMBINATION OF σ, t, τ.

	$\sigma = 5$	$\sigma = 25$	$\sigma = 50$	$\sigma = 100$
Ours	11.5%	18.5%	**18.4%**	**22.6%**
Avg. (t=1)	9.6%	**14.0** %	22.5%	41.8%
Avg. (t=2)	11.9%	14.2%	19.6%	33.2%
Avg. (t=5)	16.5%	17.3%	19.6%	26.8%
N.N. ($\tau = 1$)	**3.9%**	19.8%	39.7%	79.5%
N.N. ($\tau = 25$)	4.0%	17.8%	37.3%	76.9%
N.N. ($\tau = 50$)	5.5%	16.1%	34.9%	74.3%

B. High Frequency Preservation: Comparison to MAGIC

We revisit MAGIC under our proposed framework. MAGIC takes our prior assumption, that the true signal is likely in the low-frequencies as a fact, rather than a probabilistic statement. It can be interpreted as choosing $h(\lambda)$ in advance to be equal to $h(\lambda) = (1 - \lambda/2)^t$ (where t is a tuned parameter) rather than finding the optimal filter based combining our prior beliefs with the observed signal.

To illustrate the advantages of our method over MAGIC, we conduct a comparison using Bernoulli dropout. We generate a set of $C = 5$ cluster centers in two dimensional space. Around each, we generate $m = 200$ points. We construct

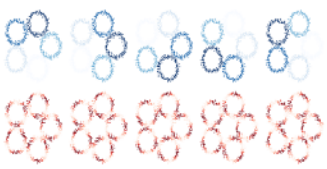

Fig. 4. Top: Low frequency signals that vary over clusters. Bottom: High frequency content that periodically varies inside clusters. Each row contains different phases.

an affinity-based graph with 1000 vertices. We then consider low frequency and high frequency signals. Low frequency signals vary between clusters, while high frequency signals vary within clusters. Finally, we randomly set a proportion p of the observations to zero for different values of p and apply each algorithm. Table 3 examines the resulting correlations between estimated and ground truth signals.

TABLE II
CORRELATIONS BETWEEN THE TRUE & MAP SIGNALS FOR EACH FREQUENCY TYPE, ALGORITHM, AND DROPOUT PROBABILITY p.

	p=0.1	p=0.5	p=0.9	p=0.95
Spectral, Low F.	**1.00**	**1.00**	**0.97**	**0.91**
MAGIC (t=1), Low F	0.59	0.57	0.46	0.38
MAGIC (t=5), Low F.	0.99	0.96	0.75	0.52
MAGIC (t=10), Low F.	0.55	0.53	0.42	0.34
Spectral, High F.	**0.87**	**0.82**	**0.55**	**0.41**
MAGIC (t=1), High F	0.47	0.46	0.38	0.34
MAGIC (t=5), High F.	0.83	0.77	0.51	0.39
MAGIC (t=10), High F.	0.43	0.41	0.35	0.27

Our algorithm consistently outperforms MAGIC, and the effect is most notable for high-frequency signals. This can be explained, at a high level, by the fact that the powered diffusion operator [6] rapidly depresses high-frequency information, which our algorithm is better able to preserve.

C. Denoising Simulated single-cell Data

TABLE III
PERCENTAGE ERROR SIMULATED SINGLE-CELL DATA (LOWER IS BETTER)

	Bernoulli	Uniform
Noisy	50.0%	28.1%
Ours	**7.9%**	**25.5%**
Local Avg	39.9%	29.3%
Low Pass	49.7%	28.8%
High Pass	100.4%	100.3%
MAGIC	40.1%	30.3%

We next apply our method to single-cell RNA sequence (scRNA-seq). scRNA-seq data involves counting mRNA molecules in each cell, which is prone to two types of noise which we test our method's ability to remove:

1) Bernoulli dropout. Because of the small number of the mRNAs molecules in the cell, there can be Bernoulli dropouts when the mRNAs are present but not captured by the experiment equipment [24].

2) Uniform noise. For a given gene, considering the fact that the failure of mRNA capturing does not happen for all the mRNAs, but only for a percentage, we model the noise as uniform - the counts are randomly reduced by a uniformly-distributed percentage [4].

From the data matrix, we build a nearest neighbor graph [4] where each vertex is a cell (modeled as a row vector of gene counts). A column of the matrix (a gene's counts on all the cells) is considered a signal on the graph that we can denoise with our models. By applying the models on all the columns, we obtain a matrix of the denoised data. We compare the denoising performance of our method with four existing methods. On top of the ground truth, we add different types of noise and then compare the performance of our method with MAGIC and other denoising methods: low-pass filter and high-pass band-limit filters defined w.r.t. $\boldsymbol{L}$, and local averaging which is a 1-step random walk on the graph using the row-normalized adjacency matrix. We compute the relative error of the denoised signals with the ground truth. In order to be able to assess the efficacy of our method, we use the bulk gene expression data of *C. elegans* containing 164 worms and 2448 genes [25] to simulate the ground truth single-cell data, because it does not have the zero-inflation as in noisy scRNA-seq data. As shown in Table III we are better able to recover the true signal than the competing methods.

V. Conclusion

We have introduced a method that denoises high-dimensional data by building a graph, treating the features as signals on the graph, and doing M.A.P estimation to recover the true features as denoised signals. We only rely on a mild prior of smoothness on the graph, making our model general and applicable to a vast variety of data modalities. We produce estimators and efficient algorithms for three types of noise common in real data: Gaussian, uniform-scaling, and Bernoulli dropout. Our model outperforms MAGIC and other methods, thanks to the modeling of the noise.

References

[1] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[2] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun, "Graph neural networks: A review of methods and applications," *AI open*, vol. 1, pp. 57–81, 2020.

[3] Kevin R Moon, Jay S Stanley III, Daniel Burkhardt, David van Dijk, Guy Wolf, and Smita Krishnaswamy, "Manifold learning-based methods for analyzing single-cell rna-sequencing data," *Current Opinion in Systems Biology*, vol. 7, pp. 36–46, 2018.

[4] David Van Dijk, Roshan Sharma, Juozas Nainys, Kristina Yim, Pooja Kathail, Ambrose J Carr, Cassandra Burdziak, Kevin R Moon, Christine L Chaffer, Diwakar Pattabiraman, et al., "Recovering gene interactions from single-cell data using data diffusion," *Cell*, vol. 174, no. 3, pp. 716–729, 2018.

[5] Kevin R Moon, David van Dijk, Zheng Wang, Scott Gigante, Daniel B Burkhardt, William S Chen, Kristina Yim, Antonia van den Elzen, Matthew J Hirn, Ronald R Coifman, et al., "Visualizing structure and transitions in high-dimensional biological data," *Nature biotechnology*, vol. 37, no. 12, pp. 1482–1492, 2019.

[6] Ronald R Coifman and Stéphane Lafon, "Diffusion maps," *Applied and computational harmonic analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[7] Antoine-Emmanuel Saliba, Alexander J Westermann, Stanislaw A Gorski, and Jörg Vogel, "Single-cell rna-seq: advances and future challenges," *Nucleic acids research*, vol. 42, no. 14, pp. 8845–8860, 2014.

[8] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang, "Single-cell rna sequencing technologies and bioinformatics pipelines," *Experimental & molecular medicine*, vol. 50, no. 8, pp. 1–14, 2018.

[9] Dragomirka Jovic, Xue Liang, Hua Zeng, Lin Lin, Fengping Xu, and Yonglun Luo, "Single-cell rna sequencing technologies and applications: A brief overview," *Clinical and Translational Medicine*, vol. 12, no. 3, pp. e694, 2022.

[10] Aleksandra A Kolodziejczyk, Jong Kyoung Kim, Valentine Svensson, John C Marioni, and Sarah A Teichmann, "The technology and biology of single-cell rna sequencing," *Molecular cell*, vol. 58, no. 4, pp. 610–620, 2015.

[11] Jacob H Levine, Erin F Simonds, Sean C Bendall, Kara L Davis, D Amir El-ad, Michelle D Tadmor, Oren Litvin, Harris G Fienberg, Astraea Jager, Eli R Zunder, et al., "Data-driven phenotypic dissection of aml reveals progenitor-like cells that correlate with prognosis," *Cell*, vol. 162, no. 1, pp. 184–197, 2015.

[12] Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija, "Integrating single-cell transcriptomic data across different conditions, technologies, and species," *Nature biotechnology*, vol. 36, no. 5, pp. 411–420, 2018.

[13] Juexin Wang, Anjun Ma, Yuzhou Chang, Jianting Gong, Yuexu Jiang, Ren Qi, Cankun Wang, Hongjun Fu, Qin Ma, and Dong Xu, "scgnn is a novel graph neural network framework for single-cell rna-seq analyses," *Nature communications*, vol. 12, no. 1, pp. 1882, 2021.

[14] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[15] Liang Hong and Ryan Martin, "Model misspecification, bayesian versus credibility estimation, and gibbs posteriors," *Scandinavian Actuarial Journal*, vol. 2020, no. 7, pp. 634–649, 2020.

[16] Daniel Spielman, "Spectral graph theory," *Combinatorial scientific computing*, vol. 18, 2012.

[17] Michael B Cohen, Rasmus Kyng, Gary L Miller, Jakub W Pachocki, Richard Peng, Anup B Rao, and Shen Chen Xu, "Solving sdd linear systems in nearly m log1/2 n time," in *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, 2014, pp. 343–352.

[18] Thomas Lipp and Stephen Boyd, "Variations and extension of the convex–concave procedure," *Optimization and Engineering*, vol. 17, pp. 263–287, 2016.

[19] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th International conference on Machine learning (ICML-03)*, 2003, pp. 912–919.

[20] Daniel A Spielman and Shang-Hua Teng, "Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems," in *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, 2004, pp. 81–90.

[21] Balas Kausik Natarajan, "Sparse approximate solutions to linear systems," *SIAM journal on computing*, vol. 24, no. 2, pp. 227–234, 1995.

[22] Hussein Hazimeh, Rahul Mazumder, and Ali Saab, "Sparse regression at scale: Branch-and-bound rooted in first-order optimization," *arXiv preprint arXiv:2004.06152*, 2020.

[23] Robert Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 58, no. 1, pp. 267–288, 1996.

[24] Wei Vivian Li and Jingyi Jessica Li, "An accurate and robust imputation method scimpute for single-cell rna-seq data," *Nature communications*, vol. 9, no. 1, pp. 997, 2018.

[25] Mirko Francesconi and Ben Lehner, "The effects of genetic variation on gene expression dynamics during development," *Nature*, vol. 505, no. 7482, pp. 208–211, 2014.