Runtime Guarantees for Regression Problems *

Hui Han Chin
CMU
hchin@cmu.edu

Aleksander Mądry †
EPFL
aleksander.madry@epfl.ch

Gary L. Miller
CMU
glmiller@cs.cmu.edu

Richard Peng ‡
CMU
yangp@cs.cmu.edu

ABSTRACT

We study theoretical runtime guarantees for a class of optimization problems that occur in a wide variety of inference problems. These problems are motivated by the LASSO framework and have applications in machine learning and computer vision.

Our work shows a close connection between these problems and core questions in algorithmic graph theory. While this connection demonstrates the difficulties of obtaining runtime guarantees, it also suggests an approach of using techniques originally developed for graph algorithms.

We show that most of these problems can be formulated as a grouped least squares problem, and give efficient algorithms for this formulation. Our algorithms rely on routines for solving quadratic minimization problems, which in turn are equivalent to solving linear systems. Some preliminary experimental work on image processing tasks are also presented.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity

Keywords

Image processing, Optimization, Regression

*Partially supported by the National Science Foundation under grant number CCF-1018463 and National Eye Institute (R01-EY01317-08, RO1-EY11289-22).

†Part of this work was done while at MIT and MSR New England

‡Partially supported by Natural Sciences and Engineering Research Council of Canada (NSERC) under grant number D-390928-2010, and by a Microsoft Fellowship. Part of this work was done while at MSR New England.

ITCS'13, January 9–12, 2013, Berkeley, California, USA.

1. INTRODUCTION

The problem of recovering a clear signal from noisy data is an important problem in signal processing. One general approach to this problem is to formulate an objective based on required properties of the answer, and then return its minimizer via optimization algorithms. The power of this method was first demonstrated in image denoising, where the total variation minimization approach by Rudin, Osher and Fatemi [36] had much success. More recent works on sparse recovery led to the theory of compressed sensing [7], which includes approaches such as the least absolute shrinkage and selection operator (LASSO) objective due to Tibshirani [40]. These objective functions have proven to be immensely powerful tools, applicable to problems in signal processing, statistics, and computer vision. In the most general form, given vector $\mathbf{y}$ and a matrix $\mathbf{A}$, one seeks to minimize:

$$\min_{\mathbf{x}} ||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 \qquad (1.1)$$
$$\text{subject to:} \quad |\mathbf{x}|_1 \leq c$$

It can be shown to be equivalent to the following by introducing a Lagrangian multiplier, λ:

$$\min_{\mathbf{x}} ||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 + \lambda|\mathbf{x}|_1 \qquad (1.2)$$

Many of the algorithms used to minimize the LASSO objective in practice are first order methods [33, 3], which updates a sequence of solutions using well-defined vectors related to the gradient. These methods are guaranteed to converge well when the matrix $\mathbf{A}$ is "well-structured". The formal definition of this well-structuredness is closely related to the conditions required by the guarantees given in the compressed sensing literature [7] for the recovery of a sparse signal. As a result, these methods perform very well on problems where theoretical guarantees for solution quality are known. This good performance, combined with the simplicity of implementation, makes these algorithms the method of choice for most problems.

However, LASSO type approaches have also been successfully applied to larger classes of problems. This has in turn led to the use of these algorithms on a much wider variety of problem instances. An important case is image denoising, where works on LASSO-type objectives predates the compressed sensing literature [36]. The matrices involved here are based on the connectivity of the underlying pixel struc-

ture, which is often a $\sqrt{n} \times \sqrt{n}$ square mesh. Even in a unweighted setting, these matrices tend to be ill-conditioned. In addition, the emergence of non-local formulations that can connect arbitrary pairs of vertices in the graph also highlights the need to handle problems that are traditionally considered ill-conditioned. We show in Appendix A that the broadest definition of LASSO problems include well-studied problems from algorithmic graph theory:

FACT 1.1. *Both the s-t shortest path and s-t minimum cut problems in undirected graphs can be solved by minimizing a LASSO objective.*

Although linear time algorithms for unweighted shortest path are known, finding efficient parallel algorithms for this has been a long-standing open problem. The current state of the art parallel algorithm for finding $1 + \epsilon$ approximate solutions, due to Cohen [10], is quite involved. Furthermore, as the reductions done in Lemma A.1 are readily parallelizable, an efficient algorithm for LASSO minimization would also lead to an efficient parallel shortest path algorithm. This suggests that algorithms for minimizing LASSO objectives, where each iteration involve simple, parallelizable operations, are also difficult. Finding a minimum s-t cut with nearly-linear running time is also a long standing open question in algorithm design. In fact, there are known hard instances where many algorithms do exhibit their worst case behavior [20]. The difficulty of these problems and the non-linear nature of the objective are two of the main challenges in obtaining fast run time guarantees for grouped least squares minimization.

Previous run time guarantees for minimizing LASSO objectives rely on general convex optimization routines [6], which take at least $\Omega(n^2)$ time. As the resolution of images are typically at least 256×256, this running time is prohibitive. As a result, when processing image streams or videos in real time, gradient descent or filtering based approaches are typically used due to time constraints, often at the cost of solution quality. The continuing increase in problem instance size, due to higher resolution of streaming videos, or 3D medical images with billions of voxels, makes the study of faster algorithms an increasingly important question.

While the connection between LASSO and graph problems gives us reasons to believe that the difficulty of graph problems also exists in minimizing LASSO objectives, it also suggests that techniques from algorithmic graph theory can be brought to bear. To this end, we draw upon recent developments in algorithms for maximum flow [9] and minimum cost flow [13]. We show that relatively direct modifications of these algorithms allows us to solve a generalization of most LASSO objectives, which we term the **grouped least squares problem**. Our algorithm is similar to convex optimization algorithms in that each iteration of it solves a quadratic minimization problem, which is equivalent to solving a linear system. The speedup over previous algorithms come from the existence of much faster solvers for graph related linear systems [38], although our approaches are also applicable to situations involving other underlying quadratic minimization problems.

The organization of this paper is as follows: In Section 2 we provide a unified optimization problem that encompasses LASSO, fused LASSO, and grouped LASSO. We then discuss known applications of the grouped least squares minimization in Section 3 and other algorithms in Section 4. Section 5 shows an algorithm for approximating grouped least squares based on the maximum flow algorithm of Christiano et al. [9]. Some experimental results demonstrating the practical feasibility of this algorithm are discussed in Section 6. An alternate algorithm with better accuracy, but worse running time dependency on the number of groups is given in Appendix C.

2. BACKGROUND AND FORMULATIONS

The formulation of our main problem is motivated by the total variation objective from image denoising. This objective has its origin in the seminal work by Mumford and Shah [32]. There are two conflicting goals in recovering a smooth image from a noisy one, namely that it must be close to the original image, while having very little noise. The Mumford-Shah function models the second constraint by imposing penalties for neighboring pixels that differ significantly. These terms decrease with the removal of local distortions, offsetting the higher cost of moving further away from the input. However, the minimization of this functional is computationally difficult and subsequent works focused on minimizing functions that are close to it.

The total variation objective is defined for a discrete, pixel representation of the image and measures noise using a smoothness term calculated from differences between neighboring pixels. This objective leads naturally to a graph $G = (V, E)$ corresponding to the image with pixels. The original (noisy) image is given as a vertex labeling $\mathbf{s}$, while the goal of the optimization problem is to recover the 'true' image $\mathbf{x}$, which is another set of vertex labels. The requirement of $\mathbf{x}$ being close to $\mathbf{s}$ is quantified by $||\mathbf{x} - \mathbf{s}||_2^2$, which is the square of the L_2 norm of the vector that's identified as noise. To this is added the smoothness term, which is a sum over absolute values of difference between adjacent pixels' labels:

$$||\mathbf{x} - \mathbf{s}||_2^2 + \sum_{(u,v)\in E} |x_u - x_v| \tag{2.3}$$

This objective can be viewed as an instance of the fused LASSO objective [41]. As the orientation of the underlying pixel grid is artificially imposed by the camera, this method can introduce rotational bias in its output. One way to correct this bias is to group the differences of each pixel with its 4 neighbors, giving terms of the form:

$$\sqrt{(x_u - x_v)^2 + (x_u - x_w)^2} \tag{2.4}$$

where v and w are the horizontal and vertical neighbor of u.

Our generalization of these objectives is based on the key observation that $\sqrt{(x_u - x_v)^2 + (x_u - x_w)^2}$ and $|x_u - x_v|$ are both L_2 norms of vectors consisting of differences of values between adjacent pixels. Each such difference can be viewed as an edge in the underlying graph, and the grouping gives a natural partition of the edges into disjoint sets $S_1 \ldots S_k$:

$$||\mathbf{x} - \mathbf{s}||_2^2 + \sum_{1\leq i\leq k} \sqrt{\sum_{(u,v)\in S_i} (x_u - x_v)^2} \tag{2.5}$$

When each S_i contain a single edge, this formulation is identical to the objective in Equation 2.3 since $\sqrt{(x_u - x_v)^2} =$

$|x_u - x_v|$. To make the first term resemble the other terms in our objective, we will take a square root of it – as we prove in Appendix D, algorithms that give exact minimizers for this variant still captures the original version of the problem. Also, the terms inside the square roots can be written as quadratic positive semi-definite terms involving $\mathbf{x}$. The simplified problem now becomes:

$$||\mathbf{x} - \mathbf{s}||_2 + \sum_{1 \leq i \leq k} \sqrt{\mathbf{x}^T \mathbf{L}_i \mathbf{x}} \tag{2.6}$$

We use $||\cdot||_{\mathbf{L}_i}$ to denote the norm induced by the PSD matrix $\mathbf{L}_i$, and rewrite each of the later terms as $||\mathbf{x}||_{\mathbf{L}_i}$. Fixed labels $\mathbf{s}_1 \ldots \mathbf{s}_k$ an also be introduced for each of the groups, with roles similar to let $\mathbf{s} = \mathbf{s}_0$. As the L_2 norm is equivalent to the norm given by the identify matrix, $||\mathbf{x} - \mathbf{s}_0||_2$ is also a term of the form $||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}$. These generalizations allows us to define our main problem:

DEFINITION 2.1. *The* **grouped least squares problem** *is:*

Input: $n \times n$ *matrices* $\boldsymbol{L}_1 \ldots \boldsymbol{L}_k$ *and fixed values* $\boldsymbol{s}_1 \ldots \boldsymbol{s}_k \in \Re^n$.

Output:

$$\min_{\boldsymbol{x}} \mathcal{OBJ}(\boldsymbol{x}) = \sum_{1 \leq i \leq k} ||\boldsymbol{x} - \boldsymbol{s}_i||_{\boldsymbol{L}_i}$$

Note that this objective allows for the usual definition of LASSO involving terms of the form $|\mathbf{x}_u|$ by having one group for each such variable with $\mathbf{s}_i = \mathbf{0}$. It is also related to group LASSO [44], which incorporates similar assumptions about closer dependencies among some of the terms. To our knowledge grouping has not been studied in conjunction with fused LASSO, although many problems such as the ones listed in Section 3 require this generalization.

2.1 Quadratic Minimization and Solving Linear Systems

Our algorithmic approach to the group least squares problem crucially depends on solving a related quadratic minimization problem. Specifically, we solve linear systems involving a weighted combination of the $\mathbf{L}_i$ matrices. Let $\mathbf{w}_1 \ldots \mathbf{w}_k \in \Re^+$ denote weights, where $\mathbf{w}_i$ is the weight on the ith group. Then the quadratic minimization problem that we consider is:

$$\min_{\mathbf{x}} \mathcal{OBJ}2(\mathbf{x}, \mathbf{w}) = \sum_{1 \leq i \leq k} \frac{1}{\mathbf{w}_i} ||\mathbf{x} - \mathbf{s}_i||^2_{\mathbf{L}_i}$$

We will use OPT2($\mathbf{w}$) to denote the minimum value that is attainable. This minimizer, $\mathbf{x}$, can be obtained using the following Lemma:

LEMMA 2.2. $\mathcal{OBJ}2(\boldsymbol{x}, \boldsymbol{w})$ *is minimized for* $\boldsymbol{x}$ *such that*

$$\left(\sum_{1 \leq i \leq k} \frac{1}{\boldsymbol{w}_i} \boldsymbol{L}_i \right) \boldsymbol{x} = \sum_{1 \leq i \leq k} \frac{1}{\boldsymbol{w}_i} \boldsymbol{s}_i$$

Therefore the quadratic minimization problem reduces to a linear system solve involving $\sum_i \frac{1}{\mathbf{w}_i} \mathbf{L}_i$, or $\sum_i \alpha_i \mathbf{L}_i$ where α is an arbitrary set of positive coefficients. In general, this can be done in $O(n^\omega)$ time where ω is the matrix multiplication constant [39, 11, 42]. When $\mathbf{L}_i$ is symmetric diagonally dominant, which is the case for image applications and most graph problems, these systems can be approximately solved to ϵ accuracy in $\tilde{O}(m \log(1/\epsilon))$ [1] time, where m is the total number of non-zero entries in the matrices [37, 38, 25, 26], and also in $\tilde{O}(m^{1/3+\theta} \log(1/\epsilon))$ parallel depth [4]. There has also been work on extending this type of approach to a wider class of systems [2], with works on systems arising from well-spaced finite-element meshes [5], 2-D trusses [12], and certain types of quadratically coupled flows [21]. For the analysis of our algorithms, we treat this step as a black box with running time $T(n, m)$. Furthermore, to simplify our presentation we assume that the solves return exact answers, as errors can be brought to polynomially small values with an extra $O(\log n)$ overhead. We believe analyses similar to those performed in [9, 21] can be adapted if we use approximate solves instead of exact ones.

3. APPLICATIONS

A variety of problems ranging from computer vision to statistics can be formulated as grouped least squares. We describe some of them below, starting with classical problems from image processing.

3.1 Total Variation Minimization

As mentioned in Section 1, one of the earliest applications of these objectives was in the context of image processing. More commonly known as total variation minimization in this setting [8], various variants of the objective have been proposed with the anisotropic objective the same as Equation 2.3 and the isotropic objective being the one shown in Equation 2.5.

Obtaining a unified algorithm for isotropic and anisotropic TV was one of the main motivations for our work. Our results lead to an algorithm that approximately minimizes both variants in $\tilde{O}(m^{4/3}\epsilon^{-8/3})$ time. It's worth noting that this guarantee does not rely on the underlying structure of the of the graph. This makes the algorithm readily applicable to 3-D images or non-local models involving the addition of edges across the image . However, when the neighborhoods are those of a 2-D image, a $\log n$ factor speedup can be obtained by using the optimal solver for planar systems given in [24].

3.2 Denoising with Multiple Colors

Most works on image denoising deals with images where each pixel is described using a single number corresponding to its intensity. A natural extension would be to colored images, where each pixel has a set of c attributes (in the RGB case, $c = 3$). One possible analogue of $|x_i - x_j|$ in this case would be $||x_i - x_j||_2$, and this modification can be incorporated by replacing a cluster involving a single edge with clusters over the c edges between the corresponding pixels.

This type of approach can be viewed as an instance image reconstruction algorithms using Markov random fields. Instead of labeling each vertex with a single attribute, a

[1] We use $\tilde{O}(f(m))$ to denote $\tilde{O}(f(m) \log^c f(m))$ for some constant c.

set of c attributes are used instead and the correlation between vertices is represented using arbitrary PSD matrices. It's worth remarking that when such matrices have bounded condition number, it was shown in [21] that the resulting least squares problem can still be solved efficiently by preconditioning with SDD matrices, yielding a similar overall running time.

3.3 Poisson Image Editing

The Poisson Image Editing method of Perez, Gangnet and Blake [34] is a very popular method for image blending. This method aims to minimize the difference between the gradient of the image and a guidance field vector $\mathbf{v}$. We show here that the grouped least square problem can be used for minimizing objectives from this framework. The objective function given in equation (6) of [34]

$$\min_{f|\Omega} \sum_{(p,q)\cap\Omega\neq\emptyset} (f_p - f_q - v_{pq})^2, \text{ with } f_p = f_p^* \ \forall p \in \partial\Omega$$

comprises mainly of terms of the form:

$$(x_p - x_q - v_{pq})^2$$

This term can be rewritten as $((x_p - x_q) - (v_{pq} - 0))^2$. So if we let $\mathbf{s}_i$ be the vector where $s_{i,p} = v_{pq}$ and $s_{i,q} = 0$, and $\mathbf{L}_i$ be the graph Laplacian for the edge connecting p and q, then the term equals to $||\mathbf{x} - \mathbf{s}_i||^2_{\mathbf{L}_i}$. The other terms on the boundary will have x_q as a constant, leading to terms of the form $||x_{i,p} - s_{i,p}||_2^2$ where $s_{i,p} = x_q$. Therefore the discrete Poisson problem of minimizing the sum of these squares is an instance of the quadratic minimization problem as described in Section 2.1. Perez et al. in Section 2 of their paper observed that these linear systems are sparse, symmetric and positive definite. We make the additional observation here that the systems involved are also symmetric diagonally dominant. The use of the grouped least squares framework also allows the possibility of augmenting these objectives with additional L_1 or L_2 terms.

3.4 Clustering

Hocking et al. [19] recently studied an approach for clustering points in d dimensional space. Given a set of points $x_1 \dots x_n \in \Re^d$, one method that they proposed is the minimization of the following objective function:

$$\min_{y_1 \dots y_n \in \Re^d} \sum_{i=1}^{n} ||x_i - y_i||_2^2 + \lambda \sum_{ij} w_{ij} ||y_i - y_j||_2$$

Where w_{ij} are weights indicating the association between items i and j. This problem can be viewed in the grouped least squares framework by viewing each x_i and y_i as a list of d variables, giving that the $||x_i - y_i||_2$ and $||y_i - y_j||_2$ terms can be represented using a cluster of d edges. Hocking et al. used the Frank-Wolfe algorithm to minimize a relaxed form of this objective and observed fast behavior in practice. In the grouped least squares framework, this problem is an instance with $O(n^2)$ groups and $O(dn^2)$ edges. Combining with the observation that the underlying quadratic optimization problems can be solved efficiently allows us to obtain an $1+\epsilon$ approximate solution in $\tilde{O}(dn^{8/3}\epsilon^{-8/3})$ time.

4. PREVIOUS ALGORITHMIC RESULTS

Due to the importance of optimization problems motivated by LASSO there has been much work on efficient algorithms for them. We briefly describe some of the previous approaches for LASSO minimization below.

4.1 Second-order Cone Programming

To the best of our knowledge, the only algorithms that provide robust worst-case bounds for the entire class of grouped least squares problems are based on applying tools from convex optimization. In particular, it is known that interior point methods applied to these problems converge in $\tilde{O}(\sqrt{k})$ iterations with each iterations requiring solving a certain linear system [6, 17]. Unfortunately, computing these solutions is computationally expensive – the best previous bound for solving one of these systems is $O(m^\omega)$ where ω is the matrix multiplication exponent. This results in fairly large $O(m^{1/2+\omega})$ total running time and contributes to the popularity of first-order methods described above in practical scenarios. We will revisit this approach in Appendix C and show an improved algorithm for the inner iterations. However, its running time still has a fairly large dependency on k.

4.2 Graph Cuts

For the anisotropic total variation objective shown in Equation 2.3, a minimizer can be found by solving a large number of almost-exact maximum flow calls [14, 22]. Although the number of iterations can be large, these works show that the number of problem instances that a pixel can appear in is small. Combining this reduction with the fastest known exact algorithm for the maximum flow problem by Goldberg and Rao [16] gives an algorithm that runs in $\tilde{O}(m^{3/2})$ time.

It's worth mentioning that both of these algorithms requires extracting the minimum cut in order to construct the problems for subsequent iterations. As a result, it's not clear whether recent advances on fast approximations of maximum flow and minimum s-t cuts [9] can be used as a black box with these algorithms. Extending this approach to the non-linear isotropic objective also appears to be difficult.

4.3 Iterative Reweighted Least Squares

An approach similar to convex optimization methods, but has much better observed rates of convergence is the iterative reweighted least squares (IRLS) method. This method does a much more aggressive adjustment each iteration and to give good performances in practice [43].

4.4 First Order Methods

The method of choice in practice are first order methods such as [33, 3]. Theoretically these methods are known to converge rapidly when the objective function satisfies certain Lipschitz conditions. Many of the more recent works on first order methods focus on lowering the dependency of ϵ under these conditions. As discussed in Section 1 and Appendix A, this direction can be considered orthogonal to our guarantees as the grouped least squares problem is a significantly more general formulation.

5. APPROXIMATE ALGORITHM USING QUADRATIC MINIMIZATIONS

In this section, we show an approximate algorithm for the grouped least squares problem. The analyses of the algorithms is intricate, but is closely based on the approximate minimum cut algorithm given by Christiano et al. [9]. The main modifications that we make are presented in this section, while the full analysis is in Appendix B. A different algorithm based on the lossy generalized flow algorithm given in [13] is presented in Appendix C. It has better error dependencies, but also takes much more time for moderate number of groups. Both of these algorithms can be viewed as reductions to the quadratic minimization problems described in Section 2.1. As a result, they imply efficient algorithms for problems where fast algorithms are known for the corresponding least squares problems.

Recall that the minimum s-t cut problem - equivalent to an L_1-minimization problem - is a special case of the grouped least squares problem where each edge belongs to its own group(i.e., $k = m$). As a result, it's natural to extend the approach of [9] to the whole spectrum of values of k by treating each group as an 'edge'.

One view of the cut algorithm from [9] is that it places a weight on each group, and minimizes a quadratic, or L_2^2 problem involving terms of the from $\frac{1}{\mathbf{w}_i}||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}^2$. Their algorithm then adjusts the weights based on the flow on each edge using the multiplicative weights update framework [1, 29]. This flow is in turn obtained from the dual of the quadratic minimization problem. We simplify this step by showing that the energy of the groups from the quadratic minimization problems can be directly used. Pseudocode of the algorithm is shown in Algorithm 1.

Algorithm 1 Algorithm for the approximate decision problem of whether there exist vertex potentials with objective at most OPT

ApproxGroupedLeastSquares

Input: PSD matrices $\mathbf{L}_1 \dots \mathbf{L}_k$, fixed values $\mathbf{s}_1 \dots \mathbf{s}_k$ for each group. Routine Solve for solving linear systems, width parameter ρ and error bound ϵ.

Output: Vector $\mathbf{x}$ such that $\mathcal{OBJ}(\mathbf{x}) \leq (1 + 10\epsilon)\text{OPT}$.

1: Initialize $\mathbf{w}_i^{(0)} = 1$ for all $1 \leq i \leq k$
2: $N \leftarrow 10\rho \log n\epsilon^{-2}$
3: **for** $t = 1 \dots N$ **do**
4: $\quad \mu^{(t-1)} \leftarrow \sum_i \mathbf{w}_i^{(t-1)}$
5: $\quad$ Use Solve to compute a minimizer for the quadratic minimization problem where $\alpha_i = \frac{1}{\mathbf{w}_i^{(t-1)}}$, let this solution be $\mathbf{x}^{(t)}$
6: $\quad$ Let $\lambda^{(t)} = \sqrt{\mu^{(t-1)}\mathcal{OBJ}2(\mathbf{x}^{(t)})}$
7: $\quad$ Update the weight of each group:
$$\mathbf{w}_i^{(t)} \leftarrow \mathbf{w}_i^{(t-1)} + \left(\frac{\epsilon}{\rho}\frac{||\mathbf{x}^{(t)} - \mathbf{s}_i||_{\mathbf{L}_i}}{\lambda^{(t)}} + \frac{2\epsilon^2}{k\rho}\right)\mu^{(t-1)}$$
8: **end for**
9: $\bar{t} \leftarrow \arg\min_{0 \leq t \leq N} \mathcal{OBJ}(\mathbf{x}^{(t)})$
10: **return** $\mathbf{x}^{(\bar{t})}$

The main difficulty of analyzing this algorithm is that the analysis of minimum s-t cut algorithm of [9] relies strongly on the existence of a solution where $\mathbf{x}$ is either 0 or 1. Our analysis extends this potential function into the fractional setting via. a function based on the Kulback-Liebler (KL) divergence [28]. To our knowledge the use of this potential with multiplicative weights was first introduced by Freund and Schapire [15], and is common in learning theory. This function can be viewed as measuring the KL-divergence between $\mathbf{w}_i^{(t)}$ and $||\bar{\mathbf{x}} - \mathbf{s}_i||_{\mathbf{L}_i}$ over all groups, where $\bar{\mathbf{x}}$ an optimum solution to the grouped least squares problem. This term, which we denote as D_{KL} is:

$$D_{KL} = \sum_i ||\bar{\mathbf{x}} - \mathbf{s}_i||_{\mathbf{L}_i} \log\left(\frac{||\bar{\mathbf{x}} - \mathbf{s}_i||_{\mathbf{L}_i}}{\mathbf{w}_i^{(t)}}\right) \tag{5.7}$$

One way to interpret this function is that $||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}$ and $\frac{1}{\mathbf{w}_i}||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}^2$ are equal when $\mathbf{w}_i = ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}$. Therefore, this algorithm can be viewed as gradually adjusts the weights to become a scaled copy of $||\bar{\mathbf{x}} - \mathbf{s}_i||_{\mathbf{L}_i}$, and D_{KL} serves a way to measure this difference. It can be simplified by subtracting the constant term given by $\sum_i ||\bar{\mathbf{x}} - \mathbf{s}_i||_{\mathbf{L}_i^{(t)}} \log(||\bar{\mathbf{x}} - \mathbf{s}_i||_{\mathbf{L}_i^{(t)}})$ and multiplying by $-1/\text{OPT}$. This gives us our key potential function, $\nu^{(t)}$:

$$\nu^{(t)} = \frac{1}{\text{OPT}} \sum_i ||\bar{\mathbf{x}} - \mathbf{s}_i||_{\mathbf{L}_i} \log(\mathbf{w}_i^{(t)}) \tag{5.8}$$

It's worth noting that in the case of the cut algorithm, this function is identical to the potential function used in [9]. We show the convergence of our algorithm by proving that if the solution produced in some iteration is far from optimal, $\nu^{(t)}$ increases substantially. Upper bounding it with a term related to the sum of weights, $\mu^{(t)}$ allows us to prove convergence. The full proof is given in Appendix B.

To simplify the analysis, we assume that the guess that we're trying to solve the decision problem on, OPT, all entries of $\mathbf{s}$, and spectrum of $\mathbf{L}_i$ are polynomially bounded in n. That is, there exist some constant d such that $-n^d \leq \mathbf{s}_{i,u} \leq n^d$ and $n^{-d}\mathbf{I} \preceq \sum_i \mathbf{L}_i \preceq n^d\mathbf{I}$ where $\mathbf{A} \preceq \mathbf{B}$ means $\mathbf{B} - \mathbf{A}$ is PSD. Some of these assumptions can be relaxed via. analyses similar to Section 2 of [9].

Theorem 5.1. *On input of an instance of $\mathcal{OBJ}$ with edges partitioned into k sets. If all parameters polynomially bounded between n^{-d} and n^d, running* ApproxGroupedLeastSquares *with $\rho = 2k^{1/3}\epsilon^{-2/3}$ returns a solution $\boldsymbol{x}$ with such that $\mathcal{OBJ}(\boldsymbol{x}) \leq \max\{(1+10\epsilon)\,OPT, n^{-d}\}$ where OPT is the value of the optimum solution.*

The additive n^{-d} case is included to deal with the case where OPT = 0, or is close to it. We believe it should be also possible to handle this case by restricting the condition number of $\sum_i \mathbf{L}_i$.

6. EVIDENCE OF PRACTICAL FEASIBILITY

We performed a series of experiments using the approximate algorithm described in Section 5 in order to demonstrate its practical feasibility. Their running times that are slower than the state of the art methods, but nonetheless reasonable. This suggests the need of further experimental works on a more optimized version, which is outside of the scope of this paper.

The SDD linear systems that arise in the quadratic minimization problems were solved using the combinatorial multigrid (CMG) solver [23, 27]. One side observation confirmed by these experiments is that for the sparse SDD linear systems that arise from image processing, the CMG solver yields good results both in accuracy and running time.

6.1 Total Variational Denoising

Total Variational Denoising is the concept of applying Total Variational Minimization as denoising process. This was pioneered by Rudin, Osher and Fatemi [36] and is commonly known as the ROF image model [8]. Our approximate algorithm from Section 5 yields a simple way to solve the ROF model and most of its variants. In Figure 1, we present a simple denoising experiment using the standard image processing data set, 'Lenna'. The main goal of the experiment is to show that our algorithm is competitive in terms of accuracy, while having running times comparable to first-order methods. On a 512×512 grayscale image, we introduce Additive White Gaussian Noise (AWGN) at a measured Signal to Noise Ratio (SNR) of 2. AWGN is the most common noise model in photon capturing sensors from consumer cameras to space satellites systems. We compare the results produced by our algorithm with those by the Split Bregman algorithm from [18] and the Gauss-Seidel variation of the fixed point algorithm from [30]. These methods minimize an objective with L_2^2 fidelity term given in Equation 2.3 while we used the variant with L_2 fidelity shown in Equation 2.6. Also, the parameters in these algorithms were picked to give the best results for the objective functions being minimized. As a result, for measuring the qualities of output images we only use the L_2 and L_1 norms of pixel-wise differences with the original image.

Our experiments were conducted on a single core 64-bit Intel(R) Xeon(R) E5440 CPU @ 2.83GHz. The non-solver portion of the algorithm was implemented in Matlab(R). On images of size 256×256, 512×512 and 1024×1024, the average running times are 2.31, 9.70 and 47.61 seconds respectively. These running times are noticeably slower than the state of the art. However, it's worth noting is that on average 45% of the total running time is from solving the SDD linear systems using the CMG solver. The rest is mostly from reweighting edges and MATLAB function calls, which should be much faster in more optimized versions. More importantly, in all of our experiments the weights are observed to converge in under 15 iterations, even for larger images of size up to 3000×3000.

6.2 Image Processing

As exemplified by the denoising with colors application discussed in Section 3.2, the grouped least squares framework can be used for a wide range of image processing tasks. Some examples of such applications are shown in Figure 2. Our denoising algorithm can be applied as a preprocessing step to segmenting images of the retina obtained from Optical Coherence Tomography (OCT) scans. Here the key is to preserve the sharpness between the nerve fiber layers and this is achieve by using a L_1 regularization term. Variations of this formulation allows one to emulate a large variety of established image preprocessing applications. For example, introducing additional L_2 terms containing differences of neighboring pixels of a patch leads to the removal of boundaries, giving an overall blurring effect. On our examples, this leads to results similar to methods that apply a filter over the image, such as Gaussian blurring. Introducing such effects using an objective function has the advantage that it can be used in conjunction with other terms. By mixing and matching penalty terms on the groups, we can preserve global features while favoring the removal of small artifacts introduced by sensor noise.

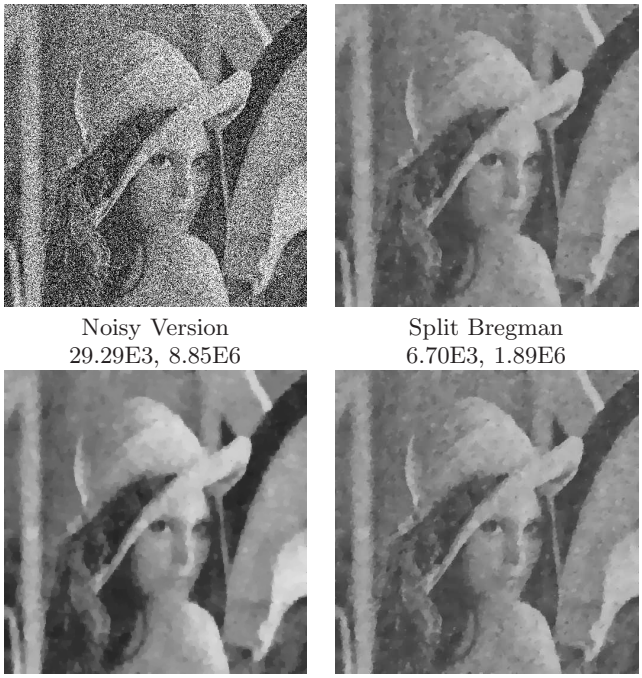

Figure 1: Outputs of various denoising algorithms on image with AWGN noise. Starting form top left in clock-wise order: noisy version, Split Bregman [18], Fixed Point [30], and Grouped Least Squares. Errors listed below each figure from left to right are L_2 and L_1 norms of differences with the original.

Examples of Poisson Image Editing mentioned in Section 3.3 are shown in Figure 3. The application is seamless cloning as described in Section 3 of [34], which aims to insert complex objects into another image. Given two images, they are blended by solving the discrete poisson equation based on a mix of their gradients and boundary values. We also added L_2 constraints on different parts of the image to give a smoother result. Below we show two examples produced by our algorithm with inputs on the left and results on the right. The input consists of locations of the foreground pictures over the background, along with boundaries (shown in red) around the objects in the foreground. These rough boundaries makes the blending of surrounding textures the main challenge, and our three examples (void/sky, sea/pool, snow/sand) are some representative situations. These examples also show that our approaches can be extended to handle multichannel images (RGB or multi-spectral) with only a few modifications.

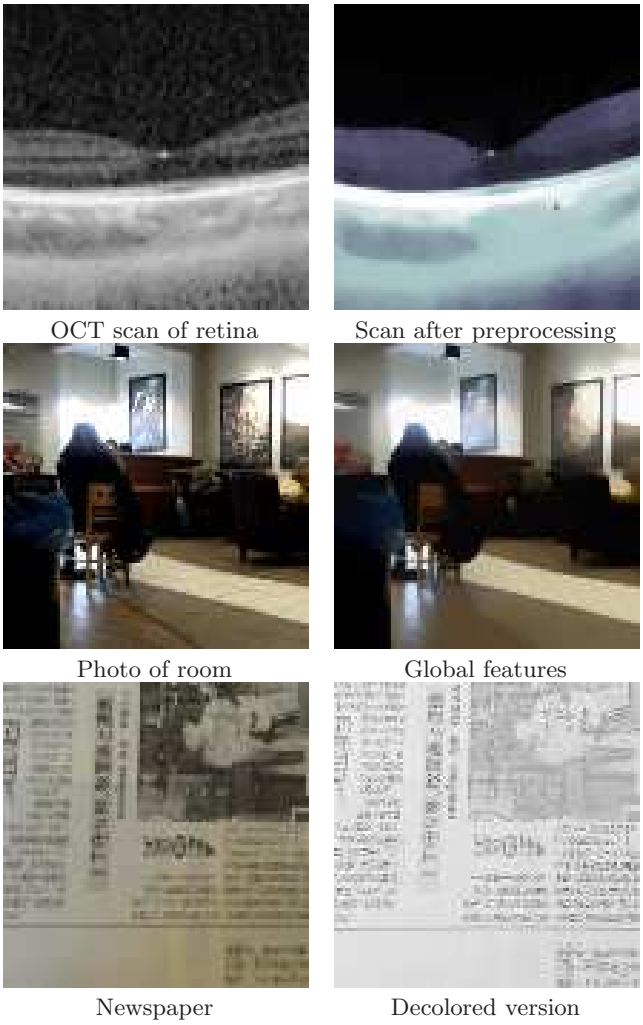

Figure 2: Applications to various image processing tasks. From top to bottom: image segmentation, global feature extraction / blurring, and decoloring.

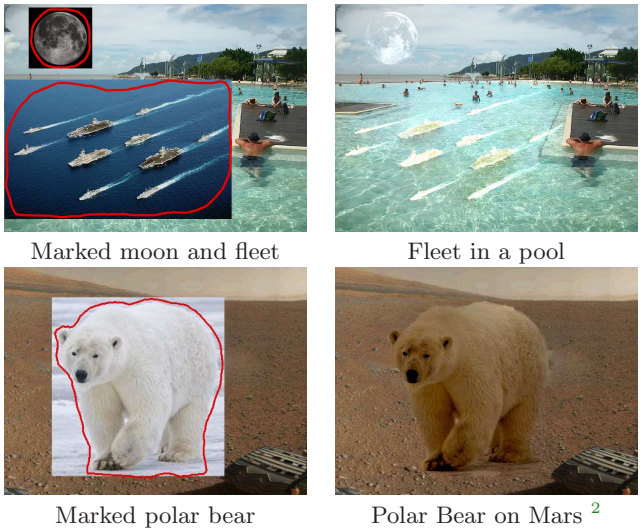

Figure 3: Examples of seamless cloning using Poisson Image Editing

7. REMARKS

We believe that the ability of our algorithm to encompass many of the current image processing algorithms represents a major advantage in practice. It allows the use of a common data structure (the underlying graph) and subroutine (linear system solvers) for many different tasks in the image processing pipeline. Theoretically, the grouped least squares problem is also interesting as it represents an intermediate problem between linear and quadratic optimization.

The performances of our algorithms given in Section 5 and Appendix C depend on k, which is the number of groups in the formulation given in Definition 2.1. Two settings of k are helpful for comparison to previous works. When $k = 1$, the problem becomes the electrical flow problem, and the running time of both algorithms are similar to directly solving the linear system. This is also the case when there is a small (constant) number of groups. The other extremum is when each edge belongs to its own group, aka. $k = m$. Here our approximate algorithm is the same as the minimum s-t cut algorithm given in [9], but our analysis for our almost-exact algorithm gives a much worse running time. This is due to the interior point algorithm generating more complicated linear systems, and occurs when most groups contain a small number of edges. As a result, more work is needed on faster almost-exact algorithms for problems with intermediate values of k. One other consequence of this dependency on k is that although the problem with smaller number of groups is no longer captured by linear optimization, the minimum s-t cut problem – that still falls within the framework of linear optimization – is in some sense the hardest problem in this class. Therefore we believe that the grouped least squares problem is a natural interpolation between the L_1 and L_2^2 optimization, and has potential to be used as a subroutine in other algorithms.

The preliminary experimental results from Section 6 show that more aggressive reweightings of edges lead to much faster convergence than what we showed for our two algorithms. Although the running time from these experiments are slower than state of the art methods, we believe the results suggest that more thorough experimental studies with better tuned algorithms are needed. Also, the Mumford-Shah functional can be better approximated by non-convex functions [32]. Objectives as hinged loss often lead to better results in practice [35], but few algorithmic guarantees are known for them. Designing algorithms with strong guarantees for minimizing these objectives is an interesting direction for future work.

Acknowledgements

The authors would like to thank Jerome Darbon, Stanley Osher, Aarti Singh and Ryan Tibshirani for pointing them to works that are relevant to the grouped least squares framework, and also an anonymous reviewer of a previous submis-

[2]Image of Mars courtesy of NASA/JPL-Caltech/MSSS

sion of this paper for pointing out an alternate view of the proof of Theorem 5.1.

8. REFERENCES

[1] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta algorithm and applications. Technical report, Princeton University, 2005. 5

[2] H. Avron, G. Shklarski, and S. Toledo. On element SDD approximability. *CoRR*, abs/0911.0547, 2009. 2.1

[3] S. R. Becker, E. J. Candes, and M. Grant. Templates for convex cone problems with applications to sparse signal recovery. *MATHEMATICAL PROGRAMMING COMPUTATION*, 3:165, 2011. 1, 4.4

[4] G. E. Blelloch, A. Gupta, I. Koutis, G. L. Miller, R. Peng, and K. Tangwongsan. Near linear-work parallel SDD solvers, low-diameter decomposition, and low-stretch subgraphs. In *Proceedings of the 23rd ACM symposium on Parallelism in algorithms and architectures*, SPAA '11, pages 13–22, New York, NY, USA, 2011. ACM. 2.1

[5] E. G. Boman, B. Hendrickson, and S. A. Vavasis. Solving elliptic finite element systems in near-linear time with support preconditioners. *CoRR*, cs.NA/0407022, 2004. 2.1

[6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Camebridge University Press, 2004. 1, 4.1, C, C.1

[7] J. Candes, E. Compressive sampling. *Proceedings of the International Congress of Mathematicians*, 2006. 1, 1

[8] T. Chan and J. Shen. *Image Processing And Analysis: Variational, Pde, Wavelet, And Stochastic Methods*. SIAM, 2005. 3.1, 6.1

[9] P. Christiano, J. A. Kelner, A. Mądry, D. Spielman, and S.-H. Teng. Electrical Flows, Laplacian Systems, and Faster Approximation of Maximum Flow in Undirected Graphs. In *Proceedings of the* 43^{rd} *ACM Symposium on Theory of Computing (STOC)*, 2011. 1, 2.1, 4.2, 5, 5, 5, 7, B, B

[10] E. Cohen. Polylog-time and near-linear work approximation scheme for undirected shortest paths. *J. ACM*, 47(1):132–166, 2000. 1

[11] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetical progressions. *J. Symbolic Computation*, 9:251–280, 1990. 2.1

[12] S. I. Daitch and D. A. Spielman. Support-graph preconditioners for 2-dimensional trusses. *CoRR*, abs/cs/0703119, 2007. 2.1

[13] S. I. Daitch and D. A. Spielman. Faster approximate lossy generalized flow via interior point algorithms. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC '08, pages 451–460, New York, NY, USA, 2008. ACM. 1, 5, A, C

[14] J. Darbon and M. Sigelle. Image restoration with discrete constrained total variation part i: Fast and exact optimization. *Journal of Mathematical Imaging and Vision*, 26(3):261–276, 2006. 4.2

[15] Y. Freund and R. E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, October 1999. 5

[16] A. V. Goldberg and S. Rao. Beyond the flow decomposition barrier. *J. ACM*, 45:783–797, September 1998. 4.2

[17] D. Goldfarb and W. Yin. Second-order cone programming methods for total variation-based image restoration. *SIAM J. Sci. Comput*, 27:622–645, 2004. 4.1, C

[18] T. Goldstein and S. Osher. The split bregman method for l1-regularized problems. *SIAM J. Img. Sci.*, 2:323–343, April 2009. 6.1, 1

[19] T. Hocking, J.-P. Vert, F. Bach, and A. Joulin. Clusterpath: an algorithm for clustering using convex fusion penalties. In L. Getoor and T. Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 745–752, New York, NY, USA, June 2011. ACM. 3.4

[20] D. S. Johnson and C. C. McGeoch, editors. *Network Flows and Matching: First DIMACS Implementation Challenge*. American Mathematical Society, Boston, MA, USA, 1993. 1

[21] J. A. Kelner, G. L. Miller, and R. Peng. Faster approximate multicommodity flow using quadratically coupled flows. In *Proceedings of the 44th symposium on Theory of Computing*, STOC '12, pages 1–18, New York, NY, USA, 2012. ACM. 2.1, 3.2

[22] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:65–81, 2004. 4.2

[23] I. Koutis and G. Miller. The combinatorial multigrid solver. Conference Talk, March 2009. 6

[24] I. Koutis and G. L. Miller. A linear work, $O(n^{1/6})$ time, parallel algorithm for solving planar Laplacians. In *Proc. 18th ACM-SIAM Symposium on Discrete Algorithms (SODA 2007)*, 2007. 3.1

[25] I. Koutis, G. L. Miller, and R. Peng. Approaching optimality for solving sdd linear systems. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 235–244, Washington, DC, USA, 2010. IEEE Computer Society. 2.1

[26] I. Koutis, G. L. Miller, and R. Peng. A nearly-m log n time solver for sdd linear systems. In *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 590–598, Washington, DC, USA, 2011. IEEE Computer Society. 2.1

[27] I. Koutis, G. L. Miller, and D. Tolliver. Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing. In *International Symposium of Visual Computing*, pages 1067–1078, 2009. 6

[28] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 1951. 5

[29] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, Feb. 1994. 5

[30] C. A. Micchelli, L. Shen, and Y. Xu. Proximity algorithms for image models: denoising. *Inverse Problems*, 27(4):045009, 2011. 6.1, 1

[31] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, Jan. 1987. A

[32] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989. 2, 7

[33] Y. NESTEROV. Gradient methods for minimizing composite objective function. CORE Discussion Papers 2007076, UniversitÃl' catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2007. 1, 4.4

[34] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics (SIGGRAPH'03)*, 22(3):313–318, 2003. 3.3, 6.2

[35] L. Rosasco, E. De, V. A. Caponnetto, M. Piana, and A. Verri. Are loss functions all the same. *Neural Computation*, 15, 2004. 7

[36] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithm. *Physica D*, 1(60):259–268, 1992. 1, 1, 6.1

[37] D. A. Spielman and S.-H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 81–90, June 2004. 2.1

[38] D. A. Spielman and S.-H. Teng. Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *CoRR*, abs/cs/0607105, 2006. 1, 2.1

[39] V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13:354–356, 1969. 2.1

[40] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996. 1

[41] 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. 2

[42] V. Vassilevska Williams. Breaking the Coppersmith-Winograd barrier. In *Proceedings of the 44th symposium on Theory of Computing*, STOC '12, 2012. 2.1

[43] B. Wohlberg and P. Rodriguez. An iteratively reweighted norm algorithm for minimization of total variation functionals. *Signal Processing Letters, IEEE*, 14(12):948 –951, dec. 2007. 4.3

[44] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006. 2

APPENDIX

A. PROOFS ABOUT GRAPH PROBLEMS AS MINIMIZING LASSO OBJECTIVES

In this section we give formal proofs that show the shortest path problem is an instance of LASSO and the minimum cut problem is an instance of fused-LASSO.

It's worth noting that our proofs do not guarantee that the answers returned are a single path or cut. In fact, when multiple solutions have the same value it's possible for our algorithm to return a linear combination of them. However, we can ensure that the optimum solution is unique using the Isolation Lemma of Mulmuley, Vazarani and Vazarani [31] while only incurring polynomial increase in edge lengths/weights. This analysis is similar to the one in Section 3.5 of [13] for finding a unique minimum cost flow, and is omitted here.

We prove the two claims in Fact 1.1 about shortest path and minimum cut separately in Lemmas A.1 and A.2.

LEMMA A.1. *Given a s-t shortest path instance in an undirected graph where edge lengths $\boldsymbol{l} : E \to \Re^+$ are integers between 1 and n^d. There is a LASSO minimization instance where all entries are bounded by $n^{O(d)}$ such that the value of the LASSO minimizer is within 1 of the optimum answer.*

Proof Our reductions rely crucially on the edge-vertex incidence matrix, which we denote using $\mathbf{B}$. Entries of this matrix are defined as follows:

$$\mathbf{B}_{e,u} = \begin{cases} -1 & \text{if u is the head of e} \\ 1 & \text{if u is the tail of e} \\ 0 & \text{otherwise} \end{cases} \tag{1.9}$$

We first show the reduction for shortest path. Then a path from s to t corresponds to a flow value assigned to all edges, $\mathbf{f} : E \to \Re$ such that $\mathbf{B}^T\mathbf{f} = \chi_{s,t}$. If we have another flow $\mathbf{f}'$ corresponding to any path from s to t, then this constraint can be written as:

$$||\mathbf{B}^T\mathbf{f} - \chi_{s,t}||_2 = \mathbf{0} \tag{1.10}$$

$$||\mathbf{B}^T(\mathbf{f} - \mathbf{f}')||_2 = \mathbf{0}$$

$$||\mathbf{f} - \mathbf{f}'||_{\mathbf{B}\mathbf{B}^T} = \mathbf{0} \tag{1.11}$$

The first constraint is closer to the classical LASSO problem while the last one is within our definition of grouped least squares problem. The length of the path can then be written as $\sum_e l_e|f_e|$. Weighting these two terms together gives:

$$\min_{\mathbf{f}} \lambda||\mathbf{f} - \mathbf{f}'||^2_{\mathbf{B}\mathbf{B}^T} + \sum_e l_e|f_e| \tag{1.12}$$

Where the maximum entry is bounded by $\max\{n^d, n^2\lambda\}$. Clearly its objective is less than the length of the shortest path, let this solution be $\bar{\mathbf{f}}$. Then since the total objective is at most n^{d+1}, we have that the maximum deviation between $\mathbf{B}^T\bar{\mathbf{f}}$ and $\chi_{s,t}$ is at most n^{d+1}/λ. Then given a spanning tree, each of these deviations can be routed to s or t at a cost of at most n^{d+1} per unit of flow. Therefore we can obtain $\mathbf{f}'$ such that $\mathbf{B}^T\mathbf{f}' = \chi_{s,t}$ whose objective is bigger by at most n^{2d+2}/λ. Therefore setting $\lambda = n^{2d+2}$ guarantees that our objective is within 1 of the length of the shortest path, while the maximum entry in the problem is bounded by $n^{O(d)}$. ■

We now turn our attention to the minimum cut problem, which can be formulated as finding a vertex labeling $\mathbf{x}^{(vert)}$ where $\mathbf{x}_s^{(vert)} = 0$, $\mathbf{x}_t^{(vert)} = 1$ and the size of the cut, $\sum_{uv\in E} |\mathbf{x}_u^{(vert)} - \mathbf{x}_v^{(vert)}|$. Since the L_1 term in the objective can incorporate single variables, we use an additional vector $\mathbf{x}^{(edge)}$ to indicate differences along edges. The minimum cut problem then becomes minimizing $|\mathbf{x}^{(edge)}|$ subject

to the constraint that $\mathbf{x}^{(edge)} = \mathbf{B}'\mathbf{x}^{(vert)'} + \mathbf{B}\chi_t$, where $\mathbf{B}'$ and $\mathbf{x}^{(vert)'}$ are restricted to vertices other than s and t and χ_t is the indicator vector that's 1 on t. The equality constraints can be handled similar to the shortest path problem by increasing the weights on the first term. One other issue is that $|\mathbf{x}^{(vert)'}|$ also appears in the objective term, and we handle this by scaling down $\mathbf{x}^{(vert)'}$, or equivalently scaling up $\mathbf{B}'$.

LEMMA A.2. *Given a s-t minimum cut instance in an undirected graph where edge weights $\boldsymbol{w} : E \to \Re^+$ are integers between 1 and n^d. There is a LASSO minimization instance where all entries are bounded by $n^{O(d)}$ such that the value of the LASSO minimizer is within 1 of the minimum cut.*

Proof

Consider the following objective, where λ_1 and λ_2 are set to n^{d+3} and n^2:

$$\lambda_1||\lambda_2\mathbf{B}'\mathbf{x}^{(vert)'} + \mathbf{B}\chi_t - \mathbf{x}^{(edge)}||_2^2 + |\mathbf{x}^{(vert')}|_1 + |\mathbf{x}^{(edge)}|_1 \tag{1.13}$$

Let $\bar{\mathbf{x}}$ be an optimum vertex labelling, then setting $\mathbf{x}^{(vert')}$ to the restriction of $n^{-2}\bar{\mathbf{x}}$ on vertices other than s and t and $\mathbf{x}^{(edge)}$ to $\mathbf{B}\mathbf{x}^{(vert)}$ makes the first term 0. Since each entry of $\bar{\mathbf{x}}$ is between 0 and 1, the additive increase caused by $|\mathbf{x}^{(vert')}|_1$ is at most $1/n$. Therefore this objective's optimum is at most $1/n$ more than the size of the minimum cut.

For the other direction, consider any solution $\mathbf{x}^{(vert)'}, \mathbf{x}^{(edge)'}$ whose objective is at most $1/n$ more than the size of the minimum cut. Since the edge weights are at most n^d and s has degree at most n, the total objective is at most $n^{d+1} + 1$. This gives:

$$\begin{aligned} &||\lambda_2\mathbf{B}'\mathbf{x}^{(vert)'} + \mathbf{B}\chi_t - \mathbf{x}^{(edge)}||_2^2 \leq n^{-1} \\ &||\lambda_2\mathbf{B}'\mathbf{x}^{(vert)'} + \mathbf{B}\chi_t - \mathbf{x}^{(edge)}||_1 \\ \leq &||\lambda_2\mathbf{B}'\mathbf{x}^{(vert)'} + \mathbf{B}\chi_t - \mathbf{x}^{(edge)}||_2 \leq n^{-1/2} \end{aligned} \tag{1.14}$$

Therefore changing $\mathbf{x}^{(edge)}$ to $\lambda_2\mathbf{B}'\mathbf{x}^{(vert)'} + \mathbf{B}\chi_t$ increases the objective by at most $n^{-1/2} < 1$. This gives a cut with weight within 1 of the objective value and completes the proof. ■

We can also show a more direct connection between the minimum cut problem and the fused LASSO objective, where each absolute value term may contain a linear combination of variables. This formulation is closer to the total variation objective, and is also an instance of the problem formulated in Definition 2.1 with each edge in a group.

LEMMA A.3. *The minimum cut problem in undirected graphs can be written as an instance of the fused LASSO objective.*

Proof Given a graph $G = (V, E)$ and edge weights **cost**, the problem can be formulated as finding a vertex labeling x such that $x_s = 0$, $x_t = 1$ and minimizing:

$$\sum_{uv \in E} \mathbf{cost}_{uv}|x_u - x_v| \tag{1.15}$$

■

B. MULTIPLICATIVE WEIGHTS BASED APPROXIMATE ALGORITHM

In this section we show that the approximate algorithm described in Section 5 finds a solution close to the optimum. For readers familiar with the analysis of the Christiano et al. algorithm [9], the following mapping of terminology might be useful:

- edge $e \to$ group i.
- flow along edge $e \to$ value of $||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}$.
- weight on edge $e \to$ weight of group i, $\mathbf{w}_i$.
- electrical flow problem $\to$ quadratic minimization problem (defined in Section 2.1)
- total energy of electrical flow / effective resistance $\to \mathcal{OBJ}2(\mathbf{w})$.

We first show that if $\lambda^{(t)}$ as defined on Line 6 of Algorithm 1 is an upper bound for $\mathcal{OBJ}(\mathbf{x}^{(t)})$. This is crucial in its use the normalizing factor in our update step on Line 7.

LEMMA B.1. *In all iterations we have:*

$$\mathcal{OBJ}(\boldsymbol{x}^{(t)}) \leq \lambda^{(t)}$$

Proof By the Cauchy-Schwarz inequality we have:

$$\begin{aligned} (\lambda^{(t)})^2 &= \left(\sum_i \mathbf{w}_i^{(t-1)}\right)\left(\sum_i \frac{1}{\mathbf{w}_i^{(t-1)}}||\mathbf{x}^{(t)} - \mathbf{s}_i||_{\mathbf{L}_i}^2\right) \\ &\geq \left(\sum_i ||\mathbf{x}^{(t)} - \mathbf{s}_i||_{\mathbf{L}_i}\right)^2 \\ &= \mathcal{OBJ}(\mathbf{x}^{(t)})^2 \end{aligned} \tag{2.16}$$

Taking square roots of both sides completes the proof. ■

At a high level, the algorithm assigns weights $\mathbf{w}_i$ for each group, and iteratively reweighs them for N iterations. Recall that our key potential functions are $\mu^{(t)}$ which is the sum of weights of all groups, and:

$$\nu^{(t)} = \frac{1}{\text{OPT}}\sum_i ||\bar{\mathbf{x}} - \mathbf{s}_i||_{\mathbf{L}_i} \log(\mathbf{w}_i^{(t)}) \tag{2.17}$$

Where $\bar{\mathbf{x}}$ is a solution such that $\mathcal{OBJ}(\bar{\mathbf{x}}) = \text{OPT}$. We will show that if $\mathcal{OBJ}(\mathbf{x}^{(t)})$, or in turn $\lambda^{(t)}$ is large, then $\nu^{(t)}$ increases at a rate substantially faster than $\log(\mu^{(t)})$. These bounds, and the relations between $\mu^{(t)}$ and $\nu^{(t)}$ are summarized below:

LEMMA B.2. *1.*

$$\nu^{(t)} \leq \log(\mu^{(t)}) \tag{2.18}$$

2.

$$\mu^{(t)} \leq \left(1 + \frac{\epsilon(1+2\epsilon)}{\rho}t\right)\mu^{(t-1)} \tag{2.19}$$

and

$$\log(\mu^{(t)}) \leq \frac{\epsilon(1+2\epsilon)}{\rho}t + \log k \tag{2.20}$$

3. *If in iteration t, $\lambda^{(t)} \geq (1+10\epsilon)OPT$ and $||\boldsymbol{x}-\boldsymbol{s}_i||_{\mathbf{L}_i} \leq \rho\frac{\boldsymbol{w}_i^{(t-1)}}{\mu^{(t-1)}}\lambda^{(t)}$ for all groups i, then:*

$$\nu^t \geq \nu^{(t-1)} + \frac{\epsilon(1+9\epsilon)}{\rho} \tag{2.21}$$

The relationship between the upper and lower potentials can be established using the fact that $\mathbf{w}_i$ is non-negative:

Proof of Lemma B.2, Part 1:

$$\begin{aligned}
\nu^{(t)} =& \frac{1}{\text{OPT}}\sum_i ||\bar{\mathbf{x}}-\mathbf{s}_i||_{\mathbf{L}_i}\log(\mathbf{w}_i^{(t)}) \\
\leq& \frac{1}{\text{OPT}}\sum_i ||\bar{\mathbf{x}}-\mathbf{s}_i||_{\mathbf{L}_i}\log\left(\sum_j \mathbf{w}_j^{(t)}\right) \\
=& \log(\mu^{(t)})\left(\frac{1}{\text{OPT}}\sum_i ||\bar{\mathbf{x}}-\mathbf{s}_i||_{\mathbf{L}_i}\right) \\
\leq& \log(\mu^{(t)})
\end{aligned} \tag{2.22}$$

■

Part 2 follows directly from the local behavior of the log function:

Proof of Lemma B.2, Part 2: The update rules gives:

$$\begin{aligned}
&\mu^{(t)} \\
=&\sum_i \mathbf{w}_i^{(t)} \\
=&\sum_i \mathbf{w}_i^{(t-1)} + \left(\frac{\epsilon}{\rho}\frac{||\mathbf{x}^{(t)}-\mathbf{s}_i||_{\mathbf{L}_i}}{\lambda^{(t)}} + \frac{2\epsilon^2}{k\rho}\right)\mu^{(t-1)} \\
&\quad \text{by update rule on Line 7 of GroupedLeastSquares} \\
=&\mu^{(t-1)} + \frac{\epsilon}{\rho}\frac{\sum_i ||\mathbf{x}^{(t)}-\mathbf{s}_i||_{\mathbf{L}_i}}{\lambda^{(t)}}\mu^{(t-1)} + \sum_i \frac{2\epsilon^2}{k\rho}\mu^{(t-1)} \\
=&\mu^{(t-1)} + \frac{\epsilon}{\rho}\frac{\mathcal{OBJ}(\mathbf{x}^{(t)})}{\lambda^{(t)}}\mu^{(t-1)} + \frac{2\epsilon^2}{\rho}\mu^{(t-1)} \\
\leq&\mu^{(t-1)} + \frac{\epsilon}{\rho}\mu^{(t-1)} + \frac{2\epsilon^2}{\rho}\mu^{(t-1)} \qquad \text{By Lemma B.1} \\
=&\left(1+\frac{\epsilon(1+2\epsilon)}{\rho}\right)\mu^{(t-1)}
\end{aligned} \tag{2.23}$$

Using the fact that $1+x \leq \exp(x)$ when $x \geq 0$ we get:

$$\begin{aligned}
\mu^{(t)} &\leq \exp\left(\frac{\epsilon(1+2\epsilon)}{\rho}\right)\mu^{(t-1)} \\
&\leq \exp\left(t\frac{\epsilon(1+2\epsilon)}{\rho}\right)\mu^{(0)} \\
&= \exp\left(t\frac{\epsilon(1+2\epsilon)}{\rho}\right)k
\end{aligned}$$

Taking logs of both sides gives Equation 2.20.

■

This upper bound on the value of μ^t also allows us to show that the balancing rule keeps the w_i^ts reasonably balanced within a factor of k of each other. The following corollary can also be obtained.

Corollary B.3. *The weights at iteration t satisfy $\boldsymbol{w}_i^{(t)} \geq \frac{\epsilon}{k}\mu^{(t)}$.*

Proof

The proof is by induction on t. When $t=0$ we have $\mathbf{w}_i^{(0)}=1$, $\mu^{(0)}=k$ and the claim follows from $\frac{\epsilon}{k}k=\epsilon<1$. When $t>1$, we have:

$$\begin{aligned}
&\mathbf{w}_i^{(t)} \\
\geq&\mathbf{w}_i^{(t-1)} + \frac{2\epsilon^2}{k\rho}\mu^{(t-1)} \qquad \text{By line 7} \\
\geq&\left(\frac{\epsilon}{k}+\frac{2\epsilon^2}{k\rho}\right)\mu^{(t-1)} \qquad \text{By the inductive hypothesis} \\
=&\frac{\epsilon}{k}\left(1+\frac{2\epsilon}{\rho}\right)\mu^{(t-1)} \\
\geq&\frac{\epsilon}{k}\left(1+\frac{\epsilon(1+2\epsilon)}{\rho}\right)\mu^{(t-1)} \\
\geq&\frac{\epsilon}{k}\mu^{(t)} \qquad \text{By Lemma B.2, Part 2}
\end{aligned} \tag{2.24}$$

■

The proof of Part 3 is the key part of our analysis. The first order change of $\nu^{(t)}$ is written as a sum of products of $\mathbf{L}_i$ norms, which we analyze via. the fact that $\mathbf{x}^{(t)}$ is the solution of a linear system from the quadratic minimization problem.

Proof of Lemma B.2, Part 3:

We make use of the following known fact about the behavior of the log function around 1:

Fact B.4. *If $0 \leq x \leq \epsilon$, then $\log(1+x) \geq (1-\epsilon)x$.*

$$\begin{aligned}
&\nu^{(t)} - \nu^{(t-1)} \\
=&\frac{1}{\text{OPT}}\sum_{1\leq i\leq k} ||\bar{\mathbf{x}}-\mathbf{s}_i||_{\mathbf{L}_i}\log\left(\mathbf{w}_i^{(t)}\right) \\
&\quad - \frac{1}{\text{OPT}}\sum_{1\leq i\leq k} ||\bar{\mathbf{x}}-\mathbf{s}_i||_{\mathbf{L}_i}\log\left(\mathbf{w}_i^{(t-1)}\right) \\
&\quad \text{By Equation 2.17} \\
=&\frac{1}{\text{OPT}}\sum_{1\leq i\leq k} ||\bar{\mathbf{x}}-\mathbf{s}_i||_{\mathbf{L}_i}\log\left(\frac{\mathbf{w}_i^{(t)}}{\mathbf{w}_i^{(t-1)}}\right) \\
\geq&\frac{1}{\text{OPT}}\sum_{1\leq i\leq k} ||\bar{\mathbf{x}}-\mathbf{s}_i||_{\mathbf{L}_i}\log\left(1+\frac{\epsilon}{\rho}\frac{||\mathbf{x}^{(t)}-\mathbf{s}_i||_{\mathbf{L}_i}}{\lambda^{(t)}}\frac{\mu^{(t-1)}}{\mathbf{w}_i^{(t-1)}}\right) \\
&\quad \text{By update rule in line 7} \\
\geq&\frac{1}{\text{OPT}}\sum_{1\leq i\leq k} ||\bar{\mathbf{x}}-\mathbf{s}_i||_{\mathbf{L}_i}\frac{\epsilon(1-\epsilon)}{\rho}\frac{||\mathbf{x}^{(t)}-\mathbf{s}_i||_{\mathbf{L}_i}}{\lambda^{(t)}}\frac{\mu^{(t-1)}}{\mathbf{w}_i^{(t-1)}} \\
&\quad \text{Since } \log(1+x) \geq (1-\epsilon)x \text{ when } 0 \leq x \leq \epsilon \\
=&\frac{\epsilon(1-\epsilon)\mu_i^{(t-1)}}{\rho\text{OPT}\lambda^{(t)}}\sum_{1\leq i\leq k}\frac{1}{\mathbf{w}_i^{(t-1)}}||\bar{\mathbf{x}}-\mathbf{s}_i||_{\mathbf{L}_i}||\mathbf{x}^{(t)}-\mathbf{s}_i||_{\mathbf{L}_i}
\end{aligned} \tag{2.25}$$

Since $\mathbf{L}_i$ forms a P.S.D norm, by the Cauchy-Schwarz inequality we have:

$$
\begin{aligned}
&||\bar{\mathbf{x}}-\mathbf{s}_i||_{\mathbf{L}_i}||\mathbf{x}^{(t)}-\mathbf{s}_i||_{\mathbf{L}_i}\\
\geq&(\bar{\mathbf{x}}-\mathbf{s}_i)^T\mathbf{L}_i(\mathbf{x}^{(t)}-\mathbf{s}_i)\\
=&||\mathbf{x}^{(t)}-\mathbf{s}_i||^2_{\mathbf{L}_i}+(\bar{\mathbf{x}}-\mathbf{x})^T\mathbf{L}_i(\mathbf{x}^{(t)}-\mathbf{s}_i) \qquad (2.26)
\end{aligned}
$$

Recall from Lemma 2.2 that since $\mathbf{x}^{(t)}$ is the minimizer to $\mathcal{OBJ}2(\mathbf{w}^{(t-1)})$, we have:

$$
\left(\sum_i \frac{1}{\mathbf{w}_i^{(t-1)}}\mathbf{L}_i\right)\mathbf{x}^{(t)}=\sum_i\frac{1}{\mathbf{w}^{(t-1)}}\mathbf{s}_i \qquad (2.27)
$$

$$
\left(\sum_i \frac{1}{\mathbf{w}_i^{(t-1)}}\mathbf{L}_i\right)(\mathbf{x}^{(t)}-\mathbf{s}_i)=\mathbf{0} \qquad (2.28)
$$

$$
(\bar{\mathbf{x}}-\mathbf{x}^{(t)})^T\left(\sum_i \frac{1}{\mathbf{w}_i^{(t-1)}}\mathbf{L}_i\right)(\mathbf{x}^{(t)}-\mathbf{s}_i)=0 \qquad (2.29)
$$

Substituting this into Equation 2.25 gives:

$$
\begin{aligned}
&\nu^{(t)}-\nu^{(t-1)}\\
\geq&\frac{\epsilon(1-\epsilon)\mu_i^{(t-1)}}{\rho\mathrm{OPT}\lambda^{(t)}}\sum_i\frac{1}{\mathbf{w}_i^{(t-1)}}||\mathbf{x}^{(t)}-\mathbf{s}_i||^2_{\mathbf{L}_i}\\
=&\frac{\epsilon(1-\epsilon)}{\rho\mathrm{OPT}\lambda^{(t)}}\mu^{(t-1)}\mathcal{OBJ}2(\mathbf{w},\mathbf{x}^{(t)})\\
=&\frac{\epsilon(1-\epsilon)}{\rho\mathrm{OPT}\lambda^{(t)}}(\lambda^{(t)})^2\\
&\text{By definition of } \lambda^{(t)} \text{ on Line 6}\\
\geq&\frac{\epsilon(1-\epsilon)(1+10\epsilon)}{\rho}\\
&\text{By assumption that } \lambda^{(t)}>(1+10\epsilon)\mathrm{OPT}\\
\geq&\frac{\epsilon(1+8\epsilon)}{\rho} \qquad (2.30)
\end{aligned}
$$

Since the iteration count largely depends on ρ, it suffices to provide bounds for ρ over all the iterations. The proof makes use of the following lemma about the properties of electrical flows, which describes the behavior of modifying the weights of a group S_i that has a large contribution to the total energy. It can be viewed as a multiple-edge version of Lemma 2.6 of [9].

LEMMA B.5. *Assume that $\epsilon^2\rho^2<1/10k$ and $\epsilon<0.01$ and let $\boldsymbol{x}^{(t-1)}$ be the minimizer for $\mathcal{OBJ}2(\boldsymbol{w}^{(t-1)})$. Suppose there is a group i such that $||\boldsymbol{x}^{(t-1)}-\boldsymbol{s}_i||_{\boldsymbol{L}_i}\geq\rho\frac{\boldsymbol{w}_i^{(t-1)}}{\mu^{(t-1)}}\lambda^{(t)}$, then*

$$
\mathit{OPT2}(\boldsymbol{w}^{(t)})\leq\exp\left(-\frac{\epsilon^2\rho^2}{2k}\right)\mathit{OPT2}(\boldsymbol{w}^{(t-1)})
$$

Proof

We first show that group i contributes a significant portion to $\mathcal{OBJ}2(\mathbf{w}^{(t-1)},\mathbf{x}^{(t-1)})$. Squaring both sides of the given condition gives:

$$
\begin{aligned}
&||\mathbf{x}^{(t-1)}-\mathbf{s}_i||^2_{\mathbf{L}_i}\\
\geq&\rho^2\frac{(\mathbf{w}_i^{(t-1)})^2}{(\mu^{(t-1)})^2}(\lambda^{(t)})^2\\
=&\rho^2\frac{(\mathbf{w}_i^{(t-1)})^2}{(\mu^{(t-1)})^2}\mu^{(t-1)}\mathcal{OBJ}2(\mathbf{w}^{(t-1)},\mathbf{x}^{(t-1)}) \qquad (2.31)\\
&\frac{1}{\mathbf{w}_i^{(t-1)}}||\mathbf{x}^{(t-1)}-\mathbf{s}_i||_{\mathbf{L}_i}\\
\geq&\rho^2\frac{\mathbf{w}_i^{(t-1)}}{\mu^{(t-1)}}\mathcal{OBJ}2(\mathbf{w}^{(t-1)},\mathbf{x}^{(t-1)})\\
\geq&\frac{\epsilon\rho^2}{k}\mathcal{OBJ}2(\mathbf{w}^{(t-1)},\mathbf{x}^{(t-1)})\\
&\text{By Corollary B.3} \qquad (2.32)
\end{aligned}
$$

Also, by the update rule we have $\mathbf{w}_i^{(t)}\geq(1+\epsilon)\mathbf{w}_i^{(t-1)}$ and $\mathbf{w}_j^{(t)}\geq\mathbf{w}_j^{(t-1)}$ for all $1\leq j\leq k$. So we have:

$$
\begin{aligned}
&\mathrm{OPT2}(\mathbf{w}^{(t)})\\
\leq&\mathcal{OBJ}2(\mathbf{w}^{(t)},\mathbf{x}^{(t-1)})\\
=&\mathcal{OBJ}2(\mathbf{w}^{(t)},\mathbf{x}^{(t-1)})-(1-\frac{1}{1+\epsilon})||\mathbf{x}^{(t-1)}-\mathbf{s}_i||^2_{\mathbf{L}_i}\\
\leq&\mathcal{OBJ}2(\mathbf{w}^{(t)},\mathbf{x}^{(t-1)})-\frac{\epsilon}{2}||\mathbf{x}^{(t-1)}-\mathbf{s}_i||^2_{\mathbf{L}_i}\\
\leq&\mathcal{OBJ}2(\mathbf{w}^{(t)},\mathbf{x}^{(t-1)})-\frac{\epsilon^2\rho^2}{2k}\mathcal{OBJ}2(\mathbf{w}^{(t-1)},\mathbf{x}^{(t-1)})\\
\leq&\exp\left(-\frac{\epsilon^2\rho^2}{2k}\right)\mathcal{OBJ}2(\mathbf{w}^{(t-1)},\mathbf{x}^{(t-1)}) \qquad (2.33)
\end{aligned}
$$

■

This means the value of the quadratic minimization problem can be used as a second potential function. We first show that it's monotonic and establish rough bounds for it.

LEMMA B.6. *$\mathit{OPT2}(\boldsymbol{w}^{(0)})\leq n^{3d}$ and $\mathit{OPT2}(\boldsymbol{w}^{(t)})$ is monotonically decreasing in t.*

Proof By the assumption that the input is polynomially bounded we have that all entries of $\mathbf{s}$ are at most n^d and $\mathbf{L}_i\preceq n^d\mathbf{I}$. Setting $\mathbf{x}_u=0$ gives $||\mathbf{x}-\mathbf{s}_i||_2\leq n^{d+1}$. Combining this with the spectrum bound then gives $||\mathbf{x}-\mathbf{s}_i||_{\mathbf{L}_i}\leq n^{2d+1}$. Summing over all the groups gives the upper bound.

The monotonicity of $\mathrm{OPT2}(\mathbf{w}^{(t)})$ follows from the fact that all weights are decreasing. ■

Combining this with the fact that $\mathrm{OPT2}(\mathbf{w}^{(N)})$ is not low enough for termination gives our bound on the total iteration count.

Proof of Theorem 5.1: The proof is by contradiction. Suppose otherwise, since $\mathcal{OBJ}(\mathbf{x}^{(N)})>\epsilon$ we have:

$$
\begin{aligned}
\lambda^{(N)}\geq&(1+10\epsilon)n^{-d}\\
\geq&2n^{-d}\mathrm{OPT} \qquad (2.34)\\
\sqrt{\mu^{(N)}\mathrm{OPT2}(\mathbf{w}^{(t)})}\geq&2n^{-d} \qquad (2.35)\\
\mathrm{OPT2}(\mathbf{w}^{(t)})\geq&\frac{4}{n^{-2d}\mu^{(N)}} \qquad (2.36)
\end{aligned}
$$

Which combined with $\text{OPT2}(\mathbf{w}^{(0)}) \leq n^{3d}$ from Lemma B.6 gives:

$$\frac{\text{OPT2}(\mathbf{w}^{(0)})}{\text{OPT2}(\mathbf{w}^{(N)})} \leq n^{5d}\mu^{(N)} \tag{2.37}$$

By Lemma B.2 Part 2, we have:

$$\begin{aligned}
&\log(\mu^{(N)}) \\
\leq&\frac{\epsilon(1+\epsilon)}{\rho}N + \log k \\
\leq&\frac{\epsilon(1+\epsilon)}{\rho}10d\rho\log n\epsilon^{-2} + \log n \\
&\qquad \text{By choice of } N = 10d\rho\log n\epsilon^{-2} \\
=&10(1+\epsilon)\epsilon^{-1}\log n + \log n \\
\leq&10d(1+2\epsilon)\epsilon^{-1}\log n \\
&\qquad \text{when } \epsilon < 0.01
\end{aligned} \tag{2.38}$$

Combining with Lemma B.5 implies that the number of iterations where $||\mathbf{x}^{(t-1)} - \mathbf{s}_i||_{\mathbf{L}_i} \geq \rho\frac{\mathbf{w}_i^{(t-1)}}{\mu^{(t-1)}}\lambda^{(t)}$ for i is at most:

$$\begin{aligned}
&\log\left(\mu^{(N)}n^{5d}\right) / \left(\frac{\epsilon^2\rho^2}{2k}\right) \\
=&10d(1+3\epsilon)\epsilon^{-1}\log n / \left(\frac{2\epsilon^{2/3}}{k^{1/3}}\right) \\
&\qquad \text{By choice of } \rho = 2k^{1/3}\epsilon^{-2/3} \\
=&8d\epsilon^{-5/3}k^{1/3}\log n \\
=&4d\epsilon^{-1}\rho\log n \leq \epsilon N
\end{aligned} \tag{2.39}$$

This means that we have $||\mathbf{x}^{(t-1)} - \mathbf{s}_i||_{\mathbf{L}_i} \leq \rho\frac{\mathbf{w}_i^{(t-1)}}{\mu^{(t-1)}}\lambda^{(t)}$ for all $1 \leq i \leq k$ for at least $(1-\epsilon)N$ iterations and therefore by Lemma B.2 Part 3:

$$\nu^{(N)} \geq \nu^{(0)} + \frac{\epsilon(1+8\epsilon)}{\rho}(1-\epsilon)N > \mu^{(N)} \tag{2.40}$$

Giving a contradiction. ■

C. ALMOST-EXACT ALGORITHM USING INTERIOR POINT ALGORITHMS

We now show improved algorithms for solving the second order cone programming formulation given in [6, 17]. It was shown in [13] that in the linear case, as with graph problems such as maximum flow, minimum cost flow and shortest path, interior point algorithms reduce the problem to solving $\tilde{O}(m^{1/2})$ symmetrically diagonally dominant linear systems. The grouped least squares formulation creates artifacts that perturb the linear systems generated by the interior point algorithms, making the resulting system both more difficult to interpret and to solve. However, the iteration count of this approach also only depends on k, [17, 6], and has a better dependency on ϵ of $O(\log(1/\epsilon))$.

There are various ways to solve the grouped least squares problem using interior point algorithms. We follow the log-barrier method, as presented in Boyd and Vandenberghe [6] here for simplicity. This formulation defines one extra variable y_i for each group and enforces $y_i \geq ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}$ using the barrier function $\phi_i(\mathbf{x}, y_i) = \log(y_i^2 - ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}^2)$. Minimizing $t \cdot (\sum_i y_i)$ for gradually increasing values of t gives the following sequence of functions to minimize:

$$f(t, \mathbf{x}, \mathbf{y}) = t\sum_i y_i - \sum_i \log(y_i^2 - ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}^2) \tag{3.41}$$

Various interior point algorithms have been proposed, one commonality that they have is finding an update direction by solving a linear system. The iteration guarantees for recovering almost-exact solution can be characterized as follows:

LEMMA C.1. *(Section 11.5.3 from [6]) A solution that's within additive ϵ of the optimum solution can be produced in $\tilde{O}(k^{1/2}\log(1/\epsilon))$ steps, each of which requires solving a linear system involving $\nabla^2 f(t, \boldsymbol{x}, \boldsymbol{y})$ for some value of t, $\boldsymbol{x}$ and $\boldsymbol{y}$.*

Since the $t\sum_i y_i$ term is linear, it can be omitted from the Hessian, leaving $\sum_i \nabla^2\phi(\mathbf{x}, y_i)$. We then check that the barrier term y_i creates a low rank perturbation to the $\mathbf{L}_i$ term, which is the Hessian for $||\mathbf{x}-\mathbf{s}_i||_{\mathbf{L}_i}^2$. By taking Schur complements and applying the Sherman-Morrison-Woodbury identity on inverses for low rank perturbations, we arrive at the following observation.

THEOREM C.2. *Suppose there is an algorithm for solving linear systems of the form $\sum_i \alpha_i \boldsymbol{L}_i$ in $T(n, m)$ time where m. For any choice of $\boldsymbol{x}, \boldsymbol{y}$, a linear system involving the Hessian of $\phi(\boldsymbol{x}, \boldsymbol{y})$, $\nabla^2\phi(\boldsymbol{x}, \boldsymbol{y})$ can be solved in $O(k^\omega + kT(n, m) + k^2 n)$ time.*

Proof

We first consider the barrier function corresponding to each group, $\phi(\mathbf{x}, y_i)$. Its gradient is:

$$\begin{aligned}
\nabla\phi(\mathbf{x}, y_i) =& \nabla - \log\left(y_i^2 - (\mathbf{x} - \mathbf{s}_i)^T\mathbf{L}_i(\mathbf{x} - \mathbf{s}_i)\right) \\
=& \frac{2}{y_i^2 - ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}^2}\begin{bmatrix} \mathbf{L}_i(\mathbf{x} - \mathbf{s}_i) \\ -y_i \end{bmatrix}
\end{aligned} \tag{3.42}$$

and its Hessian, $\nabla^2\phi(\mathbf{x}, y_i)$, is:

$$\begin{aligned}
&\frac{2}{(y_i^2 - ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}^2)^2} \\
&\left[\begin{array}{c|c} (y_i^2 - ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}^2)\mathbf{L}_i & \\ +2\mathbf{L}_i(\mathbf{x} - \mathbf{s}_i)(\mathbf{x} - \mathbf{s}_i)^T\mathbf{L}_i & 2y_i\mathbf{L}_i(\mathbf{x} - \mathbf{s}_i) \\ \hline 2y_i(\mathbf{x} - \mathbf{s}_i)^T\mathbf{L}_i & y_i^2 + ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}^2 \end{array}\right]
\end{aligned} \tag{3.43}$$

Since the variable y_i only appears in $\phi(\mathbf{x}, y_i)$, we may use partial Cholesky factorization to arrive at a linear system without it. The $n \times n$ matrix that we obtain is:

$$\begin{aligned}
&\mathbf{L}_i(y_i^2 - ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}^2) \\
&\qquad + \left(2 - \frac{4y_i^2}{y_i^2 + ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}^2}\right)\mathbf{L}_i(\mathbf{x} - \mathbf{s}_i)(\mathbf{x} - \mathbf{s}_i)^T\mathbf{L}_i \\
=&(y_i^2 - ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}^2)\left(\mathbf{L}_i - \frac{2\mathbf{L}_i(\mathbf{x} - \mathbf{s}_i)(\mathbf{x} - \mathbf{s}_i)^T\mathbf{L}_i}{y_i^2 + ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}^2}\right)
\end{aligned} \tag{3.44}$$

Since $\phi(\mathbf{x}, \mathbf{y}) = \sum_i \phi(\mathbf{x}, y_i)$, this partial Cholesky factorization of $\nabla^2\phi(\mathbf{x}, \mathbf{y})$ can be written as:

$$\sum_i \alpha_i \mathbf{L}_i - \beta_i \mathbf{u}_i \mathbf{u}_i^T \tag{3.45}$$

Where $\mathbf{u}_i = \mathbf{L}_i(\mathbf{x} - \mathbf{s}_i)$ and α_i and β_i are scalars. We can simplify solving this system using the Sherman-Morrison-Woodbury formula:

FACT C.3. *(Sherman–Morrison–Woodbury formula)*
If $\boldsymbol{A}$, $\boldsymbol{U}$, $\boldsymbol{C}$, $\boldsymbol{V}$ are $n \times n$, $n \times k$, $k \times k$ and $k \times n$ matrices respectively, then:

$$(\boldsymbol{A} + \boldsymbol{UCV})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{C}^{-1} + \boldsymbol{VA}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VA}^{-1}$$

Here we have $\mathbf{A} = \sum_i \alpha_i \mathbf{L}_i$, $\mathbf{C} = -\mathbf{I}$, $\mathbf{U}$ being the k columns vectors $\sqrt{\beta_i}\mathbf{u}_i$ concatenated and $\mathbf{V} = \mathbf{U}^T$. So the linear system that we need to evaluate becomes:

$$\mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{U}^T\mathbf{A}^{-1}\mathbf{U} - \mathbf{I})^{-1}\mathbf{U}^T\mathbf{A}^{-1} \tag{3.46}$$

The system $\mathbf{A}^{-1}\mathbf{U}$ can be found using k solves in $\mathbf{A} = \sum_i \alpha_i \mathbf{L}_i$, which is equivalent to the quadratic minimization problem. Multiplying this by $\mathbf{U}$ can be done in $O(k^2 n)$ time and gives us $\mathbf{U}^T\mathbf{A}^{-1}\mathbf{U}$. This $k \times k$ system can in turn be solved in k^ω time. The other terms can be applied to vectors in either solves in $\mathbf{A}$ or matrix multiples in $\mathbf{U}$, taking $O(T(n, m) + kn)$ time. ■

Combining this with the iteration count of $\tilde{O}(k^{1/2}\log(1/\epsilon))$ gives a total running time of $\tilde{O}((k^{\omega+1/2} + k^{3/2}T(n, m) + k^{5/2}n)\log(1/\epsilon))$.

D. OTHER VARIANTS

Although our formulation of $\mathcal{OBJ}$ as a sum of L_2 objectives differs syntactically from some common formulations, we show below that the more common formulation involving quadratic, or L_2^2 fidelity term can be reduced to finding exact solutions to $\mathcal{OBJ}$ using 2 iterations of ternary search. Most other formulations differs from our formulation in the fidelity term, but more commonly have L_1 smoothness terms as well. Since the anisotropic smoothness term is a special case of the isotropic one, our discussion of the variations will assume anisotropic objectives.

D.1 L_2^2 fidelity term

The most common form of the total variation objective used in practice is one with L_2^2 fidelity term. This term can be written as $||\mathbf{x} - \mathbf{s}_0||_2^2$, which corresponds to the norm defined by $\mathbf{I} = \mathbf{L}_0$. This gives:

$$\min_{\mathbf{x}} \quad ||\mathbf{x} - \mathbf{s}_0||_{\mathbf{L}_0}^2 + \sum_{1 \le i \le k} ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}$$

We can establish the value of $||\mathbf{x} - \mathbf{s}_0||_{\mathbf{L}_0}^2$ separately by guessing it as a constraint. Since the t^2 is convex in t, the following optimization problem is convex in t as well:

$$\min_{\mathbf{x}} \quad \sum_{1 \le i \le k} ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i} ||\mathbf{x} - \mathbf{s}_0||_{\mathbf{L}_0}^2 \le t^2$$

Also, due to the convexity of t^2, ternary searching on the minimizer of this plus t^2 would allow us to find the optimum solution by solving $O(\log n)$ instances of the above problem.

Taking square root of both sides of the $||\mathbf{x} - \mathbf{s}_0||_{\mathbf{L}_0}^2 \le t^2$ condition and taking its Lagrangian relaxation gives:

$$\min_{\mathbf{x}} \max_{\lambda \ge 0} \sum_{i=1}^{k} ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i} + \lambda(||\mathbf{x} - \mathbf{s}_0||_{\mathbf{L}_0} - t)$$

Which by the min-max theorem is equivalent to:

$$\max_{\lambda \ge 0} -\lambda t + \left(\min_{\mathbf{x}} \sum_{i=1}^{k} ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i} + \lambda ||\mathbf{x} - \mathbf{s}_0||_{\mathbf{L}_0} \right)$$

The term being minimized is identical to our formulation and its objective is convex in λ when $\lambda \ge 0$. Since $-\lambda t$ is linear, their sum is convex and another ternary search on λ suffices to optimize the overall objective.

D.2 L_1 fidelity term

Another common objective function to minimize is where the fidelity term is also under L_1 norm. In this case the objective function becomes:

$$||\mathbf{x} - \mathbf{s}_0||_1 + \sum_i \sum_{1 \le i \le k} ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}$$

This can be rewritten as a special case of $\mathcal{OBJ}$ as:

$$\sum_u \sqrt{(x_u - s_u)^2} + \sum_i \sum_{1 \le i \le k} ||\mathbf{x} - \mathbf{s}_i||_{\mathbf{L}_i}$$

Which gives, instead, a grouped least squares problem with $m + k$ groups.