An Algorithmic Framework for High Dimensional Regression with Dependent Variables

Hoyt Koepke

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2013

Reading Committee:

Marina Meila-Predoviciu, Chair

Prof. Jon Wellner

Prof. Adrian Dobra

Prof. Maryam Fazel

Program Authorized to Offer Degree:
Deptartment of Statistics

University of Washington

Abstract

An Algorithmic Framework for High Dimensional Regression with Dependent Variables

Hoyt Koepke

Chair of the Supervisory Committee:
Professor Marina Meila-Predoviciu
Department of Statistics

We present an exploration of the rich theoretical connections between several classes of regularized models, network flows, and recent results in submodular function theory. This work unifies key aspects of these problems under a common theory, leading to novel methods for working with several important models of interest in statistics, machine learning and computer vision. Most notably, we describe the full regularization path of a class of penalized regression problems with dependent variables that includes variants of the fused LASSO and total variation constrained models.

We begin by reviewing the concepts of network flows and submodular function optimization theory foundational to our results. We then examine the connections between network flows and the minimum-norm algorithm from submodular optimization, extending and improving several current results. This theory leads to a new representation of the structure of a large class of pairwise regularized models important in machine learning, statistics and computer vision. Finally, by applying an arbitrarily accurate approximation, our approach allows us to efficiently optimize total variation penalized models on continuous functions. Ultimately, our new algorithms scale up easily to high-dimensional problems with millions of variables.

TABLE OF CONTENTS

Page

LIST OF FIGURES

ACKNOWLEDGMENTS

The author wishes to express sincere appreciation to his advisor, Marina Meilă, to professors Maryam Fazel and Jon Wellner for numerous helpful discussions, and to his family and friends for their unending support during this journey.

DEDICATION

to my parents, Doris and Galen, who have never doubted that I could reach this point.

Chapter 1

FOUNDATIONAL CONCEPTS

1.1 Introduction

High-dimensional data is a central focus of modern statistical research. Recent technological advances in a variety of scientific fields for gathering and generating data, matched by rapidly increasing computing power for analysis, has attracted significant research into the statistical questions surrounding structured data. Numerous models and computational techniques have emerged recently to work with these data types.

Our primary contribution is a theoretical framework enabling the efficient optimization of models that work with high-dimensional models in which the predictors are believed to be dependent, correlated, or sparse. In particular, we propose a new approach for estimation of certain types of dependent predictors, in which the model incorporates a prior belief that many of the predictors take similar or identical values. This has been a hot topic of research in recent years, with applications in genomics, image analysis, graphical models, and several other areas. However, efficient estimation involving structured and dependent predictors has proven to be quite challenging. Our main contribution, which includes a number of related theoretical results, is a theoretical framework and algorithmic approach that unlocks a large and particularly thorny class of these models.

This chapter is laid out as follows. After laying out the context for our research in section 1.2, we describe our main results as well as lay out the full structure of this paper. Our contributions build on recent results in optimization theory and combinatorial optimization, which are covered by the next three sections.

1.2 Statistical Models and Regularization

To begin, consider a simple regression model, in which we have N observations $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N)$ and n predictors $\mathbf{u} = (u_1, u_2, ..., u_n)$, with

$$\mathbf{y}_i = \mathbf{A}\mathbf{u} + \boldsymbol{\varepsilon}_i, \qquad i = 1, 2, ..., N \tag{1.1}$$

where $\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2, ..., \boldsymbol{\varepsilon}_N$ are independent random noise vectors with $\mathrm{E}\,\boldsymbol{\varepsilon}_i = \mathbf{0}$; typically, these are assumed to be i.i.d. Gaussian. In many high dimensional contexts, the estimation of $\hat{\mathbf{u}}$ may be problematic using classical methods. For example, n may be much larger than N,

making the problem ill-posed as many possible values of $\mathbf{u}$ map to the same response. (n and N are used here instead of the more common p and n to be consistent with the optimization literature we connect this problem to.) Additionally, many predictors of interest may have a negligible or nearly identical affect on $\mathbf{y}$; eliminating or grouping these predictors is then desirable. To handle this, a number of sophisticated approaches have been proposed to incorporate variable selection or aggregation into the statistical estimation problem.

One common and well-studied approach is to add a penalty term, or regularizer, to the log-likelihood that enforces some prior belief about the structure of $\mathbf{u}$ [Bickel et al., 2006, Hastie et al., 2009]. In this setup, estimating the predictor $\mathbf{u}$ involves finding the minimizer of a log-likelihood term or loss function $\mathcal{L}$ plus a regularization term Φ:

$$\hat{\mathbf{u}} = \underset{\mathbf{u}\in\mathbb{R}^n}{\operatorname{argmin}} \ \mathcal{L}(\mathbf{u}, \mathbf{y}) + \lambda\Phi(\mathbf{u}), \tag{1.2}$$

where λ controls the strength of the regularization $\lambda\Phi(\mathbf{u})$.

Many well studied and frequently used models fall into this context. For example, if $\mathcal{L}$ is the log-likelihood from a multivariate Gaussian distribution, one of the oldest regularization techniques is the L_2-norm, which gives us the classic technique of ridge regression [Hoerl and Kennard, 1970], where

$$\hat{\mathbf{u}}_{\text{ridge}} = \underset{\mathbf{u}\in\mathbb{R}^n}{\operatorname{argmin}} \ \|\mathbf{y} - \mathbf{A}\mathbf{u}\|_2^2 + \lambda \|\mathbf{u}\|_2^2 . \tag{1.3}$$

In this example, the regularization term is typically used as an effective way of dealing with an ill-conditioned inverse problem in the least squares context or as a simple way of preventing $\mathbf{u}$ from over-fitting the model [Bishop, 2006].

In the Bayesian context, (1.2) often corresponds directly to finding the maximum a posteriori estimate in the classic likelihood-prior formulation, i.e.

$$p(\mathbf{u} \mid \mathbf{y}) \propto p(\mathbf{y} \mid \mathbf{u}) p(\mathbf{u} \,;\, \lambda, \theta) \tag{1.4}$$

$$\propto e^{-\mathcal{L}(\mathbf{u},\mathbf{y})} \times e^{-\lambda\Phi_\theta(\mathbf{u})}. \tag{1.5}$$

where λ and θ are hyperparameters controlling the behavior of the prior distribution. In this formulation, the prior captures the belief encoded by the regularization term. For example,

the ridge regression problem of (1.3) corresponds to using a standard multivariate Gaussian likelihood but assumes a 0-mean spherical Gaussian prior distribution over $\mathbf{u}$.

More recently, the Least Absolute Shrinkage and Selection Operator (LASSO), is used frequently to promote sparsity in the resulting estimator [Tibshirani, 1996, Hastie et al., 2005, 2009]. This approach uses the L_1-norm as the regularization term, i.e.

$$\hat{\mathbf{u}}_{\text{lasso}} = \underset{\mathbf{u}\in\mathbb{R}^n}{\operatorname{argmin}} \ \|\mathbf{y} - \mathbf{A}\mathbf{u}\|_2^2 + \lambda \|\mathbf{u}\|_1 \tag{1.6}$$

The LASSO problem has been analyzed in detail as a method for simultaneous estimation and variable selection, as the L_1 penalty tends to set many of the predictors to 0. The consistency and theoretical properties of this model, for both estimation and variable selection, are well studied [Meinshausen and Yu, 2009, Bunea et al., 2007, 2006, Van De Geer, 2008, van de Geer, 2007, Bickel et al., 2006]. Many variants of this problem have also been proposed and analyzed. These include the elastic net, in which $\Phi(\mathbf{u}) = \alpha \|\mathbf{u}\|_1 + (1-\alpha) \|\mathbf{u}\|_2^2$, with $\alpha \in [0, 1]$; this generalizes both ridge regression and LASSO [Zou and Hastie, 2005]. Data-adaptive versions of these estimators are analyzed in [Zou, 2006, Zou and Hastie, 2005, Zou and Zhang, 2009]. In addition, there are many efficient algorithms to quickly solve these problems, which is one of the reasons they are used frequently in practice [Friedman et al., 2008a, 2010, 2007, Efron et al., 2004, Beck and Teboulle, 2009].

1.2.1 Graph Structured Dependencies

While there are other possible regularization strategies on individual parameters, of most interest to us is the incorporation of correlation structures and dependencies among the predictors into the prior distribution or regularization term. In many cases, this translates into penalizing the differences of predictors. This has recently been attracting significant algorithmic and theoretical interest as a way to effectively handle dependency structures in the data.

In our particular context, we are interested in the MAP estimate of models with Markov Random Field prior. The resulting log-linear models consist of a collection of unary and pairwise terms that capture the dependency structure of the problem. In particular, the priors we are interested in enforce similarity between neighboring variables, where "neigh-

bor" is defined according to the graph structure of the MRF prior. The type of dependency we are looking at is determined by the

The simplest form of model dependency in the parameters is captured by the *Fused LASSO* problem [Tibshirani et al., 2005], which penalizes pairwise differences between terms in an ordered problem to be

$$\mathbf{u}^*_{\text{fused}} = \underset{\mathbf{u}\in\mathbb{R}^n}{\text{argmin}} = \|\mathbf{A}\mathbf{u}-\mathbf{y}\|_2^2 + \lambda\left[w_1\|\mathbf{u}\|_1 + w_2\sum_{i=1}^{n-1}|u_{i+1}-u_i|\right], \tag{1.7}$$

where $w_1, w_2 \geq 0$ control the balance between the L_1-norm controlling variable sparsity and the sum of ordered differences that effectively penalizes changepoints. This problem has gained some recent attention in the statistics community in the context of non-parametric regression [Dümbgen and Kovac, 2009, Cho and Fryzlewicz, 2011, Davies and Meise, 2008], group sparsity [Tibshirani et al., 2005, Bleakley and Vert, 2011], and change-point detection [Bleakley and Vert, 2011]. From the optimization perspective, several algorithms to find the solution to this problem have been proposed; we refer the reader to Liu et al. [2010], Ye and Xie [2011], Bach et al. [2012] or Friedman et al. [2007] for discussions of this particular problem.

The generalized problem, in which the pairwise terms are not required to be ordered, is of significant practical interest in both the statistics and machine learning communities. Here, the pairwise interactions are controlled with an arbitrary graph of weighted difference penalties:

$$\mathbf{u}^*_{\text{graph}} = \underset{\mathbf{u}\in\mathbb{R}^n}{\text{argmin}} = \|\mathbf{y}-\mathbf{A}\mathbf{u}\|_2^2 + \lambda\left[w_1\|\mathbf{u}\|_1 + \sum_{1\leq i<j\leq n} w_{ij}|u_i-u_j|\right], \tag{1.8}$$

where $w_1 \geq 0$ controls the sparsity of the individual predictors, and $w_{ij} \geq 0$ penalizes differences, typically between predictors known to be correlated.

This model – often with $w_1 = 0$ – arises in the context of "roughness" penalized regression in which differences between neighboring regions are penalized [Belkin et al., 2004]. Besag, Green, Higdon, and Mengersen [1995] examined pairwise interactions on Markov random fields, which describe a prior distribution on the edges of an undirected graph; the proposed models were then solved with MCMC methods. Similar problems also arise in the estimation

of sparse covariance matrices as outlined by Friedman, Hastie, and Tibshirani [2008b]; there, structure learning is the goal and the pairwise interaction terms are not formed explicitly.

Similarly, Kovac and Smith [2011] analyzes this model as an approach to nonparametric regression on a graph, although with the simpler case of $\mathbf{A} = \mathbf{I}$. He proposes an efficient active-region based procedure to solve the resulting optimization problem. In the genomics community, a version of (1.8) has been proposed as the Graph-Guided Fused LASSO problem by Kim, Sohn, and Xing [2009]. There, the pairwise interactions were estimated from the correlation structure in the quantitative traits. Their model is essentially identical to this one, though with $w_1 = 0$. There, the authors used a general quadratic solver to tackle the problem and showed good statistical performance in detecting genetic markers. The use of recent results in optimization, in particular proximal operator methods [Nesterov, 2007], have been proposed as a way of optimizing this model Chen, Kim, Lin, Carbonell, and Xing [2010], Chen, Lin, Kim, Carbonell, and Xing [2012], Bach, Jenatton, Mairal, and Obozinski [2012]. The resulting algorithms, discovered independently, are similar to the ones we propose, although we improve upon them in several ways.

A very similar model to (1.8) is proposed in Sharma, Bondell, and Zhang [2013]; there, however, the authors include an additional regularization term penalizing $|u_i + u_j|$. In addition, they give several adaptive schemes for choosing the weights in a way that depends on the correlation structures of the design matrix. This work is noteworthy in that the authors give a detailed analysis of the asymptotic convergence and estimation rate of the model when the regularization weights are chosen adaptively. Their optimization method, however, involves a simple quadratic programming setup, which can severely limit the size of the problems for which their estimator can be used.

In the context of regularized regression with pairwise dependencies, our contribution is a theoretical treatment and several algorithms for a class of models that generalizes (1.8). Namely, we allow the $|u_i|$ penalty in the L_1 norm to be replaced with an arbitrary convex piecewise linear function $\xi_i(u_i)$. In particular, we examine the estimator

$$\mathbf{u}^*_\star = \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^n} \mathcal{L}(\mathbf{u}, \mathbf{y}) + \lambda \left[\sum_i \xi_i(u_i) + \sum_{1\le i<j\le n} w_{ij}|u_i - u_j| \right], \tag{1.9}$$

which generalizes many of the above models. While our primary result is a representation of the theoretical structures underlying this optimization problem, our work motivates efficient and novel algorithms for working with these types of structure. The theory we develop leads to proofs of the correctness of our algorithms. In addition, we are able to present an algorithmic description that gives the entire regularization path for one version of this problem.

1.2.2 Total Variation Models

Total variation models are effectively an extension of the above pairwise regularization schemes to the estimation of continuous functions. In this context, we wish to estimate a function u in which the total variation of u – the integral of the norm of the gradient – is controlled. This model is used heavily in developing statistical models of images [Chambolle et al., 2010] and the estimation of density functions [Bardsley and Luttman, 2009], but arises in other contexts as well.

When dealing with total variation models, we work theoretically with general L_2-measurable functions with bounded variation. Our response becomes a function f, and our predictor becomes a function u. However, in practice, we discretize the problem by working with the functions f and u only on a discrete lattice of points, as is common in these problems. In this context, the total variation problem can be expressed in the form of (1.9) as described in chapter 5. Our novel contribution, also in chapter 5, is a thorough theoretical treatment of the accuracy of this approximation.

Let $\Omega \subset \mathbb{R}^d$ be a compact region, and let u be an L_1-integrable continuously differentiable function defined on Ω. In this context, the total variation of u is given by

$$\mathrm{TV}(u) = \int_\Omega \|\nabla u\|_2 \, \mathrm{d}\mu = \int_\Omega \|(\nabla u)(\mathbf{x})\|_2 \, \mathrm{d}\mathbf{x}. \tag{1.10}$$

More general definitions exist when u is not continuously differentiable Ambrosio and Di Marino [2012], Giusti [1984]; for simplicity, we assume this condition. Let $\mathcal{F}(\Omega)$ represent the space of L_1-integrable continuously differentiable functions of bounded variation. Here, bounded variation can be taken as the condition

$$\mathrm{TV}(u) < +\infty. \tag{1.11}$$

Again, much more general definitions exist, but this suffices for our purposes.

Formally, then, the total variation problem seeks to find an estimation function u^* that to a function $f : \mathbb{R}^d \supset \Omega \mapsto \mathbb{R}$ under a constraint or penalty on the total norm of the gradient of u.

$$u^* = \underset{u \in \mathcal{F}(\Omega)}{\operatorname{argmin}} \ \|u - f\|_2^2 + \lambda \, \mathrm{TV}(u) \tag{1.12}$$

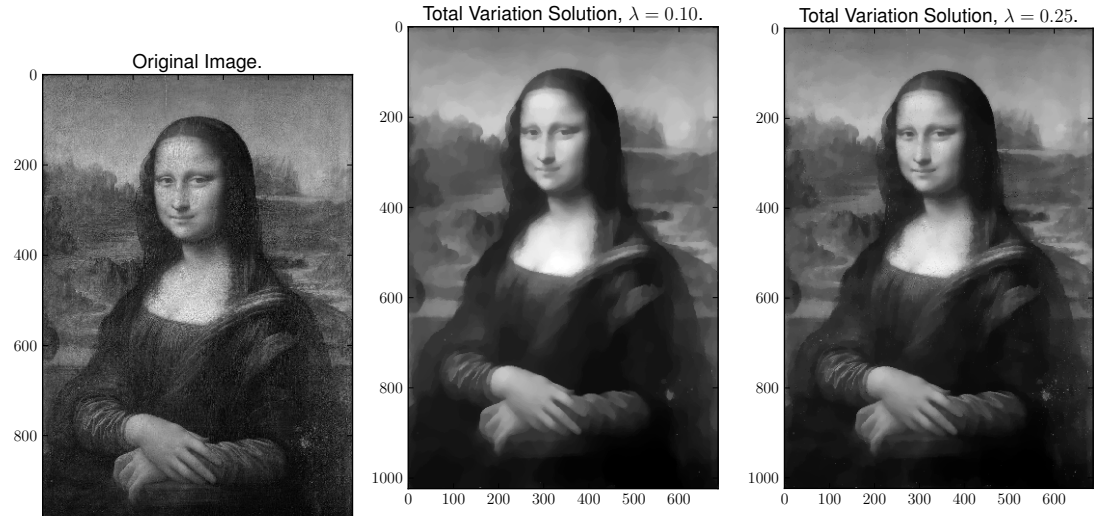
$$= \underset{u \in \mathcal{F}(\Omega)}{\operatorname{argmin}} \int_\Omega (u(\mathbf{x}) - f(\mathbf{x}))^2 \mathrm{d}\mathbf{x} + \lambda \, \mathrm{TV}(u) \tag{1.13}$$

This model – often called the Rudin-Osher-Fatimi (ROF) model after the authors – was originally proposed in Rudin et al. [1992] as a way of removing noise from images corrupted by Gaussian white noise; however, it has gained significant attention in a variety of other areas.

In the image analysis community, it has been used often as a model for noise removal, with the solution of (1.10) being the estimation of the noise-free image. In addition, the value of the total variation term in the optimal solution is often used as a method of edge detection, as the locations of the non-zero gradients tend to track regions of significant change in the image.

Beyond image analysis, a number of recent models have used total variation regularization as a general way to enforce similar regions of the solution to have a common value. In this way, it is a generalization of the behavior of the Fused LASSO in one dimension; it tends to set regions of the image to constant values. As such, it is used in MRI reconstruction, electron tomography [Goris et al., 2012, Gao et al., 2010], MRI modeling [Michel et al., 2011, Keeling et al., 2012], CT reconstruction [Tian et al., 2011], and general ill-posed problems [Bardsley and Luttman, 2009].

The total variation regularizer tends to have a smoothing effect around regions of transition, while also setting similar regions to a constant value. As such, it has proven to be quite effective for removing noise from images and finding the boundaries of sufficiently distinct regions. The locations where the norm of the gradient $\mathrm{TV}(u)$ is nonzero correspond to boundaries of the observed process in which the change is the greatest. A rigorous treatment of the theoretical aspects surrounding using this as a regularization term can be found in



(a) Original Image

(b) TV minimized image with $\lambda' = 0.1$.

(c) TV minimized image with $\lambda' = 0.25$.

Figure 1.1: An example of Total Variation Minimization for noise removal on Leonardo da Vinci's Mona Lisa. Different values of the regularization parameter produce different results, with the higher value of λ' smoothing the image less but removing less of the noise.

Bellettini, Caselles, and Novaga [2002], Ring [2000] and Chan and Esedoglu [2005]. A complete and treatment of this topic in practice can be found in a number of surveys, including Darbon and Sigelle [2006], Allard [2007, 2008, 2009]; in particular, see Caselles, Chambolle, and Novaga [2011] and Chambolle, Caselles, Cremers, Novaga, and Pock [2010]. As our purpose in this work is to present a treatment of the underlying theoretical structures, we refer the reader to one of the above references for use practical image analysis. Still, several points are of statistical interest, which we discuss now.

Several generalizations of the traditional Rudin-Osher-Fatimi model have been proposed for different models of the noise and assumptions about the boundaries of the images. Most of these involve changes to the loss or log-likelihood term but preserve the total variation term as the regularizer. In general, we can treat u^* as the estimation of

$$u^* = \underset{u \in \mathcal{F}(\Omega)}{\operatorname{argmin}} \; \mathcal{L}(u, f) + \lambda \, \mathrm{TV}(u) \tag{1.14}$$

where $\mathcal{L}$ is a smooth convex loss function.

One option is to use L_1 loss for $\mathcal{L}$, namely

$$\mathcal{L}_1 = \|f - u\|_1 , \tag{1.15}$$

as a way of making the noise more robust to outliers. This model is explored in Bect et al. [2004], Chan and Esedoglu [2005] and the survey papers above. These perform better under some types of noise and have also become popular [Goldfarb and Yin, 2009].

Similarly, many models for image denoising – often when the image comes from noisy sensing processes in physics, medical imaging, or astronomy – involve using a Poisson likelihood for the observations. Here, we wish to recover a density function where we observe Poisson counts in a number of cells with rate proportional to the true density of the underlying process. Total variation regularization can be employed to deal with low count rates. Here, total variation regularization is used to recover the major structures in the data. In this case, the optimal loss function is given by

$$\mathcal{L}(u, f) = \int_\Omega (u(\mathbf{x}) - f(\mathbf{x}) \log u(\mathbf{x}))\mathrm{d}\mathbf{x}, \qquad u \geq 0. \tag{1.16}$$

Research on this, including optimization techniques, can be found in Bardsley and Luttman [2009], Bardsley [2008], Sawatzky et al. [2009].

Also of interest in the statistics community, Wunderli [2013] replaces the squared-norm fit penalty in (1.13) with the quantile regression penalty from Koenker and Bassett Jr [1978], where loss function is replaced with

$$\mathcal{L}(u, f) = |f(\mathbf{x}) - u(\mathbf{x})| \times \left\{ \begin{array}{ll} 1 - \beta & f(\mathbf{x}) \geq u(\mathbf{x}) \\ \beta & f(\mathbf{x}) < u(\mathbf{x}) \end{array} \right\}. \tag{1.17}$$

for which the author proves the existence, uniqueness, and stability of the solution.

One of the variations of total variation minimization is the Mumford-Shah model. In this model, used primarily for segmenting images, regions of discontinuity are handled explicitly Mumford and Shah [1989]. In this model, the energy which is minimized is

$$E(u, \Gamma) = \|f - u\|_2^2 + \int_{\Omega \backslash \Gamma} \|\nabla u\|_2^2 \, \mathrm{d}\mu + v \, \|\Gamma\| \tag{1.18}$$

where Γ is a collection of curves and $\|\Gamma\|$ is the total length of these curves. Thus the function is allowed to be discontinuous on Γ; these are then taken to be the segmentations of the image. This model has also attracted a lot of attention in the vision community as a way of finding boundaries in the image [Chan and Vese, 2000, Pock et al., 2009, El Zehiry et al., 2007]. The Bayesian model of this is as a type of mixture model in which the mean function and noise variance differs between regions [Brox and Cremers, 2007]. Unlike general TV regularization, however, it is typically focused exclusively on finding curves for image segmentation, rather than on estimating the original image. While it shares a number of close parallels to our problem, particularly in the graph cut optimizations used [El Zehiry et al., 2007], our method does not appear to generalize to this problem.

The optimization of the total variation problem has garnered a lot of attention as well. An enormous number of approaches have been proposed for this problem. These largely fall into the category of graph based algorithms, active contour methods, or techniques based around partial differential equations. These are discussed in more depth in chapter 5, where we mention them in introducing our approach to the problem.

1.3 Main Results

The primary purpose of our work is to present a theory of the underlying structure connecting the optimization of the above models over dependent variables and recent results in combinatorial optimization, particularly submodular function minimization. Our theory connects and makes explicit a number of connections between known results in these fields, many of which we extend in practically relevant ways. The insights gained from our theory motivate a family of novel algorithms for working with these models. Furthermore, we are able to give a complete description of the structure of the regularization path; this result is also completely novel.

One primary practical contribution is the development of methods to efficiently optimize functions of the following form, which we denote as ($\Re_{\mathrm{B}}$)

$$\mathbf{u}^*(\lambda) = \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^n} \|\mathbf{u}-\mathbf{a}\|_2^2 + \lambda \left[\sum_i \xi_i(u_i) + \sum_{i,j} w_{ij}|u_i - u_j| \right], \qquad (\Re_{\mathrm{B}})$$

with $\mathbf{a} \in \mathbb{R}^n$, λ is a non-negative regularization parameter, $w_{ij} \geq 0$ controls the regularization of dependent variables, and $\xi_i(u_i)$ is a convex piecewise linear function, possibly the common L_1 penalty $\xi_i(u_i) = |u_i|$.

We develop a novel algorithm for ($\Re_{\mathrm{B}}$) based on network flows and submodular optimization theory, and completely solve the regularization path as well over $\lambda > 0$. The crux of the idea is to construct a collection of graph-based binary partitioning problems indexed by a continuous parameter. With our construction, we show that the points at which each node exactly gives the solution to ($\Re_{\mathrm{B}}$). We develop algorithms for this that scale easily to millions of variables, and show that the above construction also unlocks the nature of the regularization path.

These functions arise in two active areas of interest to the statistics and machine learning communities. The first is in the optimization of penalized regression functions where we wish to find

$$\mathbf{u}^*(\lambda) = \underset{\mathbf{u} \in \mathbb{R}^n}{\operatorname{argmin}} \; \mathcal{L}(\mathbf{u}, \mathbf{y}) + \lambda \left[\sum_i \xi_i(u_i) + \sum_{i,j} w_{ij} |u_i - u_j| \right], \tag{$\Re_{\mathrm{L}}$}$$

where $\mathbf{y}$ is a collection of observed response and $\mathcal{L}(\mathbf{u}, \mathbf{y})$ is a smooth, convex log-likelihood term. Throughout our work, we denote this problem as ($\Re_{\mathrm{L}}$). In the general case, we handle this problem through the use of the proximal gradient methods, a variant of sub-gradient methods, which we described below. The inner routine involves solving ($\Re_{\mathrm{B}}$).

Finally, ($\Re_{\mathrm{B}}$) also extends to the continuous variational estimation problem of Total Variation minimization from section 1.2.2. where we wish to minimize

$$u^* = \underset{\mathcal{F}(\Omega)}{\operatorname{argmin}} \; \|u - f\|_2^2 + \lambda \, \mathrm{TV}(u) \tag{$\Re_{\mathrm{TV}}$}$$

where $\mathcal{F}(\Omega)$ and $\mathrm{TV}(u)$ are defined in (1.10) – (1.13). We show in chapter 5 that this problem can be solved using the methods developed for ($\Re_{\mathrm{B}}$). Thus we not only present an efficient algorithm for ($\Re_{\mathrm{TV}}$), we also present the first algorithm for finding the full regularization path of this problem.

1.3.1 Outline

This thesis is laid out as followed. The rest of this chapter presents a survey of the background to the current problem, describing the current state of the research in the relevant areas and laying out the reasons it is of interest to the statistics and machine learning communities. The chapter ends with a concise list of our main contributions.

Chapter 2 lays out the basic framework connecting network flows, submodular function minimization, and the optimization of ($\Re_{\mathrm{B}}$). While some of the results in this chapter were discovered independently, our theory pulls them together explicitly in a common framework, and provides more direct proofs of their correctness. Additionally, our approach motivates a new algorithm to exactly solve the resulting optimization problem.

Chapter 3 presents entirely novel results. We extend the basic theory of chapter 2 using a novel extension lemma to allow for a general size-biasing measure on the different components of the input. The theoretical results are proved for general submodular functions, extending the state-of-the-art in this area. In the context of our problem, we use these results to extend the algorithmic results of chapter 2. In particular, this opens a way to handle the $\xi_i(u_i)$ term in ($\Re_{\mathrm{B}}$) above, significantly extending the results of Mairal et al. [2011] related to chapter 2.

Chapter 4, builds upon the previous chapters to form a complete description of the structure of the regularization path of ($\Re_{\mathrm{B}}$) as λ varies. We show this path is formed from a network of linear splits and joins and can be calculated exactly using a novel algorithm. The theoretic structure of the previous chapters is key to proving the correctness of this result.

Then, in chapter 5, we examine the total variation problem of (1.13), in which we find the best estimation of an observed response function while controlling the total variation of the estimator. We review a collection of known results that allows this regularizer to be encoded on a lattice-type graph structure that maps to the framework of the previous chapters. We show that this approximation can be made arbitrarily exact, and we prove rigorous bounds on the accuracy's of the approximation in terms of the fineness of the lattice. This also gives the first known algorithm to completely calculate the regularization path of the total

variation problem. Our approach is demonstrated with several experimental results.

In chapter 6, we prove a consistency result from sparse estimation problems when the individual variables are closely correlated – possibly defined as nodes on a graph, with neighbors being similar – and one wishes to recover a correlated set of these nodes that could produce the final solution. In this context, we have that the solution variables can be exchanged with other variable but with a (possible) penalty on the fit; this is far different from classical compressed sensing, which assumes that the input variables one wishes to recover are orthogonal. Here, we study convergence using a flow-based metric that permits a type of exchangability among the solutions; namely, solution values can be exchanged between groups as defined on a graph structure with some penalty. This situation encompasses many of the algorithmic approaches we give here. By giving us a solid theoretical basis for using these methods for sparse recovery, it gives a fitting concluding point to this work. Finally, in chapter 7, we give a full summary of our contributions.

1.4 Optimization of Regularized Models

Regularized models have gained significant attention in both the statistics and machine learning communities. Intuitively, they provide a way of imposing structure on the problem solution. This facilitates the accurate and tractable estimation of high-dimensional variables or other features, such as change points, that are difficult to do in a standard regression context. Furthermore, these methods have been well studied both from the theoretical perspective – correct estimation is guaranteed under a number of reasonable assumptions – and the optimization perspective – efficient algorithms exist to solve them. Not surprisingly, these approaches have gained significant popularity in both the statistics and machine learning communities. We will describe several relevant examples below.

In the general context, we consider finding the minimum of

$$\gamma(\mathbf{u}) = \mathcal{L}(\mathbf{u}, \mathbf{y}) + \lambda\Phi(\mathbf{u}), \tag{1.19}$$

where $\mathcal{L}(\mathbf{u}, \mathbf{y})$ is a smooth convex Lipshitz loss function, $\Phi(\mathbf{u})$ is a regularization term, and $\lambda \geq 0$ controls the amount of the regularization. In statistical models, $\mathcal{L}$ is typically given by the log-likelihood of the model. The regularization term $\Phi(\mathbf{u})$ is required to be convex

but is not required to be smooth; in practice it is is often a regularization penalty that enforces some sort of sparseness or structure. A common regularization term is simply the L_1 norm on $\mathbf{u}$, i.e.

$$\Phi_1(\mathbf{u}) = \|\mathbf{u}\|_1 \tag{1.20}$$

which gives the standard LASSO problem proposed in [Tibshirani, 1996] and described in section 1.2.

A number of algorithms have been proposed for the optimization of this particular problem Friedman et al. [2010], Efron et al. [2004]. Most recently, a method using proximal operators has been proposed called the Fast Iterative Soft-Threshold Algorithm [Beck and Teboulle, 2009]; this achieves theoretically optimal performance – identical to the $\mathcal{O}(1/k^2)$ lower bound of the convergence rate by iteration of optimization of general smooth, convex functions [Nesterov, 2005]. The method is based on the proximal operators we discuss next, and is described in detail, along with theoretical guarantees, in Beck and Teboulle [2009]. This is the method that immediately fits with our theoretical results.

1.4.1 Proximal Operators

Proximal gradient methods Nesterov [2007], Beck and Teboulle [2009] are a very general approach to optimizing (1.19). At a high level, they can be understood as a natural extension of gradient and sub-gradient based methods designed to deal efficiently with the non-smooth component of the optimization. The motivation behind proximal methods is the observation that the objective $\gamma(\mathbf{u})$ is the composition of two convex functions, with the non-smooth part isolated in one of the terms. In this case, the proximal problem takes the form

$$\gamma_P(\mathbf{u}, \hat{\mathbf{u}}) = f(\hat{\mathbf{u}}) + (\mathbf{u} - \hat{\mathbf{u}})^T \nabla f(\hat{\mathbf{u}}) + \lambda\Phi(\mathbf{u}) + \frac{L}{2}\|\mathbf{u} - \hat{\mathbf{u}}\|_2^2 \tag{1.21}$$

where we abbreviate $f(\mathbf{u}) = \mathcal{L}(\mathbf{u}, \mathbf{y})$. L is the Lipshitz constant of f, which we assume is differentiable. We rewrite this as

$$\gamma'_P(\mathbf{u}, \hat{\mathbf{u}}) = \frac{1}{2}\left\|\mathbf{u} - \left[\hat{\mathbf{u}} - \frac{1}{L}\nabla f(\hat{\mathbf{u}})\right]\right\|_2^2 + \frac{\lambda}{L}\Omega(\mathbf{u}). \tag{1.22}$$

The minimizing solution of (1.22) forms the update rule. Intuitively, at each iteration, proximal methods linearize the objective function around the current estimate of the solution,

$\hat{\mathbf{u}}$, then update this with the solution of the proximal problem. As long as (1.21) can be optimized efficiently, the proximal operator methods give excellent performance.

While this technique has led to a number of algorithmic improvements for simpler regularizers, it has also opened the door to the practical use of much more complicated regularization structures. In particular, there has been significant interest in more advanced regularization structures, particularly ones involving dependencies between the variables. These types of regularization structures are the primary focus of our work.

1.4.2 Related Work

Finally, of most relevance to our work here, is a series of recent papers by Francis Bach and others, namely Bach, Jenatton, Mairal, and Obozinski [2012], Mairal, Jenatton, Obozinski, and Bach [2010], and Mairal, Jenatton, Obozinski, and Bach [2011], that independently discover some of our results. In these papers, combined with the more general work of Bach, Jenatton, Mairal, and Obozinski [2011], Bach [2010b] and Bach [2010a, 2011], several of the results we present here were independently proposed, although from a much different starting point. In particular, he proposes using parametric network flows to solve the proximal operator for $\Phi(\mathbf{u}) = \sum_{i,j} w_{ij}|u_i - u_j|$ and that this corresponds to calculating the minimum norm vector of the associated submodular function. These results are discussed in more detail in chapter 2.

While our results were discovered independently from these, we extend them in a number of important ways. First, we propose a new algorithm to exactly solve the linear parametric flow problem; this is a core algorithm in these problems. Second, through the use of the weighting scheme proposed in chapter 3, we develop a way to include general convex linear functions directly in the optimization procedure. This expands the types of regularization terms that can be handled as part of the proximal operator scheme.

1.5 Combinatorial Optimization and Network Flows

The final piece of background work we wish to present forms the last pillar on which our theory is built. Core to the theory and the algorithms is the simple problem of finding the minimum cost partitioning of a set of nodes $\mathcal{V}$ in which relationships between these nodes

are defined by a graph connecting them. The contribution laid out in the next chapters is built on a fundamental connection between a simple combinatorial problem – finding a minimum cost cut on a graph – and the convex optimization problems described in section 1.4.

A fundamental problem in combinatorial optimization is finding the minimum cost cut on a directed graph. This problem is defined by $(\mathcal{V}, \mathcal{E}, \mathbf{c}, s, t)$, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of directed edges connecting two nodes in $\mathcal{V}$. We consistently use n to denote the number of nodes and, without loss of generality, assume $\mathcal{V} = \{1, 2, ..., n\}$, i.e. the nodes are refered to using the first n counting numbers. Similarly, the edges are denoted using pairs of vertices; thus $\mathcal{E} \subseteq \{(i, j) : i, j \in \mathcal{V}\}$. Here, (i, j) denotes an edge in the graph going from i to j. For an *undirected graph*, we simply assume that $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$, and that $c_{ij} = c_{ji}$ for all $i, j \in \mathcal{V}$. $\mathbf{c}$ associates a cost with each of the edges in $\mathcal{E}$. I.e. maps from a pair of edges to a non-negative cost, i.e. $c_{ij} \in \mathbb{R}^+$ for $(i, j) \in \mathcal{E}$.

The letters s and t here denote specific nodes not in the node set $\mathcal{V}$. For reasons that will become clear in a moment, s is called the *source* node and t is called the *sink* node. We treat s and t as valid nodes in the graph and define the cost associated with an edge from s to i as c_{si}; analogously, c_{it} denotes the cost of an edge from i to the sink t. These nodes are treated specially in the optimization, however; the *minimum cut problem* is the problem of finding a cut in the graph separating s and t such that the total cost of the edges cut is minimal. Formally, we wish to find a set $S^* \subseteq \mathcal{V}$ satisfying

$$S^* \in \operatorname*{Argmin}_{S \subseteq \mathcal{V}} \left[\sum_{\substack{i \in S \\ j \in (\mathcal{V} \setminus S)}} c_{ij} \right] + \left[\sum_{i \in S} c_{it} \right] + \left[\sum_{i \in \mathcal{V} \setminus S} c_{si} \right] \tag{1.23}$$

where Argmin with a capital A returns the set of minimizers, as there may be multiple partitions S^* achieving the minimum cost.

The above lays out the basic definitions needed for our work with graph structures. This problem is noteworthy, however, as it can be solved easily by finding the maximal flow on the graph – one of the fundamental dualities in combinatorial optimization is the fact that the cut edges defining the minimum cost partition are the saturated edges in the maximum flow from s to t in the equivalent network flow problem. It is also one of the simplest

practical examples of a submodular function, another critical component of our theory. We now describe these concepts.

1.5.1 Network Flows

The network flow problem is the problem of pushing as much "flow" as possible through the graph from s to t, where the capacity of each edge is given by the cost mapping $\mathbf{c}$ above. The significance of this problem is the fact that it map directly to finding the minimum cost partition in a graph [Dantzig and Fulkerson, 1955, Cormen et al., 2001, Kolmogorov and Zabih, 2004]. The saturating edges of a maximizing network flow – those edges limiting any more flow from being pushed through the graph – defines the optimal cut in the minimum cut partitioning. This result is one of the most practical results of combinatorial optimization, as many problems map to the partitioning problem above, and one and simple algorithms exist for solving network flow problems.

In the network flow problem, we wish to construct a mapping $\mathbf{z}$ that represents "flow" from s to t. A mapping $\mathbf{z}$ is a valid flow if, for all nodes in $\mathcal{V}$, the flow going into the each node is the same as the flow leaving that node. Specifically,

Definition 1.5.1 ((Valid) Flow)**.** *Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{c}, s, t)$ as described above. Then z_{ij}, $i, j \in \{s, t\} \cup \mathcal{V}$, is a flow on $\mathcal{G}$ if*

$$0 \leq z_{ij} \leq c_{ij} \;\; \forall\, i, j \in \mathcal{V} \tag{1.24}$$

$$\sum_{i \in \mathcal{V} \cup \{s\}} z_{ij} = \sum_{k \in \mathcal{V} \cup \{t\}} z_{jk}, \tag{1.25}$$

where for convenience, we assume that $z_{ii} = 0$ and $z_{ij} = c_{ij} = 0$ if $(i, j) \notin \mathcal{E}$.

Furthermore, we say that an edge (i, j) is saturated if $z_{ij} = c_{ij}$, i.e. the flow on that edge cannot be increased.

Since we frequently talk about flows in a more informal sense, we use the term "valid flow" to reference this formal definition.

It is easy to show that the amount of flow leaving s is the same as the amount of flow leaving t, i.e.

$$\text{Total Flow} = \sum_{i \in \mathcal{V}} z_{si} = \sum_{i \in \mathcal{V}} z_{it}. \tag{1.26}$$

this *total flow* is what we wish to maximize in the maximum flow problem, i.e. we wish to find a *maximal flow* $\mathbf{z}^*$ such that

$$\mathbf{z}^* \in \underset{\mathbf{z} \,:\, \mathbf{z} \text{ is a valid flow on } \mathcal{G}}{\operatorname{Argmin}} \sum_{i \in \mathcal{V}} z_{si}. \tag{1.27}$$

The canonical *min-cut-max-flow* theorem [Dantzig and Fulkerson, 1955, Cormen et al., 2001] states that the set of edges saturated by all possible maximal flows defines a minimum cut in the sense of (1.23) above. The immediate practical consequence of this duality is that we are able to find the minimum cut partitioning quickly as fast algorithms exist for finding a maximal flow $\mathbf{z}^*$. We outline one of these below, but first we generalize the idea of a *valid flow* in two practically relevant ways.

1.5.2 Preflows and Pseudoflows

Two extensions of the idea of a flow $\mathbf{z}$ on a graph relaxes the equality in the definition of a flow. Relaxing the inequality is key to several algorithms, and forms some interesting connections to the theory we outline. The equality constraint is effectively replaced with an excess$(\cdot)$ function that gives the excess flow at each node; if the total amount of flow into and out of a node i is equal, then excess$(i) = 0$. Formally,

Definition 1.5.2 (Preflow)**.** *Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{c}, s, t)$ as in definition 1.5.1. Then z_{ij}, $i, j \in \{s, t\} \cup \mathcal{V}$, is a flow on $\mathcal{G}$ if*

$$0 \leq z_{ij} \leq c_{ij} \;\; \forall\, i, j \in \mathcal{V} \tag{1.28}$$

$$\sum_{i \in \mathcal{V} \cup \{s\}} z_{ij} \geq \sum_{k \in \mathcal{V} \cup \{t\}} z_{jk}. \tag{1.29}$$

With this definition, we define the excess$(\cdot)$ function as

$$\text{excess}(i) = \left[\sum_{i \in \mathcal{V} \cup \{s\}} z_{ij} \right] - \left[\sum_{k \in \mathcal{V} \cup \{t\}} z_{jk} \right]. \tag{1.30}$$

It is easy to see that if $\mathbf{z}$ is a preflow, excess$(i) \geq 0$ for all nodes i. Maintaining a valid preflow is one of the key invariants in the common push-relabel algorithm discussed below.

A *Pseudoflow* is the weakest definition of a flow; it relaxes (1.29) completely, allowing there to be both excesses and deficits on the nodes of the graph. However, it does add in an

additional constraint, namely that all edges connected to the source and sink are saturated. Formally,

Definition 1.5.3 (Pseudoflow)**.** *Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{c}, s, t)$ as in definition 1.5.1. Then z_{ij}, $i, j \in \{s, t\} \cup \mathcal{V}$, is a flow on $\mathcal{G}$ if*

$$0 \leq z_{ij} \leq c_{ij} \ \ \forall\, i, j \in \mathcal{V} \tag{1.31}$$

$$z_{si} = c_{si} \ \ \forall\, i \in \mathcal{V} \tag{1.32}$$

$$z_{it} = c_{it} \ \ \forall\, i \in \mathcal{V} \tag{1.33}$$

Note that now the excess(i) function can be either positive or negative. A pseudoflow has some interesting geometric and algorithmic properties, It was described in Hochbaum [1998] and an efficient algorithm for solving network flows based on pseudoflows was presented in Hochbaum [2008]. We mention it here to preview our results in chapter 2: the pseudoflow matches up directly with the base polytope, one of the fundamental structures in submodular function optimization.

1.5.3 Network Flow Algorithms

Finding the minimum cut or the maximum flow on a network is one of the oldest combinatorial optimization problems, and, as such, it is also one of the most well studied. It would be impossible to detail the numerous algorithms to solve the network flow problem here – Schrijver [2003] lists over 25 *survey* papers on different network flow algorithms. Many of these are tailored for different types of graphs (sparse vs. dense) and other variations of the problem.

Perhaps the most well-known algorithm for Network flow analysis is the push-relabel method. Variants of it achieve the best known performance guarantees for a number of problems of interest. It is simple to implement; essentially, each node has a specific height associated with it that represents (informally) the number of edges in the shortest unsaturated path to the sink. Each node can have an excess amount of flow from the source. At each iteration, a node with excess either pushes flow along an unsaturated edge to a neighbor with a lesser height, or increments its height. Termination occurs when the source

node is incremented to $|\mathcal{V}|+1$, where $|\mathcal{V}|$ is the number of nodes – at this point, there is no possible path to the sink and the maximum cut can be found. For more information on this, along with other information on this algorithm, see Cormen et al. [2001] or Schrijver [2003]. This algorithm runs at the core of our network flow routines. This algorithm is guaranteed to run in polynomial time; in the general case, it runs in $\mathcal{O}\Big(|\mathcal{V}|^2\,|\mathcal{E}|\Big)$ time, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of edges. However, it can be improved to $\mathcal{O}\big(|\mathcal{V}|\,|\mathcal{E}|\log(\big|\mathcal{V}^2\big|\,/\,|\mathcal{E}|)\big)$ using dynamic tree structures [Schrijver, 2003].

1.5.4 The Parametric Flow Problem

One variation of the regular network flow problem is the *parametric flow problem* [Gallo et al., 1989]. The general parametric flow problem is a simple modification of the maximum flow problem given above, except that now c_{si} is replaced with a monotonic non-decreasing function $c_{si}(\beta)$,

$$\mathbf{z}^*(\beta) \in \underset{\substack{\mathbf{z}\,:\,\mathbf{z}\text{ is a valid flow on }\mathcal{G}(\beta);\\ \text{Capacity of edge }(s,i)\text{ given by }c_{si}(\beta)}}{\operatorname{Argmin}} \sum_{i\in\mathcal{V}} z_{si}. \tag{1.34}$$

Gallo et al. [1989] showed that this problem could be solved for a fixed sequence $\beta_1 < \beta_2 < \cdots < \beta_m$ in time proportional to the time of a single run of the network flow problem by exploiting a nestedness property of the solutions. Namely, as β increase, the minimum cut of the optimal solution moves closer on the graph to the source. More formally, if $\beta_1 < \beta_2$, then

$$S^*(\beta_1) \subseteq S^*(\beta_2), \tag{1.35}$$

where $S^*(\beta)$ is an optimal cut at β in the sense of (1.23).

If $c_{si}(\beta)$ is a linear function with positive slope, we call this problem the the *linear parametric flow problem.* In chapters 2 and 3, we develop an exact algorithm for this problem that does not require a predetermined sequence of β to solve. This method is one of the key routines in the algorithms for general optimization.

1.5.5 Connections to Statistical Problems

One of the recent applications of graph partitioning is in finding the minimum energy state of a binary Markov random field. In particular, if we can model the pairwise interactions over binary variables $x_1, ..., x_n$ on the graph using unary and pairwise potential functions $E_i^1(x_i)$ and $E_{ij}^2(x_i, x_j)$, then the minimum energy solution can be found using the maximum cut on a specially formulated graph, provided that $E_{ij}^2(0,0) + E_{ij}^2(1,1) \leq E_{ij}^2(0,1) + E_{ij}^2(1,0)$ for all i, j. This application of network flow solvers has had substantial impact in the computer vision community, where pairwise potential functions satisfying this condition are quite useful.

We discuss the details of this application in chapter 2, where we start with the theory behind these connections as first building block of our other results. Ultimately, we show that network flow algorithms can be used to solve not only this problem, but the much more general optimization problems described in section 1.4 as well. Through these connections, we hope to bring the power of network flow solvers into more common use in the statistical community.

1.6 Submodular Functions

The problem of finding a minimal partition in a graph is a special case of a much larger class of combinatorial optimization problems called *submodular function minimization.* Like the problem of finding a minimal-cost partitioning of a graph given in (1.23), this optimization problem involves finding the minimizing set over a submodular function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$, where $2^{\mathcal{V}}$ denotes the collection of all subsets of a *ground set* $\mathcal{V}$. Given this, f is submodular if, for all sets $S, T \subseteq \mathcal{V}$,

$$f(S) + f(T) \geq f(S \cap T) + f(S \cup T). \tag{1.36}$$

This condition is seen as the discrete analogue of convexity. It is sufficient to guarantee that a minimizing set S^* can be found in polynomial time complexity.

An alternative definition of submodularity involves the idea of *diminishing returns* [Fujishige, 2005]. Specifically f is submodular if and only if for all $S \subseteq T \subset \mathcal{V}$, and for all

$i \in \mathcal{V} \backslash T$,

$$f(S \cup \{i\}) - f(S) \leq f(T \cup \{i\}) - f(T). \tag{1.37}$$

This property is be best illustrated with a simple example. Let $A_1, A_2, ..., A_n \subseteq \mathcal{A}$ be n subsets of a larger set $\mathcal{A}$, and define

$$f_{\text{coverage}}(S) = \left| \bigcup_{i \in S} A_i \right|, \tag{1.38}$$

so $f_{\text{coverage}}(S)$ measures the coverage of the set $\cup_{i \in S} A_i$. In this context, it is easy to see that $f_{\text{coverage}}(S)$ satisfies (1.37).

$f_{\text{coverage}}(S)$ is an example of a *monotone* submodular function as adding new elements to S is only going to increase the value of $f_{\text{coverage}}(S)$. However, many practical examples do not fall into this category. In particular, the graph cut problem given in equation (1.23) above is non-monotone submodular:

$$f_{\text{gc}}(S) = \left[\sum_{\substack{i \in S \\ j \in (\mathcal{V} \backslash S)}} c_{ij} \right] + \left[\sum_{i \in S} c_{it} \right] + \left[\sum_{i \in \mathcal{V} \backslash S} c_{si} \right]. \tag{1.39}$$

We examine this particular example in more detail in chapter 2, where we prove f_{gc} is indeed submodular.

Submodular function optimization has gained significant attention lately in the machine learning community, as many other practical problems involving the sets can be phrased as submodular optimization problems. It has been used for numerous applications in computer vision, language modeling [Lin and Bilmes, 2010, 2012], clustering [Narasimhan, Jojic, and Bilmes, 2005, Narasimhan and Bilmes, 2007], computer vision [Jegelka and Bilmes, 2011, 2010], and many other domains. This field is quite active, both in terms of theory and algorithms, and we contribute some novel results to both areas in chapter 3.

The primary focus of our work has been developing a further connection between these graph problems and submodular optimization theory. Our approach, however, is the reverse of much of the previous work. Ultimately, we attempted to map the network flow problems back to the submodular optimization problems. Surprisingly, this actually opened the door to several new theoretical results for continuous optimization, and, in particular, to efficient solutions of the optimization problems given in section 1.2.

1.6.1 Some Formalities

The theory surrounding submodular optimization, and combinatorial optimization in general, is quite deep. Many of the results underlying submodular optimization require a fairly substantial tour of the theory of the underlying structures; good coverage of these results is found in [Fujishige, 2005] and [Schrijver, 2003]. Most of these results are not immediately relevant to our work, so we leave them to the interested reader. However, several additional results are needed for some of the proofs we use later.

As with the discussion of problem, we assume that $\mathcal{V} = \{1, 2, 3, ..., n\}$; our notation intentionally matches that of the minimum cut problem defined above and is consistent throughout our work. We refer to $\mathcal{V}$ as the *ground set*. Formally, f can be restricted to map from a collection of subsets of $2^{\mathcal{V}}$, which we refer to consistently as $\mathcal{D}$, with $\mathcal{D} \subseteq 2^{\mathcal{V}}$. In the context of submodular functions, $\mathcal{D}$ must be closed under union and intersection and include $\mathcal{V}$ as an element[Fujishige, 2005]. In general, and except for some of our proofs in chapter 3, $\mathcal{D}$ can be thought of as $2^{\mathcal{V}}$.

1.6.2 Submodular Function Optimization

The first polynomial time algorithm for submodular function optimization was described in [Grötschel et al., 1993]; it used the ellipsoid method from linear programming [Chvtal, 1983]. While sufficient to prove that the algorithm can be solved in polynomial time, it was impractical to use on any real problems. The first *strongly* polynomial time algorithms – polynomial in a sense that does not depend on the values in the function – were proposed independently in Schrijver [2000] and Iwata, Fleischer, and Fujishige [2001]. The algorithm proposed by Schrijver runs in $\mathcal{O}\big(n^8 + \gamma n^7\big)$, where γ refers to the complexity of evaluating the function. The latter algorithm is $\mathcal{O}\big(\gamma n^7 \log n\big)$, which may be better or worse depending on γ. The weakly polynomial version of this algorithm runs in $\mathcal{O}\big(\gamma n^5 \log M\big)$, where M is the difference between maximum and minimum function values. Research in this area, however, is ongoing – a strongly polynomial algorithm that runs in $\mathcal{O}\big(n^6 + n^5\gamma\big)$ has been proposed by Orlin [2009].

In practice, the minimum norm algorithm – also called the Fujishige-Wolfe Algorithm –

is generally much faster, although it does not have a theoretical upper bound on the running time [Fujishige, 2005]. We discuss this algorithm in detail, as it forms the basis of our work. However, there are clear cases that occur in practice where this algorithm does not seem to improve upon the more complicated deterministic ones – in Jegelka, Lin, and Bilmes [2011], a running time of $\mathcal{O}(n^7)$ was reported. Thus the quest for practical algorithms for this problem continues; it is an active area of research.

Additionally, several other methods for practical optimization have been proposed for general submodular optimization or for special cases that are common in practice. Stobbe and Krause [2010] proposed an efficient method for submodular functions that can be represented as a decomposable sum of smaller submodular functions given by $g_i(|U_i \cap S|)$, where g_i is convex. In this particular case, the function can be mapped to a form that permits the use of nice numerical optimization techniques; however, many submodular functions cannot be minimized using this technique, and it can also be slow [Jegelka et al., 2011]. Along with analyzing the deficiencies of existing methods, Jegelka, Lin, and Bilmes [2011] propose a powerful approach that relies on approximating the submodular functions with a sequence of graphs that permit efficient optimization. This method is quite efficient on a number of practical problems, likely because the graph naturally approximates the underlying structure present in many real-world problems.

Most recently, in Iyer, Jegelka, and Bilmes [2013], another practical optimization method is proposed; at its core is a framework for both submodular minimization and maximization based on a notion of discrete sub-gradients and super-gradients of the function. While not having a theoretical upper bound itself, it performs efficiently in practice. This method is also noteworthy in that it can constrain the solution space in which other exact solvers operate, providing substantial speedups.

1.6.3 Geometrical Structures and the Minimum Norm Algorithm

A number of geometrical structures underpin the theory of submodular optimization. In particular, an associated *polymatroid* is defined as the set of points in $|\mathcal{V}|$-dimensional Euclidean space with sums of sets of the dimensions constrained by the function value of the

associated set.

For notational convenience, for a vector $\mathbf{x} \in \mathbb{R}^n$ and set $S \subseteq \mathcal{V}$, define

$$\mathbf{x}(S) = \sum_{i \in S} x_i. \tag{1.40}$$

In this way, $\mathbf{x}(S)$ forms a type of unnormalized set measure.

Now the polymatroid associated with f is defined as

$$P(f) = \left\{ \mathbf{x} \in \mathbb{R}^{|\mathcal{V}|} : \mathbf{x}(S) \leq f(S) \ \ \forall \, S \in \mathcal{D} \right\}, \tag{1.41}$$

$P(f)$ is fundamental to the theory behind submodular function optimization.

In our theory, we work primarily with the *base* of the polymatroid $P(f)$, denoted by $B(f)$. This is the extreme $(|\mathcal{V}| - 1)$-dimensional face of $P(f)$; it is defined as

$$B(f) = \left\{ \mathbf{x} \in \mathbb{R}^{|\mathcal{V}|} : \mathbf{x} \in P(f), \mathbf{x}(\mathcal{V}) = f(\mathcal{V}) \right\}. \tag{1.42}$$

In the case where the domain $\mathcal{D} = 2^{\mathcal{V}}$, $B(f)$ is a linear, convex compact set. Thus it is often refered to as the *Base Polytope* of f as it is compact.

The base $B(f)$ is particularly important for our work, as one of the key results from submodular function optimization is the *minimum norm algorithm*, which states effectively that sets S^* minimizing f are given by the sign of the point in $B(f)$ closest to the origin. This surprising result, while simple to state, takes a fair amount of deep theory to prove for general f; we state it here:

Theorem 1.6.1 ([Fujishige, 2005], Lemma 7.4.)**.** *For submodular function f defined on $2^{\mathcal{V}}$, let*

$$\mathbf{y}^* = \operatorname*{argmin}_{\mathbf{y} \in B(f)} \|\mathbf{y}\|_2 \tag{1.43}$$

and let $S_1^* = \{i : y_i^* < 0\}$ and $S_2^* = \{i : y_i^* \leq 0\}$. Then both S_1^* and S_2^* minimize $f(S)$ over all subsets $S \subseteq \mathcal{V}$. Furthermore, for all $S^\dagger \subseteq \mathcal{V}$ such that $f(S^\dagger) = \min_{S \in \mathcal{V}} f(S)$, $S_1^* \subseteq S^\dagger \subseteq S_2^*$.

A central aspect of our theory relies on this result. In general, it as one way of working the geometric structure of the problem. It turns out that in the case of graph partitioning, $B(f)$

takes on a particularly nice form. This allows us to very quickly solve the minimum norm problem here. We show the resulting minimum norm vector also gives us the full solution path over weighting by the cardinality of the minimizing set. While this basic result was independently discovered in Mairal, Jenatton, Obozinski, and Bach [2011], our approach opens several doors for theoretical and algorithmic improvements, which we outline in the next chapters.

Chapter 2

THE COMBINATORIAL STRUCTURE OF DEPENDENT PROBLEMS

2.1 Overview

The foundational aspect of our work is the result established in this chapter, namely an exploration of network flow minimization problems in terms of their geometric representation on the base polytope of a corresponding submodular problem. This representation is not new; it is explored in some depth as an illustrative example in [Fujishige, 2005] and connected to the minimum norm problems in Mairal et al. [2011]. Our contribution, however minor, is based on a simple transformation of the original problem that yields a particularly intuitive geometric form. The fruit of this transformation, however, is a collection of novel results of theoretic and algorithmic interest; in particular, we are able to exactly find the optimal $\mathbf{u}^*$ over the problem

$$\mathbf{u}^*(\lambda) = \underset{\mathbf{u}\in\mathbb{R}^n}{\operatorname{argmin}} \ \|\mathbf{u}-\mathbf{a}\|_2^2 + \lambda\sum_{i,j} w_{ij}|u_i - u_j|. \tag{2.1}$$

Recall that this problem was discussed in depth in section 1.4; here, $\mathbf{a} \in \mathbb{R}^n$ is given, and $\lambda \in \mathbb{R}^+$ and the weights $w_{ij} \in \mathbb{R}^+$ control the regularization.

In this chapter, we lay the theoretical foundation of this work, denoting connections to other related or previously known results. The contribution at the end is a strongly polynomial algorithm for the solution of a particular parametric flow problem; this algorithm follows naturally from this representation. In the next chapter, we extend the theory to naturally allow a convex piecewise-linear function $\xi_i(u_i)$ to be included in (2.1) to match ($\mathfrak{R}_\mathrm{B}$). The theory is extended in chapter 4 to allow computation of the full regularization path over λ, and applied to the total variation minimization problem in chapter 5.

Before presenting the results for the real-valued optimization of (2.1), we must first present a number of theoretical results using the optimization over sets $\mathcal{V} = \{1, 2, ..., n\}$. We begin with the simplest version of this optimization – finding the minimizing partition of a graph – and extend this result to general submodular functions later.

2.2 Basic Equivalences

We proceed by defining and establishing equivalences between three basic forms of the minimum cut problem on a graph. Recall from section 1.5 that the minimum cut problem

is the problem of finding a set $S^* \subseteq \mathcal{V}$ satisfying

$$S^* \in \underset{S \subseteq \mathcal{V}}{\text{Argmin}} \left[\sum_{\substack{i \in S \\ j \in (\mathcal{V} \setminus S)}} c_{ij} \right] + \left[\sum_{i \in S} c_{it} \right] + \left[\sum_{i \in \mathcal{V} \setminus S} c_{si} \right]. \tag{2.2}$$

As we mentioned earlier, several other important problems can be reduced to this form; in particular, finding the lowest energy state of a binary Markov random field, when the pairwise potential functions satisfy the submodularity condition, is equivalent to this problem. Our task now is to make this explicit.

We here show equivalences between three versions of the problem. The first is $\mathcal{P}_E$, which gives the standard energy minimization formulation, i.e. finding the MAP estimate of

$$p(\mathbf{x}) \propto \exp\left[\sum_{(i,j) \in \mathcal{E}} E^2_{ij}(x_i, x_j) + \sum_{i \in \mathcal{V}} E^1_i(x_i) \right], \qquad \mathbf{x} \in \{0,1\}^n \tag{2.3}$$

which is equivalent to finding

$$\mathbf{x}^* \in \underset{\mathbf{x} \in \{0,1\}^n}{\text{Argmin}} \sum_{(i,j) \in \mathcal{E}} E^2_{ij}(x_i, x_j) + \sum_{i \in \mathcal{V}} E^1_i(x_i). \tag{2.4}$$

$\mathcal{P}_Q$ gives the formulation as a quadratic binary minimization problem; here, the problem is to find $\mathbf{x} \in \{0,1\}^n$ that minimizes $\mathbf{x}^T \mathbf{Q} \mathbf{x}$ for an $n \times n$ matrix $\mathbf{Q}$. This version forms a convenient form that simplifies much of the notation in the later proofs. Finally, we show it is equivalent to the classic network flow formulation, denoted $\mathcal{P}_N$. This states the original energy minimization problem as the minimum *st*-cut on a specially formulated graph structure. The equivalence of these representations is well known [Kolmogorov and Zabih, 2004] and widely used, particularly in computer vision applications. We here present a different parametrization of the problem which anticipates the rest of our results.

The other unique aspect of our problem formulation is the use of a *size biasing term*; specifically, we add a term $\beta |S|$ to the optimization problem, where $\beta \in \mathbb{R}$ can be positive or negative. This term acts similarly to a regularization term in how it influences the optimization, but to think of it this way would lead to confusion as the true purpose of this formulation is revealed later in this chapter – ultimately, we show equivalence between the values of β at which the set membership of a node flips and the optimal values of the continuous optimization problem of ($\Re_B$).

Theorem 2.2.1. *Let $G = \mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V}$ is the set of vertices (assume $\mathcal{V} = \{1, ..., n\}$) and $\mathcal{E} \subseteq \{(i, j) : i, j \in \mathcal{V}\}$ is the set of edges. Without loss of generality, assume that $i < j \ \forall (i, j) \in \mathcal{E}$. Define $\mathcal{S}_E^*(\beta)$, $\mathcal{S}_Q^*(\beta)$, and $\mathcal{S}_N^*(\beta)$ as the sets of optimizing solutions to the following three problems, respectively:*

Energy Minimization: $\mathcal{P}_E(\beta)$**.** *Given an energy function $\mathbf{E}_1 = (E_i(x_i) : i \in \mathcal{V})$ defined for each vertex $i \in \mathcal{V}$ and a pairwise energy function $\mathbf{E}_2 = (E_{i,j}(x_i, x_j) : (i, j) \in \mathcal{E})$ defined for each edge $(i, j) \in \mathcal{E}$, with $E_{ij}(0, 0) + E_{ij}(1, 1) \leq E_{ij}(0, 1) + E_{ij}(1, 0)$, let*

$$\mathbf{X}^*(\beta) = \underset{\mathbf{x}\in\{0,1\}^n}{\text{Argmin}} \sum_{i\in\mathcal{V}} (E_i(x_i) - \beta x_i) + \sum_{\substack{(i,j)\in\mathcal{E} \\ i<j}} E_{ij}(x_i, x_j) \tag{2.5}$$

and let

$$\mathcal{S}_E^*(\beta) = \big\{\{i : x_i^* = 1\} \ : \mathbf{x}^* \in \mathbf{X}^*(\beta)\big\}. \tag{2.6}$$

Quadratic Binary Formulation: $\mathcal{P}_Q(\beta)$**.** *Given $\mathbf{E}_1$ and $\mathbf{E}_2$ as in $\mathcal{P}_E(\beta)$, define the $n \times n$ matrix $\mathbf{Q} = [q_{ij}]$ as:*

$$q_{ij} = \begin{cases} E_{ij}(1, 1) + E_{ij}(0, 0) - E_{ij}(0, 1) - E_{ij}(1, 0) & i < j \\ 0 & \textit{otherwise} \end{cases} \tag{2.7}$$

$$q_{ii} = (E_i(1) - E_i(0)) + \sum_{i'<i \ : \ (i',i)\in\mathcal{E}} (E_{i',i}(0, 1) - E_{i',i}(0, 0)) + \sum_{j>i \ : \ (i,j)\in\mathcal{E}} (E_{ij}(1, 0) - E_{ij}(0, 0)). \tag{2.8}$$

Suppose $q_{ij} \leq 0$ for $i \neq j$, and let

$$\mathbf{X}^* = \underset{\mathbf{x}\in\{0,1\}^n}{\text{Argmin}} \ \mathbf{x}^T(\mathbf{Q} - \beta\mathbf{I})\,\mathbf{x}, \tag{2.9}$$

and let

$$\mathcal{S}_Q^*(\beta) = \big\{\{i : x_i^* = 1\} \ : \mathbf{x}^* \in \mathbf{X}^*(\beta)\big\}. \tag{2.10}$$

Minimum Cut Formulation: $\mathcal{P}_N(\beta)$**.** *Let $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ be an augmented undirected graph with $\mathcal{V}' = \mathcal{V} \cup \{s, t\}$, where s and t represent source and sink vertices, respectively, and $\mathcal{E}' = \mathcal{E} \cup \{(s, i) : i \in \mathcal{V}\} \cup \{(i, t) : i \in \mathcal{V}\} \cup \{(j, i) : (i, j) \in \mathcal{E}\}$. Define capacities*

c_{ij}, $(i,j) \in \mathcal{E}$ on the edges as:

$$c_{si} = c_{si}(\beta) = [a_i(\beta)]^+ \quad c_{ji} = c_{ij} = -\frac{q_{ij}}{2},\ i < j \quad c_{jt} = c_{jt}(\beta) = [a_j(\beta)]^- \tag{2.11}$$

where

$$a_i(\beta) = \frac{1}{2} \sum_{i' : i' < i} q_{i',i} + (q_{ii} - \beta) + \frac{1}{2} \sum_{j\ :\ ie < j} q_{ij}. \tag{2.12}$$

Then the set of minimum cut solutions $\mathcal{S}_N^*(\beta)$ is given by

$$\mathcal{S}_N^*(\beta) = \underset{\substack{S \subset \mathcal{V}' \\ s \in S,\, t \in \mathcal{V}' \setminus S}}{\text{Argmin}} \sum_{(i,j) \in \delta(S, \mathcal{V}' \setminus S)} c_{ij}(\beta). \tag{2.13}$$

Then $\mathcal{P}_E(\beta)$, $\mathcal{P}_Q(\beta)$, and $\mathcal{P}_N(\beta)$ are equivalent in the sense that any minimizer of one problem is also a minimizer of the others; specifically,

$$\mathcal{S}_E^*(\beta) = \mathcal{S}_Q^*(\beta) = \mathcal{S}_N^*(\beta) \tag{2.14}$$

Proof. A reformulation of known results [Kolmogorov and Zabih, 2004], but given in section 2.7 for convenience. □

The primary consequence of this theorem is that solving the energy minimization problem can be done efficiently and exactly due to several types of excellent network flow solvers that make solving problems with millions of nodes routine [Boykov and Kolmogorov, 2004, Cormen et al., 2001, Schrijver, 2003]. Because of this, numerous applications for graphcuts have emerged in recent years for computer vision and machine learning. Our purpose, in part, is to expand the types of problems that can be handled with network flow solvers, and problems of interest in statistics in particular.

In our work, we alternate frequently between the above representations. For construction, the energy minimization problem is nicely behaved. In the theory, we typically find it easiest to work with the quadratic binary problem formulation due to the algebraic simplicity of working in that form. Again, however, each of these is equivalent; when it does not matter which form we use, we refer to the problem an solution set as $\mathcal{S}^*(\beta)$ and $\mathcal{P}(\beta)$ respectively.

2.2.1 Connections to Arbitrary Network Flow Problems

While theorem 2.2.1 lays out the equivalence between $\mathcal{P}_E(\beta)$ and $\mathcal{P}_Q(\beta)$ and a specific form of network flow problem $\mathcal{P}_N(\beta)$, for completeness we show that any network flow problem can be translated into the form $\mathcal{P}_N(\beta)$ and thus $\mathcal{P}_E(\beta)$ and $\mathcal{P}_Q(\beta)$. The two distinguishing aspects of $\mathcal{P}_N(\beta)$ are the facts that each node is connected to either the source or the sink, and that all the edges are symmetric, i.e. $c_{ij} = c_{ji}$ for all $i \neq j$. It is thus sufficient to show that an arbitrary flow problem can be translated to this form. For conciseness, assume that $\beta = 0$; the results can be adapted for other β easily.

Theorem 2.2.2. *Any minimum cut problem on an arbitrary, possibly directed graph can be formulated as a quadratic binary problem of the form $\mathcal{P}_Q(\beta = 0)$ as follows:*

1. *For all edges (i, j) such that $c_{ij} > c_{ji}$, add a path $s \to j \to i \to t$ with capacity $c_{ij} - c_{ji}$. That edge is now undirected in the sense that both directions have the same capacity, and the edges in the minimum cut are unchanged, as these paths will simply be eliminated by flow along that path.*
2. *Set $q_{ij} = c_{ij}$ for $i < j$ and $q_{ij} = 0$ for $i > j$.*
3. *Given q_{ij}, set $q_{ii} = (c_{si} - c_{it}) - 2\left[\sum_{i':i'<i} q_{i',i} + \sum_{j\,:\,i<j} q_{ij}\right]$.*

Using these steps, any minimum cut problem can be translated to $\mathcal{P}_Q$ in the sense that the set of minimizing solutions is identical.

Proof. Network flows are additive in the sense that increasing or decreasing the capacity of each edge in any path from s to t by a constant amount does not change the set of minimizing solutions of the resulting problem, even if new edges are added [Cormen et al., 2001, Schrijver, 2003]. Thus step (1) is valid. The rest follows from simple algebra. □

2.3 Network Flows and Submodular Optimization

Recall from section 1.6 that the minimum cut problem is a subclass of the more general class of submodular optimization problems. In the context of $\mathcal{P}_Q(\beta)$, it is easy to state a direct proof of this fact, additionally showing that here the submodularity of the pairwise terms is also necessary for general submodularity.

Theorem 2.3.1. *The minimization problem $\mathcal{P}_Q(\beta)$ can be expressed as minimization of a function $f_\beta : 2^{\mathcal{V}} \mapsto \mathbb{R}$, with*

$$f_\beta(S) = \sum_{\substack{i<j \\ i,j \in S}} q_{ij} + \sum_{i \in S} (q_{ii} - \beta). \tag{2.15}$$

Then f_β is submodular if and only if $q_{ij} \leq 0 \;\; \forall\, i, j \in S, i < j$.

Proof. One immediate proof of the *if* part follows from the fact that $\mathcal{P}_Q(\beta)$ can be expressed as the sum of submodular pairwise potential terms, and the sum of pairwise submodular functions is also submodular [Fujishige, 2005]. The direct proof, including both directions, is a simple reformulation of known results see [Kolmogorov and Zabih, 2004], but given in section 2.7 on page 47 for convenience. □

2.3.1 Geometric Structures

We are now ready to present the theory that explicitly describes the geometry of $\mathcal{P}_Q(\beta)$, which extends to both $\mathcal{P}_E(\beta)$ and $\mathcal{P}_N(\beta)$, in the context of submodular function optimization. This theory is not new; several abstract aspects of it have been thoroughly explored by Schrijver [2003] and Fujishige [2005]. In our case, however, the exact form of the problem presented in theorem 2.2.1 was carefully chosen to yield nice properties when this connection is made explicit. From these, a number of desirable properties follow immediately.

The rest of this section is arranged as follows. First, we show that the base polytope $B(f_\beta)$ is a reduction of all pseudoflows (see definition 1.5.3) on the form of the cut problem $\mathcal{P}_N(\beta)$ from theorem 2.2.1. $B(f_\beta)$, described in section 1.6, has special characteristics for our purposes, as the minimum norm algorithm described in section 1.6 provides a convenient theoretical tool to examine the structure of $\mathcal{P}_Q(\beta)$. Our key result is to show that the minimum norm vector – the L_2-projection of the origin onto $B(f_\beta)$ – depends on β only through a simple, constant shift. Thus this vector allows us to immediately compute the minimum cut directly for any β, and we are thus able to compute the minimum cut solution as well for any β as well.

Theorem 2.3.2 (Structure of $B(f_\beta)$)**.** *Let f_β be defined in theorem 2.3.1* (2.15)*, and let*

$$\mathcal{A} = \left\{ \boldsymbol{\alpha} \in \mathcal{M}_{n\times n} : \left\{ \begin{array}{ll} |\alpha_{ij}| \leq |q_{ij}| & i < j \\ \alpha_{ij} = 0 & \textit{otherwise} \end{array} \right\} \right. . \tag{2.16}$$

Let $r_i(\boldsymbol{\alpha})$, $i = 1, ..., n$, be defined as follows:

$$r_i(\boldsymbol{\alpha}) = q_{ii} + \frac{1}{2}\sum_{i'<i} \left(q_{i',i} + \alpha_{i'i}\right) + \frac{1}{2}\sum_{j\,:\,i<j} (q_{ij} - \alpha_{ij}), \tag{2.17}$$

and denote $\mathbf{r}(\boldsymbol{\alpha}) = (r_1(\boldsymbol{\alpha}), ..., r_n(\boldsymbol{\alpha})) \in \mathbb{R}^n$. Then the base of the polymatroid associated with f_β, $B(f_\beta)$, is given by

$$B(f_\beta) = \left\{ \mathbf{r}(\boldsymbol{\alpha}) - \beta : \boldsymbol{\alpha} \in \mathcal{A} \right\}, \tag{2.18}$$

and the full polymatroid polytope is given by

$$P(f_\beta) = \left\{ \mathbf{y} : y_i \leq y_i' \ \forall\, i \textit{ for some } \mathbf{y}' \in B(f_\beta) \right\}. \tag{2.19}$$

Proof. Proceeds with straightforward albeit tedious algebra. Proved in section 2.7 on page 47.

□

In light of this, the minimum norm problem on the graph structure is as follows. The immediate corollary to the min-norm theorem is that $\boldsymbol{\alpha}^*(\beta)$ yields the optimum cut $S^*(\beta)$ of the corresponding network flow problem, $\mathcal{P}_N(\beta)$:

Theorem 2.3.3 (Minimum Norm Formulation of $\mathcal{P}_N(\beta)$)**.** *Let $\mathcal{A}$ and $\mathbf{r}(\boldsymbol{\alpha})$ be given by equations* (2.16) *and* (2.17)*, respectively. Then the min-norm problem $\mathcal{P}_N(\beta)$ associated with $\mathcal{P}_Q(\beta)$ is defined by*

$$\boldsymbol{\alpha}^*(\beta) = \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathcal{A}} \|\mathbf{r}(\boldsymbol{\alpha}) - \beta\|_2 \tag{2.20}$$

Then any optimal cut $S^*(\beta)$ solving $\mathcal{P}_G(\beta)$, as given in theorem 2.2.1, satisfies:

$$\{i \in \mathcal{V} : r_i(\boldsymbol{\alpha}^*(\beta)) < \beta\} \subseteq S^*(\beta) \subseteq \{i \in \mathcal{V} : r_i(\boldsymbol{\alpha}^*(\beta)) \leq \beta\}. \tag{2.21}$$

Proof. Follows immediately from theorems 2.2.1 and 2.3.1 characterizing the cut problem as a submodular function optimization problem, theorem 2.3.2 describing the structure of this problem, and theorem 1.6.1 to characterize the solution. □

The optimal $\boldsymbol{\alpha}^*$ in the above formulation has some surprising consequences that motivate the rest of our results. In particular, we show that the optimal $\boldsymbol{\alpha}^*$ in equation (2.20) is independent of β; this is the key observation that allows us to find the entire regularization path over β. Formally, this result is given in theorem 2.3.7. However, we need other results first.

2.3.2 Connection to Flows

The representation in terms of $\boldsymbol{\alpha}$ is significant partly as the values of $\boldsymbol{\alpha}$ effectively form a pseudoflow in the sense of Hochbaum [2008] (see section 1.5.2). Recall that a pseudoflow extends the concept of a flow by allowing all the nodes to have both excesses and deficits. In addition, a pseudoflow assumes that all edges from the source to nodes in the graph are saturated, possibly creating excesses at these nodes, and all edges connected to the sink are similarly saturated, possibly creating deficits at these nodes.

Theorem 2.3.4. *Consider the problem $\mathcal{P}_N(\beta)$. For any $\alpha_{ij} \in \mathbb{R}$, with $(i,j) \in \mathcal{E}$, let α_{ij} represent the flow on each edge c_{ij}, with $\alpha_{ij} > 0$ indicating flow from i to j, and $\alpha_{ij} < 0$ indicating flow from j to i. Then $\boldsymbol{\alpha} \in \mathcal{A}$ defines a pseudoflow on the graph structure indexed by $\mathcal{A}$. Furthermore, $r_i(\boldsymbol{\alpha}) - \beta = \text{excess}(i)$ is the (possibly negative) excess at node i in the sense of* (1.30).

Proof. The pseudoflow condition that all edges from the source node and to the sink node are saturated is immediately implied by the fact that $r_i(\mathbf{0}) - \beta w_i = (c_{si} - c_{it}) - \beta$. The flow conditions, then follow from the edge capacity being $c_{ij} = c_{ji} = -q_{ij}$ and $-|q_{ij}| \leq \alpha_{ij} \leq |q_{ij}|$. □

Corollary 2.3.5. *Every pseudoflow on the graph defined by theorem 2.2.1 maps to a point in the base polytope $B(f_\beta)$, and every point in $B(f_\beta)$ is given by at least one pseudoflow.*

Proof. Follows immediately from theorem 2.3.4. □

2.3.3 Structure of the Complete Solution

The theorem above has a number of important consequences detailed in the next few sections. The most immediate consequence comes when we consider the structure of the optimal solution of the minimum norm algorithm specialized to the network flow problem $\mathcal{P}_N(\beta)$; this effectively allows us to derive a way of solving $\mathcal{P}_N(\beta)$ for all β.

Lemma 2.3.6 (Optimal solutions to $\mathcal{P}_N(\beta)$)**.** *Then $\boldsymbol{\alpha}^*$ is an optimal solution to $\mathcal{P}_N(\beta)$ if and only if for all i, j, $i < j$, the following condition holds:*

$$\left\{ \begin{array}{ccc} \alpha_{ij}^* = |q_{ij}| & \iff & r_i(\boldsymbol{\alpha}^*) \geq r_j(\boldsymbol{\alpha}^*) \\ -|q_{ij}| \leq \alpha_{ij}^* \leq |q_{ij}| & \iff & r_i(\boldsymbol{\alpha}^*) = r_j(\boldsymbol{\alpha}^*) \\ \alpha_{ij}^* = -|q_{ij}| & \iff & r_i(\boldsymbol{\alpha}^*) \leq r_j(\boldsymbol{\alpha}^*) \end{array} \right\}. \tag{2.22}$$

In particular, the optimum value of $\boldsymbol{\alpha}^*$ in this case is independent of β.

Proof. First, $\mathcal{A}$ is convex as each dimension α_{ij} is bounded independently. Thus the objective of $\mathcal{P}_N(\beta)$ is minimizing a convex function over a convex domain. Therefore, it suffices to prove that equation (2.22) can be satisfied if and only if $\boldsymbol{\alpha}^*$ is a local minimum of $\|\mathbf{r}(\boldsymbol{\alpha}) - \beta\|_2^2$. As $\|\mathbf{r}(\boldsymbol{\alpha}) - \beta\|_2^2$ is differentiable w.r.t. $\boldsymbol{\alpha}$, this is equivalent to showing that either the gradient is 0 or $\boldsymbol{\alpha}^*$ is on the boundary of $\mathcal{A}$ and all coordinate-wise derivatives point outside the domain $\mathcal{A}$.

First, define $g_{ij}(\boldsymbol{\alpha})$ as the gradient of $\|\mathbf{r}(\boldsymbol{\alpha}) - \beta\|_2^2$ w.r.t. α_{ij}:

$$g_{ij}(\boldsymbol{\alpha}) = \frac{\partial}{\partial \alpha_{ij}} \|\mathbf{r}(\boldsymbol{\alpha}) - \beta\|_2^2 = 2 \sum_k (r_k(\boldsymbol{\alpha}) - \beta) \frac{\partial}{\partial \alpha_{ij}} r_k(\boldsymbol{\alpha}) \tag{2.23}$$

Now

$$\frac{\partial}{\partial \alpha_{ij}} r_k(\boldsymbol{\alpha}) = \left\{ \begin{array}{ll} 1 & k = i \\ -1 & k = j \\ 0 & \text{otherwise} \end{array} \right. . \tag{2.24}$$

Thus,

$$g_{ij}(\boldsymbol{\alpha}) = 2\left[(r_i(\boldsymbol{\alpha}) - \beta) - (r_j(\boldsymbol{\alpha}) - \beta)\right] \tag{2.25}$$

$$= 2\left[r_i(\boldsymbol{\alpha}) - r_j(\boldsymbol{\alpha})\right], \tag{2.26}$$

and the following condition holds for all $\boldsymbol{\alpha}$:

$$\left\{ \begin{array}{lcl} g_{ij}(\boldsymbol{\alpha}) < 0 & \iff & r_i(\boldsymbol{\alpha}) > r_j(\boldsymbol{\alpha}) \\ g_{ij}(\boldsymbol{\alpha}) = 0 & \iff & r_i(\boldsymbol{\alpha}) = r_j(\boldsymbol{\alpha}) \\ g_{ij}(\boldsymbol{\alpha}) > 0 & \iff & r_i(\boldsymbol{\alpha}) < r_j(\boldsymbol{\alpha}) \end{array} \right\}. \tag{2.27}$$

Matching these conditions to those in (2.22) shows that $\boldsymbol{\alpha}^*$ as given defines a local, and thus global, optimum of $\mathcal{P}_N(\beta)$. In particular, note that this criterion is independent of β, completing the proof. □

Invariance to β

The invariance of the optimal $\boldsymbol{\alpha}^*$ to β allows us to characterize the solution space of optimal cuts as the level sets of $\mathbf{r}(\boldsymbol{\alpha}^*)$. The core result, as well as our algorithm, is based on this intuition.

Theorem 2.3.7 (). **I.** *$\boldsymbol{\alpha}^*$ is the optimal solution to* (2.20) *if and only if for all $\beta \in \mathbb{R}$, all optimal cuts $S^*_\beta \in \mathcal{S}^*(\beta)$ for $\mathcal{P}(\beta)$ satisfy*

$$U_1(\beta) = \{i \in \mathcal{V} : r_i(\boldsymbol{\alpha}^*) < \beta\} \subseteq S^*_\beta \subseteq \{i \in \mathcal{V} : r_i(\boldsymbol{\alpha}^*) \leq \beta\} = U_2(\beta) \tag{2.28}$$

II. *Furthermore, for all β, $U_1(\beta)$ is the unique smallest minimizer of $\mathcal{P}(\beta)$ and $U_2(\beta)$ is the unique largest minimizer.*

Proof. Part (I) follows as a direct consequence of theorem 2.3.3 and the invariance of the minimum norm problem to changes in β as given in lemma 2.3.6. Part (II) is then an immediate consequence of the minimum norm algorithm. □

This theorem is intuitively important as the key values that permit a connection to the continuous problems are values of β at which the membership of the different nodes change. This theorem tells us that these points are given by the values of the minimum norm vector, here given as $\mathbf{r}(\boldsymbol{\alpha}^*)$.

Monotonicity

One immediate corollary of theorem 2.3.7 is a monotonicity property on the optimal sets, used in the continuous optimization theory we present later:

Corollary 2.3.8. *Let $\beta_1 < \beta_2$. Then for all $S_1^* \in \mathcal{S}^*(\beta_1)$ and $S_2^* \in \mathcal{S}^*(\beta_2)$,*

$$S_1^* \subset S_2^*. \tag{2.29}$$

Proof. Follows immediately from the equivalence of the optimizing sets to the level sets of the minimum norm vector given in theorem 2.3.7. □

2.4 Beyond Network Flows

Theorem 2.3.7 above was discovered independently for the full case of general submodular functions by Nagano et al. [2011]. There, the authors similarly showed that the level sets of the minimum norm algorithm give the solutions for the $f(S) - \beta\,|S|$ problem. While the approach those authors take is different and more involved, we give a shorter, alternative proof. We use a simple argument following from the fact that $B(f)$ constrains the minimum norm vector $\mathbf{y}$ to a constant total sum. The result is that the constant offset in the minimum norm objective drops out of the optimization. More formally:

Theorem 2.4.1 (Invariance of General Submodular Functions to β)**.** **I.** *Let f be a general submodular function. Then $\mathbf{y}^*$ is the optimal solution to the minimum norm problem*

$$\mathbf{y}^* = \operatorname*{argmin}_{\mathbf{y} \in B(f)} \|\mathbf{y}\|_2^2 \tag{2.30}$$

if and only if $\forall\, \beta \in \mathbb{R}$, the set of optimizing solutions

$$S^*(\beta) = \operatorname*{argmin}_{S \in \mathcal{D}} f(S) - \beta\,|S| \tag{2.31}$$

satisfies

$$U_1(\beta) = \{i \in \mathcal{V} : y^* < \beta\} \subseteq S^*(\beta) \subseteq \{i \in \mathcal{V} : y^* \leq \beta\} = U_2(\beta). \tag{2.32}$$

II. *Furthermore, for all β, $U_1(\beta)$ is the unique minimal solution to* (2.20) *and $U_2(\beta)$ is the unique maximal solution in $\mathcal{S}^*(\beta)$.*

Proof. Consider the submodular function $f_\beta(S) = f(S) - \beta\,|S|$. It is easy to show that

$$B(f_\beta) = \{\mathbf{x} - \beta\mathbf{1} : \mathbf{x} \in B(f)\}\,. \tag{2.33}$$

Denote by $\mathbf{y}^*(\beta)$ the minimum norm solution for f_β. Then the submodular problem for $\mathbf{y}^*(\beta)$ is given by

$$\begin{aligned}
\mathbf{y}^*(\beta) &= \operatorname*{argmin}_{\mathbf{y}\in B(f_\beta)} \|\mathbf{y}\|_2^2 && (2.34)\\
&= \left[\operatorname*{argmin}_{\mathbf{v}\in B(f)} \|\mathbf{v} - \beta\mathbf{1}\|_2^2\right] + \beta\mathbf{1} && (2.35)\\
&= \left[\operatorname*{argmin}_{\mathbf{v}\in B(f)} \left\{\left(\sum_{i\in\mathcal{V}} {v_i}^2\right) - 2\beta\left(\sum_{i\in\mathcal{V}} v_i\right)\right\} + |\mathcal{V}|\,\beta^2\right] + \beta\mathbf{1} && (2.36)\\
&= \left[\operatorname*{argmin}_{\mathbf{v}\in B(f)} \left\{\left(\sum_{i\in\mathcal{V}} {v_i}^2\right)\right\} - 2\beta f(\mathcal{V}) + |\mathcal{V}|\,\beta^2\right] + \beta\mathbf{1} && (2.37)\\
&= \left[\operatorname*{argmin}_{\mathbf{v}\in B(f)} \|\mathbf{v}\|_2^2\right] + \beta\mathbf{1} && (2.38)
\end{aligned}$$

where steps (2.35)–(2.36) follow by definition of $B(f)$, causing the terms dependent on β to drop out by way as constants under the optimization.

From this, we have that the optimal minimum norm vector for f_β is just the minimum norm vector for f shifted by β, immediately implying part (I). Similarly, part (II) follows immediately from the minimum norm theorem. □

This result is used in several other sections as well, and has a number of practical implications for size-constrained optimizations and related problems. For a full treatment of related implications, see [Nagano et al., 2011].

2.5 Exact Algorithm for Constant Parametric Flows

Using the above theory, we now wish to present a viable algorithm to calculate the reduction, and hence all cuts, for each node. The idea is simple and follows immediately from the similarity of the minimum cut problem $\mathcal{P}_N(\beta)$ to the structure of $B(f_\beta)$ as detailed in theorem 2.3.4. As each level set of $\mathbf{r}(\boldsymbol{\alpha}^*)$ defines an optimum cut in the graph, we can adjust all of the unary potentials by $r_\mu = \text{mean}_{i\in S}$, chosen to bisect the reduction values,

Algorithm 1: AlphaReduction

Input: Submodular $\mathbf{Q}$.

Output: $\boldsymbol{\alpha}^*$, the minimizer in $\mathcal{A}$ of $\|\mathbf{r}(\boldsymbol{\alpha})\|_2^2$.

// *Begin by calling* BisectReductions *below on the full set* $\mathcal{V}$ *to get* $\boldsymbol{\alpha}$.

return BisectReductions *(*$T = \mathcal{V}$, $\boldsymbol{\alpha} = \mathbf{0}$, $\mathbf{Q}$*)*

// $\boldsymbol{\alpha}_{[T]}$ *denotes* $\boldsymbol{\alpha}$ *restricted to edges with both nodes in* T.

BisectReductions (T, $\boldsymbol{\alpha}$, $\mathbf{Q}$)

$r_\mu \leftarrow \text{mean}_{i \in T}\, r_i(\boldsymbol{\alpha})$, $r_{\min} \leftarrow \min_{i \in T} r_i(\boldsymbol{\alpha})$

if $r_\mu = r_{\min}$ **then return** $\boldsymbol{\alpha}_{[T]}$ // *Done; We are on a single level set.*

$\mathcal{E}_T \leftarrow \{(i,j) : i,j \in T\}$

$S_T^* \leftarrow$ Minimum cut on $(T, \mathcal{E}_T)$, with capacities formed from $(\mathbf{Q}_{[T]} - \text{diag}(r_\mu))$ by theorem 2.2.1.

// *Fix the flow on edges in the cut by adjusting the source/sink capacities of each node, then removing those edges.*

for $i \in S_T^*$, $j \in T \backslash S_T^*$, $i < j$ **do** $\alpha_{ij} \leftarrow -q_{ij}$, $q_{ii} \leftarrow q_{ii} - q_{ij}$, $q_{ij} \leftarrow 0$

for $i \in T \backslash S_T^*$, $j \in S_T^*$, $i < j$ **do** $\alpha_{ij} \leftarrow q_{ij}$, $q_{jj} \leftarrow q_{jj} + q_{ij}$, $q_{ij} \leftarrow 0$

// *Recursively solve on the two partitions to fix the other* $\boldsymbol{\alpha}$*'s.*

$\boldsymbol{\alpha}_{[S_T^*]} \leftarrow$ BisectReductions (S_T^*, $\boldsymbol{\alpha}$, $\mathbf{Q}$)

$\boldsymbol{\alpha}_{[T \backslash S_T^*]} \leftarrow$ BisectReductions ($T \backslash S_T^*$, $\boldsymbol{\alpha}$, $\mathbf{Q}$)

return $\boldsymbol{\alpha}_{[T]}$

and solve the resulting cut problem. By the max-flow min-cut theorem, all edges crossing the cut are saturated. Specifically,

$$\forall\, i, j,\ i < j, \text{ such that } r_i(\boldsymbol{\alpha}) \leq r_\mu < r_j(\boldsymbol{\alpha}),\ \alpha^*_{ij} = -q_{ij}, \tag{2.39}$$

$$\forall\, i, j,\ i < j, \text{ such that } r_i(\boldsymbol{\alpha}) > r_\mu \geq r_j(\boldsymbol{\alpha}),\ \alpha^*_{ij} = q_{ij} \tag{2.40}$$

As these α_{ij} are optimal in the final solution $\boldsymbol{\alpha}^*$, they can be fixed by permanently adding their values to the corresponding $r_i(\boldsymbol{\alpha})$ and removing them from consideration in the optimization. This then bisects the nodes, allowing us to treat these two subsets separately when solving for the rest of the bisections. The validity of this bisection can also be seen by the optimality of the minimum norm solution as described by theorem 2.3.7.

Algorithm 1 can be summarized as follows. We first start by considering the entire set of nodes, setting the working set $S = \mathcal{V}$. At each step, we recursively partition the working set S using a minimum cut as follows:

1. If $\mathbf{r}(\boldsymbol{\alpha})$ is constant in S, then return. We're done.
2. Otherwise, set up a network flow problem to bisect the nodes and find a minimum cut. Once a minimum cut is found, set all the edges in the cut to their saturated values.
3. Repeat on the two resulting subsets of nodes.

Pseudocode for this algorithm is presented in Algorithm 1.

Theorem 2.5.1 (Correctness of Algorithm 1.)**.** *After the termination of Algorithm 1, all values of $\boldsymbol{\alpha}$ are set such that $\|\mathbf{r}(\boldsymbol{\alpha})\|_2$ is minimized over $\boldsymbol{\alpha} \in \mathcal{A}$.*

Proof. Let $i, j,\ i \neq j$, be any pair of nodes such that $q_{ij} \neq 0$, and let $\boldsymbol{\alpha}^\dagger$ be the solution returned by Algorithm 1.

First, suppose that $r_i(\boldsymbol{\alpha}^*) = r_j(\boldsymbol{\alpha}^*)$. Then trivially, the optimality criteria of Lemma 2.3.6 is satisfied.

Next, suppose that $r_i(\boldsymbol{\alpha}) \neq r_j(\boldsymbol{\alpha})$, and first suppose that $r_i(\boldsymbol{\alpha}) > r_j(\boldsymbol{\alpha})$. Then, by the termination condition of the recursion in BisectReductions, nodes i and j must have been separated by a valid minimum cut for some r_μ. However, as all edges crossing a minimum cut are saturated by the max-flow-min-cut theorem, $\alpha_{ij} = |q_{ij}|$. Thus condition

(2.22) in Lemma 2.3.6 is satisfied. Similarly, if $r_i(\boldsymbol{\alpha}) < r_j(\boldsymbol{\alpha})$, then $\alpha_{ij} = -|q_{ij}|$, indicating a flow of $|q_{ij}|$ from j to i; again, this satisfies (2.22).

As the above holds for any pairs of nodes i, j, the optimality criteria of Lemma 2.3.6 is satisfied globally, proving the correctness of the algorithm. □

This algorithm is quite efficient in practice, and it forms an core routine of the total variation minimization algorithm given in chapter 5, where we present full experiments and some comparisons with existing approaches.

2.6 Extensions to Real-valued Variables

One of the intriguing consequences of the above theory, and one that opens new doors to efficiently optimizing several other classes of functions, comes as the result of being able to map other correlated data to this framework. In general, interactions between terms can be very difficult to work with in practice. However, the above theory allows us to exactly find the optimizer of a large class of general functions. These functions may not necessarily be smooth.

Our approach connects closely to several recent results discovered independently by Mairal [Mairal et al., 2011] and Bach [Bach, 2010a], which connect some of these problems to an older result by Hochbaum [Hochbaum and Hong, 1995]. The last of these papers effectively establishes an equivalence between a class of quadratic objective functions and some types of network flow algorithms, although the equivalence to parametric flows isn't really explored. However, this result was used by Bach in Bach [2010a] to note that the minimum norm problem of the network flow problem can be solved using classical methods for solving parametric flow problems [Gallo et al., 1989], and the implications of this for structured sparse recovery are explored in Mairal et al. [2011]. The end result, discovered independently, parallels the theorem we present below, albeit with a different algorithm.

In contrast, while less general, the algorithm we presented in 1 gives the exact change points immediately, and the theoretical framework presented surrounding this problem is more thoroughly explored here. However, the primary practical improvement we provide comes in the next chapter when we incorporate the use of piecewise-linear convex penalty

terms as well.

Theorem 2.6.1. *Suppose* $\gamma : \mathbb{R}^n \to \mathbb{R}^+$ *can be expressed as*

$$\gamma(\mathbf{u}) = \|\mathbf{u} - \mathbf{a}\|_2^2 + \lambda \sum_{i,j} w_{ij}|u_i - u_j| \tag{2.41}$$

where $\mathbf{u} \in \mathbb{R}^n$ *is the variable we wish to optimize over and* $\mathbf{a} \in \mathbb{R}^n$, $\lambda > 0$, *and* $w_{ij} \geq 0$ *are given. Without loss of generality, assume that* $i < j$. *Then the minimizer*

$$\mathbf{u}^* = \underset{\mathbf{u}\in\mathbb{R}^n}{\operatorname{argmin}}\ \gamma(\mathbf{u}) \tag{2.42}$$

can be found exactly using Algorithm 1 with

$$q_{ii} = a_i \tag{2.43}$$

$$q_{ij} = \frac{1}{2} w_{ij} \lambda \tag{2.44}$$

Then $\mathbf{u}^* = \mathbf{r}(\boldsymbol{\alpha}^*)$.

Proof. First, we can write $\gamma(\mathbf{u})$ as

$$\gamma(\mathbf{u}) = \frac{1}{\lambda}\mathbf{u}^T\mathbf{u} - \frac{2}{\lambda}\mathbf{a}^T\mathbf{u} + \sum_{i,j} w_{ij}|u_i - u_j| + \text{const} \tag{2.45}$$

$$\propto \frac{1}{2}\mathbf{u}^T\mathbf{u} - \mathbf{a}^T\mathbf{u} + \sum_{i,j} \frac{\lambda w_{ij}}{2}|u_i - u_j| + \text{const} \tag{2.46}$$

Now, define $\mathbf{1}_{\{\mathbf{u}\leq\beta\}}$ as the cut vector of $\mathbf{u}$ at β, given by

$$\mathbf{1}_{\{\mathbf{u}\leq\beta\}} = \left(\mathbf{1}_{\{u_i\leq\beta\}} : i = 1, 2, ..., n\right) \subseteq \{0, 1\}^n, \tag{2.47}$$

Now, assume that $u_i \in [-M, M]$ for $i = 1, 2, ..., n$, where M is sufficiently large. This is reasonable, as $\lambda^{-1}\mathbf{u}^T\mathbf{u}$ dominates the optimization asymptotically. This enables us to write

the terms in the above expression as

$$\frac{1}{2}\mathbf{u}^T\mathbf{u} = \frac{1}{2}\sum_{i=1}^{n} u_i^2 = \text{const} + \sum_{i=1}^{n}\int_{-M}^{M} \beta \mathbf{1}_{\{\beta < u_i\}}\,\mathrm{d}\beta \tag{2.48}$$

$$= \text{const} + \sum_{i=1}^{n}\int_{-M}^{M} (-\beta) \mathbf{1}_{\{u_i \leq \beta\}}\,\mathrm{d}\beta \tag{2.49}$$

$$-\mathbf{a}^T\mathbf{u} = \text{const} - \sum_{i=1}^{n}\int_{-M}^{M} \mathbf{a}^T \mathbf{1}_{\{\beta < u_i\}}\,\mathrm{d}\beta \tag{2.50}$$

$$= \text{const} + \sum_{i=1}^{n}\int_{-M}^{M} \mathbf{a}^T \mathbf{1}_{\{u_i \leq \beta\}}\,\mathrm{d}\beta \tag{2.51}$$

$$\sum_{i,j} \frac{\lambda w_{ij}}{2}|u_i - u_j| = \int_{-M}^{M} \sum_{i,j} \frac{\lambda w_{ij}}{2} \mathbb{I}\Big\{\mathbf{1}_{\{u_i \leq \beta\}} \neq \mathbf{1}_{\{u_j \leq \beta\}}\Big\}\mathrm{d}\beta. \tag{2.52}$$

Putting this together gives us

$$\gamma(\mathbf{u}) \propto \text{const} + \sum_i \int_{-M}^{M} \left\{ \left[(\mathbf{a} - \beta)^T \big(\mathbf{1}_{\{\mathbf{u} \leq \beta\}}\big) + \left[\sum_{i,j} \frac{\lambda w_{ij}}{2} \mathbb{I}\Big\{\mathbf{1}_{\{u_i \leq \beta\}} \neq \mathbf{1}_{\{u_j \leq \beta\}}\Big\} \right] \right] \right\} \mathrm{d}\beta \tag{2.53}$$

However, the first term in the given approach maps directly to the unary $q_{ii} - \beta$ terms in the graph problem earlier, with the $\mathbf{1}_{\{\mathbf{u} \leq \beta\}}$ term indexing the cut from source to node or from node to sink. Thus, for a given β, $\mathbf{1}_{\{u_i \leq \beta\}}$ is a binary variable with $\mathbf{1}_{\{u_i \leq \beta\}}$ indicating that the edge from source to node i is cut, incurring the appropriate cost in the overall function.

Similarly, $\mathbb{I}\Big\{\mathbf{1}_{\{u_i \leq \beta\}} \neq \mathbf{1}_{\{u_j \leq \beta\}}\Big\}$ is 1 for the range of β in which the edge connecting nodes i and j is cut; i.e. where the indicator variables of set membership differ. The cost of cutting this edge is given by $\frac{1}{2}\lambda w_{ij}$, and it is incurred by all points in the range of β between $\min(u_i, u_j)$ and $\max(u_i, u_j)$. Integrating this gives us $\frac{1}{2}\lambda w_{ij}|u_i - u_j|$. This establishes (2.53).

Now, it remains to show that finding the minimum energy cut solution of the above problem at all β is equivalent to finding $\mathbf{u}^*$. For this step, define a set of functions mapping $\mathbb{R}$ to $\{0, 1\}$ as:

$$\mathscr{S} = \{\mathbb{I}\{\cdot \leq t\} : t \in \mathbb{R}\}. \tag{2.54}$$

Now, we can rewrite the minimization of $\gamma(\mathbf{u})$ as

$$\min_{u\in[-M,M]^n} \gamma(\mathbf{u}) \tag{2.55}$$

$$\propto \text{const} + \min_{u\in[-M,M]^n} \int_{-M}^{M} \left\{ \left[(\mathbf{a}-\beta)^T \big(\mathbf{1}_{\{\mathbf{u}\le\beta\}}\big) + \left[\sum_{i,j} \frac{\lambda w_{ij}}{2} \mathbb{I}\Big\{ \mathbf{1}_{\{u_i\le\beta\}} \neq \mathbf{1}_{\{u_j\le\beta\}} \Big\} \right] \right] \right\} \mathrm{d}\beta \tag{2.56}$$

$$= \text{const} + \min_{\mathbf{s}\in\mathscr{S}^n} \int_{-M}^{M} \left\{ \big[(\mathbf{a}-\beta)^T \mathbf{s}(\beta)\big] + \left[\sum_{i,j} \frac{\lambda w_{ij}}{2} \mathbb{I}\{s_i(\beta) \neq s_j(\beta)\} \right] \right\} \mathrm{d}\beta \tag{2.57}$$

where $\mathbf{s} = s_1, s_2, ..., s_n,\ s_i \in \mathscr{S}$ replaces the optimization over the changepoints u_i. For convenience, denote the term in the integral by

$$h(\beta, \mathbf{x}) = \big[(\mathbf{a}-\beta)^T \mathbf{x}\big] + \left[\sum_{i,j} \frac{\lambda w_{ij}}{2} \mathbb{I}\{x_i \neq x_j\} \right] \tag{2.58}$$

where $\mathbf{x} \in \{0,1\}^n$. Now it is easy to see that

$$\min_{\mathbf{s}\in\mathscr{S}^n} \int_{-M}^{M} h(\beta, \mathbf{s}(\beta)) \mathrm{d}\beta \geq \int_{-M}^{M} \left\{ \min_{\mathbf{x}\in\{0,1\}^n} h(\beta, \mathbf{x}) \right\} \mathrm{d}\beta. \tag{2.59}$$

However, by corollary 2.3.8, we know that for any $\beta_1 < \beta_2$, with

$$\mathbf{x}^*(\beta_1) \in \operatorname*{Argmin}_{\mathbf{x}\in\{0,1\}^n} h(\beta_1, \mathbf{x}) \tag{2.60}$$

$$\mathbf{x}^*(\beta_2) \in \operatorname*{Argmin}_{\mathbf{x}\in\{0,1\}^n} h(\beta_2, \mathbf{x}), \tag{2.61}$$

we have the following implications for all i:

$$\text{if } x_i^*(\beta_1) = 0, \quad \text{then} \quad x_i^*(\beta_2) = 0 \tag{2.62}$$

$$\text{if } x_i^*(\beta_2) = 1, \quad \text{then} \quad x_i^*(\beta_1) = 1. \tag{2.63}$$

This, however, is exactly the constraint implied in the set $\mathscr{S}$. Thus (2.61) holds with equality:

$$\min_{\mathbf{s}\in\mathscr{S}^n} \int_{-M}^{M} h(\beta, \mathbf{s}(\beta)) \mathrm{d}\beta = \int_{-M}^{M} \left\{ \min_{\mathbf{x}\in\{0,1\}^n} h(\beta, \mathbf{x}) \right\} \mathrm{d}\beta. \tag{2.64}$$

However, this minimization is exactly the problem solved earlier; namely, by theorem 2.3.7, we find the optimal set for each β. Thus, the net result of this operation is that $\mathbf{u}^* = \mathbf{r}(\boldsymbol{\alpha}^*)$. Letting $M \to \infty$ completes the theorem. □

2.7 Proofs

In this section, we give a number of the proofs needed for the previous theorems; they are given here for readability.

Proof of Theorem 2.2.1. To show the equivalence between $\mathcal{P}_E(\beta)$ and $\mathcal{P}_Q(\beta)$, it is sufficient to verify

$$\sum_{i\in\mathcal{V}}(E_i(x_i)-\beta x_i)+\sum_{\substack{(i,j)\in\mathcal{E}\\ i<j}}E_{ij}(x_i,x_j)=\mathbf{x}^T(\mathbf{Q}-\mathrm{diag}(\beta))\mathbf{x}+\sum_i E_i(0)+\sum_{i<j\,:\,(i,j)\in\mathcal{E}}E_{ij}(0,0) \tag{2.65}$$

for arbitrary $\mathbf{x}\in\{0,1\}^n$. To simplify notation, assume that $E_{ij}(x_i,x_j)=0$ for all $(i,j)\notin\mathcal{E}$. Then

$$\sum_{i,j}\mathbf{x}^T(\mathbf{Q}-\mathrm{diag}(\beta))\mathbf{x}+\sum_i E_i(0)+\sum_{i<j\,:\,(i,j)\in\mathcal{E}}E_{ij}(0,0) \tag{2.66}$$

$$=\sum_i x_i(q_{ii}-\beta)+\sum_{i<j}q_{ij}x_ix_j+\sum_i E_i(0)+\sum_{i<j\,:\,(i,j)\in\mathcal{E}}E_{ij}(0,0) \tag{2.67}$$

$$\begin{aligned}=&\sum_i\big[x_i(\,\mathrm{E}_i(1)-\beta-E_i(0))+E_i(0)\big]\\ &+\sum_{i<j}\big[x_ix_j(E_{ij}(1,1)+E_{ij}(0,0)-E_{ij}(0,1)-E_{ij}(1,0))\\ &\qquad+x_i(E_{ij}(1,0)-E_{ij}(0,0))+x_j(E_{ij}(0,1)-E_{ij}(0,0))\big]\\ &+\sum_{i<j}E_{ij}(0,0)\end{aligned} \tag{2.68}$$

$$=\sum_i\left\{\begin{array}{ll}E_i(1)-\beta & x_i=1\\ E_i(0) & x_i=0\end{array}\right\}+\sum_{i<j}\left\{\begin{array}{ll}E_{ij}(1,1) & x_i=x_j=1\\ E_{ij}(1,0) & x_i=1,x_j=0\\ E_{ij}(0,1) & x_i=0,x_j=1\\ E_{ij}(0,0) & x_i=0,x_j=0\end{array}\right\} \tag{2.69}$$

$$=\sum_{i\in\mathcal{V}}(E_i(x_i)-\beta x_i)+\sum_{\substack{(i,j)\in\mathcal{E}\\ i<j}}E_{ij}(x_i,x_j). \tag{2.70}$$

Thus the values of the objective function differ by a constant amount for all values $\mathbf{x}$, so the set of minimizers is identical.

Now, to verify the equivalence of $\mathcal{P}_Q(\beta)$ and $\mathcal{P}_N(\beta)$, let x_i be the associated indicator vector, given by:

$$x_i = \mathbb{I}[i \in (S \backslash \{s\})] \quad \Leftrightarrow \quad S = \{i : x_i = 1\} \tag{2.71}$$

It then suffices to verify that

$$\sum_{(i,j)\in\delta(S(\beta),\mathcal{V}\backslash S(\beta))} c_{ij} = \mathbf{x}(\beta)^T(\mathbf{Q} - \mathrm{diag}(\beta))\mathbf{x}(\beta) + \sum_i c_{i,t} \tag{2.72}$$

for all $\mathbf{x} \in \{0,1\}^n$ and corresponding cuts $\{s\} \subseteq S \subseteq \mathcal{V}'$:

$$\begin{aligned}
\mathrm{Cost}(S, \mathcal{V}\backslash S) &= \sum_{i\in S, j\in T} c_{ij} + \sum_{i\in T} c_{si} + \sum_{i\in S} c_{it} && (2.73)\\
&= \left[\sum_{i,j\,:\,j\in T} c_{ij} - \sum_{i,j\in T} c_{ij}\right] + \sum_{i\in T}(c_{si} - c_{it}) + \sum_i c_{it} && (2.74)\\
&= \left[\sum_{i'<i\,:\,i\in T} c_{i',i} + \sum_{i<j\,:\,i\in T} c_{ij}\right] + \sum_{i\in T}(c_{si} - c_{ti}) \\
&\qquad + 2\sum_{i<j\,:\,i,j\in T}(-c_{ij}) + \sum_i c_{it} && (2.75)\\
&= \sum_{i'<i\,:\,i\in T}\left(-\frac{1}{2}q_{i',i}\right) + \sum_{i<j\,:\,i\in T}\left(-\frac{1}{2}q_{ij}\right) \\
&\qquad + \sum_{i\in T}\left[q_{ii} + \frac{1}{2}\sum_{i'\,:\,i'<i} q_{i',i} + \frac{1}{2}\sum_{j\,:\,i<j} q_{ij}\right] + 2\sum_{i<j\,:\,i,j\in T}\frac{1}{2}q_{ij} + \sum_i c_{it} && (2.76)\\
&= \mathbf{x}^T(\mathbf{Q} - \mathrm{diag}(\beta))\mathbf{x} + \sum_i c_{it}. && (2.77)
\end{aligned}$$

Again, the values of the objective functions differ by a constant amount for all $\mathbf{x}$, proving the theorem. □

Proof of Theorem 2.3.1. From [Fujishige, 2005], we know that f_β is submodular if and only if, for all $T \subset U$ and $v \in S\backslash U$,

$$f_\beta(T \cup \{v\}) - f_\beta(T) \geq f_\beta(U \cup \{v\}) - f_\beta(U). \tag{2.78}$$

For convenience, assume that the β is absorbed into the diagonal elements of q_{ii}. Now

$$[f_\beta(T \cup \{v\}) - f_\beta(T)] - [f_\beta(U \cup \{v\}) - f_\beta(U)] \tag{2.79}$$

$$= \left[\sum_{i,j \in T} q_{ij} + \sum_{i \in T} (q_{iv} + q_{vi}) + q_{vv} - \sum_{i,j \in T} q_{ij} \right] - \left[\sum_{i,j \in U} q_{ij} + \sum_{i \in U} (q_{iv} + q_{vi}) + q_{vv} - \sum_{i,j \in U} q_{ij} \right] \tag{2.80}$$

$$= \left[\sum_{i \in T} (q_{iv} + q_{vi}) \right] - \left[\sum_{i \in U} (q_{iv} + q_{vi}) \right] \tag{2.81}$$

$$= - \sum_{i \in U \setminus T} (q_{iv} + q_{vi}). \tag{2.82}$$

The theorem immediately follows; this is non-negative if $q_{ij} \leq 0 \ \forall\, i < j$, and T, U, v can be easily chosen to make it negative if $\exists\, i, j$ such that $q_{ij} > 0$. □

Proof of theorem 2.3.2. For ease of notation, assume that βw_i is absorbed into q_{ii}. For ease of algebra, we first prove a simpler result, then show it immediately implies the desired one.

Define

$$\mathcal{W} = \left\{ \mathbf{w} \in \mathcal{M}_{n \times n} : \left\{ \begin{array}{ll} q_{ij} \leq w_{ij} \leq 0 & \text{if } i < j \\ w_{ij} = 0 & \text{otherwise} \end{array} \right\} \right. . \tag{2.83}$$

(Recall that $q_{ij} \leq 0$.) Then, we show that the associated polymatroid $P(f_\beta)$ can be represented by

$$P(f_\beta) = \left\{ \mathbf{y} : \exists\, \mathbf{w} \in \mathcal{W} \text{ s.t. } \ \forall\, i,\ y_i \leq q_i + \sum_{i' \,:\, i' < i} w_{i',i} + \sum_{j \,:\, i < j} (q_{ij} - w_{ij}) \right\}. \tag{2.84}$$

To show that $P(f_\beta)$ is the base of f, we must show that $\forall\ \mathbf{y} \in P(f_\beta)$ and $\forall\ U \subseteq \mathcal{V}$, $\sum_{i \in U} y_i \leq f(U)$ and that $\forall\, U \subseteq \mathcal{V}$, there exists a point $\mathbf{y} \in P(f_\beta)$ such that $\sum_{i \in U} y_i = f(U)$ [Schrijver, 2003].

To show equation (2.84), we first prove that $\ \forall\, \mathbf{y} \in P(f_\beta),\ \sum_{i \in U} y_i \leq f(U)\ \forall\, U \subseteq S$.

Now

$$\sum_{i\in U} y_i \leq \sum_{i\in U}\left[q_{ii} + \sum_{i' \,:\, i'<i} w_{i',i} + \sum_{j \,:\, i<j}(q_{ij} - w_{ij})\right] \tag{2.85}$$

$$\leq \sum_{i\in U}\left[q_{ii} + \sum_{i'\in U \,:\, i'<i} w_{i',i} + \sum_{j\in U \,:\, i<j}(q_{ij} - w_{ij})\right] \tag{2.86}$$

$$= \sum_{i\in U} q_{ii} + \sum_{i,j\in U} q_{ij} = f(U) \tag{2.87}$$

It remains to show that $\forall U \subseteq \mathcal{V},\ \exists \mathbf{y} \in P(f_\beta)$ such that $\sum_{i\in U} y_i = f(U)$. For this, set

$$w^*_{ij} = \begin{cases} q_{ij} & i \in U,\, i < j \\ 0 & \text{otherwise} \end{cases} \tag{2.88}$$

Clearly, $\mathbf{w}^* \in \mathcal{W}$. Setting $\mathbf{y}^*$ to be the extreme point of $P(f_\beta)$ using $\mathbf{w} = \mathbf{w}^*$, we have, for all $U \subseteq \mathcal{V}$,

$$\sum_{i\in U} y^*_i = \sum_{i\in U}\left[q_{ii} + \sum_{i' \,:\, i'<i} w^*_{i',i} + \sum_{j \,:\, i<j}(q_{ij} - w_{ij})\right] = \sum_{i\in U} q_{ii} + \sum_{i',i\in U \,:\, i'<i} q_{i',i} = f(U) \tag{2.89}$$

Thus (2.84) defines the polymatroid polytope of f_β. Similarly, we show that the base of $P(f_\beta)$ can be given by

$$B(f_\beta) = \left\{\mathbf{y} :\ \exists \mathbf{w} \in \mathcal{W} \text{ s.t. } \ \forall i,\ y_i = q_{ii} + \sum_{i' \,:\, i'<i} w_{i',i} + \sum_{j \,:\, i<j}(q_{ij} - w_{ij})\right\} \tag{2.90}$$

To show this, note that, trivially, $B(f_\beta) \subset P(f_\beta)$. It remains to show that for all $\mathbf{y} \in B(f_\beta)$, $\sum_i y_i = f(\mathcal{V})$ for all $\mathbf{w} \in \mathcal{W}$:

$$\sum_i y_i = \sum_i q_{ii} + \sum_{i',i \,:\, i'<i} w_{ij} + \sum_{i,j \,:\, i<j}(q_{ij} - w_{ij}) = \sum_i q_{ii} + \sum_{i,j \,:\, i<j} q_{ij} = f(S) \tag{2.91}$$

Finally, the theorem follows by replacing w_{ij} with $\frac{1}{2}(\alpha_{ij} - q_{ij})$ in the above results and bringing the βw_i outside of q_{ii} and $\mathbf{r}(\boldsymbol{\alpha})$. □

Chapter 3

UNARY REGULARIZERS AND NON-UNIFORM SIZE MEASURES

3.1 Introduction

In many contexts, it is helpful to use regularization terms or weighting terms on the solution to control the final behavior of the result. In the previous chapter, we discussed the simplest case in the submodular function context, namely weighting the problem against the cardinality of the solution set. We proved this for the case of graph-based submodular problems and extended that argument to the full submodular function context. In this chapter, we extend this result to the case where the size biasing term $\beta\,|S|$ term is replaced with a weighted size biasing term $\beta\mathbf{w}(S)$. Here,

$$\mathbf{w}(S) = \sum_{i \in S} w_i \geq 0 \tag{3.1}$$

is the weighted size biasing term on the function. We then wish to find

$$\mathcal{S}^*(\beta) = \operatorname*{Argmin}_{S \subseteq \mathcal{V}} \; f(S) - \beta \mathbf{w}(S) \tag{3.2}$$

for all $\beta \in \mathbb{R}$ and all positive weight measures $\mathbf{w}$.

The results of this chapter are entirely novel. We prove that the optimal solution to an alternate construction of the minimum norm theorem yields the entire solution path for all β. Like the last chapter, we find a vector $\mathbf{z}^*$ on the base $B(f)$ such that every solution is given by a zero-crossing of $[\mathbf{z}^* - \beta\mathbf{w}]$. We then show that this allows us to include more detailed structures in the continuous optimization problem; in particular, we are able to incorporate the piecewise-linear convex penalty term $\xi_i(u_i)$ in ($\Re_{\mathrm{B}}$) directly into our optimization.

Our result is based around a very simple technique to augment the original graph such that auxiliary variables "attract" parts of the regularization influence of β and transfer it to associated variables in the original problem. We then show that it is possible to translate this result into arbitrary weights by taking several well-controlled limits. The end result is an algorithm for solving (3.2) for arbitrary positive weights.

As this result is novel and holds for general submodular functions, we prove it for the general case first, then extend it to the special case of the linear parametric flow problem. Analogously to algorithm 1, the algorithm we develop here solves this problem exactly. The fact that the solution is exact also allows us to solve ($\Re_{\mathrm{B}}$).

3.2 Encoding Weights by Augmentation

The primary tool used for introducing weights into the optimization of the level sets of the function is to augment the original problem with additional variables. When $\beta = 0$, these variables do not contribute to the solution values of the base set of nodes. In the network flow interpretation, they have no connection to the source or sink – but they are subject to the influence by β in the resulting solutions.

With the proper construction, it is possible to guarantee that these augmented nodes always have the same reduction value as the nodes they are augmenting; this allows us to construct a graph such that these values then translate back into weights on the β terms. The primary tool used is the following lemma, which forms the basis of the rest of our results.

Lemma 3.2.1. *Let $\mathcal{V} = \{1, 2, ..., n\}$, and let $f(S)$ be a bounded submodular function defined on $\mathcal{D} \subseteq 2^{\mathcal{V}}$. (Recall that $\mathcal{D}$ is closed under intersection and union.)*

Let $\mathbf{w} \in \{1, 2, ...\}^{|\mathcal{V}|}$ be a vector of positive integer weights, and set $W = \sum_i (w_i - 1)$. Denote $\mathcal{V}_{\mathbf{w}} = \mathcal{V} \cup \{n+1, ..., n+W\}$. Fix $M_\beta \in \mathbb{R}^+$ and set $M >> M_\beta$ sufficiently large. Then,

I. *For all $\beta \in [-M_\beta, M_\beta]$,*

$$\underset{S \in \mathcal{D}}{\text{Argmin}}\ f(S) - \beta \mathbf{w}(S) = \left\{ T^* \cap \mathcal{V} : T^* \in \underset{T \subseteq \mathcal{V}_{\mathbf{w}} \,:\, T \cap \mathcal{V} \in \mathcal{D}}{\text{Argmin}} f_{\mathbf{w}}(T) - \beta \left| T \right| \right\} \tag{3.3}$$

where Argmin *returns the set of minimizing sets, $f_{\mathbf{w}}(T)$ is submodular and given by*

$$f_{\mathbf{w}}(T) = f(T \cap \mathcal{V}) + M \sum_{i \in \mathcal{V}} \sum_{j \in K_i} \left[\mathbf{1}_{\{i \in T\}} + \mathbf{1}_{\{j \in T\}} - 2\mathbf{1}_{\{i,j\} \subseteq T} \right], \tag{3.4}$$

and K_i is a block of indices of length $w_i - 1$, given by

$$K_i = \left\{ \left(n + \sum_{k<i} (w_k - 1) \right), ..., \left(n + \sum_{k<i} (w_k - 1) \right) + (w_i - 1) \right\}. \tag{3.5}$$

II. *Define*

$$\mathcal{D}_{\mathbf{w}} = \left\{ S \cup T : S \in \mathcal{D},\ T = \bigcup_{i \in S} K_i \right\}. \tag{3.6}$$

Then $\mathcal{D}_{\mathbf{w}}$ *is a distributed lattice and* (3.3) *can be replaced by*

$$\underset{S\in\mathcal{D}}{\text{Argmin}}\ f(S)-\beta\mathbf{w}(S)=\left\{T^{*}\cap\mathcal{V}:T^{*}\in\ \underset{T\in\mathcal{D}_{\mathbf{w}}}{\text{Argmin}}\ f(T\cap\mathcal{V})-\beta\left|T\right|\right\}. \tag{3.7}$$

III. *Furthermore,*

$$f_{\mathbf{w}}(T)=f(T\cap\mathcal{V})\ \textit{for all}\ T\in\mathcal{D}_{\mathbf{w}}. \tag{3.8}$$

Proof. (**(I)**) First, we establish that $f_{\mathbf{w}}$ is indeed a submodular function defined on a distributed lattice. As given, $f_{\mathbf{w}}$ is defined on

$$\mathcal{D}'=\left\{T\in 2^{\mathcal{V}_{\mathbf{w}}}:T\cap\mathcal{V}\in\mathcal{D}\right\}. \tag{3.9}$$

$$=\left\{S\cup U:S\in\mathcal{D},\ U\in 2^{\mathcal{V}_{\mathbf{w}}\backslash\mathcal{V}}\right\}. \tag{3.10}$$

This is a distributed lattice, as it is the product of the distributed lattices $\mathcal{D}$ and $2^{\mathcal{V}_{\mathbf{w}}\backslash\mathcal{V}}$. Now, let

$$g_{ij}(U)=M\cdot\left(\mathbf{1}_{\{i\in T\}}+\mathbf{1}_{\{j\in T\}}-2\mathbf{1}_{\{i,j\}\subseteq T}\right) \tag{3.11}$$

for $i\in\mathcal{V}$ and $j\in\mathcal{V}_{\mathbf{w}}\backslash\mathcal{V}$. It is easy to see that g_{ij} is a submodular function, as

$$2M=g(\{i\})+g(\{j\})\geq g(\{i,j\})+g(\emptyset)=0 \tag{3.12}$$

for any i,j. Thus $f_{\mathbf{w}}(T)$ is a submodular function on $\mathcal{D}'$, as it is the sum of $W+1$ submodular functions.

Now, we establish (3.3) inductively. Define a sequence of W integers $k_1,k_2,...,k_W$ satisfying

$$w_i=\#\left\{j:k_j=w_i\right\}+1\ \ \forall\, i\in\mathcal{V}. \tag{3.13}$$

Such a set may be chosen by reversing K_i in (3.5) above.

Let $\mathcal{V}_m=\mathcal{V}\cup\{n+1,...,n+m\}$. Define $f_{0,\beta}(S)=f'_{0,\beta}(S)=f(S)-\beta\left|S\right|$, and let

$$f_{m,\beta}(S)=f_{m-1,\beta}(S)-\beta\mathbf{1}_{\{k_j\in S\}}. \qquad S\subseteq\mathcal{V} \tag{3.14}$$

$$f'_{m,\beta}(T)=f'_{m-1,\beta}(T\cap\mathcal{V}_{m-1})+g_{k_m,n+m}(T)-\beta\mathbf{1}_{\{n+m\in T\}}, \qquad T\subseteq\mathcal{V}_m \tag{3.15}$$

We intend to show that for $m = 1, ..., W$,

$$\underset{S\subseteq\mathcal{D}}{\text{Argmin}}\ f_{m,\beta}(S) + h(S) = \left\{\mathcal{V}\cap T^* : T^* \in \left[\underset{T\subseteq\mathcal{V}_m\,:\,T\cap\mathcal{V}\in\mathcal{D}}{\text{Argmin}} f'_{m,\beta}(T) + h(T\cap\mathcal{V})\right]\right\} \quad (3.16)$$

for all modular functions h on $\mathcal{V}$.

Fix β and h. Trivially, (3.16) is true for the ground case $m = 0$. Now suppose (3.16) is true for $m-1$, and let

$$T^* \in \underset{T\subseteq\mathcal{V}_m\,:\,T\cap\mathcal{V}\in\mathcal{D}}{\text{Argmin}} f'_{m,\beta}(T) + h(T) \quad (3.17)$$

be any minimizer of $f'_{m,\beta}$.

To show that the elements k_m and $n+m$ are tied in T^*, assume the opposite – suppose that either $[k_m \in T^*$ and $n+m \notin T^*]$ or $[k \notin T^*$ and $n+m \in T^*]$. This, however, contradicts the optimality of T^* for sufficiently large M, as the value of the minimum can always be improved by M if element $n+m$ is included or excluded so that its membership in T^* matches that of k_m. Thus, for optimal T^*, $k_m \in T^*$ if and only if $n+m \in T^*$.

We then have that for any minimizer T^* of $f'_{m,\beta} + h$,

$$f'_{m,\beta}(T^*) = f'_{m-1,\beta}(T^*\cap\mathcal{V}_{m-1}) - \beta\mathbf{1}_{\{k_m\in T^*\}}\,, \quad (3.18)$$

as $g_{k_m,n+m}(\{k_m, n+m\}) = g_{k_m,n+m}(\emptyset) = 0$.

Now let $h'(S) = h(S) - \beta\mathbf{1}_{\{k_m\in T^*\}}$. Then

$$\underset{S\subseteq\mathcal{D}}{\text{Argmin}}\ f_{m,\beta}(S) + h(S) \quad (3.19)$$

$$= \underset{S\subseteq\mathcal{D}}{\text{Argmin}}\ f_{m-1,\beta}(S) + h'(S) \quad (3.20)$$

$$= \left\{\mathcal{V}\cap T^* : T^* \in \left[\underset{T\subseteq\mathcal{V}_{m-1}\,:\,T\cap\mathcal{V}\in\mathcal{D}}{\text{Argmin}} f'_{m-1,\beta}(T) + h'(T\cap\mathcal{V})\right]\right\} \quad (3.21)$$

$$= \left\{\mathcal{V}\cap T^* : T^* \in \left[\underset{T\subseteq\mathcal{V}_m\,:\,T\cap\mathcal{V}\in\mathcal{D}}{\text{Argmin}} f'_{m,\beta}(T) + h(T\cap\mathcal{V})\right]\right\} \quad (3.22)$$

where (3.20) - (3.21) follows by the inductive assumption. As (3.20) - (3.21) holds for any modular function $h'(S)$, we have that (3.19) - (3.22) is true as well for any $h(S)$, proving (3.16) for $m = 1, 2, ..., W$.

Now, it is easy to show that

$$f'_{W,\beta}(T) = f(T \cap \mathcal{V}) + \sum_{i=1}^{W} g_{k_i,n+i}(T) - \beta\,|T| \tag{3.23}$$

$$= f_{\mathbf{w}}(T) - \beta\,|T| \tag{3.24}$$

For all $T \in \mathcal{D}'$, and

$$f_{W,\beta}(S) = f(S) - \beta \sum_{i \in \mathcal{V}} w_i \mathbf{1}_{\{i \in S\}} \tag{3.25}$$

$$= f(S) - \beta \mathbf{w}(S) \tag{3.26}$$

for all $S \in \mathcal{D}$, proving the first part of the theorem.

(**(II)**) To show (II), note that the membership of each item in $\mathcal{V}_{\mathbf{w}} \backslash \mathcal{V}$ exactly matches the membership of an item in $\mathcal{V}$; as $\mathcal{D}$ is a distributed lattice, $\mathcal{D}_{\mathbf{w}}$ is also a distributed lattice that is closed under intersection and union.

It remains to show that (3.7) is equivalent to (3.3). This follows directly from noting that for M sufficiently large, $\mathcal{D}_{\mathbf{w}}$ can be written as

$$\mathcal{D}_{\mathbf{w}} = \{T \subseteq \mathcal{V}_{\mathbf{w}} : T \cap \mathcal{V} \in \mathcal{D},\ f_{\mathbf{w}}(T) \leq M/2\}\,. \tag{3.27}$$

As we have already argued that in any optimal solution T^* of $f_{\mathbf{w}}(T)$, the M terms cancel out, so any minimizer of $f_{\mathbf{w}}(T)$ must be in $\mathcal{D}_{\mathbf{w}}$, proving (II).

(**(III)**) Finally, in this case, for all $\{j,k\}$ such that $\exists\, S \in \mathcal{D}_{\mathbf{w}}$ with $\{j,k\} \subseteq S$, $g_{jk}(S) = 0$, proving (3.8) and completing the proof. □

The above lemma is noteworthy as it provides a way to theoretically augment the original problem in a way that alters the original problem such that the relative influence of the β scaling can be altered. In particular, in the augmented problem, the size of the evaluation set $|T|$ includes these augmented nodes – since they are included deterministically based on the values in the unaugmented set $\mathcal{V}$, the unaugmented node is effectively counted multiple times. This allows us to weight the nodes separately.

In our context, when dealing with graph structures, this corresponds to adding a collection of single nodes with no connections other than an effectively infinite capacity edge

connecting each to one of the base nodes. As this edge ties the nodes together in any cut solution, the influence of the global weighting parameter β on this auxiliary node is simply transferred to the attached node. The next two theorems extend this result to the minimum norm vector $\mathbf{y}^*$, and an approximation lemma extends this to general weight vectors.

3.3 Structure and Solution of the Optimization Problem

The central result of this section is a weighted version of the minimum norm problem. Under this construction, the solution to the original minimum norm problem is the same as problem a $\beta\,|S|$ weighting term, but with additional nodes. However, the level sets of the resulting vector yield the optimal minimizing sets $f(S) - \beta\mathbf{w}(S)$ for all values of the parameter β.

Definition 3.3.1 (Weighted Minimum Norm Problem)**.** *For a submodular function f defined on $\mathcal{D} \subseteq 2^{\mathcal{V}}$, and positive weights $\mathbf{w} \in \mathbb{R}^n$, $\mathbf{w} > 0$, the weighted minimum norm problem is given by*

$$\mathbf{z}^* = \operatorname*{argmin}_{\mathbf{z} \in B(f)} \sum_{i \in \mathcal{V}} \frac{z_i^2}{w_i}. \tag{3.28}$$

and we call the solution vector $\mathbf{z}^*$ the Weighted Minimum Norm Vector. Furthermore, if the weights $\mathbf{w}$ are restricted to be positive integers, then this is called the Integer Weighted Minimum Norm Problem.

We use the solution to this problem in an analogous way to the use of the minimum norm vector of the last chapter. The theorems in this section show that $\mathbf{z}^*$ gives the entire solution path to $f(S) - \beta\mathbf{w}(S)$ over β.

Theorem 3.3.2 (Structure)**.** *Let f be a submodular function on $\mathcal{D}$, and let $\mathbf{w}$, $\mathcal{V}_{\mathbf{w}}$, $f_{\mathbf{w}}$,K_i, and $\mathcal{D}_{\mathbf{w}}$ be as defined in lemma 3.2.1 (in particular, the elements of $\mathbf{w}$ are integers). Let $\kappa_i = K_i \cup \{i\}$, so $\mathbf{x}(\kappa_i) = x_i + \sum_{j \in K_i} x_j$. Then*

I. *The polymatroid associated with $f_{\mathbf{w}}$ is given by*

$$P(f_{\mathbf{w}}) = \left\{ \mathbf{x} \in \mathbb{R}^{|\mathcal{V}_{\mathbf{w}}|} : \ \forall\, S \in \mathcal{D},\ \sum_{i \in S} \mathbf{x}(\kappa_i) \leq f(S) \right\} \tag{3.29}$$

and the associated base polymatroid is given by

$$B(f_{\mathbf{w}}) = \left\{ \mathbf{x} \in \mathbb{R}^{|\mathcal{V}_{\mathbf{w}}|} : \mathbf{x} \in P(f_{\mathbf{w}}),\ \mathbf{x}(\mathcal{V}_{\mathbf{w}}) = \sum_{i \in \mathcal{V}} \mathbf{x}(\kappa_i) = f(\mathcal{V}) \right\}, \tag{3.30}$$

II. *Let*

$$\mathbf{y}^* = \underset{\mathbf{y} \in B(f_{\mathbf{w}})}{\operatorname{argmin}} \ \|\mathbf{y}\|_2^2 . \tag{3.31}$$

Then

$$y_j^* = y_i^* \text{ for all } j \in K_i \tag{3.32}$$

III. *Furthermore, let* $\mathbf{z}^*$ *be the solution to the integer weighted minimum norm problem, i.e.*

$$\mathbf{z}^* = \underset{\mathbf{z} \in B(f)}{\operatorname{argmin}} \sum_i \frac{z_i^2}{w_i}. \tag{3.33}$$

Then

$$y_i^* = \frac{z_i^*}{w_i} \text{ for all } i \in \mathcal{V}. \tag{3.34}$$

Furthermore a vector $\mathbf{z}^*$ *is the optimal solution to* (3.33) *if and only* $\mathbf{y}^*$*, as given by* (3.34)*, is the optimal solution to* (3.31)*.*

Proof. (**(I)**) Recall that the definition of the polymatroid associated with the submodular function $f_{\mathbf{w}}$ is defined as

$$P(f_{\mathbf{w}}) = \left\{ \mathbf{x} \in \mathbb{R}^{|\mathcal{V}_{\mathbf{w}}|} : \ \ \forall T \in \mathcal{D}_{\mathbf{w}}, \mathbf{x}(T) \leq f_{\mathbf{w}}(T) \right\} \tag{3.35}$$

Now, by equation (3.8) in lemma 3.2.1,

$$f_{\mathbf{w}}(T) = f(T \cap \mathcal{V}) \tag{3.36}$$

for all $T \in \mathcal{D}_{\mathbf{w}}$. Furthermore, for all $j \in K_i$, $i \in T$ if and only if $j \in T$. Thus the condition $\mathbf{x}(T) \leq f_{\mathbf{w}}(T)$ is equivalent to

$$\sum_{i \in T \cap \mathcal{V}} \mathbf{x}(\kappa_i) \leq f(T \cap \mathcal{V}) \tag{3.37}$$

and the condition $\mathbf{x}(\mathcal{V}_{\mathbf{w}}) = f_{\mathbf{w}}(\mathcal{V}_{\mathbf{w}})$ is equivalent to

$$\mathbf{x}(\mathcal{V}_{\mathbf{w}}) = \sum_{i \in \mathcal{V}} \mathbf{x}(\kappa_i) = f(\mathcal{V}). \tag{3.38}$$

This proves part (I)

(**(II)-(III)**) Now, to prove part (II), consider minimizing $\|\mathbf{y}\|_2^2$ over $B(f_{\mathbf{w}})$. Under the constraints, we can express this problem as

$$\begin{aligned} (\star) \quad \text{minimize} \quad & \sum_{i \in \mathcal{V}} \left[\sum_{j \in \kappa_i} y_j^2 \right] \\ \text{such that} \quad & \sum_{i \in S} \mathbf{y}(\kappa_i) \leq f(S) \;\; \forall \, S \in \mathcal{D} \\ & \sum_{i \in \mathcal{V}} \mathbf{y}(\kappa_i) = f(\mathcal{V}) \end{aligned} \tag{3.39}$$

Now, let $z_i = \mathbf{y}(\kappa_i)$, and define

$$h_i(\mathbf{y}, t) = \min \left\{ \sum_{j \in \kappa_i} y_j^2 \, : \, \sum_{j \in \kappa_i} y_j = t \right\} \tag{3.40}$$

The objective $(\star)$ in (3.39) above can then be expressed as

$$(\star) = \text{minimize} \sum_{i \in \mathcal{V}} h_i(\mathbf{y}, \mathbf{y}(\kappa_i)). \tag{3.41}$$

This, however, has an analytic solution, as the L_2-norm above is minimized when $\mathbf{y}$ is equal on the set κ_i, i.e.

$$y_j = \frac{t}{|\kappa_i|} \;\; \forall \, j \in \kappa_i. \tag{3.42}$$

Thus

$$h_i(\mathbf{y}, t) = h_i(t) = |\kappa_i| \left(\frac{t}{|\kappa_i|} \right)^2 = \frac{t^2}{|\kappa_i|} = \frac{t^2}{w_i}. \tag{3.43}$$

This proves equation (3.32), and allows us to rewrite (3.39) as

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in \mathcal{V}} z_i^2 / w_i \\ \text{such that} \quad & \mathbf{z}(S) \leq f(S) \;\; \forall \, S \in \mathcal{D} \\ & \mathbf{z}(\mathcal{V}) = f(\mathcal{V}) \end{aligned} \tag{3.44}$$

which is exactly identical to the optimization in equation (3.33). Given that $\mathbf{y}$ is constant on each κ_i, we have already proved equation (3.34) by setting $t = z_i$ in equation (3.42). $\square$

The important concept behind this theorem and its corollaries is that it demonstrates a direct connection between the minimum norm problem on the augmented problem $f_{\mathbf{w}}$ and the original problem f. This connection allows us to build the theory of the weighted problem directly upon the original theory, essentially using those results.

3.4 On the Use of General Positive Weights

The goal of this section is to extend the above results on integer $\mathbf{w}$ to all positive real numbers. This allows us to do a number of interesting things, particularly in the case of network flow algorithms. It also extends the state of the known theory on general submodular function minimization outside of the cases we are interested in. We here state the form of theorem 3.3.2 for general $\mathbf{w}$, then discuss some of the implications for the case of network flows and the continuous optimization problems introduced in chapter 1. In particular, this theorem allows us a way to include the piecewise linear $\xi_i(u_i)$ term in $(\Re_{\mathrm{B}})$.

Theorem 3.4.1. *Let f be a submodular function defined on $\mathcal{D} \subset 2^{\mathcal{V}}$, and let $\mathbf{w} \in \mathbb{R}^{|\mathcal{V}|}$ be strictly positive, finite weights. Then*

I. *Let $\mathbf{z}^*$ be the optimal solution to the weighted minimum norm problem, i.e.*

$$\mathbf{z}^* = \operatorname*{argmin}_{\mathbf{z} \in B(f)} \sum_{i \in \mathcal{V}} \frac{z_i^2}{w_i}, \tag{3.45}$$

and, for all $\beta \in \mathbb{R}$, let

$$U_1(\beta) = \{i \in \mathcal{V} : z_i^* - \beta w_i < 0\} \tag{3.46}$$

$$U_2(\beta) = \{i \in \mathcal{V} : z_i^* - \beta w_i \leq 0\} \tag{3.47}$$

and let

$$\mathcal{S}^*(\beta, \mathbf{w}) = \operatorname*{Argmin}_{S \in \mathcal{D}} f(S) - \beta \mathbf{w}(S). \tag{3.48}$$

Then $\mathbf{z}^*$ is the optimal solution to (3.45) *if and only if, for all $\beta \in \mathbb{R}$ and all $S^* \in \mathcal{S}^*(\beta, \mathbf{w})$,*

$$U_1(\beta) \subseteq S^* \subseteq U_2(\beta) \tag{3.49}$$

II. *Furthermore, for all β, $U_1(\beta)$ is the unique minimal solution to* (3.48) *and $U_2(\beta)$ is the unique maximal solution in $\mathcal{S}^*(\beta, \mathbf{w})$.*

Before proving this theorem, we must establish a lemma that shows convergence of the mapping in (3.45) under various perturbations. The reason that this lemma is needed is

that the crux of the proof of theorem 3.4.1 involves showing that all positive $\mathbf{w} \in \mathbb{R}^n$ can be seen as the limit of a sequence of problems with integer $\mathbf{w} \in \mathbb{Z}^n$. The crux of this lemma is found in Wets [2003], where a convenient theorem shows that mappings of the type (3.45) are locally Lipshitz-continuous with respect to perturbations of the objective function. This continuity is sufficient to ensure that the limiting argument employed in the proof of theorem 3.4.1 is valid.

Lemma 3.4.2. *Let D be a convex set, and let*

$$g^*(\boldsymbol{\delta}, \mathbf{w}) = \operatorname*{argmin}_{\mathbf{z} \in D} \sum_{i=1}^{n} \frac{(z_i + \delta_i)^2}{w_i} \tag{3.50}$$

for $\boldsymbol{\delta}, \mathbf{w} \in \mathbb{R}^n$, with $\mathbf{w} \geq \eta > 0$ bounded away from 0. Then $g^*(\boldsymbol{\delta}, \mathbf{w})$ is a locally Lipshitz-continuous function of $\boldsymbol{\delta}$ and $\mathbf{w}$. In other words, for every point $(\boldsymbol{\delta}, \mathbf{w}) \in \mathbb{R}^n \times [\eta, \infty)^n$, there exists a neighborhood $U \subset (\boldsymbol{\delta}, \mathbf{w})$, $(\boldsymbol{\delta}, \mathbf{w}) \in U$, such that g^* is Lipshitz-continuous on U.

Proof. Define $h : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R} \cap \{\infty\}$ as an extended version of the function g:

$$h(\boldsymbol{\delta}, \mathbf{w}, \mathbf{z}) = \begin{cases} \sum_{i=1}^{n} \frac{(z_i+\delta_1)^2}{w_i} & \mathbf{z} \in B(f) \\ \infty & \text{otherwise} \end{cases}. \tag{3.51}$$

The lemma follows from showing that h satisfies the properties of Theorem 3.4 of Wets [2003], which states that the inf-mapping of (3.50) is locally Lipshitz continuous if mild conditions on h are satisfied. These conditions follow immediately as h is Lipshitz continuous in $\mathbf{w}$, $\boldsymbol{\delta}$, and $\mathbf{z}$, and the domain of $\mathbf{z}$ is not affected by the $\mathbf{w}$ and $\boldsymbol{\delta}$. This gives us the desired result. □

Proof of theorem 3.4.1. With this lemma in place, we are now ready to provide the proof of theorem 3.4.1. The proof proceeds in two stages. First, we show that it is valid for positive rational $\mathbf{w} \in \mathbb{Q}^n_{++}$, where $\mathbb{Q}_{++} = \{x \in \mathbb{Q} : x > 0\}$. Then, we use a carefully constructed limiting argument extends this to all real weights.

Step 1: First, assume that the weights are positive rationals. Thus we may write

$$w_i = \frac{N_i}{M_i}, \tag{3.52}$$

where $N_i \in \mathbb{Z}_{++}$ and $M_i \in \mathbb{Z}_{++}$. Then, let

$$M = \prod_{i \in \mathcal{V}} M_i, \tag{3.53}$$

so Mw_i is an integer for all $i \in \mathcal{V}$. Now note that we can immediately map the original problem to this form by simply setting

$$w_i' = Mw_i \tag{3.54}$$

$$\beta' = \beta / M \tag{3.55}$$

Then $\beta w_i = \beta' w_i'$, with w_i' being an integer.

Let $\mathcal{V}_{\mathbf{w}'}$, $\mathcal{D}_{\mathbf{w}'}$, and $f_{\mathbf{w}'}$ be as defined in theorem 3.3.2, and let

$$\mathbf{z}^* = \operatorname*{argmin}_{\mathbf{z} \in B(f)} \sum_{i \in \mathcal{V}} \frac{z_i^2}{w_i} = \operatorname*{argmin}_{\mathbf{z} \in B(f)} \frac{1}{M} \sum_{i \in \mathcal{V}} \frac{z_i^2}{w_i} = \operatorname*{argmin}_{\mathbf{z} \in B(f)} \sum_{i \in \mathcal{V}} \frac{z_i^2}{w_i'} \tag{3.56}$$

and let

$$\mathbf{y}^* = \frac{\mathbf{z}^*}{\mathbf{w}'}. \tag{3.57}$$

By theorem 3.3.2, we know that $\mathbf{z}^*$ is the optimal solution to (3.56) if and only if $\mathbf{y}^*$ is the optimal solution to

$$\mathbf{y}^* = \operatorname*{argmin}_{\mathbf{y} \in B(f_{\mathbf{w}'})} \|\mathbf{y}\|_2^2 . \tag{3.58}$$

Now, by theorem 3.3.2-(III) and theorem 2.4.1-(I), we have that $\mathbf{y}^*$ is optimal for (3.58) if and only if $\forall\, \beta' \in \mathbb{R}$, all optimal solutions

$$T^*(\beta') = \operatorname*{argmin}_{T \in \mathcal{D}_{\mathbf{w}'}} \left(f_{\mathbf{w}'}(T) - \beta' \right) \tag{3.59}$$

satisfy

$$U_1'(\beta') = \left\{ i \in \mathcal{V} : y_i^* - \beta' < 0 \right\} \subseteq T^*(\beta') \subseteq \left\{ i \in \mathcal{V} : y_i^* - \beta' \leq 0 \right\} = U_2'(\beta'). \tag{3.60}$$

However, substituting $y_i^* = z_i^* / w_i'$ into equation (3.60) immediately gives us the equivalence

$$\left\{ i \in \mathcal{V} : z_i^* - \beta' w_i' < 0 \right\} \subseteq T^*(\beta') \subseteq \left\{ i \in \mathcal{V} : z_i^* - \beta' w_i' \leq 0 \right\}. \tag{3.61}$$

Letting $T^\dagger(\beta) = T^*(\beta / M)$, and recalling that $\beta' w_i' = \beta w_i$, we have that

$$U_1(\beta) = \{ i \in \mathcal{V} : z_i^* - \beta w_i < 0 \} \subseteq T^\dagger(\beta) \subseteq \{ i \in \mathcal{V} : z_i^* - \beta w_i \leq 0 \} = U_2(\beta). \tag{3.62}$$

Thus $U_1'(\beta') = U_1(\beta)$ and $U_2'(\beta') = U_2(\beta)$, and we have proved (I) for $\mathbf{w} \in \mathbb{Q}^n_{++}$.

Part (II) follows similarly. $\mathbf{y}^*$ in (3.58) is the minimum norm vector for the extended problem $f_{\mathbf{w}'}$, so by theorem 2.4.1-(II) $U_1'(\beta')$ and $U_2'(\beta')$ are the smallest and largest minimizers of $f(S) - \beta'\mathbf{w}'(S) = f(S) - \beta\mathbf{w}(S)$. However, $U_1'(\beta') = U_1(\beta)$ and $U_2'(\beta') = U_2(\beta)$, so $U_1(\beta)$ and $U_2(\beta)$ satisfy part (II). Thus we have proved the theorem for rational $\mathbf{w}$.

Step 2: Now, it remains to show that this result extends to all real weights as well. This is more difficult than it immediately seems, as $\mathbf{z}^*$ depends on $\mathbf{w}$ through an optimization over $B(f)$, and, unless $\mathcal{D} = 2^{\mathcal{V}}$, $B(f)$ is not necessarily bounded. Thus it takes some care to show that the sets generated by the inequalities in (3.46) and (3.47) are the limit points corresponding to a sequence of rational $\mathbf{w}$, and that they are the smallest and largest minimizers of $f(S) - \beta\mathbf{w}(S)$ for all β.

First, for convenience, define $f_{\tilde{\mathbf{w}},\beta} : \mathcal{D} \mapsto \mathbb{R}$ as

$$f_{\tilde{\mathbf{w}},\beta} = f(S) - \beta\tilde{\mathbf{w}}(S) \tag{3.63}$$

and recall that this is a submodular function for any $\tilde{\mathbf{w}} \in \mathbb{R}^n_{++}$. Let $\mathbf{y}^*(\tilde{\mathbf{w}}, \beta)$ be the corresponding minimum norm vector:

$$\mathbf{y}^*(\tilde{\mathbf{w}}, \beta) = \operatorname*{argmin}_{\mathbf{y} \in B(f_{\tilde{\mathbf{w}},\beta})} \|\mathbf{y}\|_2^2 . \tag{3.64}$$

Similarly, denote $\mathbf{z}^*$ as a function of $\tilde{\mathbf{w}}$ as well, i.e.

$$\mathbf{z}^*(\tilde{\mathbf{w}}) = \operatorname*{argmin}_{\mathbf{z} \in B(f)} \sum_{i \in \mathcal{V}} \frac{z_i^2}{\tilde{w}_i}. \tag{3.65}$$

Now suppose that $\mathbf{w} \notin \mathbb{Q}^n$, so the above proof for rational $\mathbf{w}$ does not apply. As $\mathbb{Q}$ is dense on $\mathbb{R}$ and $\mathbf{w} > 0$, there exists a sequence of positive rationals $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, ..., \boldsymbol{\omega}_i \in \mathbb{Q}^n$ such that

$$\lim_{i \to \infty} \boldsymbol{\omega}_i = \mathbf{w}. \tag{3.66}$$

By the minimum norm theorem and the fact that we have proved the theorem in question for all rational $\boldsymbol{\omega}_m$, we know that, for all i and m,

$$U_1(\beta, \boldsymbol{\omega}_m) = \{i : z_i^*(\boldsymbol{\omega}_m) - \beta\omega_{m,i} < 0\} = \{i : y_i^*(\boldsymbol{\omega}_m, \beta) < 0\} \tag{3.67}$$

$$U_2(\beta, \boldsymbol{\omega}_m) = \{i : z_i^*(\boldsymbol{\omega}_m) - \beta\omega_{m,i} \leq 0\} = \{i : y_i^*(\boldsymbol{\omega}_m, \beta) \leq 0\} . \tag{3.68}$$

where we have made the dependence of the minimal and maximal sets $U_1(\beta)$ and $U_2(\beta)$ in (3.46) and (3.47) on $\boldsymbol{\omega}_m$ explicit since we are working with a sequence of problems given by $\boldsymbol{\omega}_m$. Thus, by theorem 2.4.1, for all $\tilde{\mathbf{w}} \in \{\mathbf{w}, \boldsymbol{\omega}_1, \boldsymbol{\omega}_2, ...\}$,

$$U_1(\beta, \tilde{\mathbf{w}}) = \{i : y_i^*(\tilde{\mathbf{w}}, \beta) < 0\} \tag{3.69}$$

and

$$U_2(\beta, \tilde{\mathbf{w}}) = \{i : y_i^*(\tilde{\mathbf{w}}, \beta) \leq 0\} \tag{3.70}$$

are the unique smallest and largest minimizing sets of $f_{\tilde{\mathbf{w}},\beta}$, respectively. Thus we are done if we can show that, for all β,

$$\lim_{m \to \infty} \quad \mathbf{z}^*(\boldsymbol{\omega}_m) - \beta\boldsymbol{\omega}_m = \mathbf{z}^*(\mathbf{w}) - \beta\mathbf{w} \tag{3.71}$$

and

$$\lim_{m \to \infty} \mathbf{y}^*(\boldsymbol{\omega}_m, \beta) = \mathbf{y}^*(\mathbf{w}, \beta), \tag{3.72}$$

as this immediately implies that (3.67) and (3.68) hold in the limit as well.

First, let D be a convex domain, and consider the function $g_D^* : \mathbb{R}^n \times (0, \infty) \mapsto D$, where

$$g_D^*(\boldsymbol{\delta}, \mathbf{w}) = \operatorname*{argmin}_{\mathbf{z} \in D} \sum_{i=1}^{n} \frac{(z_i + \delta_i)^2}{w_i}. \tag{3.73}$$

As $w_i > 0$ is fixed, we know that $g_D^*(\boldsymbol{\delta}, \mathbf{w})$ is locally Lipshitz-continuous in $\mathbf{w}$ by lemma 3.4.2. Thus, for all fixed D and sequences $(\tilde{\boldsymbol{\delta}}_m, \tilde{\mathbf{w}}_m)$ such that

$$(\tilde{\boldsymbol{\delta}}_m, \tilde{\mathbf{w}}_m) \to (\boldsymbol{\delta}, \mathbf{w}) \quad \text{as} \quad m \longrightarrow \infty, \tag{3.74}$$

we have that

$$g_D^*(\tilde{\boldsymbol{\delta}}_m, \tilde{\mathbf{w}}_m) \to g_D^*(\boldsymbol{\delta}_m, \mathbf{w}_m) \quad \text{as} \quad m \longrightarrow \infty. \tag{3.75}$$

Now, we may assume w.l.o.g. that $\boldsymbol{\omega}_m > 0$. Then we immediately have that

$$\mathbf{z}^*(\boldsymbol{\omega}_m) = g_{B(f)}^*(\mathbf{0}, \boldsymbol{\omega}_m) \longrightarrow g_{B(f)}^*(\mathbf{0}, \mathbf{w}) = \mathbf{z}^*(\mathbf{w}) \quad \text{as} \quad m \longrightarrow \infty, \tag{3.76}$$

proving (3.71).

To show (3.72), let $\boldsymbol{\delta}_m = \boldsymbol{\omega}_m - \mathbf{w}$, and note that

$$B(f_{\boldsymbol{\omega}_m,\beta}) = \Big\{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}(S) \leq f(S) - \beta\boldsymbol{\omega}_m(S) \;\; \forall\, S \in \mathcal{D}, \\ \mathbf{x}(\mathcal{V}) = f(\mathcal{V}) - \beta\boldsymbol{\omega}_m(\mathcal{V})\Big\} \tag{3.77}$$

$$= \Big\{\mathbf{x} + \beta\left[\mathbf{w} - \boldsymbol{\omega}_m\right] \; : \mathbf{x}(S) \leq f(S) - \beta\boldsymbol{\omega}_m(S) - \beta\left[\mathbf{w}(S) - \boldsymbol{\omega}_m(S)\right] \;\; \forall\, S \in \mathcal{D}, \\ \mathbf{x}(\mathcal{V}) = f(\mathcal{V}) - \beta\boldsymbol{\omega}_m(\mathcal{V}) - \beta\left[\mathbf{w}(\mathcal{V}) - \boldsymbol{\omega}_m(\mathcal{V})\right]\Big\} \tag{3.78}$$

$$= \Big\{\mathbf{x} + \beta\left[\mathbf{w} - \boldsymbol{\omega}_m\right] \; : \mathbf{x}(S) \leq f(S) - \beta\mathbf{w}(S) \;\; \forall\, S \in \mathcal{D}, \\ \mathbf{x}(\mathcal{V}) = f(\mathcal{V}) - \beta\mathbf{w}(\mathcal{V})\Big\} \tag{3.79}$$

$$= \left\{\mathbf{x} - \beta\boldsymbol{\delta}_m : \mathbf{x} \in B(f_{\mathbf{w},\beta})\right\}. \tag{3.80}$$

Thus

$$\mathbf{y}^*(\boldsymbol{\omega}_m, \beta) = \operatorname*{argmin}_{\mathbf{y} \in B(f_{\boldsymbol{\omega}_m,\beta})} \|\mathbf{y}\|_2^2 \tag{3.81}$$

$$= \operatorname*{argmin}_{\mathbf{y}' \in B(f_{\mathbf{w},\beta})} \left\|\mathbf{y}' - \beta\boldsymbol{\delta}_m\right\|_2^2 \tag{3.82}$$

$$= g^*_{B(f_{\mathbf{w},\beta})}(\beta\boldsymbol{\delta}_m, \mathbf{1}). \tag{3.83}$$

Thus, since $\boldsymbol{\delta}_m \to \mathbf{0}$,

$$\mathbf{y}^*(\boldsymbol{\omega}_m, \beta) = g^*_{B(f_{\mathbf{w},\beta})}(\beta\boldsymbol{\delta}_m, \mathbf{1}) \longrightarrow g^*_{B(f_{\mathbf{w},\beta})}(\mathbf{0}, \mathbf{1}) = \mathbf{y}^*(\mathbf{w}, \beta) \quad \text{as} \quad m \longrightarrow \infty, \tag{3.84}$$

proving (3.72). The theorem is proved. □

The above problem allows us to generalize the previous results of size-penalized submodular optimization to general weighted penalties. This result may have several significant practical implications; several of these we explore later in the context of the network flow analysis results.

An interesting corollary to the above theorem is that the original formulation of the minimum norm problem is still valid when the norm being optimized over is reweighted. It may be that this would open up an way to remove some of the numerical difficulties often encountered with the minimum norm problem [Jegelka et al., 2011]. More formally,

Corollary 3.4.3 (Validity of Weighted Minimum Norm.)**.** *Let* $\mathbf{z}^*$ *be the optimal value of the weighted minimum norm problem, with* $\mathbf{w} \in \mathbb{R}^{|\mathcal{V}|}$, $\mathbf{w} > 0$. *Let* $U_1 = \{i : z_i^* < 0\}$ *and* $U_2 = \{i : z_i^* \leq 0\}$, *Then both* U_1 *and* U_2 *minimize* $f(U)$ *over all subsets* $U \in \mathcal{D}$. *Furthermore, for all* $U^\dagger \in \mathcal{D}$ *such that* $f(U^\dagger) = \min_{U \in \mathcal{V}}$, $U_1 \subseteq U^\dagger \subseteq U_2$. *In other words, the weighted minimum norm vector* $\mathbf{z}^*$ *may be substituted for the original minimum norm vector.*

Proof. Set $\beta = 0$ in theorem 3.4.1. □

3.4.1 Handling the Case of $w_i = 0$

One of the challenging aspects here is that we might be interested in the case of $w_i = 0$. In theory, this can be easily handled by simply allowing w_i to be so small that its effect on the problem is negligibly different from $w_i = 0$; in other words, we can see it as the limit $w_i \searrow 0$. In practice, this leads to numerical issues. Thus we propose here a numerically stable method to work with $w_i = 0$ by investigating the limiting behavior.

Theorem 3.4.4. *Let* $\mathbf{w} \geq 0$ *and define* $Q = \{i \in \mathcal{V} : w_i = 0\}$. *Then let*

$$\mathbf{z}_Q^* = \min_{\mathbf{z} \in B(f)} \sum_{i \in Q} z_i^2. \tag{3.85}$$

and let

$$\mathbf{z}^* = \operatorname*{argmin}_{\substack{\mathbf{z} \in B(f) \\ \sum_{i \in Q} \|\mathbf{z}[Q]\|_2 = \left\|\mathbf{z}_Q^*[Q]\right\|_2}} \sum_{i \notin Q} \frac{z_i^2}{w_i}. \tag{3.86}$$

where $\mathbf{z}[Q]$ *denotes the vector of elements of* $\mathbf{z}$ *in* Q. *Then, for*

$$\mathbf{z}_\varepsilon^* = \operatorname*{argmin}_{\mathbf{z} \in B(f)} \sum_{i \in \mathcal{V}} \frac{z_i^2}{\max(\varepsilon, w_i)}, \tag{3.87}$$

we have that

$$\mathbf{z}_\varepsilon^* \to \mathbf{z}^* \text{ as } \varepsilon \searrow 0 \tag{3.88}$$

Proof. Consider the form of the optimization problem in (3.87). For ε sufficiently small, we have that

$$\mathbf{z}_\varepsilon^* = \operatorname*{argmin}_{\mathbf{z} \in B(f)} \left[\sum_{i \notin Q} \frac{z_i^2}{w_i}\right] + \varepsilon^{-1} \left[\sum_{i \in Q} z_i^2\right] \tag{3.89}$$

As $\varepsilon^{-1} \nearrow \infty$, the optimum value of $\mathbf{z}$ is constrained to be on the simplex in which $\sum_{i \in Q} z_i^2$ is minimal. This value is given by C_Q above, and this is the constraint that is enforced explicitly in (3.86). Since everything is continuous, it is valid to take the limit as $\varepsilon \searrow 0$. Thus the theorem is proved. □

It is outside the current realm of our investigation how to implement this in the inner workings of the general minimum norm algorithm; however, we will revisit this issue later when proving the correctness of the network flow version of the weighted reduction algorithm.

3.5 Network Flow Solutions

We now turn our attention to the specific case of network flows. The linear parametric flow problem is similar to the flow problem described earlier, except that now we allow the capacity functions – analogous to the unary energy terms – to be a non-decreasing linear function of the weighting term β. Previously, we treated this global weighting term as having equal influence on all nodes. This chapter considers the case where the influence of β has a different weight on each node. Specifically, we replace β with βw_i in theorem 2.2.1. In this case, we are still able to compute the entire path directly. This result, while interesting in its own right, also sets the stage for our later results for total variation minimization.

To be specific, we extend the problems in theorem 2.2.1 as follows:

Energy Minimization: $\mathcal{P}_E(\beta, \mathbf{w})$. Let $\mathbf{E}_1$ and $\mathbf{E}_2$ be defined as in theorem 2.2.1, and let

$$\mathbf{X}^*(\beta, \mathbf{w}) = \operatorname*{Argmin}_{\mathbf{x} \in \{0,1\}^n} \sum_{i \in \mathcal{V}} (E_i(x_i) - \beta w_i x_i) + \sum_{\substack{(i,j) \in \mathcal{E} \\ i<j}} E_{ij}(x_i, x_j) \tag{3.90}$$

and let

$$\mathcal{S}_E^*(\beta, \mathbf{w}) = \big\{ \{i : x_i^* = 1\} \ : \mathbf{x}^* \in \mathbf{X}^*(\beta, \mathbf{w}) \big\}. \tag{3.91}$$

Quadratic Binary Formulation: $\mathcal{P}_Q(\beta, \mathbf{w})$. Given $\mathbf{Q}$ defined as in theorem 2.2.1, let

$$\mathbf{X}^*(\beta, \mathbf{w}) = \operatorname*{Argmin}_{\mathbf{x} \in \{0,1\}^n} \mathbf{x}^T (\mathbf{Q} - \beta \operatorname{diag} \mathbf{w})\, \mathbf{x}, \tag{3.92}$$

and let

$$\mathcal{S}_Q^*(\beta, \mathbf{w}) = \big\{\{i : x_i^* = 1\} \ : \mathbf{x}^* \in \mathbf{X}^*(\beta, \mathbf{w})\big\}. \tag{3.93}$$

Minimum Cut Formulation: $\mathcal{P}_C(\beta, \mathbf{w})$. Let the graph structure be defined as in theorem 2.2.1, and define capacities c_{ij}, $(i, j) \in \mathcal{E}$ on the edges as:

$$c_{si} = c_{si}(\beta) = [a_i(\beta, \mathbf{w})]^+ \quad c_{ji} = c_{ij} = -\frac{q_{ij}}{2},\ i < j \quad c_{jt} = c_{jt}(\beta) = [a_j(\beta, \mathbf{w})]^- \tag{3.94}$$

where

$$a_i(\beta, \mathbf{w}) = \frac{1}{2} \sum_{i' : i' < i} q_{i',i} + (q_{ii} - \beta w_i) + \frac{1}{2} \sum_{j \ : \ i < j} q_{ij}. \tag{3.95}$$

Then the minimum cut solution $S_C^*(\beta)$ is given by

$$S_C^*(\beta) = \underset{\substack{S \subset \mathcal{V}' \\ s \in S,\, t \in \mathcal{V}' \setminus S}}{\text{Argmin}} \sum_{(i,j) \in \delta(S, \mathcal{V}' \setminus S)} c_{ij}(\beta, \mathbf{w}). \tag{3.96}$$

It is a simple matter to show that the above are equivalent:

Theorem 3.5.1. *$\mathcal{P}_E^*(\beta, \mathbf{w})$, $\mathcal{P}_Q^*(\beta, \mathbf{w})$, and $\mathcal{P}_C^*(\beta, \mathbf{w})$ from the above description are equivalent in the sense that any minimizer of one problem is also a minimizer of the others, i.e.*

$$\mathcal{S}_E^*(\beta, \mathbf{w}) = \mathcal{S}_Q^*(\beta, \mathbf{w}) = \mathcal{S}_C^*(\beta, \mathbf{w}). \tag{3.97}$$

Proof. Replace β with βw_i or $\beta \mathbf{w}$ as appropriate in the proof of theorem 2.2.1. □

However, it is a much more complicated endeavor to show that this formulation can be solved exactly in a similar manner to the previous result. In the end, we prove the following result: analogously to before, we find an optimal pseudoflow $\boldsymbol{\alpha}^*$ such that the zero crossings of $\mathbf{r}(\boldsymbol{\alpha}^*) - \beta \mathbf{w}$, with $\mathbf{r}(\boldsymbol{\alpha})$ defined as in 2.3.2, give the level sets of the augmented problem. The purpose of the current chapter is to define these relationships explicitly.

The above result works as well for general network flow solutions as well. In this case, we have the following immediate corollary to theorem 3.4.1:

Corollary 3.5.2. *Let* $\mathbf{w}$ *be a collection of positive weights. Then* $\boldsymbol{\alpha}^*$ *is the optimal pseudoflow solution to the weighted minimum norm problem*

$$\boldsymbol{\alpha}^* = \operatorname*{argmin}_{\boldsymbol{\alpha} \in \mathcal{A}} \sum_{i \in \mathcal{V}} \frac{r^2(\boldsymbol{\alpha})}{w_i} \tag{3.98}$$

if and only if $\forall\, \beta \in \mathbb{R}$*, the optimal cut* $S^*(\beta, \mathbf{w})$ *for* $\mathcal{P}_G$ *satisfies*

$$\{i \in \mathcal{V} : r_i(\boldsymbol{\alpha}^*) < \beta w_i\} \subseteq S^*(\beta, \mathbf{w}) \subseteq \{i \in \mathcal{V} : r_i(\boldsymbol{\alpha}^*) \leq \beta w_i\}\,. \tag{3.99}$$

Proof. Replace $B(f)$ with its representation for network flow problems as given in theorem 2.3.2. □

An alternate version of this, which is handy for the proofs, is the following:

Corollary 3.5.3. $\boldsymbol{\alpha}^\dagger$ *is optimal for a problem if and only if, for all* $\beta \in \mathbb{R}$*, the following condition holds for each ordered pair* $i, j \in \mathcal{V}$*,* $i < j$*:*

$$\textit{if} \quad r_i(\boldsymbol{\alpha}^\dagger) - \beta w_i \leq 0 < r_i(\boldsymbol{\alpha}^\dagger) - \beta w_j, \quad \textit{then} \quad \alpha_{ij} = -|q_{ij}| \tag{3.100}$$

$$\textit{if} \quad r_i(\boldsymbol{\alpha}^\dagger) - \beta w_i > 0 \geq r_i(\boldsymbol{\alpha}^\dagger) - \beta w_j, \quad \textit{then} \quad \alpha_{ij} = |q_{ij}| \tag{3.101}$$

Proof. Follows immediately from noting that criteria (3.99) in corollary 3.5.2 specifies that if there exists a β that separates two scaled reductions, then there is valid cut in the associated flow problem that separates these nodes. This, however, is equivalent to the condition on the α terms given by (3.100) or (3.101). □

3.5.1 Anchoring Nodes with Differing or Infinite Weights

While the original interpretation of the weights is to vary the influence of each node in the regularization path, a natural extension in the network flow context is to use these weights to fix nodes at particular reduction values – informally, to make them less responsive to the influence of the external flow during the optimization. This is done by scaling both the base reduction value q_{ii} and the β term by w_i instead of just β. In this context, then, replace $\mathbf{r}(\boldsymbol{\alpha})$ with $\mathbf{r}^{\mathbf{w}}(\boldsymbol{\alpha})$, defined as

$$r_i^{\mathbf{w}}(\boldsymbol{\alpha}) = w_i q_{ii} + \frac{1}{2}\left[\sum_{i' < i} \left(q_{i',i} + \alpha_{i'i}\right) + \sum_{j\,:\,i<j} \left(q_{ij} - \alpha_{ij}\right)\right] \tag{3.102}$$

and the node membership changes sign at

$$S^*(\beta) = \mathbb{I}\{r_i^{\mathbf{w}}(\boldsymbol{\alpha}^*) - \beta w_i \leq 0\} \tag{3.103}$$

$$= \mathbb{I}\left\{ [q_{ii} - \beta] + \frac{1}{2w_i} \left[\sum_{i'<i} \left(q_{i',i} + \alpha^*_{i'i} \right) + \sum_{j\,:\,i<j} \left(q_{ij} - \alpha^*_{ij} \right) \right] \right\} \tag{3.104}$$

As a result, it is possible to anchor nodes at a particular value by setting the corresponding weights to a high value and adjusting the corresponding source-to-sink weights to match. In particular, taking the limit as $w_i \to \infty$ in (3.104) causes the membership to be fixed at the internal reduction value. In this case, it effectively fixes these nodes at a pre-determined reduction value, even though the node still influences other nodes in the flow. In particular, this fact is used below to handle the case where we wish to look at the L_1 norm of a reduction value, as we explain later.

3.5.2 Algorithm

In the unweighted algorithm, one of the key routines was shifting the reduction level of all the nodes by the average on that region. This allowed us to effectively bisect the problem at that value of β; this in turn allowed us to set some of the edges as saturated, fixing them at their extreme values. When the resulting solution on a region is that the flow solution sets all reductions to the average value – zero in the scaled version – that region is connected in the final solution, and we can set it to its common value.

Now that we are putting weights on the nodes, the idea of an "average" is redefined. Now, it is the value of the β_μ such that, if all the reductions values were in the same contiguous block, $r_i(\boldsymbol{\alpha}) - \beta_\mu w_i$ would equal 0. This means that when β is increased or decreased, all of these nodes would flip at the same time.

Formally, this β value on a set T is given by:

$$\beta_\mu(T, \boldsymbol{\alpha}) = \frac{\sum_{i \in T} r_i(\boldsymbol{\alpha}) w_i}{\sum_{i \in T} w_i}. \tag{3.105}$$

This behavior is consistent with the augmentation scheme discussed earlier – a node with weight 2 would count the same as 2 nodes in the "average."

Now, when solving the resulting flow equation, it is important that the reduction values are scaled away from the mean such that shifts in the flow affect the reduction value on the correct scale. Since all we care about are the zero-crossings of the nodes, we then rescale all the starting reduction values away from the mean by w_i^{-1}. In other words, we end up setting the unary potential value ρ_i used in calculating the cut to

$$\rho_i = \frac{1}{w_i}\left[w_i q_{ii} - \beta_\mu(T, \boldsymbol{\alpha})\right] = q_{ii} - \frac{1}{w_i}\beta_\mu(T) \tag{3.106}$$

Note that $\sum_{i \in T} w_i \rho_i = 0$, as we would expect, and that when used to split the regions, β_μ will always separate the starting reductions ρ_i into positive and negative components, ensuring a non-trivial cut solution when this is incorporated into the algorithm. As before, the resulting cut dictates the edges that will be saturated; this is how the solution in the rest of the problem is tracked.

In the case of $w_i = 0$, these nodes simply do not enter into the average; In rescaling them away from the mean, we set them to extreme positive or negative numerical values. This approach is consistent with theorem 3.4.4; it will force the squared reduction value in these nodes to be minimized.

Thus we can define a weighted version of the bisection routine, WeightedBisection-Cut, in the previous chapter to work with weighted nodes:

Note that what is important once we are done with the algorithm is not the resulting values of $r_\mu(T)$, but that the edges spanning the cut are saturated. These are fixed per the same logic as in section 2.5.

Theorem 3.5.4 (Correctness of Algorithm 3)**.** *Upon termination, algorithm 3 correctly finds* $\boldsymbol{\alpha}^*$*.*

Proof. This proof is nearly identical to the proof of algorithm 1; the difference is that we instead work with the condition of optimality given in corollary 3.5.3.

Note first that for $w_i = 0$, the method used to set the values of ρ_i on these nodes effectively sets them to their numerical extremes, ensuring that all operations seek to pull them as far as possible towards zero. Thus we are consistent with the behavior of theorem 3.4.4.

Algorithm 2: WeightedBisectionCut

Input: Submodular $\mathbf{Q}$, weights $\mathbf{w}$, subset of nodes T.

Output: Bisecting cut $S \subseteq T$ or $\emptyset$ if all nodes in same partition.

$\boldsymbol{\rho} \leftarrow \mathbf{0}$

if $\sum_{i \in T} w_i = 0$ **then**

// In this case, the optimal cut that divides the reductions into the positive or negative components is all that matters; $w_i = 0$ means this result is true for all β.

for $i \in T$ **do** $\rho_i \leftarrow q_{ii}$,

else

$$\mu \leftarrow \frac{\sum_{i \in T} q_{ii} w_i}{\sum_{i \in T} w_i}$$

.

// M here denotes the largest numerically stable number.

for $i \in T$ **do**

$$\rho_i \leftarrow \begin{cases} q_{ii} - w_i^{-1} \beta_\mu(T) & w_i > 0 \\ M \operatorname{sign}(\beta_\mu(T)) & w_i = 0 \end{cases},$$

if $\boldsymbol{\rho}_{[T]} = 0$ **then return** $\emptyset$ // These nodes are all at a common reduction level already.

$\mathbf{Q}' \leftarrow \mathbf{Q}_{[T]}$ with diagonal replaced by $\boldsymbol{\rho}_{[T]}$.

$S_T^* \leftarrow$ Minimum cut on $(T, \mathcal{E}_{[T]})$, with capacities formed from $\mathbf{Q}'$ by theorem 2.2.1.

return S_T^*

Let i, j, $i < j$, be any pair of nodes such that $q_{ij} \neq 0$, and let $\boldsymbol{\alpha}^\dagger$ be the solution returned by Algorithm 3.

First, suppose there existed a β such that

$$r_j(\boldsymbol{\alpha}^\dagger) - \beta w_j = r_i(\boldsymbol{\alpha}^\dagger) - \beta w_i = 0 \tag{3.107}$$

Then trivially, the optimality criterion of Lemma 3.5.3 is satisfied.

Algorithm 3: FindWeightedReductions

Input: Submodular $\mathbf{Q}$, vector of non-negative weights $\mathbf{w}$.

Output: $\boldsymbol{\alpha}^*$ satisfying condition **??**.

// *Begin by calling* WeightedBisectReductions *below on the full set* $\mathcal{V}$ *to get* $\boldsymbol{\alpha}$.

return WeightedBisectReductions *(*$T = \mathcal{V}$, $\boldsymbol{\alpha} = \mathbf{0}$, $\mathbf{Q}$*)*

// *Note:* $\boldsymbol{\alpha}_{[T]}$ *denotes* $\boldsymbol{\alpha}$ *restricted to edges with both nodes in* T.

WeightedBisectReductions (T, $\boldsymbol{\alpha}$, $\mathbf{Q}$)

$S_T^* \leftarrow$ WeightedBisectionCut$(\mathbf{Q}, \mathbf{w}, T)$

if $S_T^* = \emptyset$ **or** $S_T^* = T$ **then**

// We are done on this set.

return $\boldsymbol{\alpha}_{[T]}$

// *Fix the flow on edges in the cut by adjusting the source/sink capacities of each node, then removing those edges.*

for $i \in S_T^*$, $j \in T \backslash S_T^*$, $i < j$ **do** $\alpha_{ij} \leftarrow -q_{ij}$, $q_{ii} \leftarrow q_{ii} - q_{ij}$, $q_{ij} \leftarrow 0$

for $i \in T \backslash S_T^*$, $j \in S_T^*$, $i < j$ **do** $\alpha_{ij} \leftarrow q_{ij}$, $q_{jj} \leftarrow q_{jj} + q_{ij}$, $q_{ij} \leftarrow 0$

// *Recursively solve on the two partitions to fix the other* $\boldsymbol{\alpha}$*'s.*

$\boldsymbol{\alpha}_{[S_T^*]} \leftarrow$ WeightedBisectReductions (S_T^*, $\boldsymbol{\alpha}$, $\mathbf{Q}$)

$\boldsymbol{\alpha}_{[T \backslash S_T^*]} \leftarrow$ WeightedBisectReductions ($T \backslash S_T^*$, $\boldsymbol{\alpha}$, $\mathbf{Q}$)

return $\boldsymbol{\alpha}_{[T]}$

Next, suppose that there exists a β such that

$$r_i(\boldsymbol{\alpha}^\dagger) - \beta w_i \leq 0 < r_j(\boldsymbol{\alpha}^\dagger) - \beta w_j \tag{3.108}$$

Then, according to the termination criterion of WeightedBisectionCut – in which it returns $S_T^* = T$ or $S_T^* = \emptyset$ – the edge between nodes i and j will be saturated in such a way that the reduction value of $r_i(\boldsymbol{\alpha}^\dagger)$ is maximized and the reduction value of $r_j(\boldsymbol{\alpha}^\dagger)$ is minimized, i.e. $\alpha_{ij} = -|q_{ij}|$. Thus condition (3.100) in corollary 3.5.3 is satisfied. Similarly,

if the order in 3.108 is reversed, then condition (3.101) is satisfied.

As the above holds for any pairs of nodes i, j, the optimality criteria of corollary 3.5.3 are satisfied globally, proving the correctness of the algorithm. □

3.6 General Optimization

This section improves upon the result of theorem 2.6.1 given at the end of chapter 2: using the weighting method above, it is possible to anchor weights at specific values in the optimization. This indicates that the reduction levels of these nodes do not vary in the optimization even though they are treated the same way as the other nodes. This provides a simple way of incorporating an L_1 fit penalty into our optimization. Formally, the following corollary to theorem 2.6.1 lays this out:

Theorem 3.6.1. *Consider* ($\Re_B$)*: suppose* $\gamma : \mathbb{R}^n \to \mathbb{R}^+$ *can be expressed as*

$$\gamma(\mathbf{u}) = \text{const} + \|\mathbf{u} - \mathbf{a}\|_2^2 + \lambda \left[\sum_i \xi_i(u_i) + \sum_{i,j} w_{ij}|u_i - u_j| \right] \tag{3.109}$$

with $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$*,* $\lambda \geq 0$*, and* $w_{ij} \geq 0$*, and where* ξ_i *is a convex piecewise-linear function. Then the minimizer*

$$\mathbf{u}^* = \underset{\mathbf{u} \in \mathbb{R}^n}{\operatorname{argmin}} \ \gamma(\mathbf{u}) \tag{3.110}$$

can be found exactly using Algorithm 3.

Proof. First, to accommodate the $\xi_i(u_i)$ functions, suppose, without loss of generality, that each function is defined by a sequence of m_i inflection points

$$-\infty = b_{i0} < b_{i1} < b_{i2} < \cdots < b_{im_i} = \infty \tag{3.111}$$

and an associated line slope θ_{ij} between each, so

$$\theta_{ij} = \frac{\xi_i(b_{ij}) - \xi_i(b_{i,j-1})}{b_{ij} - b_{i,j-1}}. \tag{3.112}$$

Then, since $\xi_i(x)$ is linear, it is easy to show that at any point x, $\xi_i(x)$ can be expressed as the integral over sums of step functions; if c_i^- and c_i^+ are the values of these step functions,

$$\xi_i(u_i) = \text{const} + \int_{-M}^{u_i} \left\{ \left[\sum_{j:b_{ij} \leq x} c_{ij}^+ \right] + \left[\sum_{j:b_{ij} > x'} c_{ij}^- \right] \right\} \mathrm{d}x. \tag{3.113}$$

The exact values of c_{ij}^{+} and c_{ij}^{-} can be found easily by solving a simple linear system, and they can satisfy $c_{ij}^{+} + c_{ij}^{-} = 0$ by adjusting the constant term in front of the integral; this ensures these pairwise terms satisfy the submodularity condition. Thus $\xi_i(\cdot)$, can be encoded on the graph by creating m_i auxiliary nodes $u'_{i,j}$ fixed at reduction values $b_{i1}, ..., b_{i,m_i}$ using the method described in section 3.5.1. Setting the overall weight of the edge using

$$E_2(x_i, x'_{ij}) = \begin{bmatrix} 0 & c_{ij}^{+} \\ c_{ij}^{-} & 1 \end{bmatrix} \tag{3.114}$$

ensures that the cost of a cut at any region between b_{ij} and $b_{i,j+1}$ incurs cost

$$\left[\sum_{j' \leq j} c_{ij'}^{+}\right] + \left[\sum_{j' > j} c_{ij'}^{-}\right]. \tag{3.115}$$

Thus, in the final integral, the reduction level u_i incurs the penalty $\text{const} + \xi(u_i)$. The proof then follows identically to theorem 2.6.1 with some minor modifications – namely, for the auxiliary nodes, we are actually working with:

$$\mathbf{1}_{\{M \cdot u_{n+i} \leq M \cdot \beta\}} \tag{3.116}$$

but this easily converts to

$$\mathbf{1}_{\{u_{n+i} \leq \beta\}}. \tag{3.117}$$

As the L_2 fit terms on the auxiliary nodes become constant in the limit, the rest of the proof of theorem 2.6.1 goes through for this form as well, giving us the desired result. □

3.7 Conclusion

In this chapter, we extended the existing theory of size-constrained submodular optimization first proposed by [Nagano et al., 2011] to the weighted case. This theoretical tool has several important consequences. In the case of network flows, it gives us the ability to make nodes more or less affected by the optimization process. This opens the door to the general optimization problem given in theorem 3.6.1. In the next chapter, we extend these results to develop a full treatment of the entire regularization path over all λ.

Chapter 4

GENERALIZED REGULARIZATION PATHS

4.1 The Generalized Linear Parametric Flow Problem

In the previous chapters, we developed a routine to efficiently solve ($\Re_{\mathrm{B}}$). The end purpose of this chapter is to present a complete characterization of the regularization path over λ. Because of the intrinsic connections of this problem with the flow-based algorithms and reduction structures presented in chapters 2 and 3, we begin with slight reformulations of $\mathbf{r}(\boldsymbol{\alpha})$ and related.

For notational brevity, we shorten q_{ii} to q_i and denote

$$\boldsymbol{\imath}_{\{i>j\}} = \begin{cases} 1 & i > j \\ -1 & \text{otherwise} \end{cases} . \tag{4.1}$$

To efficiently take care of the ordering on $\boldsymbol{\alpha}$ – in writing α_{ij}, we assume that $i < j$ – denote

$$\alpha_{(i,j)} = \alpha_{\min(i,j),\max(i,j)}. \tag{4.2}$$

Now, we assume that the reduction terms that govern these levels are given by

$$r_i(\boldsymbol{\alpha}\ ;\ \tau) = \tau q_i + \rho_i + \sum_{i,j} \boldsymbol{\imath}_{\{i>j\}}\, \alpha_{(i,j)} \tag{4.3}$$

where τ is defined on an interval $[\tau_{\min}, \tau_{\max}]$. In this problem where we still constrain the flow variable $\boldsymbol{\alpha}$ to be on $\mathcal{A}$, i.e. $|\alpha_{ij}| \leq |q_{ij}|$ for all $i < j$. Recall that, for the optimal $\boldsymbol{\alpha}^* = \boldsymbol{\alpha}^*(\tau, \mathbf{w})$, $\mathbf{r}(\boldsymbol{\alpha}^*(\tau, \mathbf{w})\ ;\ \tau)$ in (4.3) gives us the optimal cut for all β throught the zero-crossings of $\mathbf{r}(\boldsymbol{\alpha}^*(\tau, \mathbf{w})\ ;\ \tau) - \beta\mathbf{w}$.

While the weights on the β parameter, $\mathbf{w}$, are positive, we make no assumptions about q_i or ρ_i except that they are real numbers. In particular, they can be positive or negative. Given these constraints, the goal of this chapter is to present an algorithm to efficiently calculate the optimum value of $\mathbf{r}$ for all $\tau \in [\tau_{\min}, \tau_{\max}]$ and $\beta \in \mathbb{R}$. For τ fixed, the form of $\mathbf{r}$ given in (4.3) is immediately compatible with the original form of the reduction given in equation (2.17) of theorem 2.3.2. Applications of this include the ability to completely solve the regularization path for ($\Re_{\mathrm{B}}$) and, using the model presented in the next chapter, the total variation problem ($\Re_{\mathrm{TV}}$).

Note, as well, that a number of related problems fit into this exact formulation. In particular, the total variation problem of the next chapter can be solved using this approach.

4.2 Structure of the Solution Path

Building on the results of the previous chapters, we are able to describe the complete properties of the solution path. Namely, we trace the levels of the reduction terms as a function of the path parameter τ; this permits an easy transformation into the cuts at any point in the function.

The fundamental problem we examine is to trace the reduction values $\mathbf{r}$ while the unary term is a linear function. In this context, the model we assume for the reduction values is a function of τ

$$\mathbf{r}^*(\tau) = \mathbf{r}(\boldsymbol{\alpha}^*(\tau, \mathbf{w}) \; ; \; \tau) = \rho_i + \tau q_i + \sum_{i,j} \boldsymbol{\imath}_{\{i>j\}} \, \alpha^*_{(i,j)}(\tau, \mathbf{w}) \tag{4.4}$$

where $\boldsymbol{\alpha}^*(\tau, \mathbf{w})$ is the optimal solution of the weighted minimum norm problem at τ, described in detail in definition 3.3.1:

$$\boldsymbol{\alpha}^*(\tau, \mathbf{w}) = \operatorname*{argmin}_{\boldsymbol{\alpha} \in \mathcal{A}} \sum_{i \in \mathcal{V}} \frac{\mathbf{r}^2(\boldsymbol{\alpha} \; ; \; \tau)}{w_i}. \tag{4.5}$$

For convenience, we abbreviate $\mathbf{r}(\boldsymbol{\alpha}^*(\tau, \mathbf{w}) \; ; \; \tau)$ as $\mathbf{r}^*(\tau)$.

Our contribution is to solve this exactly for a range of τ, denoted here as $[\tau_{\min}, \tau_{\max}]$, immediately giving the optimal cut for all $\tau \in [\tau_{\min}, \tau_{\max}]$ and $\beta \in \mathbb{R}$. This range of τ may be effectively infinite if needed; for sufficiently large or small τ, the solution is completely dominated by the τq_i term and does not change for more extreme values.

At a high level, we begin by solving the problem exactly for one τ at an end point using one of the previously described algorithms, either FindWeightedReductions or FindReductions depending on whether weights are specified. We then track the evolution of the regularization path as a function of τ. The structure of the path is a collection of joins and splits of tied regions of the reduction values $\mathbf{r}^*(\tau)$; it furthermore turns out that the value of these regions is linear in between each of the split and merge points. Thus our algorithm is based primarily around calculation of the next split and merge points as we move a global τ_c through $[\tau_{\max}, \tau_{\min}]$. The final result is collection of of linear segments that exactly describe the regularization path. We also maintain the optimality of $\boldsymbol{\alpha}$ throughout the computation.

The fundamental unit we work with here is a *Reduction Region*: a tied, connected set of nodes that have a common reduction level – defined below as the value of β where each node changes membership in the optimal cut. The path is given in terms of these reduction regions. As such, they are valid for a certain range of τ dependent upon an internal consistency property and the external behavior of the the neighboring reduction regions. Formally,

Definition 4.2.1 (Reduction Region)**.** *A collection of one or more nodes $\mathcal{R} \subseteq \mathcal{V}$ is called a reduction region w.r.t. a particular optimal reduction level $\mathbf{r}^*(\tau)$ if*

1. *Consistency: $w_i^{-1} r_i^*(\tau) = const$ for all $i \in \mathcal{R}$.*

2. *Optimality: For all $j \notin \mathcal{R}$ such that there exists a node $i \in \mathcal{R}$ where $q_{ij} \neq 0$, $w_j^{-1} r_j^*(\tau) \neq w_i^{-1} r_i^*(\tau)$. Thus optimality in the sense of corollary 3.5.2 on page 70.*

3. *Connectedness: For any pair of nodes $i, j \in \mathcal{R}$, there exists a path of nodes in $\mathcal{R}$ connecting i with j.*

Implicit in this definition is the idea of a *Reduction Level*, defined here as $w_i^{-1} r_i^*(\tau)$. This is equal to the value of β at which the reduction region together changes set membership in the optimal solution $S^*(\beta, \tau)$. Formally, for a region $\mathcal{R}$, we denote the reduction level as this β:

Definition 4.2.2 (Reduction Level)**.** *For a Reduction Region $\mathcal{R}$, $w_i^{-1} r_i^*(\tau)$ is constant in that region; call this value the Reduction Level, denoted by*

$$\beta_{\mathcal{R}}(\tau) = w_i^{-1} r_i^*(\tau) \text{ for all } i \in \mathcal{R} \tag{4.6}$$

Informally, in the solution path, the primary invariant condition we exploit is that as τ changes, the ordering amongst the neighboring reduction levels $\beta_{\mathcal{R}}(\tau)$ doesn't change without a corresponding change in the cut pattern of the solution. This can happen in one of two ways:

1. Two neighboring regions $\mathcal{R}_1$ and $\mathcal{R}_2$ join to become a common level with a shared reduction level.

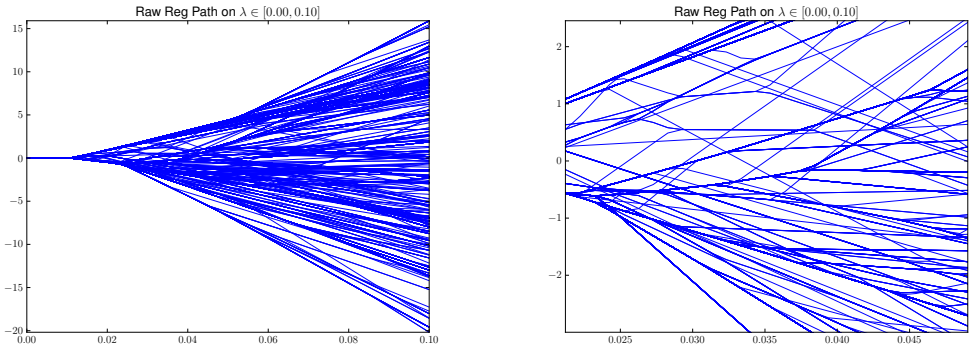


Figure 4.1: The regularization path as given by our algorithm for one example of the total variation problem described in chapter 5, with the lines weighted by the number of nodes in that reduction level. The image on the right is a zoom of the middle section of the left. This problem is a specific instance of the general class of problems we describe here. In particular, the unary terms completely die away at $\tau = 0$, hence the single reduction level on the left.

2. A single region $\mathcal{R}$ splits into two seperate regions with different reduction levels.

The reduction value for a region given the cut pattern and relative orderings on the region, is easily computable; it is just a linear function of τ and the other constants in the problem definition. The join points are thus easy to compute; these simply correspond to points of equality in the reduction values $\beta_{\mathcal{R}}$ of neighboring regions. Split points are harder to compute; they correspond to the τ value at which a transshipment problem on in that region becomes infeasible – in other words, at the point where it is no longer possible to construct a flow that maintains the region at a common reduction level under the capacity restrictions on the edges. Finding the regularization path corresponds to enumerating the network of splits and joins amongst the various levels of the reduction.

An example of this problem, including the network of splits and joins, is shown in Figure (4.1). This problem is a preview of the regularization path of the total variation regularization path on the *castle* image, as described in detail in the next chapter. This problem is a special case of the one described here in that $\rho_i = 0$, so the unary terms

completely die away at $\tau = 0$. In this example, one can clearly see the network of splits and joins among the reduction levels, as well as the linearity between these points.

4.2.1 Computing Reduction Levels

Now, to aid in our algorithms, we define several other variables that make the reduction of the various values more straightforward to compute. We define the external influence on a node in a region $\mathcal{R}$ as

$$\gamma_i = \sum_{j \in \mathcal{V} \setminus \mathcal{R}} \boldsymbol{\imath}_{\{i>j\}} \, \alpha^*_{(i,j)}(\tau, \mathbf{w}). \tag{4.7}$$

where we can similarly set

$$\gamma_{\mathcal{R}} = \gamma_{\mathcal{R}}(\tau) = \sum_{i \in \mathcal{R}} w_i^{-1} \gamma_i \tag{4.8}$$

where $\gamma_{\mathcal{R}}(\tau)$ represents the external influence on $\beta_{\mathcal{R}}(\tau)$ for the region as encoded by the optimal $\boldsymbol{\alpha}$ values. Since, by definition, nodes outside this reduction region are at a different level, the connecting edges are all saturated (see corollary 3.5.3). Thus

$$\gamma_{\mathcal{R}} = \gamma_{\mathcal{R}}(\tau) = \sum_{i \in \mathcal{R}} \gamma_i = \sum_{\substack{i \in \mathcal{R} \\ j \in \mathcal{V} \setminus \mathcal{R}}} w_i^{-1} \boldsymbol{\imath}_{\left\{r_i^*(\tau) > r_j^*(\tau)\right\}} |q_{ij}|. \tag{4.9}$$

Note that as long as the ordering of the reduction value $\beta_{\mathcal{R}'}$ of the other regions relative to $\beta_{\mathcal{R}}$ do not change – that those regions with a larger or smaller reduction levels stay that way – this is simply a constant. Since the invariance of the level ordering among neighbors is one of the conditions for $\mathcal{R}$ to be defined, we drop the dependence of $\gamma_{\mathcal{R}}$ on τ.

For the other terms, we simply use the subscript $\mathcal{R}$ to denote summation over a region:

$$\rho_{\mathcal{R}} = \sum_{i \in \mathcal{R}} w_i^{-1} \rho_i \tag{4.10}$$

$$q_{\mathcal{R}} = \sum_{i \in \mathcal{R}} w_i^{-1} q_i \tag{4.11}$$

As long as the collection of regions doesn't change, these edges are saturated at a fixed value, as gauranteed by the theorems shown in the previous chapters. Then, the level of a region can be computed as

$$\beta_{\mathcal{R}}(\tau) = \frac{1}{|\mathcal{R}|} \left[\gamma_{\mathcal{R}} + \rho_{\mathcal{R}} + \tau q_{\mathcal{R}} \right]. \tag{4.12}$$

Since this just a linear function of τ, we can easily track the exact reduction levels by tracking the pattern of cuts and splits among the reduction regions. The next few sections make this explicit.

4.2.2 Theoretical Foundation

Before presenting the algorithm, we here present the theoretical justification behind the algorithms given in this section. In general, we wish to make explicit and formal the informal statement previously given: The optimal solution is given by the relative orderings of $\beta_{\mathcal{R}}(\tau)$, and this optimality is preserved between any split and join events. The primary theorem that we use to support our results is the following.

Theorem 4.2.3. *Let $\mathscr{R} = (\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_n)$ be a list of reduction regions covering $\mathcal{V}$, each valid with respect to τ_c. Let $\mathscr{R}$ have a partial ordering such that for all neighboring regions $\mathcal{R}_i, \mathcal{R}_j$, $i < j$, we have*

$$\beta_{\mathcal{R}_i}(\tau_c) < \beta_{\mathcal{R}_j}(\tau_c). \tag{4.13}$$

Suppose $\tau_C \in [\tau_{\min}, \tau_{\max}]$, and let $\tau_0 \in [\tau_{\min}, \tau_c)$. Then $\mathscr{R}$ gives the optimal solution for all $\tau \in [\tau_0, \tau_c]$ if and only if

I. For all $\tau \in [\tau_0, \tau_c]$, the ordering condition (4.13) *amongst all neighboring regions $\mathcal{R}_i, \mathcal{R}_j$, $i < j$, is preserved, i.e.*

$$\beta_{\mathcal{R}_i}(\tau) < \beta_{\mathcal{R}_j}(\tau). \tag{4.14}$$

II. WeightedBisectionCut *(Algorithm 2), restricted to $\mathcal{R}$ and with unary potentials given by $\rho_i + \tau_0 q_i + \gamma_i$, returns no cut.*

This theorem is noteworthy in that it gives us conditions under which we can ensure that segments in the regularization path are valid. Note that, by definition, a reduction region $\mathcal{R}$ w.r.t. τ assumes that the problem is solved optimally at that τ. Thus the above theorem gives us a way to gaurantee that the sections of the path are indeed optimal.

Background on Transhipment Problems

Before proving 4.2.3, we review some algorithmic theory that underpins some of our results.

The b-transshipment problem Schrijver [2003] is a network flow problem in which the goal is to distribute excess at some nodes to deficits at other nodes. This is simply a rewording of the original formulation of the network flow problem that we have been working with in various forms up til now. We mention it now as we use one of the fundamental results that allows us to easily prove one of the theorems:

Theorem 4.2.4 (Gale's Theorem, also Corollary 11.2g of [Schrijver, 2003])**.** *Let $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be a directed graph and let $c_{ij} \geq 0$ be the capacity associated with each edge $(i, j) \in \mathcal{E}$. Let b_i give the excess flow at node i, with $\mathbf{b}(\mathcal{V}) = \sum_{i \in \mathcal{V}} b_i = 0$. Then there exists a b-transshipment flow $\mathbf{z}$ satisfying $0 \leq z_{ij} \leq c_{ij}$ if and only if, for all $U \subseteq \mathcal{V}$,*

$$c(\delta^{out}(U)) \geq b(U), \tag{4.15}$$

where

$$c(\delta^{out}(U)) = \sum_{\substack{i \in U \\ j \in \mathcal{V} \setminus U}} c_{ij} \tag{4.16}$$

is the capacity of all edges leaving U.

This theorem is used in proving the following lemma. Intuitively, this lemma shows that if the weighted bisection cut routine indicates that $\mathcal{R}$ is a common region with the same reduction level at both τ_0 and τ_1 – i.e. it is always in the same cut – and that the aspects of the problem external to $\mathcal{R}$ do not change in that range, then is also tied for any τ between τ_0 and τ_1 as well. Formally,

Lemma 4.2.5. *Fix $\mathcal{R}$, τ_0, and τ_1, with $\tau_0 < \tau_1$. Suppose that* WeightedBisectionCut *restricted to $\mathcal{R}$ indicates that the nodes in $\mathcal{R}$ at are at a common reduction level at both τ_0 and τ_1. Then they are at a common reduction level for all $\tau \in [\tau_0, \tau_1]$.*

Proof. The condition that WeightedBisectionCut indicates that all nodes are on the same side of the cut is equivalent to the condition that the constructed flow problem, when

recast as a b-transshipment problem, has a valid solution. By theorem 4.2.4, this is true if and only if, for all $U \subseteq \mathcal{R}$,

$$c(\delta^{out}(U)) \geq b(U), \tag{4.17}$$

where $c(\delta^{out}(U))$ is the capacity of the edges out of a subset of nodes, and $b(U)$ is the total sum of the excess in that region. Here, the excess is given by

$$b_\tau(U) = \sum_{i \in U} w_i \cdot (\gamma_i + \rho_i + \tau q_{ii} - \beta_{\mathcal{R}}(\tau)), \tag{4.18}$$

which is a linear function of τ. The capacity term is given by the pairwise q_{ij} terms and is a constant. By assumption, then, we know that for all $U \subseteq \mathcal{R}$, both

$$c(\delta^{out}(U)) \geq b_{\tau_0}(U) \tag{4.19}$$

$$c(\delta^{out}(U)) \geq b_{\tau_1}(U) \tag{4.20}$$

Thus,

$$c(\delta^{out}(U)) \geq t \cdot b_{\tau_0}(U) + (1-t) \cdot b_{\tau_1}(U) \tag{4.21}$$

for all $t \in [0,1]$. However, since $b_\tau(U)$ is a linear function of τ, for all $\tau \in [\tau_0, \tau_1]$, there exists a $t \in [0,1]$ such that

$$b_\tau(U) = t \cdot b_{\tau_0}(U) + (1-t) \cdot b_{\tau_1}(U) \tag{4.22}$$

Thus, by theorem 4.2.4, this means that the given b-transshipment problem is feasible, which in turn proves the lemma. □

Proof of Theorem 4.2.3

We are now in a position to prove the main result enabling the algorithm.

Proof of theorem 4.2.3. Trivially, if these reduction regions represent the optimal solution, then, by corollary 3.5.3, these conditions are met.

Now, suppose that both I and II are true. Then, by lemma 4.2.5, we know that condition II is true for all $\tau \in [\tau_0, \tau_c]$ as well. Since this is true for all the regions on $[\tau_0, \tau_c]$, then the conditions for optimality, as given in 3.5.3, are met, proving the theorem. □

Now that this theorem is proved, we give the algorithm in two steps.

Algorithm 4: BUILDREGULARIZATIONPATH

Input: Full network structure and variation information as defined by $\mathcal{V}$, $\mathbf{Q}$, and $\boldsymbol{\rho}$; interval on which to calculate $[\tau_{\min}, \tau_{\max}]$.
Output: Full information about reduction levels for all τ in $[\tau_{\min}, \tau_{\max}]$.

// Initialize an empty heap ordered by τ as the key value.
$H \leftarrow$ Empty Heap

// Get all the initial reduction levels.
$\mathbf{r} \leftarrow$ ALPHAREDUCTION$(\mathbf{Q}, \boldsymbol{\rho}, \mathbf{q})$

// The function to add a new region to the heap.
function ADDPOTENTIALEVENTSTOHEAP$(\mathcal{R}, \tau_c)$

  // Add in a possible split to the heap.
  τ, S_1, $S_2 \leftarrow$ CHECKFORREGIONSPLIT$(\mathcal{R})$
  if $\tau > \tau_{\min}$ **then** PUSH $\{\mathcal{R}, \text{“}Split''\text{”}, (S_1, S_2)\}$ **on** H **with key** τ.
  // Add in all possible joins to the heap.
  for $\mathcal{R}'$ *in* R_{active} **do**
    if $\mathcal{R}' \neq \mathcal{R}$ **and** $\mathcal{R}$ *borders* $\mathcal{R}'$ **then**
      τ, $\mathcal{R}_{\text{other}} \leftarrow$ CHECKFORJOIN$(\mathcal{R}, \mathcal{R}', \tau_c)$
      if $\tau \in [\tau_{\min}, \tau_{\max}]$ **then**
        PUSH $\{\mathcal{R},$ “Join” $, \mathcal{R}_{\text{other}}\}$ **on** H **with key** τ

// Form empty lists to track the active regions and the completed segments.
$R_{\text{active}} \leftarrow [],\ R_{\text{complete}} \leftarrow []$

for *each contiguous region* $\mathcal{R}$ *with same reduction* **do** APPEND $\mathcal{R}$ **to** R_{active}

for $\mathcal{R}$ *in* R_{active} **do** ADDPOTENTIALEVENTSTOHEAP$(\mathcal{R})$

while *heap is not empty* **do**
  τ_c, $\mathcal{R}$, type, info $\leftarrow$ **HeapTopValue** (H)

  if *type* = *“Split”* ***and*** $\mathcal{R} \in R_{active}$ **then**
    S_1, $S_2 \leftarrow$ info
    $\mathcal{R}_1$,$\mathcal{R}_2 \leftarrow$ APPLYSPLIT$(\mathcal{R}, S_1, S_2)$

    REMOVE $\mathcal{R}$ **from** R_{active}
    APPEND $\{\tau_c,$ “Split”$, \mathcal{R}, \mathcal{R}_1, \mathcal{R}_2\}$ **to** R_{complete}

    ADDPOTENTIALEVENTSTOHEAP$(\mathcal{R}_1)$
    ADDPOTENTIALEVENTSTOHEAP$(\mathcal{R}_2)$

    APPEND $\mathcal{R}_1$, $\mathcal{R}_2$ **to** R_{active}

  if *type* = *“Join”* **then**
    $\mathcal{R}_{\text{other}} \leftarrow$ info
    if $\mathcal{R}, \mathcal{R}_{other} \in R_{active}$ **then**
      REMOVE $\mathcal{R}$, $\mathcal{R}_{\text{other}}$ **from** R_{active}

      $\mathcal{R}_0 \leftarrow$ APPLYJOIN$(\mathcal{R}, \mathcal{R}_{\text{other}})$
      APPEND $\{\tau_c,$ “Join”$, \mathcal{R}_0, \mathcal{R}, \mathcal{R}_{\text{other}}\}$ **to** R_{complete}

      ADDPOTENTIALEVENTSTOHEAP$(\mathcal{R}_0)$
      APPEND $\mathcal{R}$ **to** R_{active}

return $R_{complete}$

4.2.3 Higher Level Algorithm

The higher level algorithm, given in algorithm 4, procedes by first solving the complete reduction at one of the endpoints using one of the algorithms developed in the previous

chapters. In our case, we chose to calculate it starting at $\tau_{\max}$ and move from larger values of τ to smaller values. The reason has to do with the nature of the total variation minimization problem, for which this is the target application. There, $\tau_{\min} = 0$, and we expect the problem to be joined into one large reduction set at that point. Joins are much cheaper to compute than splits, so we start at $\tau_{\max}$, where there are the most reduction levels, and move to $\tau_{\min}$, where there are the fewest. To make the resulting discussion easier, we thus assume that we calculate the regularization path in terms of decreasing τ, i.e. from right to left. However, the algorithm works equally well for both directions.

Once we have an initial solution, we partition the nodes into regions based on contiguous, common reduction values – each region is formed by a connected set of nodes with the same reduction value, bordered by other regions with differing reduction values. Once these regions are calculated, the algorithm attempts to move a current value of τ, here refered to as τ_c, towards $\tau_{\min}$. The gaurantee at each stage is that all split and join points to the right of the current τ have been calculated; thus at all times the regularization path is known completely on the interval $[\tau_c, \tau_{\max}]$.

The algorithm keeps track of the regularization path in terms of the split and join points. At each of these points, we store the τ value, the reduction value, and what nodes are in each region at each of these points. Because the reduction value of each of the nodes is linear between the points, this information is sufficient to reconstruct the entire path.

These events are tracked by use of a central priority queue that keeps track of the next event affecting the regularization path. The top of this heap is the next event that alters the regularization path, be it a split or join point. The algorithm procedes by repeatedly popping the top point. If it is invalid – i.e. the region it referred to has already been involved in a split or join – then we ignore it and go on to the next one. If it is a join, then it creates the new region, checks to see when/if it splits and what neighboring regions it could join to, then adds these events to the central heap. If the popped event is a split, then it creates the two new regions and does the same for each of these. At the end of this process – when the heap is empty – we have our regularization path.

Formally, the algorithm is given in Algorithm 4. The details of some of the functions, namely CheckForJoin and CheckForRegionSplit are given in the next sections. The

Algorithm 5: CHECKFORJOIN

Input: Two regions $\mathcal{R}_1$ and $\mathcal{R}_2$; current operating path index τ_c

Output: $\tau_{\min}$ if there is no join in $(\tau_{\min}, \tau_c]$.

$\tau \leftarrow \frac{\frac{1}{|\mathcal{R}_1|}\left[\gamma\mathcal{R}_1 + \rho\mathcal{R}_1\right] - \frac{1}{|\mathcal{R}_2|}\left[\gamma\mathcal{R}_2 + \rho\mathcal{R}_2\right]}{\frac{q\mathcal{R}_2}{|\mathcal{R}_2|} - \frac{q\mathcal{R}_1}{|\mathcal{R}_2|}}$..

if $\tau \in (\tau_{\min}, \tau_c]$ **then**
    return τ

else
    return $\tau_{\min}$

rest of the routines listed – APPLYSPLIT and APPLYJOIN – simply abstract the straightforward internal bookkeeping routines.

4.2.4 Computing Joins

The joins are relatively simple to compute; they are just the intersection points where two neighboring levels become equal. Namely, the join lambda for two regions can be computed as

$$\tau^* = \frac{\frac{1}{|\mathcal{R}_1|}\left[\gamma\mathcal{R}_1 + \rho\mathcal{R}_1\right] - \frac{1}{|\mathcal{R}_2|}\left[\gamma\mathcal{R}_2 + \rho\mathcal{R}_2\right]}{\frac{q\mathcal{R}_2}{|\mathcal{R}_2|} - \frac{q\mathcal{R}_1}{|\mathcal{R}_2|}}. \tag{4.23}$$

Since we are moving from right to left, we can define a simple way to check for joins as givin in Algorithm 5.

4.2.5 Computing Splits

Splits are harder to compute than joins. This is true both computationally and theoretically, as computing a join requires solving a simple linear equation, but calculating a split requires finding the smallest τ on which a particular network flow problem becomes infeasible. The principle theorem we are interested in allows us to do a type of bisection to find the smallest τ at which this network flow problem becomes infeasible. This gives us the value at which the region splits, and the corresponding cut gives the resulting partitions of the node.

Informally, the algorithm is simply as follows:

Algorithm 6: CheckForSplit

Input: A region $\mathcal{R}$ and lower bound $\tau_{\min}$.

Output: $\tau_{\min}$ if there is no split in $(\tau_{\min}, \tau_c]$; otherwise, the τ at which the split occurs.

$S^*_{\text{split}} \leftarrow \emptyset$

Function ProbeLowerBound(τ)

$\mathbf{Q}' \leftarrow \mathbf{Q}$ restricted to $\mathcal{R}$, with diaganols given by diag $(\rho_i + \tau q_i + \gamma_i)$.

$S^* \leftarrow$ WeightedBisectionCut($\mathbf{Q}, \mathcal{R}$)

if $S^* = \emptyset$ **or** $S^* = \mathcal{R}$ **then**
return τ

else

// Record this split, overwriting any previous splits so the resulting one is the one valid at the returned τ.

$S^*_{\text{split}} \leftarrow S^*$

$T^* \leftarrow \mathcal{R} \backslash S^*$.

// Find the τ at which the flow across this cut becomes feasible.

$c \leftarrow$ Total Flow from S^* to T^*.

$$\tau' \leftarrow \frac{\left| \frac{\gamma_{S^*} + c}{|S^*|} - \frac{\gamma_{T^*} - c}{|T^*|} \right| + c}{\left| \frac{q_{S^*}}{|S^*|} - \frac{q_{T^*}}{|T^*|} \right|}$$

return ProbeLowerBound(τ')

return ProbeLowerBound($\tau_{\min}$), S^*_{split}

1. Set $\tau_q = \tau_{\min}$.

2. Set unary values of $\mathcal{R}$ to $\rho_i + q_i \tau_q$.

3. Run WeightedBisectionCut (Algorithm 2) on $\mathcal{R}$.

4. If there is a valid cut, calculate the τ such that the difference between the excess and deficit on opposing sides of the cut equals the value of the cut. Set τ_q to this value

and goto step (2).

5. If there is no cut, then $\mathcal{R}$ is valid on $[\tau_q, \tau_c]$ and splits beyond τ_o. We're done.

Step (4) is valid as the value of the split does not change in terms of τ, but the unary terms – the excess and deficits – are a linear function of τ. From this, it is then possible to obtain the pivot τ in which the region $\mathcal{R}$ will be disected by a specified cut. For τ greater than this, the amount of flow entering and exiting on either side of the cut is insufficient to saturate it, and for τ less than this, the edges in the cut must necessary be saturated. At this point, however, it is necessary to rerun the network flow algorithm, as a different cut might split $\mathcal{R}$ for higher τ. If there is no cut, we have our split point. This algorithm is detailed in algorithm 6.

As explained above, this algorithm recursively tries to solve the network flow problem at a given τ. If there is a cut there, it attempts to find the τ above which this cut cannot be saturated. This cut is found at the τ where the difference in level between S^* and T^* equals the value of the cut c, i.e. the τ where

$$[\gamma_{S^*} + \tau q_{S^*} - c - |S^*|\,\beta_{\mathcal{R}}(\tau)] - [\gamma_{T^*} + \tau q_{T^*} - c - |T^*|\,\beta_{\mathcal{R}}(\tau)] = 0. \tag{4.24}$$

The solution to this equation is given in the above algorithm.

4.2.6 Algorithmic Correctness

At this point, our work is finished. The bulk of the main theory was proved by theorem 4.2.3; we here only have to clean up the final work.

Theorem 4.2.6 (Correctness of Algorithm 4)**.** *The output of Algorithm 4 yields the entire solution path over* $\tau \in [\tau_{\min}, \tau_{\max}]$.

Proof. Follows immediately from theorem 4.2.3, as Algorithm 4 ensures that conditions I and II are met. □

The running time of the algorithm is difficult to analyze. It involves solving a network flow problem at the beginning of each segment, thus a clear upper bound is given by

$\mathcal{O}(N \cdot I \cdot B)$, where N is the cost of calling the network flow solver, I upper bounds the number of times the network flow solver has to be called on a given section to find the split point, and B is the number of distinct line segments in the graph. The problem with getting a better bound is that the number of line segments – as lower bounded by the number of crossings of the 0 line – can be exponential in the worst case [Kolmogorov et al., 2007]. However, in practice, we have found that this algorithm is fairly efficient. We discuss these results in the next chapter where we bring all of these algorithms together to complete the algorithm for total variation.

4.3 General Optimization

Finally, given the previous work on optimization for general functions as given in the previous chapters, we give here the analogous theorem for optimization of ($\Re_{\mathrm{B}}$) along the path of varying τ.

Theorem 4.3.1. *Consider* ($\Re_{\mathrm{B}}$)*:*

$$\mathbf{u}^*(\lambda) = \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^n} \|\mathbf{u}-\mathbf{a}\|_2^2 + \lambda \left[\sum_i \xi_i(u_i) + \sum_{i,j} w_{ij}|u_i - u_j| \right], \tag{4.25}$$

with $\lambda > 0$*,* $\mathbf{a} \in \mathbb{R}^n$*,* $w_{ij} \geq 0$*, and* $\xi_i(u_i)$ *is convex piecewise linear. Let* $0 < \lambda_{\min} < \lambda_{\max}$*.*

Then algorithm 4 gives the complete regularization path for $\lambda \in [\lambda_{\min}, \lambda_{\max}]$*. Specifically,*

$$\mathbf{u}^*(\lambda) = \sqrt{\lambda}\mathbf{r}(\boldsymbol{\alpha}^*(\lambda^{-1/2}\ ;\ \mathbf{w}), \lambda^{-1/2}) \tag{4.26}$$

where $\mathbf{r}(\boldsymbol{\alpha}^*(\tau\ ;\ \mathbf{w}), \tau)$ *is calculated using algorithm 4 on the interval* $\left[\lambda_{\max}^{-1/2}, \lambda_{\min}^{-1/2}\right]$*.*

Proof. We begin by rewriting (4.27) as

$$\mathbf{u}^\dagger(\lambda) = \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^n} \left\| \mathbf{u} - \frac{\mathbf{a}}{\sqrt{\lambda}} \right\|_2^2 + \left[\sum_i \xi_i(u_i) + \sum_{i,j} w_{ij}|u_i - u_j| \right], \tag{4.27}$$

However, by theorem 3.6.1, the unary terms are now given by $\mathbf{a}\lambda^{-1/2}$. Using algorithm 4, we can calculate this on $[\lambda_{\min}, \lambda_{\max}]$ by setting $\tau_{\min} = \lambda_{\max}^{-1/2}$ and $\tau_{\max} = \lambda_{\min}^{-1/2}$. This approach finds the optimal estimation of $\mathbf{a}\lambda^{-1/2}$; the transformation

$$\mathbf{u}^*(\lambda) = \sqrt{\lambda} \cdot \mathbf{u}^\dagger(\lambda) \tag{4.28}$$

gives us the desired quantity. The correctness of this approach follows immediately 4.2.6 and □

4.4 Conclusion

In this chapter, we have outlined a new method for completely enumerating the solution path to the problem ($\Re_B$), where the unary terms are general linear functions. This generalizes the classic parametric flow problem described in [Gallo et al., 1989] in which the unary terms are required to be monotonically increasing. In our formulation, this solution is given by the cut at $\beta = 0$. However, it is still necessary to calculate the full solution path, as the cut at $\beta = 0$ is determined largely by the behavior of the rest of the reduction levels. It is not surprising that the solution we propose is much more complicated than the one in [Gallo et al., 1989], as the cut pattern can have quite complicated patterns as a function of τ (or λ in the context of ($\Re_B$)). Furthermore, since we find the levels simultaneously for all reduction levels, this method is applicable to the problem ($\Re_B$) as described previously in theorem 4.3.1.

Chapter 5

TOTAL VARIATION MINIMIZATION

5.1 Structure of the Total Variation Minimization Problem

In this chapter, we use the previous result to develop a method of solving the Total Variation minimization problem. Recall that the problem we are interested in, ($\Re_{\mathrm{TV}}$), is given by

$$u = \underset{u \in \mathcal{F}_2(\Omega)}{\operatorname{argmin}} \ \lambda' \|f - u\|_2^2 + \mathrm{TV}(u) \tag{5.1}$$

$$= \underset{u \in \mathcal{F}_2(\Omega)}{\operatorname{argmin}} \ \lambda' \int_\Omega \left[f(\mathbf{x}) - u(\mathbf{x})\right]^2 \mathrm{d}\mathbf{x} + \int_\Omega \|\nabla\, u(x)\|_2 \,\mathrm{d}x \tag{5.2}$$

where $\Omega \subset \mathbb{R}^d$ is a closed domain of $\mathbb{R}^d$ and $\mathcal{F}(\Omega)$ is the space of L_2-integrable functions on *Omega* with bounded variation as defined in section 1.2.2. To be consistent with the literature on this problem, we call $\|f - u\|_2^2$ the *fidelity* term and $\mathrm{TV}(u)$ the *regularizer* term.

Here, the total variation term is defined as

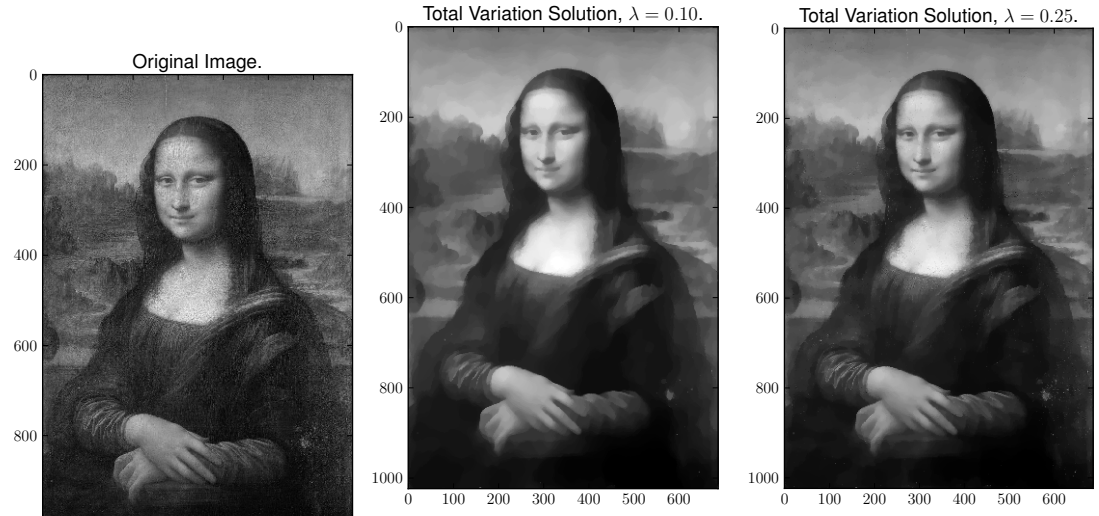
$$\mathrm{TV}(u) = \int_\Omega \|\nabla\, u(\mathbf{x})\|_2 \,\mathrm{d}\mathbf{x} \tag{5.3}$$

To stay consistent with the literature, we put the regularization parameter λ' on the fit term $\|f - u\|_2^2$. Thus for λ' sufficiently small, u is simply the mean of f over Ω; for λ' sufficiently large, the total variation term u is negligible, and $u = f$. The $\mathrm{TV}(u)$ term acts like an $L1$ regularizer on differences and enforces sparsity in the number of distinct values of u. The final result looks like collections of distinct levels; these end up corresponding to the reduction values of the previous chapters. An example of using total variation for Gaussian noise removal is found in Figure (5.1).

This problem has proven quite difficult to work with in practice, even though it is a staple in the image analysis and computer vision community. The problem was first proposed in [Rudin et al., 1992], and the original paper has over 5900 citations[1]. Not surprisingly, approaches to this problem are numerous. Many of these approaches are outlined in the next section.

Our contribution here is to present the first complete and tractable description of the regularization path of the optimal solution to (5.1) over λ'. This is based heavily on the framework laid out in the previous chapters; with those in place, our purpose now is to

[1]Statistic from Google Scholar, August 2013.



(a) Original Image

(b) TV minimized image with $\lambda' = 0.1$.

(c) TV minimized image with $\lambda' = 0.25$.

Figure 5.1: An example of Total Variation Minimization for noise removal on Leonardo da Vinci's Mona Lisa. Different values of the regularization parameter produce different results, with the higher value of λ' smoothing the image less but removing less of the noise.

simply translate the total variation functional into the form of the problem given in theorem 4.3.1.

While we work with a discretized approximation of the function in which we consider it only on a lattice structure $\Omega_L \subseteq \Omega$, this approximation can be made arbitrarily accurate at the expense of computational complexity. In making this approximation, we are in good company, as virtually all of the existing approaches do the same. A further approximation is required to accommodate the total variation term; however this can also be made arbitrarily accurate. We explore these in detail in the next few sections. In fact, our main contribution in this chapter – beyond that which is already laid out elsewhere – is a proper theoretical treatment of the approximation accuracy of using the presented approach to total variation minimization.

The structure of this chapter is as follows. We begin with a discussion of related work motivating our approach. The theory in this chapter deals with translating the problem

above into a form that matches the previous framework given in the earlier chapters. We begin by discussing the geometric translation of the current problem to the general theory of the previous chapters. We present a method of encoding the total variation term TV(u) using cuts on a specific patterned graph defined over a lattice covering the function domain Ω. Once in this form, we can cast the problem in terms of the algorithms in the previous chapters; in particular, the regularization path of (5.1) is a function of the solution path for the algorithm presented in chapter 4. In section 5.4, we present a detailed treatment of the approximation accuracy of our approach on a 2d lattice, bounding the overall convergence rate as a function of the lattice size and general properties of the line. In section 5.6 we present several simple experiments showing the validity of our approach.

5.2 Graph-based Approaches to TV Minimization

The use of graph cuts as an approach to optimizing the total variation problem is fairly recent and has attracted a fair amount of attention, largely since efficient algorithms exist for network flow optimization. Variants of the same approach we present here were proposed independently in Chambolle and Darbon [2009] and Goldfarb and Yin [2009] (see also Darbon and Sigelle [2006,?], Chambolle [2004] and Darbon and Sigelle [2005]). These approaches use the parametric flow algorithm of Gallo et al. [1989], which operates over fixed function points; while it works for general monotonic functions, more work is required to find the exact changepoints which then determine the level sets of the function [Kolmogorov et al., 2007]. Thus these approaches discretize the output levels and solve the flow at each point. In contrast, ours solves the flow exactly.

One of the issues with encoding the problem on a lattice structure is that the approximations introduced can indeed affect the quality of the solution in negative ways. Many of the methods used lead to *metrification* errors in which the translation of the metric onto the graph structure causes non-trivial issues at the local level. The approach we discuss here from Boykov and Kolmogorov [2003] is a more precise and rigorous attempt to get around these issues, even though it may cause non-trivial memory and computational requirements for the problem. For a full treatment of the issues around some of these, see Couprie et al. [2011].

5.2.1 Beyond the Current Formulation

A number of methods have proposed extensions to the total variation problem. Many of these are motivated by issues encountered when these approaches are applied in practice to various domains. Furthermore, as one of the principal uses of total variation in computer vision is to find differentials in the image corresponding to edges, a number of refinements to the problem have been proposed for both practical and algorithmic improvements.

One of the main approaches to this problem, and other related topological minimization problems, is the use of "snakes" [Kass et al., 1988], adaptive splines optimized with an objective that places them along high-gradient contours of the original function. A number of extensions to this model have been proposed, and these seem to be used heavily in practice. A number of formulations for image analysis and segmentations are used in practice.

One of the recent approaches to this problem that has gained significant interest is the use of *continuous* max-flow models in which the flow has a geometric interpretation in $\mathbb{R}^d$ space Appleton and Talbot [2006], Couprie et al. [2011]. These approaches share a lot in common with the active contour approaches above. Also, Couprie et al. [2011] proposes ways of bridging the discrete graph cut world with the continuous versions to combine the computational speed of the discrete method with the accuracy of the continuous methods.

Furthermore, the theory detailed below – with the exception of the approximation bound, which is specific to 2 dimensions – extends naturally to higher dimensions. In particular, Kolmogorov and Boykov [2005] describes how to encode a very general class of Riemannian metrics onto an $\mathbb{R}^d$ lattice. These have been adapted in many cases to create algorithms that distort the topology in response to local properties of the function [Appleton and Talbot, 2006, Chambolle and Darbon, 2009] to compensate for the tendency of TV minimization to "round" corners. However, it also allows the theory to apply much more generally beyond the simple 2d examples here.

5.3 Geometric Approximations and Problem Representations

The main approximation we use is to only examine the functions f and u by their values on a discrete set of points in Ω laid out as the points in an $\mathbb{R}^d$ lattice, with δ_L giving the

separation between axis-aligned neighboring points. Call this set of points Ω_L, defined by

$$\Omega_L = \Omega \cap \left\{ \delta_L \cdot \mathbf{z} : \mathbf{z} \in \mathbb{Z}^d \right\}, \tag{5.4}$$

where $\mathbb{Z}$ is the set of integers. This approximation is relatively easy, and the grid can be made arbitrarily fine in practice. Of course a finer resolution leads to additional computational complexity. In practice, however, this resolution is typically dictated by other factors, especially in image analysis where we observe f in terms of the discrete pixel locations.

Ultimately, each of the locations in Ω_L maps to a node on the graph. Thus, since we have attached a distinctly geometric meaning to the nodes in the graph, we index them by their geometric locations instead of by their indices. Thus we map all locations $\mathbf{x} \in \Omega_L$ on which we are dealing with f and u to one of the values in $\mathcal{V} = \{1, 2, ..., n\}$ used in previous chapters. This trivial mapping is simply for notational convenience.

The total variation problem and our approach work fine in higher dimensions; however, for simplicity and intuition, we present our results and experiments in the context of a function defined in $\mathbb{R}^2$. All our results, however, naturally extend to higher dimensions.

Additionally, in presenting our approach, there are other relevant effects in practice that we defer to the discussed literature on the subject. In particular, edge effects are known to affect the calculation of the TV(u) term near the edges. This has been given a full treatment in Kolmogorov and Boykov [2005] and Bellettini, Caselles, and Novaga [2002]; we refer the interested reader there. Furthermore, as with all regularized optimization problems, choosing the best optimization parameter λ' can be tricky. There are a number of ways to do this in practice, many of which are tailored to their particular domains, and we refer the interested reader to Caselles, Chambolle, and Novaga [2011] for discussions of doing this. Note that our work makes using many of these other results much easier, as the optimal solution becomes immediately available for all λ' once we have found the regularization path.

5.3.1 Expressing the TV(u) *Integral Using Line Lengths*

The second approximation deals with encoding the total variation penalty on the graph. This is much more complicated, and only recently a method was proposed for encoding

an arbitrarily accurate approximation to TV(u) on a graph [Chambolle and Darbon, 2009, Goldfarb and Yin, 2009]. The theoretical seed of this approach is given in Boykov and Kolmogorov [2003], Kolmogorov and Boykov [2005]. The basic idea is to first use the *Coarea* formula from functional analysis, which allows the integral of a derivative over an area to be expressed as the integral over lengths of perimeters of the levelsets of accompanying functions. Then, one uses another theorem, the Crofton formula from integral geometry, to approximate these perimeter calculations as discrete cuts in a graph. We make this connection explicit in the next few sections.

The Co-Area Formula and Hausdorff Measure

The co-area formula [Federer, 1959, Fleming and Rishel, 1960] is a well known result from real analysis. For an L1-integrable Lipshitz function $u : \Omega \mapsto \mathbb{R}$, with $\Omega \subseteq \mathbb{R}^d$, the basic form of the co-area formula states that

$$\int_{\Omega} \|\nabla u(\mathbf{x})\|_2 \, \mathrm{d}\mathbf{x} = \int_{-\infty}^{\infty} \mathcal{H}^{d-1}\big(\Omega \cup u^{-1}(t)\big)\mathrm{d}t \tag{5.5}$$

where $u^{-1}(t) = \big\{\mathbf{x} \in \mathbb{R}^d : u(\mathbf{x}) = t\big\}$ is the contour of the function u at level t; this can be the empty set if no values of u equal t.

$\mathcal{H}^{d-1}(A)$ is the $d-1$ Hausdorff measure of the set A. The d-dimensional Hausdorff measure of A [Federer, 1969], denoted as $\mathcal{H}^d(A)$, is defined as the limit of the d-dimensional Hausdorf-δ measure of the set A, denoted as $\mathcal{H}^d_\delta(A)$. In this case, $\mathcal{H}^d_\delta(A)$ is defined in terms of the minimal sum of the diameters in a countable covering set. Formally,

Definition 5.3.1. *Let*

$$\mathcal{H}^d_\delta(A) = \inf_{\substack{A_1, A_2, \ldots \in \mathcal{A}_\delta \\ A \subseteq \cup_i A_i}} \sum_{i=1}^{\infty} a_d \left[\frac{\mathrm{diam}(A_i)}{2}\right]^d \tag{5.6}$$

where

$$\mathcal{A}_\delta = \Big\{A \subset \mathbb{R}^d : \mathrm{diam}(A) \leq \delta\Big\} \tag{5.7}$$

and a_d is the volume of the unit ball in d-dimensions, equal to

$$a_d = \frac{\pi^{d/2}}{\Gamma(d/2 + 1)}. \tag{5.8}$$

Then the Hausdorff measure of A is equal to

$$\mathcal{H}^d(A) = \lim_{\delta \searrow 0} \mathcal{H}^d_\delta(A). \tag{5.9}$$

The Hausdorff measure generalizes Euclidean distance in the sense that it is proportional to the Lebesgue measure of that set. If A is a line, $\mathcal{H}^1(A)$ measures its length; if A is a region in 2-d space, then $\mathcal{H}^2(A)$ is proportional to the area of A. For more information on this, we refer the reader to [Federer, 1969] or any of a number of books on metric geometry.

For $d = 2$, $\left(u^{-1}(t)\right)$ traces a 1d contour in 2d space, and equation (5.5) simplifies down to

$$\begin{aligned} \mathrm{TV}(u) &= \int_\Omega \|\nabla u(\mathbf{x})\|_2^2 \, \mathrm{d}\mathbf{x} && (5.10)\\ &= \int_{-\infty}^{\infty} \text{length}\left(u^{-1}(\beta)\right) \mathrm{d}\beta. && (5.11) \end{aligned}$$

Using this formula, one can express the total variation as an integral over the lengths of the curves.

Example

A simple example of this theorem is as follows. Consider the following 2d function:

$$f(x,y) = \begin{cases} 1 - \max(|x|,|y|) & |x| \le 1, |y| \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.12}$$

This function looks like a symmetric pyramid with a center at $(0,0)$. For this, the magnitude of the gradient is 1 at every point, so

$$\begin{aligned} \int_\Omega \|\nabla f(\mathbf{x})\|_2 \, \mathrm{d}\mathbf{x} &= \int_{-1}^{1}\int_{-1}^{1} \sqrt{\left(\left(\frac{\partial}{\partial x} f\right)(x,y)\right)^2 + \left(\left(\frac{\partial}{\partial y} f\right)(x,y)\right)^2} \, \mathrm{d}x\mathrm{d}y && (5.13)\\ &= \int_{-1}^{1}\int_{-1}^{1} \sqrt{\mathbf{1}_{\{|x|>|y|\}} + \mathbf{1}_{\{|x|<|y|\}}} \, \mathrm{d}x\mathrm{d}y && (5.14)\\ &= \int_{-1}^{1}\int_{-1}^{1} 1 \, \mathrm{d}x\mathrm{d}y && (5.15)\\ &= 2 \cdot 2 = 4 && (5.16) \end{aligned}$$

In this formulation, define the inverse image of f as

$$f^{-1}(t) = \begin{cases} \{(x,y) : \max|x|,|y| = 1-t\} & t \in [0,1] \\ \emptyset & \text{otherwise} \end{cases} \tag{5.17}$$

For $t \in [0,1]$, this defines a square of side length $2(1-t)$, given by the 2-d ℓ_∞ ball. Otherwise, the inverse image is empty. Thus

$$\text{length}\left(f^{-1}(t)\right) = \begin{cases} 8(1-t) & t \in [0,1] \\ 0 & \text{otherwise} \end{cases} \tag{5.18}$$

and

$$\int_{-\infty}^{\infty} \text{length}\left(f^{-1}(t)\right) = \int_0^1 8(1-t) = 4 \tag{5.19}$$

exactly as expected.

5.3.2 Approximation of Line Lengths using Graph Cuts

Informally, to express this as a formula representable on a graph, we use the technique of [Boykov and Kolmogorov, 2003] to reformulate the right hand side of equation (5.11) in terms of a measure over the number of lines crossed when the lattice has a consistent connectivity pattern. In our case, the lines crossed will instead be the edges on the graph structure cut at a particular reduction level. In properly setting the weights, the length of the cut will be expressed as the sum of the weights of the edges cut. It turns out, then, that the reduction levels naturally include the integral in the Coarea formula, giving us a natural version of the total variation. In this way, we end up encoding the total variation term directly into the cost of the graph, and our framework provides the immediate translation into a global optimization of (5.1).

The primary theoretical tool allowing us to do this transformation is a formula from integral geometry called the Crofton or Cauchy-Crofton formula. Intuitively, it allows us to find the length of a curve in terms of lines crossed in a homogeneous random linefield. Define a line by a normal vector emerging from the origin at angle θ, with a distance to line of ρ. The primary form of the Cauchy-Crofton formula in 2 dimensions [Do Carmo and

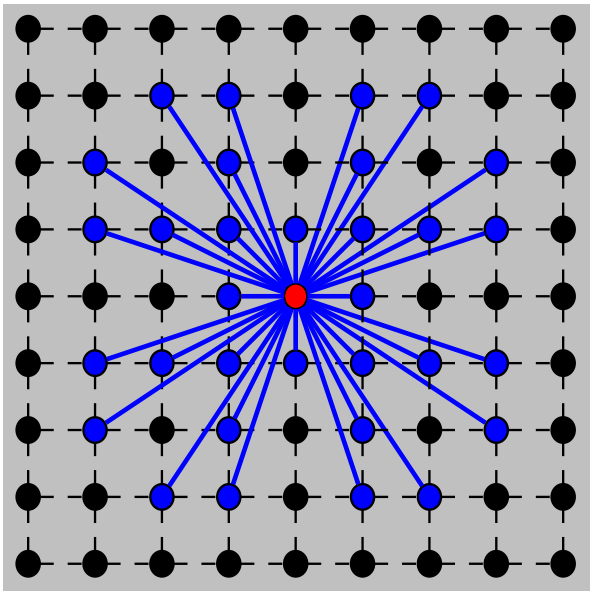


Figure 5.2: A pattern of connectivity for a single node on a 2-d grid; this pattern is repeated for every node in the lattice Ω_L. The families of lines we work with are formed by all the lines oriented at a given angle.

Do Carmo, 1976] is then

$$\text{length}\,(C) = \frac{1}{2}\int n_c \mathrm{d}\mathcal{L} \tag{5.20}$$

$$= \frac{1}{2}\int n_c(\rho,\theta)\mathrm{d}\rho\mathrm{d}\theta \tag{5.21}$$

where $n_c(\rho,\theta)$ is the number of times that a line defined by ρ and θ crosses the curve C (If C is a closed curve, then $n_c(\rho,\theta)$ is almost surely a multiple of 2). Lines are defined by ρ, their distance to the origin, and θ, the angle of incidence of the shortest-distance normal vector off the x-axis. The measure over the lines, $\mathcal{L}$, is the standard Lebesgue measure for lines. With this measure, the density of lines at a given angle is homogeneous, and the density across angles is also homogeneous. It has the unique property that it is location invariant; in particular, the reference point for the origin is arbitrary.

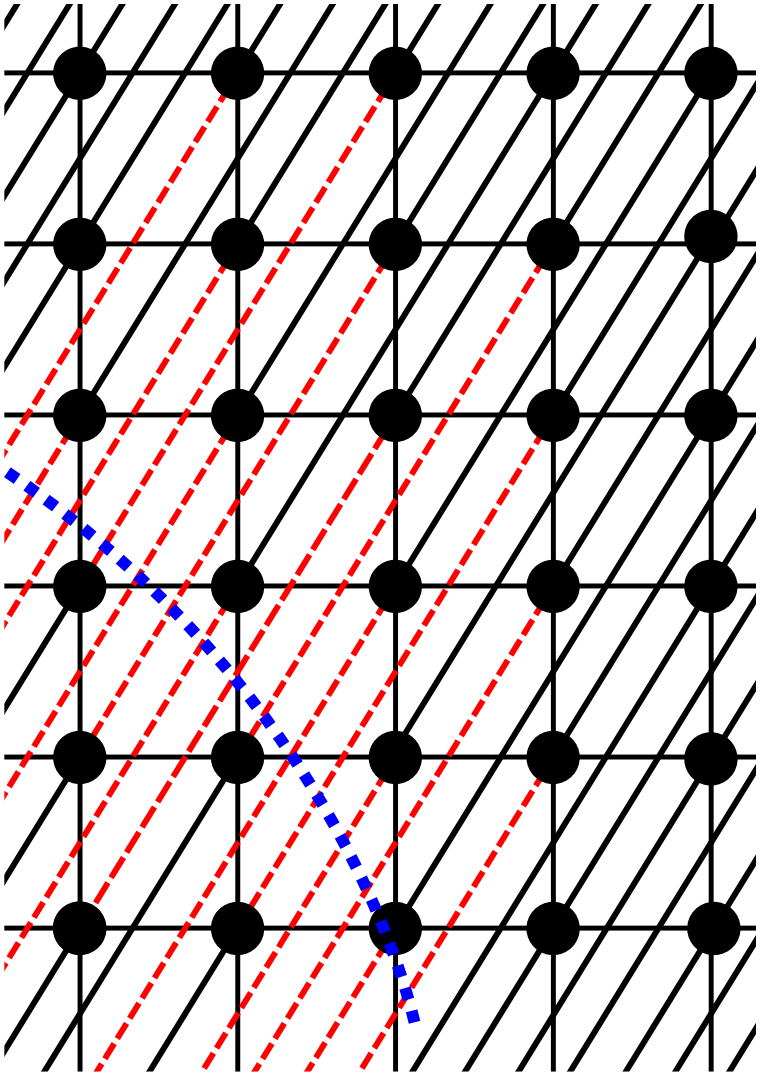


Figure 5.3: One of the families of lines formed by the connectivity pattern of Figure (5.2). With this family of lines, the distance the blue dotted curve travels in the distance normal to the lines can be approximately determined from counting the number of line segments it crosses; these segments lines are shown as dashed red lines.

The construction of a collection of edges on the graph, and accompanying weights, follows the work of [Boykov and Kolmogorov, 2003]. To keep matters simple, we deal with a regular pattern of connectivity on Ω_L. In particular, each node is connected to nearby neighbors in a regular pattern.

An example of such a connectivity pattern is shown in Figure (5.2). This pattern is repeated for all nodes in the lattice Ω_L. As such, each of the lines in this connectivity pattern that have distinct angles form a family of line segments, as shown in Figure (5.3). There, the approximate length of the blue dotted curve in the direction normal to the line family can be determined from the number of line segments crossed. The Cauchy-Crofton formula formalizes this method of determining the length of curves. Our approach is to approximate the exact formula by discretizing the integral of (5.21) with weighted sums over different sets of lines. In the limit, when there are infinite such families of lines covering all angles, the lines in each family become arbitrarily close, and the segments are arbitrarily small, our approximation to (5.21) becomes exact.

5.4 Theory

We describe in detail here the method given in [Boykov and Kolmogorov, 2003] to encode length$(u^{-1}(t))$ as part of the cost of a network flow optimization. While their work in that paper, and subsequent work in [Kolmogorov and Boykov, 2005], extend this construction to the case of general Riemannian metrics in higher dimensions, our purpose is to connect the problem to the algorithms and results of the previous chapters so we restrict ourselves to the simple 2d case.

The two free variables in our discussion are the lattice grid size δ_L and the radius of the connectivity pattern R. First, define a set of angles Φ_S in terms of the grid of points Ω_L and radius R in the first quadrant

$$\Phi_S = \left\{ \tan^{-1}(y/x) : x, y \in \mathbb{R}^+, x > 0, \sqrt{x^2 + y^2} \leq \frac{R}{\delta_L} \right\} \tag{5.22}$$

and (to get around the peculiarities of the $\tan^{-1}$ function) the full collection of angles is given by

$$\Phi = \Phi_S \cup \left\{ \frac{\pi}{2} + \phi : \phi \in \Phi_S \right\}. \tag{5.23}$$

Finally, let $\phi_1, \phi_2, ..., \phi_K \in \Phi$ be the ordered enumeration of the values in Φ, so

$$0 = \phi_1 < \phi_2 < \phi_3 < \cdots < \phi_K < \pi \tag{5.24}$$

As each line k is connected to the another node in which it emerges at an angle $\pi+\phi_k$ there, we only include lines with angles less than π. With this construction, each of the angles ϕ_k forms a distinct set of lines.

Now, let

$$\Delta\phi_k = \frac{\phi_{k+1}+\phi_k}{2} - \frac{\phi_k+\phi_{k-1}}{2} \tag{5.25}$$

be the solid angle defined by the measure of angles that are closest to ϕ_k.

The challenge now is to use these distinct families of lines to approximate the integral in (5.21). Each k thus indexes a set of line segments of all the line segments with slope $\tan(\phi_k)$. Recall that one such family is detailed in Figure (5.3). Now, call this collection $\mathcal{L}_k$. Formally, we can define $\mathcal{L}_k$ as

$$\begin{aligned}\mathcal{L}_k = \Big\{\text{Line segments } \ell = (\mathbf{x}_1, \mathbf{x}_2) \quad : \quad & \angle(\ell) = \phi_k, && (5.26)\\ & \mathbf{x}_1, \mathbf{x}_2 \in \Omega_L, && (5.27)\\ & \|\mathbf{x}_1 - \mathbf{x}_2\|_2 \leq R, && (5.28)\\ & \nexists\, t \in [0,1] \text{ such that } (t\mathbf{x}_1 + (1-t)\mathbf{x}_2) \in \Omega_L \Big\} && (5.29)\end{aligned}$$

where $\angle(\ell)$ is the angle between the line ℓ and the positive x-axis. The criteria in (5.29) ensures that the line segments are the shortest possible ones at that angle. All line segments for a given family have a constant offset $\mathbf{x}_2 - \mathbf{x}_1$, call this $\mathbf{e}_k$.

Let $\Delta\rho_k$ denote the distance between neighboring parallel line segments in $\mathcal{L}_k$; this is given as

$$\Delta\rho_k = \frac{\delta_L^2}{\|\mathbf{e}_k\|_2}. \tag{5.30}$$

With these tools in place, (5.21) can be approximated as

$$\begin{aligned}\text{length}\,(C) &= \frac{1}{2}\int_0^{\pi}\int_{-\infty}^{\infty} n_C(\rho,\phi)\mathrm{d}\rho\mathrm{d}\phi && (5.31)\\ &\simeq \frac{1}{2}\sum_{k=1}^{K}\left[\sum_{i=-\infty}^{\infty} n_{k,C}(i\cdot\Delta\rho_k)\Delta\rho_k\right]\Delta\phi_k && (5.32)\end{aligned}$$

where $n_{k,C}(d)$ counts the number of line segments in $\mathcal{L}_k$ crossed by the curve C at a normal distance d from the origin:

$$n'_{k,C}(d) = \#\left\{\ell \in \mathcal{L}_k : \ell \cap C \neq \emptyset, \text{dist}(\ell, \mathbf{0}) = d\right\}. \tag{5.33}$$

This can be simplified down to

$$\text{length}\,(C) \simeq \frac{1}{2}\sum_{k=1}^{K}\Delta\rho_k \cdot \#\left\{\ell \in \mathcal{L}_k : \ell \cap C \neq \emptyset\right\}\Delta\phi_k \tag{5.34}$$

$$= \sum_{k=1}^{K} n_{k,C}\frac{\delta_L^2 \cdot \Delta\phi_k}{2\left\|\mathbf{e}_k\right\|_2} \tag{5.35}$$

where $n'_{k,C}$ simply counts the total number of line segments crossed by C in $\mathcal{L}_k$:

$$n'_{k,C} = \#\left\{\ell \in \mathcal{L}_k : \ell \cap C \neq \emptyset\right\} \tag{5.36}$$

Let

$$w_k = \frac{\delta_L^2 \cdot \Delta\phi_k}{2\left\|\mathbf{e}_k\right\|_2} \tag{5.37}$$

be the contribution of each line in $\mathcal{L}$ towards length(C). Thus our final form for the approximation of the line lengths on the graph is

$$\text{graphlength}(C) = \sum_{k=1}^{K} w_k n'_{k,C} \tag{5.38}$$

$$= \sum_{\mathbf{x}\in\Omega_L}\sum_{k=1}^{K} w_k \mathbf{1}_{\{\mathbf{x}+\mathbf{e}_k\in\Omega_L\}}\mathbb{I}\{C \text{ crosses the line } (\mathbf{x}, \mathbf{x}+\mathbf{e}_k)\} \tag{5.39}$$

where in (5.39) we sum over all the edges in the graph that cross the line. In the final algorithm, the edges in $\mathcal{L}_k$ crossed by the curve C are the ones cut in the optimal partition.

5.4.1 Formation of the Total Variation Problem

Given the equations above, we have that

$$TV(u) = \int_{-\infty}^{\infty} \text{length}(u^{-1}(\beta))\mathrm{d}\beta \tag{5.40}$$

$$\simeq \int_{-\infty}^{\infty} \text{graphlength}(u^{-1}(\beta))\mathrm{d}\beta \tag{5.41}$$

and that graphlength$(u^{-1}(\beta))$ is determined by the cost of the cut on the graph with edges formed by $\mathcal{L}_k$, Thus

$$\int_{-\infty}^{\infty} \text{graphlength}(u^{-1}(\beta))\mathrm{d}\beta = \int_{-\infty}^{\infty}\sum_{\mathbf{x}\in\Omega_L}\sum_{k} w_k \mathbf{1}_{\{\mathbf{x}+\mathbf{e}\in\Omega_L\}}\mathbb{I}\left\{\mathbf{1}_{u_{\mathbf{x}}>\beta} \neq \mathbf{1}_{u_{\mathbf{x}+\mathbf{e}_k}\leq\beta}\right\}\mathrm{d}\beta \tag{5.42}$$

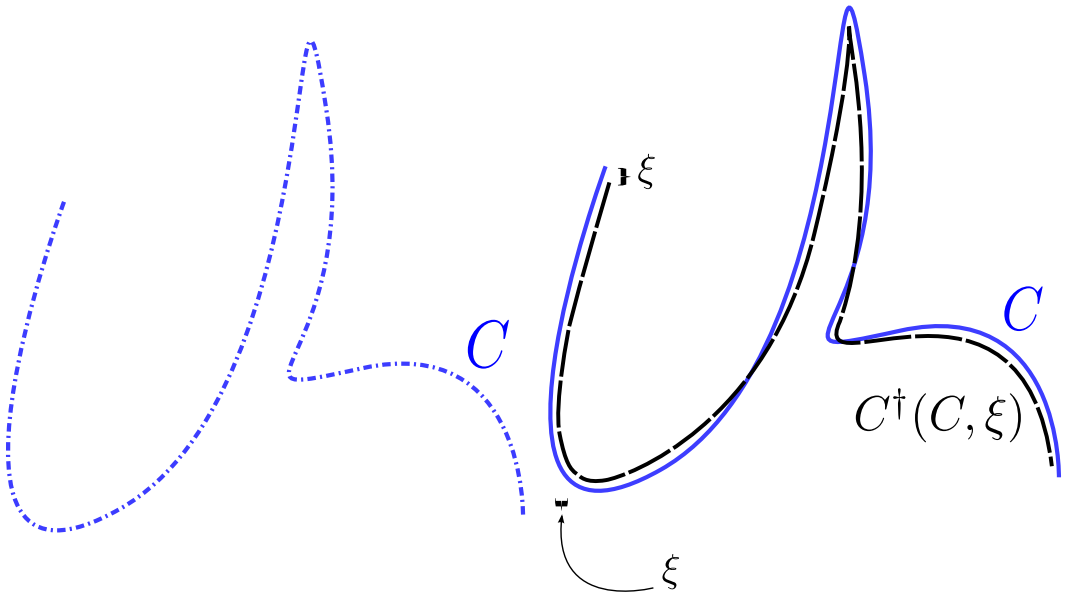


Figure 5.4: An example curve C (a) and a minimal covering curve $C^\dagger(C,\xi)$.

where (5.42) encodes the cost of the cut by indicating whether the contour $u^{-1}(\beta)$ passed between nodes $u_{\mathbf{x}_1}$ and $u_{\mathbf{x}_1+\mu_k}$. This formulation, however, immediately satisfies the conditions of theorem 2.6.1, allowing it to be calculated within our framework. Furthermore, corollary 4.3.1 gives us the full regularization path for this function through Algorithm 4.

5.5 Approximation Accuracy

It remains to quantify exactly the quality of the approximation to the true total variation. It has been stated in Boykov and Kolmogorov [2003] that the above approximation converges pointwise to the true value as $[\max_k \|\mathbf{e}_k\|_2]$, $[\max_k \Delta\rho_k]$, and $[\max_k \Delta\phi_k]$ go to zero, given that the curve is continuous and differentiable. However, the proof is omitted there, and no convergence rates are given. Here, we seek to improve on the work by proving convergence rates of the approximation to the true value. This result is novel.

5.5.1 A Complexity Measure for C

For part of the proof below, it is necessary to control the complexity of the curve. The definition of curve complexity we present below captures the inherent difficulty in approximating the exact $n_C(\rho, \phi)$, which exactly counts the number of times C crosses the line defined by ρ and ϕ, with $n'_{k,C}(\rho)$, which approximates this by counting the number of length-$\|\mathbf{e}_k\|_2$ line segments in $\mathcal{L}_k$ crossed by C. The latter inherently undercounts. We bound this by constructing a minimal covering curve that is at most δ away from the original curve.

We can bound this undercounting by looking at minimal curves that cover the original curve at a distance at most η, which we set to $\|\mathbf{e}_k\|_2$ later, away from C. The length of these curves – or unions of curves, rather – provide a lower bound for the total distance as we show below. Two such covering curves are shown in Figure (5.4) for different η. As $\eta \to 0$, this covering curve more closely matches C.

Formally,

Definition 5.5.1 (Curve Approximability Complexity)**.** *Let C be a curve in Ω. Now, given ξ, we wish to define a set of curves that live within ξ of C:*

$$\mathscr{C}^\dagger(C, \xi) = \left\{ \textit{Connected curves } C' : \ \forall \mathbf{x} \in C, \exists \mathbf{x}' \in C' \textit{ such that } \left\|\mathbf{x} - \mathbf{x}'\right\|_2 \leq \xi \right\}. \quad (5.43)$$

Now, we can take the minimum:

$$C^\dagger(C, \xi) = \underset{C' \in \mathcal{C}^\dagger}{\operatorname{argmin}} \ \mathcal{H}^1(C') \quad (5.44)$$

where $\mathcal{H}^1(C')$ is the Hausdorff measure of C' defined above.

Now, define

$$\mathcal{C}_\xi(C) = \frac{\mathcal{H}^1(C) - \mathcal{H}^1(C^\dagger(C, \xi))}{\xi} = \frac{\mathcal{H}^1(C) - \min_{C' \in \mathcal{C}^\dagger} \mathcal{H}^1(C')}{\xi} \quad (5.45)$$

and

$$\mathcal{C}_\infty(C) = \lim_{\xi \searrow 0} \mathcal{C}_\xi(C) = \lim_{\xi \searrow 0} \frac{\mathcal{H}^1(C) - \min_{C' \in \mathcal{C}^\dagger} \mathcal{H}^1(C')}{\xi} \quad (5.46)$$

as the finite and limiting Curve Complexities of C.

Now, it is easy to show that for some simple curves, $\mathcal{C}_\infty(C)$ is easy to find. For straight lines, it is equal to 2, as the only places undercounted are the ends. For a perfect circle, it

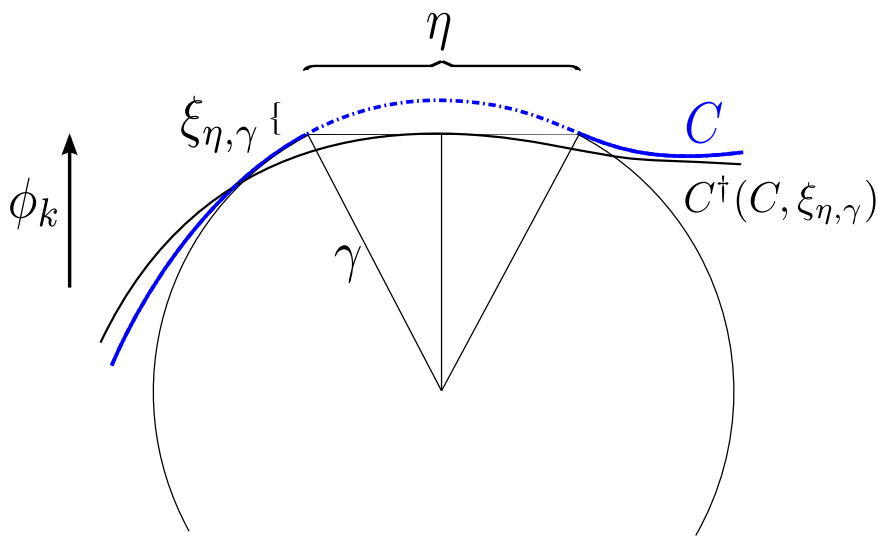


Figure 5.5: The diagram explaining lemma 5.5.2.

is equal to 2π – the length of the covering curve $C^\dagger(C,\xi)$ for C equal to a circle of radius r is simply $2\pi(r-\xi)$.

First, to aid in some of the bounds of the theorem below, we define a general version of the line collection $\mathcal{L}_k$ to allow for line segments at the same angle as those in $\mathcal{L}_k$, but spaced off of the ones dictated by the grid points on Ω_L. Specifically, let $\mathcal{L}_k^\dagger$ be defined as

$$\mathcal{L}_k^\dagger = \left\{(\mathbf{x}_1,\mathbf{x}_2) : (\mathbf{x}_1 + t\mathbf{e}_k^\perp, \mathbf{x}_2 + t\mathbf{e}_k^\perp) \in \mathcal{L}_k \text{ for some } t \in [0,\Delta\rho_k]\right\} \tag{5.47}$$

where $\mathbf{e}_k^\perp$ is the unit vector perpendicular to $\mathbf{e}$. Based on this, let $n_{k,C}^\dagger(\rho)$ be a continuous version of $n'_{k,C}(\rho)$ defined at all ρ:

$$n_{k,C}^\dagger(\rho) = \#\left\{\ell \in \mathcal{L}_k^\dagger : \ell \cap C \neq \emptyset, \text{dist}(\ell,\mathbf{0}) = \rho\right\} \tag{5.48}$$

Note that for $\rho = i\cdot\Delta\rho_k$ for $i \in \mathcal{Z}$, $n_{k,C}^\dagger(\rho) = n'_{k,C}(\rho)$.

In light of this, we can prove the following:

Lemma 5.5.2. *Let C have minimum curvature γ^{-1} in the sense that any circle of radius $\gamma-\varepsilon$ touches C at at most one location for all $0<\varepsilon<\gamma$, unless C crosses the interior of the circle. Furthermore, let $\eta = \|\mathbf{e}_k\|_2$. Then for all ρ and $\eta<\gamma$,*

$$\int_{-\infty}^{\infty} n_{C^\dagger(C,\xi_{\eta,\gamma})}(\rho,\phi_k)\mathrm{d}\rho \leq \int_{-\infty}^{\infty} n'_{k,C}(\rho)\mathrm{d}\rho \leq \int_{-\infty}^{\infty} n_C(\rho,\phi_k)\mathrm{d}\rho, \tag{5.49}$$

where

$$\xi_{\eta,\gamma} = \frac{\eta}{2}\tan\left[\frac{1}{2}\sin^{-1}\left(\frac{\eta/2}{\gamma}\right)\right]. \tag{5.50}$$

Proof. The proof of the upper bound is trivial, as $n'_{k,C}(\rho)$ inherently undercounts. To prove the lower bound, we consider the places where $n'_{k,C}(\rho)$ may undercount. Since C has minimum curvature γ^{-1}, this can only occur at locations at the upper chord of circles as shown in Figure (5.5) with chord length η. Now, from this, it is easy to see that removing the section of C above this chord from the count of the total line length lower bounds $\int n'_{k,C}(\rho)\mathrm{d}\rho$. However, this is in turn lower bounded by the same count but of the curve $C^{\dagger}(C,\xi_{\eta,\gamma})$, since it uniformly minimizes $\int n_C(\rho,\phi_k)\mathrm{d}\rho$ among curves that are within $\xi_{\eta,\gamma}$ of C, where $\xi_{\eta,\gamma}$ is the distance between that chord and the top of the curve C. More formally,

$$\int_{-\infty}^{\infty} n'_{k,C}(\rho)\mathrm{d}\rho \geq \int_{-\infty}^{\infty} n_{k,C\setminus\{\mathbf{s}\,:\,\mathbf{s}\text{ is in chordal section}\}}(\rho,\phi_k)\mathrm{d}\rho \geq \int_{-\infty}^{\infty} n_{C^{\dagger}(C,\xi_{\eta,\gamma})}(\rho,\phi_k)\mathrm{d}\rho, \tag{5.51}$$

which in turn proves the first result.

The second result follows immediately from noting that

$$\sum_{k=1}^{K}\Delta\phi_k\int_{-\infty}^{\infty} n_{C^{\dagger}}(\rho,\phi) \leq (1+\max_k \Delta\phi_k^2)\int_0^{\pi}\int_{-\infty}^{\infty} n_{C^{\dagger}}(\rho,\phi) \tag{5.52}$$

□

Balancing R and δ_L

The second building block for our theory of convergence rates is a way to optimally balance the convergence rate to the solution between R and δ_L. For fixed δ_L, increasing R gives higher resolution in the directional information, but provides less accuracy in resolving more complicated curve shapes. On the other hand, simply decreasing δ_L gives more accuracy, but does not provide additional correction to the error incurred in estimating the direction of the gradient. Thus it is needed to understand the tradeoff between these two sources of error.

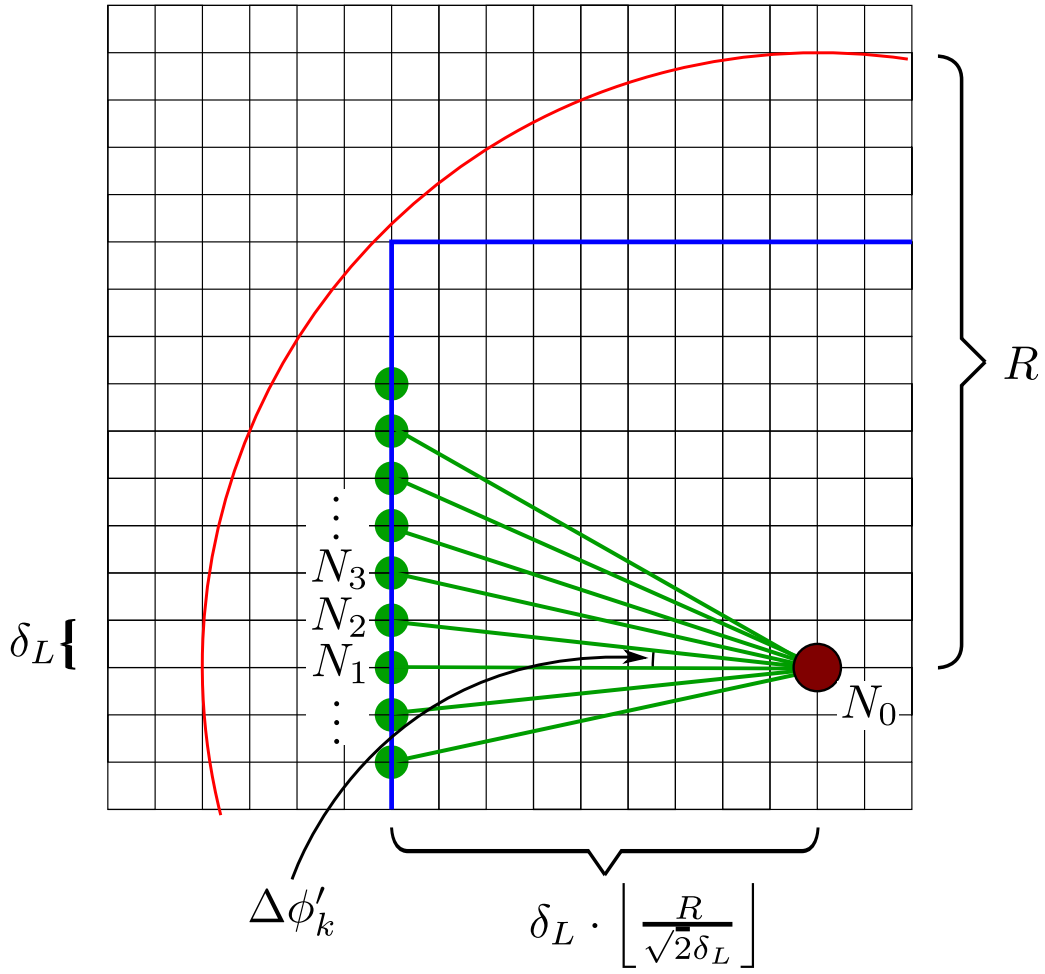


Figure 5.6: The construction used in lemma 5.5.3 to bound the maximum difference in angle between two line families in terms of δ_L and R.

Lemma 5.5.3. *Let Ω_L be a lattice with distance between grid nodes δ_L. Fix R, and let the connectivity pattern be given by $\mathcal{L}_k$ above. Then*

$$\max_k |\Delta\phi_k| \leq \left\lfloor \frac{R}{\sqrt{2}\delta_L} \right\rfloor^{-1} \tag{5.53}$$

Proof. First, we construct a simple bound using a geometric construction of shown in Figure (5.6). Consider a root node N_0, used for measuring the angles in the approximation, and let nodes $N_1, N_2, ...$, lie on the largest square pattern on the grid Ω_L that can be enscribed in a circle of radius R around the root node as shown in in Figure (5.6). Now, we know

that all of these nodes form valid end points for the connectivity pattern; thus there exists a line to each of these points or to one closer to N_0 but having the same angle.

On the square, it can be shown easily that the largest difference in angle, $\Delta\phi'_k$, is given by the angle between N_1 and N_2. However, as all these nodes are part of the connectivity pattern formed by all nodes in the circle as well, we know that this upper bounds $\Delta\phi_k$ for all k. This immediately gives us

$$\Delta\phi_k \leq \Delta\phi'_k = \tan^{-1}\left(\frac{\delta_L}{\delta_L \cdot \left\lfloor \frac{R}{\sqrt{2}\delta_L} \right\rfloor}\right) = \tan^{-1}\left(\left\lfloor \frac{R}{\sqrt{2}\delta_L} \right\rfloor^{-1}\right) \leq \left\lfloor \frac{R}{\sqrt{2}\delta_L} \right\rfloor^{-1}, \tag{5.54}$$

proving the lemma. □

Convergence Rate of Approximation

We here present bounds on the convergence rate of the approximation of the graph-encoded length measurement to the true value of the curve length. In this theory, we effectively control the radius of the connectivity kernel as the grid resolution goes down. In this particular case, the bound is a function of the length of the curve, but it goes to zero as the grid size shrinks. This effectively proves that we can indeed use the above approximation to $\mathrm{TV}(u)$.

Theorem 5.5.4. *Let δ_L define a connectivity pattern on Ω_L as defined by $\mathcal{L}_k$, and let ϕ_k, $\Delta\phi_k$, $\Delta\rho_k$, and w_k be defined above. Let C be a curve in Ω with maximum curvature γ^{-1} and $\xi = \xi_{\eta,\gamma}$ given by* (5.54)*. Let $R = Z\delta_L^{2/3}$ for some constant Z. Then*

$$|\mathrm{length}(C) - \mathrm{graphlength}(C)| \leq C\delta_L^{2/3}(\mathrm{length}(C) + \mathcal{C}_\xi(C)) + \mathcal{O}(\delta_L), \tag{5.55}$$

where C is a constant that depends only on Z.

Proof. From the above equations, recall that

$$\mathrm{length}\,(C) = \frac{1}{2}\int_0^\pi \int_{-\infty}^{\infty} n_C(\rho,\phi)\mathrm{d}\rho\mathrm{d}\phi \tag{5.56}$$

$$\mathrm{graphlength}(C) = \frac{1}{2}\sum_{k=1}^{K}\left[\sum_{i=-\infty}^{\infty} n'_{k,C}(i\cdot\Delta\rho_k)\Delta\rho_k\right]\Delta\phi_k \tag{5.57}$$

There are three approximations here, which we treat independently: (1) The discretization of the integral over ρ, (2) the discretization of the integral over the angle ϕ, and (3) the approximation of $n_c(\rho, \phi)$, the count of line crossings, by $n'_{k,C}$, the number of line segments in $\mathcal{L}_k$ crossed.

Now, we can bound the original term with a union bound:

$$|\text{length}(C) - \text{graphlength}(C)| \tag{5.58}$$

$$= \frac{1}{2}\left|\int_0^\pi \int_{-\infty}^{\infty} n_C(\rho,\phi)\mathrm{d}\rho\mathrm{d}\phi - \sum_{k=1}^{K}\left[\sum_{i=-\infty}^{\infty} n'_{k,C}(i\cdot\Delta\rho_k)\Delta\rho_k\right]\Delta\phi_k\right| \tag{5.59}$$

$$(\star 1) \qquad \leq \frac{1}{2}\left|\int_0^\pi \int_{-\infty}^{\infty} n_C(\rho,\phi)\mathrm{d}\rho\mathrm{d}\phi - \sum_{k=1}^{K}\int_{-\infty}^{\infty} n_C(\rho,\phi_k)\Delta\phi_k\mathrm{d}\rho\right| \tag{5.60}$$

$$(\star 2) \qquad + \frac{1}{2}\left|\sum_{k=1}^{K}\int_{-\infty}^{\infty} n_C(\rho,\phi_k)\Delta\phi_k\mathrm{d}\rho - \sum_{k=1}^{K}\int_{-\infty}^{\infty} n^\dagger_{k,C}(\rho)\Delta\phi_k\mathrm{d}\rho\right| \tag{5.61}$$

$$(\star 3) \qquad + \frac{1}{2}\left|\sum_{k=1}^{K}\int_{-\infty}^{\infty} n^\dagger_{k,C}(\rho)\Delta\phi_k\mathrm{d}\rho - \sum_{k=1}^{K}\left[\sum_{i=-\infty}^{\infty} n^\dagger_{k,C}(i\cdot\Delta\rho_k)\Delta\rho_k\right]\Delta\phi_k\right| \tag{5.62}$$

We now bound each of these separately to get the final result.

$$(\star 1) = \frac{1}{2}\left|\int_0^\pi \int_{-\infty}^{\infty} n_C(\rho,\phi)\mathrm{d}\rho\mathrm{d}\phi - \sum_{k=1}^{K}\int_{-\infty}^{\infty} n_C(\rho,\phi_k)\Delta\phi_k\mathrm{d}\rho\right| \tag{5.63}$$

$$= \frac{1}{2}\left|\int_0^\pi \left[\int_{-\infty}^{\infty} n_C(\rho,\phi)\mathrm{d}\rho\right]\mathrm{d}\phi - \sum_{k=1}^{K}\left[\int_{-\infty}^{\infty} n_C(\rho,\phi_k)\mathrm{d}\rho\right]\Delta\phi_k\right| \tag{5.64}$$

$$\leq \frac{1}{2}\left|\sum_{k=1}^{K}\Delta\phi_k \sup_{\phi\in\Phi_k}\left|\int_{-\infty}^{\infty} n_C(\rho,\phi)\mathrm{d}\rho - \int_{-\infty}^{\infty} n_C(\rho,\phi_k)\mathrm{d}\rho\right|\right| \tag{5.65}$$

where Φ_k refers to the range of ϕ considered by ϕ_k, i.e.

$$\Phi_k = \left[\frac{\phi_{k-1}+\phi_k}{2}, \frac{\phi_k+\phi_{k+1}}{2}\right]. \tag{5.66}$$

Continuing,

$$(\star 1) \leq \frac{1}{2}\left|\sum_{k=1}^{K}\Delta\phi_k\left\{\sup_{\phi\in\Phi_k}\left[\sup_{\substack{\text{curve } C'\subset\Omega \\ \text{length}(C')=\text{length}(C)}}\left|\int_{-\infty}^{\infty} n_{C'}(\rho,\phi)\mathrm{d}\rho - \int_{-\infty}^{\infty} n_{C'}(\rho,\phi_k)\mathrm{d}\rho\right|\right]\right\}\right| \tag{5.67}$$

Now,

$$\sup_{\phi\in\Phi_k}\left[\sup_{\substack{\text{curve } C'\subset\Omega\\ \text{length}(C')=\text{length}(C)}}\left|\int_{-\infty}^{\infty} n_{C'}(\rho,\phi)\mathrm{d}\rho-\int_{-\infty}^{\infty} n_{C'}(\rho,\phi_k)\mathrm{d}\rho\right|\right] \tag{5.68}$$

$$= \left|\text{length}(C)\cos\left(\Delta\rho_k\right)-\text{length}(C)\right| \tag{5.69}$$

$$\leq \left|\text{length}(C)\left(1-\Delta\rho_k^2\right)-\text{length}(C)\right| \tag{5.70}$$

$$= \text{length}(C)\Delta\rho_k^2 \tag{5.71}$$

where steps (5.69)-(5.70) were found by noting that the curve C' that maximized the measurement difference between the two terms was a line perfectly aligned with ϕ_k; the supremum difference was found by the angle in Φ_k maximally

from ϕ. Thus we have that

$$(\star 1) \leq \frac{1}{2}\sum_{k=1}^{K}\Delta\phi_k\,\text{length}(C)\Delta\phi_k^2 \tag{5.72}$$

$$\leq \frac{\pi}{2}\,\text{length}(C)\left[\sup_k \Delta\phi_k^2\right] \tag{5.73}$$

as $\sum_k \Delta\phi_k = \pi$.

Now, we focus our attention to term ($\star 2$). Here, the error comes from possibly under counting the crossings by the fact that the counts in $n^{\dagger}_{k,C}(\rho,\phi_k)$ are determined by segments of length $\|\mathbf{e}_k\|_2$, and each segment can contribute at most 1 count, even if the line crosses

it multiple times. This requires use of the complexity measure given in definition 5.5.1.

$$(\star 2) = \frac{1}{2}\left|\sum_{k=1}^{K}\int_{-\infty}^{\infty} n_C(\rho,\phi_k)\Delta\phi_k \mathrm{d}\rho - \sum_{k=1}^{K}\int_{-\infty}^{\infty} n'_{k,C}(\rho,\phi_k)\Delta\phi_k \mathrm{d}\rho\right| \tag{5.74}$$

$$\leq \frac{1}{2}\left|\sum_{k=1}^{K}\Delta\phi_k\int_{-\infty}^{\infty}\left[n_C(\rho,\phi_k) - n^{\dagger}_{k,C}(\rho)\right]\mathrm{d}\rho\right| \tag{5.75}$$

$$\leq \frac{1}{2}\left|\sum_{k=1}^{K}\Delta\phi_k\int_{-\infty}^{\infty}\left[n_C(\rho,\phi_k) - n_{C^{\dagger}(C,\xi)}(\rho,\phi_k)\right]\mathrm{d}\rho\right| \tag{5.76}$$

$$\leq \frac{1}{2}\left|\int_{0}^{\pi}\int_{-\infty}^{\infty}\left[n_C(\rho,\phi_k) - n_{C^{\dagger}(C,\xi)}(\rho,\phi_k)\right]\mathrm{d}\rho\right| + \frac{1}{2}\left[\max_k \Delta\phi_k^2\right]\text{length}\, C^{\dagger}(C,\xi) \tag{5.77}$$

$$\leq \frac{1}{2}\left[R\cdot\mathcal{C}_{\xi}(C) + \left(\max_k \Delta\phi_k^2\right)\text{length}\left(C^{\dagger}(C,\xi)\right)\right] \tag{5.78}$$

$$\leq \frac{1}{2}\left[R\cdot\mathcal{C}_{\xi}(C) + \left(\max_k \Delta\phi_k^2\right)\text{length}\,(C)\right] \tag{5.79}$$

$$\tag{5.80}$$

where step (5.76) follows from the lemma 5.5.2 and (5.77) follows using the same argument given in (5.69)-(5.70) above. (5.78) follows as R upper bounds $\|\mathbf{e}_k\|_2$ in lemma 5.5.2. This gives us our bound for $(\star 2)$.

The easiest term to bound is $(\star 3)$, which can be bound in similar fashion to the first, as the maximum error incurred on it equal to the length of the curve times $\Delta\rho_k$. This is a very generous bound, as in reality it will only miss certain inflexion points. However, $\Delta\rho_k$ decreases at a faster rate than some of the other terms, so this is acceptable.

Putting this together gives us

$$|\text{length}(C) - \text{graphlength}(C)| \leq \left[\max_k \Delta\rho_k + 2\max_k \Delta\phi_k^2\right]\text{length}(C) + R\cdot\mathcal{C}_{\xi}(C) \tag{5.81}$$

$$= \left[\max_k \frac{\delta_L^2}{\|\mathbf{e}_k\|_2} + 2\max_k \Delta\phi_k^2\right]\text{length}(C) + R\cdot\mathcal{C}_{\xi}(C) \tag{5.82}$$

$$\leq \left[\delta_L + 2\cdot\left\lfloor\frac{R}{\sqrt{2}\delta_L}\right\rfloor^{-2}\right]\text{length}(C) + R\cdot\mathcal{C}_{\xi}(C) \tag{5.83}$$

where the last step follows from lemma 5.5.3. Letting $R = Z\delta_L^{\frac{2}{3}}$, where Z is a constant, we

have that, for sufficiently small δ_L,

$$2 \cdot \left\lfloor \frac{R}{\sqrt{2}\delta_L} \right\rfloor^{-2} = 2 \cdot \left\lfloor \frac{Z\delta_L^{2/3}}{\sqrt{2}\delta_L} \right\rfloor^{-2} \tag{5.84}$$

$$\leq \left[\frac{Z}{\sqrt{2}\delta_L^{2/3}} - 1 \right]^{-2} \tag{5.85}$$

$$2 \cdot \left\lfloor \frac{R}{\sqrt{2}\delta_L} \right\rfloor^{-2} = 2 \cdot \left\lfloor \frac{Z\delta_L^{2/3}}{\sqrt{2}\delta_L} \right\rfloor^{-2} \tag{5.86}$$

$$\leq \left[\frac{Z}{\sqrt{2}\delta_L^{2/3}} - 1 \right]^{-2} \tag{5.87}$$

$$\leq \frac{4}{Z^2}\delta_L^{2/3} + \mathcal{O}\left(\delta_L^{4/3}\right) \tag{5.88}$$

Thus

$$|\text{length}(C) - \text{graphlength}(C)| \leq \delta_L^{2/3}\frac{4}{Z^2}\,\text{length}(C) + Z\delta_L^{2/3} \cdot \mathcal{C}_\xi(C) + \mathcal{O}(\delta_L) \tag{5.89}$$

$$\leq \max\left(\frac{4}{Z^2}, Z\right) \cdot \delta_L^{2/3} \cdot (\text{length}(C) + \mathcal{C}_\xi(C)) + \mathcal{O}(\delta_L) \tag{5.90}$$

which proves the result.

□

5.6 Algorithm and Experiments

Based on the above results, our algorithm is simple. We first cast the total variation problem in terms of algorithm 4. This, however, means that we are actually calculating a $\mathbf{u}^*$ that matches $\lambda' f$, thus we divide the output of algorithm 4 by λ' to get the final result.

We wish to present several proof-of-concept experiments to demonstrate our approach. While we illustrate our approach using several images, our purpose is not to compare against the numerous image segmentation or noise removal techniques from the computer vision community, but rather to illustrate the properties of our algorithm and the related theory. As such, images give a nice way to present our results; however, what we illustrate here applies more generally.

The software used for this approach was implemented in Python/C++ and released as

the open source package *LatticeFlow*.[2] It implements the Algorithms introduced in chapters 2, 3, 4, along with the graph construction for total variation minimization presented here. The core network flow algorithm was the standard push-relabel algorithm described in chapter 1.

We present here algorithmic results on three separate images: the *Mona Lisa*, which was briefly presented as an example earlier; *Truffles*, a picture of homemade, melt-in-your-mouth chocolate truffles which, sadly, have now been eaten; and *Castle*, a picture of Bodium castle in East Sussex, England. Several TV-regularized constructions, as well as the edges found, are presented along with timings and regularization paths. All results used the connectivity pattern used to form the lattice to approximate the total variation term is the one shown in Figure (5.2).

Figures (5.7) and (5.8) show a more detailed analysis of the Mona Lisa painting used already in Figure (5.1) as an example. In Figure (5.7), we show the painting for four different values of λ'. Here, f is the gray-scale version of the image, and u is result of solving ($\Re_{\mathrm{TV}}$). This image is quite noisy due to age, so we attempt to recover a function u that eliminates this noise.

As can be seen, the quality and characteristics of the final result varies significantly in terms of the different values. Since the original image is corrupted by noise (in this case, the wear and tear of time), we seek to recover the true image using the total variation regularization. In Figure (5.8), we show detailed renderings of the regularization path for these images. Here they are quite dense – due to the smooth nature of the image, there are numerous individual values in the total variation path. The figures shown are simply a smoothed rendering of the total lines paths; as can be seen, there are millions of path segments.

The image used here had dimension 512×344. It took 43.8 seconds to calculate the entire regularization path on $[0.1, 0.25]$ on an 2.9 GHz Intel laptop; in contrast, it took 6 minutes 45 seconds to calculate 100 values between 0.1 and 0.25. This translates to a significant speedup if more than 10 or so values of λ' need to be calculated, which is definitely a typical

[2]Available from http://github.com/hoytak/latticeflow

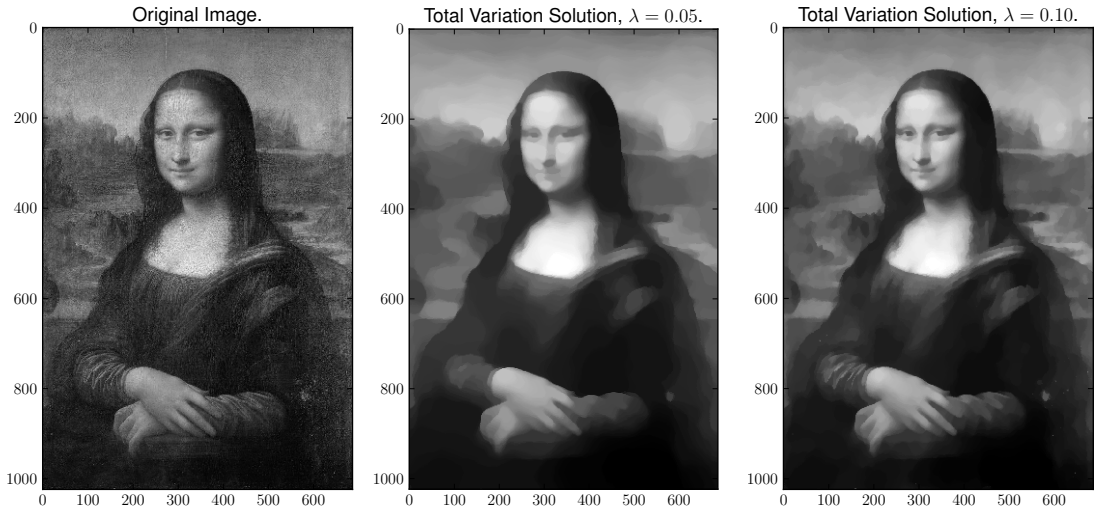
case when λ' is chosen by cross-validation or other automatic methods. The regularization path takes significantly longer to calculate for lower values of λ' as the split calculations are on much larger regions and these take a long time to calculate. However, many optimizations are possible, particularly with the network flow solver, so these timings could be significantly improved.

Similarly, results on the Truffles image demonstrates the properties of the total variation minimization with several different λ'. Here, f is the true image. Here, as the value of λ' decreases, the results clearly show how the TV regularization will pull out the most significant regions. In addition, the TV term gives well defined boundaries for these regions, as shown in Figure (5.10).

Finally, Figure (5.11) shows similar results on the Bodium castle image. Here, the total variation terms give excellent detection of the significant boundaries of the image. This illustrates one of the reasons TV minimization is used for edge and object detection.

5.7 Conclusion

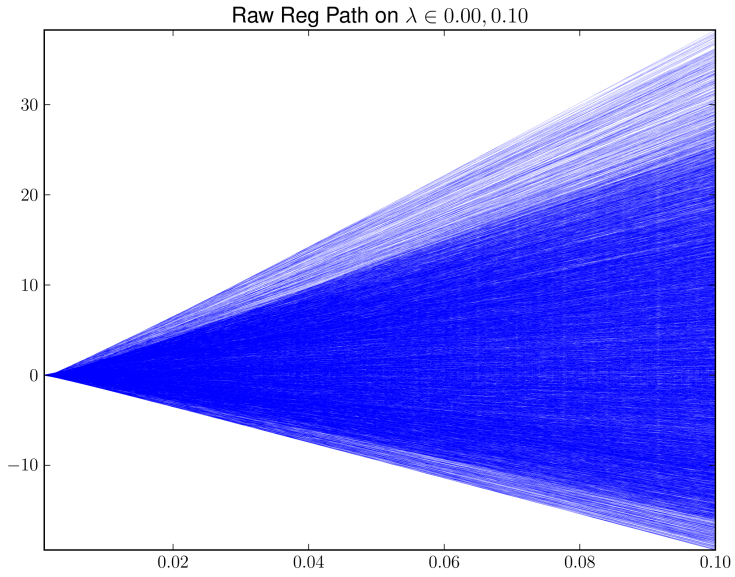
In this chapter, we have presented an algorithm to calculate the total variation problem using a lattice-based approximation to translate the problem into a form easily solvable using our methods. In two dimensions, we rigorously show accuracy bounds on this approximation. In addition, using this formulation, we show that the entire regularization path can be solved exactly by the algorithms given in the previous chapter. Finally, we demonstrate that it is indeed effective at solving the full total variation problem by demonstrating its use on a number of image problems. Numerous improvements are possible, but our purpose is primarily to validate the theory laid out in this and previous chapters, which we do successfully.



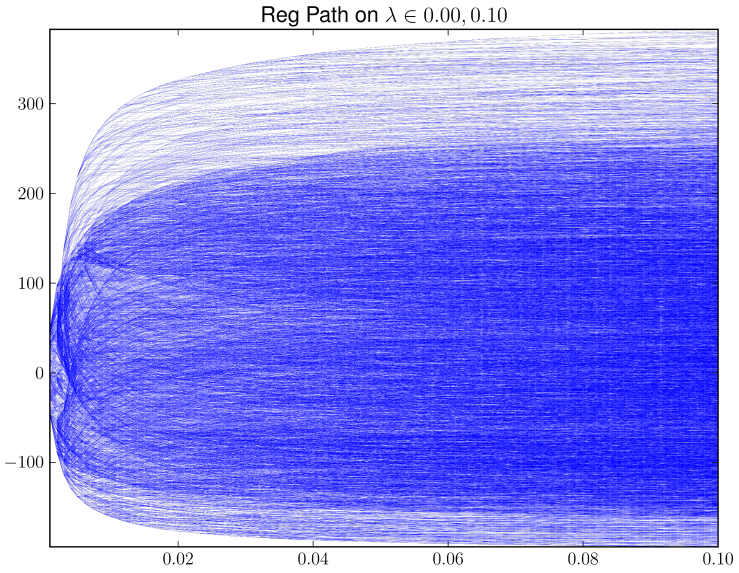
(a) Original Image (b) $\lambda' = 0.05$ (c) $\lambda' = 0.1$

(d) $\lambda' = 0.15$ (e) $\lambda' = 0.2$ (f) $\lambda' = 0.25$

Figure 5.7: Total variation solution to the *Mona Lisa* image.

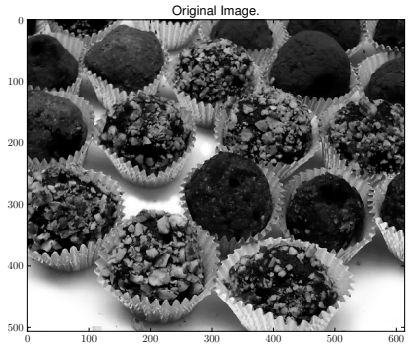


(a) Solution path from algorithm 4, untransformed linear solution.

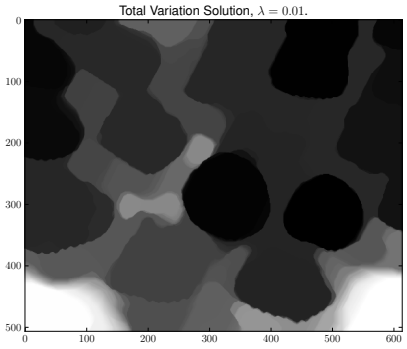


(b) Regularization Path of Total Variation Minimization Problem, transformed by λ'^{-1}.

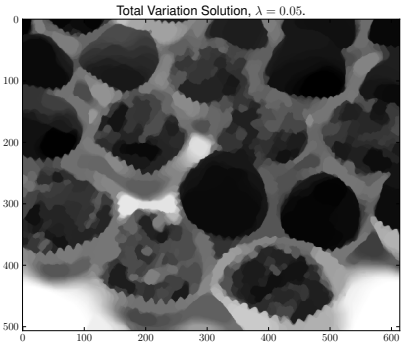
Figure 5.8: Total variation solution paths for the *Mona Lisa* image.



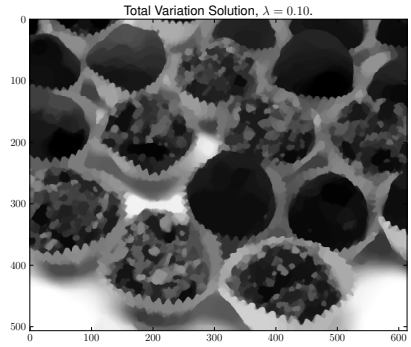
(a) Original Image



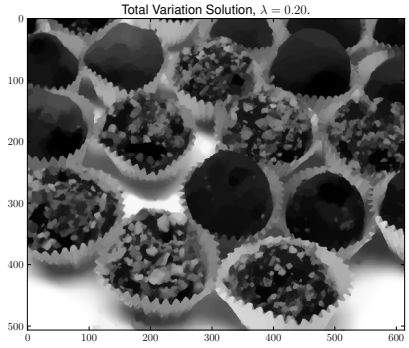
(b) $\lambda' = 0.01$



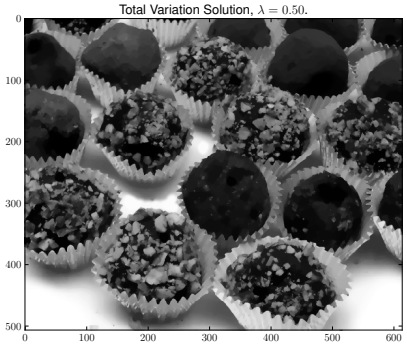
(c) $\lambda' = 0.05$



(d) $\lambda' = 0.1$

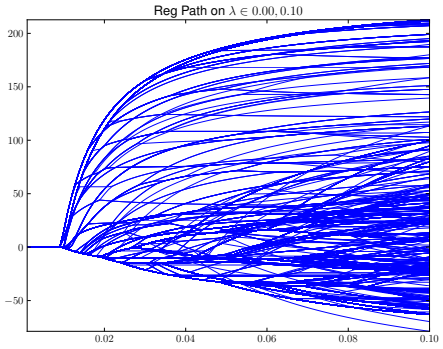


(e) $\lambda' = 0.2$

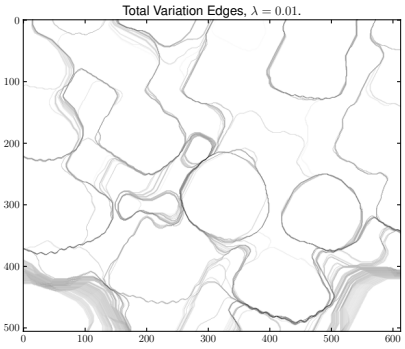


(f) $\lambda' = 0.5$

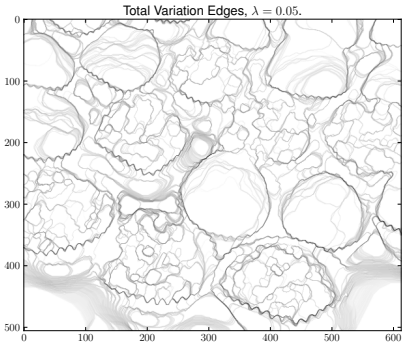
Figure 5.9: Total variation solution on the *Truffles* image.



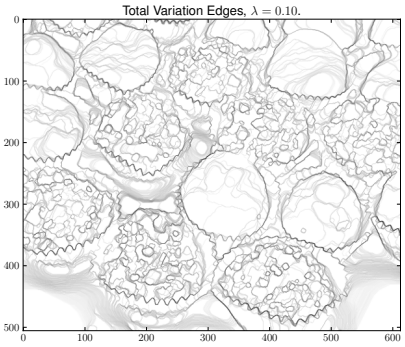
(a) Regularization Path (major regions only).



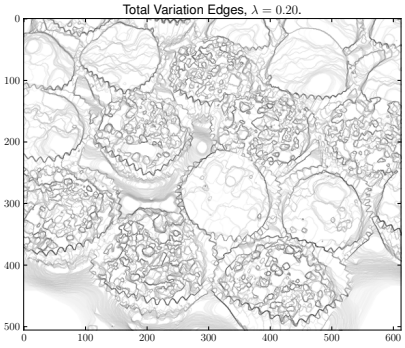
(b) $\lambda' = 0.01$



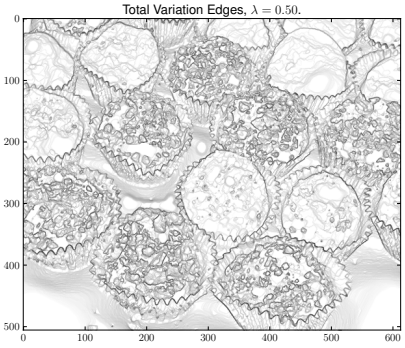
(c) TV term with $\lambda' = 0.05$



(d) TV term with $\lambda' = 0.1$



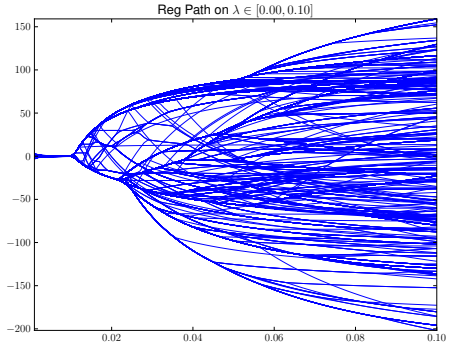
(e) TV term with $\lambda' = 0.2$



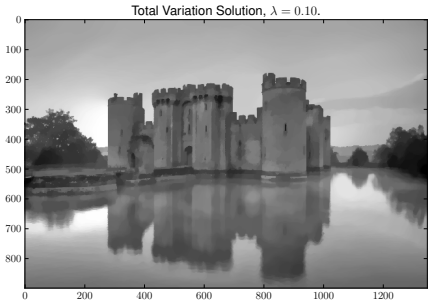
(f) TV term with $\lambda' = 0.5$

Figure 5.10: Total variation of the minimal solution on the *Truffles* image.

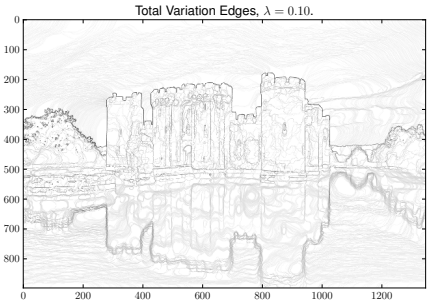
(a) Bodium Castle, Original



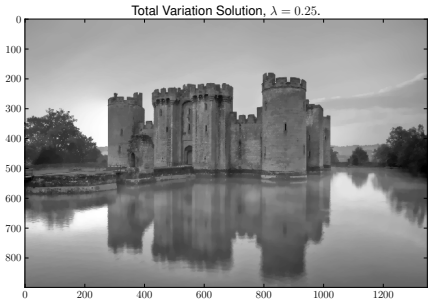
(b) Regularization Path, $\lambda' \in [0, 0.1]$, major regions only.



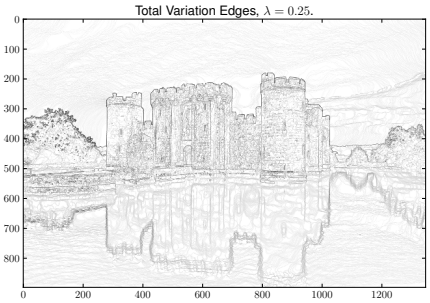
(c) TV solution with $\lambda' = 0.1$.



(d) $TV(u^*)$ term in solution at $\lambda' = 0.1$.



(e) TV solution with $\lambda' = 0.25$.



(f) $TV(u^*)$ term in solution at $\lambda' = 0.25$.

Figure 5.11: TV Minimization results for the *Castle* image.

Chapter 6

CONSISTENCY OF SPARSE RECOVERY WITH DEPENDENT VARIABLES

6.1 Recovery Consistency for Regularized Dependent Variables

The previous chapters have focused largely on the optimization questions surrounding ($\Re_{\mathrm{B}}$), and their extensions ($\Re_{\mathrm{L}}$), and ($\Re_{\mathrm{TV}}$). In this chapter, we address the issue of consistency. Namely, we wish to prove an extension of the classical consistency results for regular LASSO to accommodate structured priors given by the graph structures involved in defining ($\Re_{\mathrm{L}}$).

in a way that accommodates the recovery in structural topologies similar to the ones laid out previously.

In this chapter, we show consistency of sparse recovery as measured by a general L_1-like metric satisfying several assumptions. Specifically, we show oracle inequality bounds on the recovery rate of an estimator $\hat{\theta}_n$ from a "true" θ_n^* as determined by the solution to a penalized empirical risk problem. Formally, we show that with high probability, the following bound holds:

$$\mathrm{dist}\Big(\hat{\theta}_n,\ \theta_n^*\Big)\ \leq \Delta\zeta_n, \tag{6.1}$$

where Δ is a constant and $\zeta_n \to 0$ at a rate determined by the conditioning of the problem and the regularization parameters. The details are messy, and rely heavily on empirical process theory, but the end result is very general and offers some insight into the asymptotic behavior of the problem.

The metric $\mathrm{dist}(\cdot,\ \cdot)$ we hope to use is a type of "earthmover" metric [Karloff et al., 2006, Levina and Bickel, 2001, Andoni et al., 2008, Rubner et al., 2000], in which the metric distance is given by the minimum cost max-flow on a weighted bipartite graph between parameter values. This metric can be weighted to reduce to L_1 distance under our assumptions. However, it can also incorporate spatial dependencies in its results.

There are several recent results in this vein. One of the more rigorous results on consistency comes from Rinaldo [2009], where the authors examine and prove the consistency of the fused lasso. Other recent results have addressed the consistency of the graphical fused Lasso [Yang et al., 2012, Chen et al., 2010], which generalizes the fused Lasso in that it allows difference penalties defined as a graph; a similar result to ours. However, the results they prove in those papers are much less general. Chen et al. [2010] in particular derives the asymptotic distribution of the estimator as the sample size goes to infinity. However, the

result we prove is a bound that includes the $p >> n$ case. The question we wish to address is close to the one in Sharma et al. [2013]; however, their result is mainly a computational treatment of the subject. Our approach is purely theoretical and makes use of several deep results in empirical process theory.

6.2 Mathematical Preliminaries

The proof and results we present below uses the proof technique found in Van De Geer [2008], and many of these results are based heavily on a number of inequalities found in empirical process theory. We review the needed results as necessary, but introducing some notation is important.

First, it is common to use measures (e.g. P or P_n) as operators to denote expectation with respect to that measure. In other words,

$$P\,f = \mathrm{E}_P\,f(X) = \int f(x)\mathrm{d}P(x). \tag{6.2}$$

Similarly, when we use the empirical process, we assume a random sample $X_1, X_2, ..., X_n$. Then

$$P_n f = \int f(x)\mathrm{d}P_n(x) = \frac{1}{n}\sum_{i=1}^{n} f(X_i), \tag{6.3}$$

which is just the empirical mean of f when the random samples are drawn from P.

Similarly, we are interested in bounding the maximal difference between, for example, results obtained on the empirical and results on the true distribution. In this case, we might be interested in bounding our convergence by looking at the most problematic function in a class of functions. In that case, we would be interested in bounding

$$\mathscr{E}_{\mathcal{F}} = sup_{f\in\mathcal{F}}|P\,f - P_n\,f|, \tag{6.4}$$

where $\mathcal{F}$ is some class of P-measurable functions. Since this is a common case, we define a supremum norm over measures as the above, denoting

$$\|Y\|_{\mathcal{F}} = \sup_{f\in\mathcal{F}}|Y\,f|. \tag{6.5}$$

Thus

$$\|P - P_n\|_{\mathcal{F}} = \sup_{f\in\mathcal{F}}|P\,f - P_n\,f|. \tag{6.6}$$

6.2.1 Some Inequalities

Much of the critical work in empirical processes has been developing uniform analogues of standard inequalities. Such inequalities have proved to be the bread and butter of work in empirical processes. Among these, concentration inequalities [Massart, 2007, Klein and Rio, 2005, Panchenko, 2003, Giné and Koltchinskii, 2006], which bound a random quantity around their expectation, have been particularly useful. In essence, these inequalities allow one to control an empirical process in terms of its expectation, which can often be more convenient to work with.

Talagrand's Concentration Inequality

One of the most significant breakthroughs in Empirical Process was a uniform version of Bernstein's inequality for bounded empirical processes. This was first solved by Talagrand [1996] and has been extended and reformulated several times by Bousquet [Bousquet, 2002], Klein and Rio [Klein and Rio, 2005], Massart [Massart, 2007, 2000], and others.

Theorem 6.2.1 (Talagrand's Concentration Inequality)**.** *Let $x_1, ..., X_n$ be independent random variables in S. For any class of functions $\mathcal{F}$ on S that is uniformly bounded by a constant $U > 0$ and for all $t > 0$,*

$$\mathrm{P}\left(\left|\left\|\sum_{i=1}^{n} f(X_i)\right\|_{\mathcal{F}} - \mathrm{E}\left\|\sum_{i=1}^{n} f(X_i)\right\|_{\mathcal{F}}\right| \geq t\right) \leq K \exp\left[-\frac{t}{KU}\log\left(1+\frac{tU}{V}\right)\right] \tag{6.7}$$

where K is a universal constant and V is any number satisfying

$$V \geq \mathrm{E}\sup_{f\in\mathcal{F}}\sum_{i=1}^{n} f^2(X_i) \tag{6.8}$$

The following two bounds of the expected value of an empirical process around their expectation are particularly useful; we use the upper bound in the L1 consistency proof later on. Both are refinements of Talagrand's original concentration inequalities. The first of these is due to Bousquet [Bousquet, 2002], and the second is due to Klein and Rio [Klein and Rio, 2005].

Theorem 6.2.2 (Concentration Inequalities)**.** *Let $\mathcal{F}$ be a class of measurable functions from S to $[0,1]$, and let $\sigma^2 = \sup_{f\in\mathcal{F}}(Pf^2 - (Pf)^2)$. Then the following bounds hold for all $t > 0$:*

$$\mathrm{P}\left(\|P_n - P\|_{\mathcal{F}} \leq \mathrm{E}\,\|P_n - P\|_{\mathcal{F}} + \sqrt{\frac{2t}{n}(\sigma^2 + 2\,\mathrm{E}\,\|P_n - P\|_{\mathcal{F}})} + \frac{t}{n}\right) \leq e^{-t} \tag{6.9}$$

$$\mathrm{P}\left(\|P_n - P\|_{\mathcal{F}} \geq \mathrm{E}\,\|P_n - P\|_{\mathcal{F}} - \sqrt{\frac{2t}{n}(\sigma^2 + 2\,\mathrm{E}\,\|P_n - P\|_{\mathcal{F}})} - \frac{t}{3n}\right) \leq e^{-t} \tag{6.10}$$

Contraction Inequalities

Another very practical inequality due to Talagrand in [Ledoux and Talagrand, 1991] allows functions that contract differences to be removed from a Rademacher sequence. The following version is from [Koltchinskii, 2009]:

Theorem 6.2.3 (Contraction Inequality)**.** *Let $T \subset \mathbb{R}^n$, and let $f_i : \mathbb{R} \mapsto \mathbb{R}$, $i = 1, ..., n$ be functions such that $f_i(0) = 0$ and*

$$|f_i(u) - f_i(v)| \leq |u - v|, u, v \in \mathbb{R} \tag{6.11}$$

(if this is true, then f_i is a contraction). Then for all convex, nondecreasing functions $\phi : \mathbb{R}^+ \mapsto \mathbb{R}^+$,

$$\mathrm{E}\,\phi\left(\sup_{t\in T}\left|\sum_{i=1}^{n}\varepsilon_i f_i(t_i)\right|\right) \leq \mathrm{E}\,\phi\left(\sup_{t\in T}\left|\sum_{i=1}^{n}\varepsilon_i t_i\right|\right) \tag{6.12}$$

Proof. Proof in [Ledoux and Talagrand, 1991] or [Koltchinskii, 2009]. □

Symmetrization Theorem

We give uniform version of classical symmetrization inequalities that use Rademacher random variables to create an analogous symmetric distribution. The version here, used in [Van De Geer, 2008], is taken from [Van der Vaart and Wellner, 1996].

Theorem 6.2.4 (Symmetrization Theorem)**.** *Let $Z_1, ..., Z_n$ be independent random variables with values in $\mathcal{Z}$, and let $\varepsilon_1, ..., \varepsilon_n$ be a Rademacher sequence independent of $Z_1, ..., Z_n$.*

Then

$$\mathrm{E}\left[\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^{n}\left(f(Z_i)-\mathrm{E}\,f(Z_i)\right)\right|\right]\leq 2\,\mathrm{E}\left[\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^{n}\varepsilon_i f(Z_i)\right|\right] \tag{6.13}$$

Proof. See [Van der Vaart and Wellner, 1996]. □

Now that we have some preliminary results in place, we wish to put them to work in the context of a penalized sparse regression problem.

6.2.2 Empirical Risk Minimization for Sparse L_1 Optimization

As the focus of this paper, we wish to show consistency of estimation of a sparse regression problem using the tools of empirical risk minimization. Suppose we have a class of functions $\mathcal{F}=\{f_\theta:\theta\in\Theta\}$ indexed by a parameter $\theta\in\Theta\subseteq\mathbb{R}^m$, with Θ convex, and

$$f_\theta(\cdot)\equiv\sum_{k=1}^{m}\theta_k\psi_k(\cdot),\quad \theta\in\Theta \tag{6.14}$$

For a set of basis functions $\{\psi_k\}_{k=1}^m$, where we assume that $\mathrm{E}\,\psi_k^2=1\ \forall\,k$ and $\|\psi\|_\infty\leq K_m$ for some constant K. Thus Θ and $\{\psi\}$ define a linear space of functions on $\mathcal{X}$.

The unique aspect of this problem is that the dimension m of the recovery variable can be significantly greater than the sample size n. We handle this case by employing the finite sample bounds from empirical process theory discuss previously.

We formulate the sparse regression problem as follows. We assume that we observe a sequence of $(X_i,Y_i)\in\mathcal{X}\times\mathcal{Y}$. We then wish to show consistency of the following estimator:

$$\hat{\theta}_n=\operatorname*{argmin}_{\theta\in\Theta}\left[\frac{1}{n}\sum_{i=1}^{n}L(f_\theta(X_i),Y_i)\right]+\lambda_n\|\theta\|_1 \tag{6.15}$$

$$=\operatorname*{argmin}_{\theta\in\Theta}\ \mathcal{E}_n(f_\theta)+\lambda_n\|\theta\|_1 \tag{6.16}$$

where $L(f(X_i),Y_i)=L_f$ is a convex loss function satisfying a set of regularity assumptions described in the next section. Now, consider a target function

$$f^0\equiv\operatorname*{argmin}_{f\in\mathbf{F}}\ PL_f, \tag{6.17}$$

where $\mathbf{F}\supseteq\mathcal{F}$ and P is the distribution function of (X,Y). We wish to show that if f^0 can be approximated well by a function f_{θ^*} where only a few of the terms in θ^* are non-zero, then

the estimator $\hat{\theta}_n$ will roughly reproduce this sparsity pattern. Specifically, our proof shows the convergence of $\hat{\theta}_n$ to the solution of the estimation problem θ_n^* (with a slightly different penalty term) as if the true sparsity pattern is known. Thus this result is an example of an oracle inequality in that it shows that the estimator based on the empirical distribution is not significantly worse than if the true model were known.

More formally, we show that the empirical risk of the estimated model is not too far off of the true model, i.e., with high probability

$$\mathcal{E}(f_{\hat{\theta}_n}) \leq C \min_{\theta \in \Theta} [\mathcal{E} f_\theta + \nu(\theta)] \tag{6.18}$$

The additional term, $\nu(\theta)$, refers to the estimation error. This term incorporates inherent aspects of the problem due to the dimension and sparsity. [Van De Geer, 2008] reports that this term usual is of order $\log m \times |\text{support}\, \theta| \, / n$, where the extra log term multiplying the typical n^{-1} estimation convergence rate is due to the fact that part of the optimization is "deciding" which parameters are relevant.

This proof closely follows a similar proof by Sara Van de Geer given in Van De Geer [2008]. However, it differs in one major and one minor way. Less importantly, we simplify a few of the constants to make the proofs more readable while still preserving much of their generality. More importantly, however, we generalize her proof to measure the closeness of two parameters $\theta_1, \theta_2 \in \Theta$ in terms of a more general metric $\text{dist}(\theta_1,\ \theta_2)$ instead of simply by the L1 distance $\|\theta_1 - \theta_2\|_1$. This metric needs to be "similar" to L_1 distance in that it must reduce to $\|\cdot\|_1$ when one component is 0 – see the formal assumptions in section 6.2.2 – but it allows for several useful extensions, as will be discussed. The results are shown under a similar set of regularity conditions on the loss function and an additional set of assumptions about the metric, which we discuss in the following sections.

The main result, theorem 6.2.14, will be given after a set of regularity assumptions on the loss function and on the metric.

Assumptions on the Loss Function

The lemmas and theorems in the next section all assume the following regularity conditions on the loss function.

A1. **Convex.** We assume the loss function $L(f_\theta(x), y)$ is convex in x for all $y \in \mathcal{Y}$, and $\forall y \in \mathcal{Y}$ and $\theta \in \Theta$.

A2. **Continuity.** The loss function L satisfies the following Lipshitz continuity condition:

$$|L(f_{\theta_1}(x), y) - L(f_{\theta_2}(x), y)| \leq |f_{\theta_1}(x) - f_{\theta_2}(x)| \tag{6.19}$$

Note that in taking the Lipshitz constant to be one, there is no loss of generality as L can be rescaled without changing the problem.

A3. The basis functions forming the class $\mathcal{F}$ are bounded by a constant K_m:

$$K_m = \max_{1 \leq k \leq m} \|\psi_k\|_\infty < \infty \tag{6.20}$$

A4. The excess risk around f^0 is well behaved. Specifically, there exists a strictly convex $G : \mathbb{R}^+ \mapsto \mathbb{R}^+$, such that $G(0) = 0$, $G \nearrow$ on $\mathbb{R}^+$, and

$$\mathcal{E}(f_\theta) \geq G(\left\|f_\theta - f^0\right\|_2) \tag{6.21}$$

Denote the convex conjugate of this function as $G^\dagger$.

A5. There exists a function $D(\cdot)$ on the subsets of the index set $\{1, ..., m\}$ such that for all subsets $A \subset \{1, ..., m\}$ and all $\theta_1, \theta_2 \in \Theta$, we have that

$$\text{dist}_{[A]}(\theta_1,\ \theta_2)\ \leq \sqrt{D(A)}\, \|f_{\theta_1} - f_{\theta_2}\|_2\,, \tag{6.22}$$

where $\text{dist}_{[A]}(\theta_1,\ \theta_2)$ denotes the metric $\text{dist}(\cdot,\ \cdot)$ restricted to the subspace defined by the dimensions in A.

Remark 6.2.5 (Lipshitz continuity)**.** *First, the Lipshitz condition is not as critical an assumption as it might first seem. In particular, one common case – that of squared error loss – does not satisfy it uniformly on* $\mathbb{R}$*. However, this can be handled effectively through assumptions on* P *or truncation. See [Van De Geer, 2008] for details.*

Remark 6.2.6 (Conditioning of Results)**.** *The conditioning of the problem plays a significant role in the estimation accuracy and rate, and many of these issues are collected into the functions G and D. As the primary instance of this, the conditioning of the basis set $\{\psi_k\}$ is an important consideration in estimation accuracy. Consider the $m \times m$ matrix*

$$\mathbf{\Sigma} = [\,\mathrm{E}\,\psi_k \psi_\ell]_{k,\ell \in \{1,\ldots,m\}} \tag{6.23}$$

and suppose it has minimum eigenvalue $\sigma^2_{\min}$. It is then easy to verify that $D(A) = |A|\,/\beta^2$ satisfies the requirement of assumption (A5) (see [Van De Geer, 2008] and [Tarigan and van de Geer, 2006]). Intuitively, the more orthogonal the basis set is, the tighter the concentration bounds on the empirical risk and the better the estimation accuracy.

Assumptions on Metrics

In the next section, we present a proof that extends the consistency proof of [Van De Geer, 2008] to the case where recovery accuracy is measured in terms of metrics – satisfying certain assumptions – rather than norms. The goal of this section is to describe the assumptions of the metric under which we proved this assumption.

Extending the general lasso results to the case of metrics can be desirable if some of the parameters are highly correlated, a situation that can arise in, for example, spatial density estimation. In this case, variables spatially close to each other are also likely to have similar indexing basis functions. For example, a version of the "earthmover" metric [Karloff et al., 2006, Levina and Bickel, 2001, Andoni et al., 2008, Rubner et al., 2000] can be weighted to reduce to $L1$ distance under the needed assumptions and cost terms, yet incorporate spatial similarity in results. This type of metric is one candidate for our result.

The proof also requires us to use a *conditional* or restricted metric on which only certain components of the initial vectors are considered. We denote this conditional metric as $\mathrm{dist}_{[A]}(\theta_1,\, \theta_2)$ to say that the metric is computed only on the subset of indices A. Similarly, $\mathrm{dist}_{[\theta_3]}(\theta_1,\, \theta_2)$ denotes that the metric is computed only on the subset of indices on which θ_3 is non-zero, and we use $[\theta]^c$ to denote the complement of the support, i.e. $\{i : \theta_i = 0\}$.

We require several conditions on how the metrics relate to the problem and to the $L1$ norm that allow us to move between them and work with the $L1$ penalty term in several key parts of the proof. These are:

M1. **Reduction to L1.** $\text{dist}(\theta,\ 0) = \|\theta\|_1$.

M2. **Subadditivity of conditioning.** $\text{dist}(\theta_1,\ \theta_2) \leq \text{dist}_{[A]}(\theta_1,\ \theta_2) + \text{dist}_{[A^c]}(\theta_1,\ \theta_2)$

M3. **Invariance to ignored parameters.** $\text{dist}_{[[\theta_2]^c]}(\theta_1,\ 0) = \text{dist}_{[[\theta_2]^c]}(\theta_1,\ \theta_2)$

M4. **Convexity.** $\forall s \in [0,1], \theta_0, \theta_1 \in \Theta,\ \text{dist}(s\theta_1 + (1-s)\theta_2,\ \theta_2) \leq s\ \text{dist}(\theta_1,\ \theta_2)$

Furthermore, we assume the existence of a mapping between norms based on corresponding radii of a neighborhood of $\mathcal{F}$. This will be used in a couple key places, when the only available bound is on the metric, to jump to an alternate value of θ that is closer in $L1$ space to a specific value.

Definition 6.2.7 (Neighborhood Basis)**.** *Let*

$$\mathcal{F}_M = \{f_\theta : \theta \in \Theta, \text{dist}(\theta,\ \theta_0) \leq M\} \tag{6.24}$$

be the class of functions with parameters not too different from a fixed, given point θ_0 as indexed by the radius M pall in parameter space.

Definition 6.2.8 (Context Sensitve Mapping)**.** *We define a context sensitive mapping γ that controls the diameter of sets similar to $\mathcal{F}_M$ as measured by the $L1$ difference versus the metric* $\text{dist}(\cdot,\ \cdot)$ *.*

$$\gamma(\theta_0, M) = \sup\left\{\inf_{\theta:\|f_\theta - f\|_2 = 0} \|\theta - \theta_0\|_1\ :\ f \in \mathcal{F}_M(\theta_0)\right\} \tag{6.25}$$

Denote a ratio-type lower bound on this value with

$$\Gamma(M) = M / \max_{\theta\in\Theta} \gamma(\theta_0, M) \tag{6.26}$$

and denote by Γ_0 the constant

$$\Gamma_0 = \max_M \Gamma(M) \tag{6.27}$$

Note that both these constants reduce to 1 when you let $\text{dist}(\theta_1,\ \theta_2) = \|\theta_1 - \theta_2\|_1$.

This distance metric is capable of capturing solutions that, for example, spread the non-zero mass over a region. This can be done by setting up a flow metric between the true solution and the spatially similar one nearby, while having the cost be a function of the distance. This also extends the classic consistency results for Lasso and other sparse sensing domains in which there is no inherent ordering or patterning amongst the variables; incorrectly swapping the non-zero-ness of two variables was as bad as swapping any others.

Proof of Consistency

We first give a set of constants and associated quantities for which this proof is valid. Fix $t > 0$, Fix $d_0 > 1$ and $\delta \in (0, 1)$, and let

$$d = \frac{1 + d_0}{1 - d_0} d_0 \tag{6.28}$$

$$a_n = \sqrt{\frac{2\log(2m)}{n}} + \frac{\log(2m)}{n} K_m \tag{6.29}$$

$$\lambda_n(t) = 8a_n \left(1 + t\sqrt{2(1 + 8a_n\Gamma_0 K_m)} + \frac{8}{3} t^2 a_n \Gamma_0 K_m \right) \tag{6.30}$$

$$\nu_n(t, \delta, \theta) = 2\delta G^{\dagger}(2\lambda_n(t)\sqrt{D(\text{support}\ (\theta))}) \tag{6.31}$$

$$\theta_n^* = \underset{\theta \in \Theta}{\text{argmin}}\ \mathcal{E}(f_\theta) + \nu_n(t, \delta, \theta) \tag{6.32}$$

$$\zeta_n(t, \delta) = \left[(1 + \delta)\mathcal{E}\left(f_{\theta_n^*}\right) + \nu_{\theta_n^*} \right] / \lambda_n(t) \tag{6.33}$$

Step 1: Bounding the Difference Between Empirical and True

Lemma 6.2.9. *Let $\mathcal{F}_M$ be as defined in definition 6.2.7, and let*

$$Z(M) = \sup_{f \in \mathcal{F}_M} (P_n - P)(L(f, \cdot) - L(f_{\theta_0}, \cdot)) \tag{6.34}$$

$$= \sup_{f \in \mathcal{F}_M} (\mathcal{E}_n - \mathcal{E})(f - f_{\theta_0}) \tag{6.35}$$

Then $\forall\, t > 0$,

$$\text{P}(Z(M) \geq M\lambda_n(t)/2) \leq e^{-16nt^2 a_n^2 \Gamma^2(M)} \tag{6.36}$$

Proof. First, we apply the Rademacher symmetrization inequality to get

$$\mathrm{E}\, Z(M) \leq 2\,\mathrm{E}\left\{ \sup_{f\in\mathcal{F}_M} \left| \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i (L(f(X_i),Y_i) - L(f_{\theta_0}(X_i),Y_i)) \right| \right\}. \tag{6.37}$$

Now L is Lipshitz continuous, and without loss of generality, we can assume that the Lipshitz constant is 1. We can then apply the contraction theorem to get

$$\mathrm{E}\left\{ \sup_{f\in\mathcal{F}_M} \left| \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i (L(f(X_i),Y_i) - L(f_{\theta_0}(X_i),Y_i)) \right| \mid \mathbf{X},\mathbf{Y} \right\} \tag{6.38}$$

$$\leq 2\,\mathrm{E}\left\{ \sup_{f\in\mathcal{F}_M} \left| \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i (f(X_i) - f_{\theta_0}(X_i)) \right| \mid \mathbf{X},\mathbf{Y} \right\} \tag{6.39}$$

However,

$$\left| \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i (f(X_i) - f_{\theta_0}(X_i)) \right| \tag{6.40}$$

$$\leq \left[\sum_{k=1}^{m} \left|\theta_k - \theta_k^0\right| \right] \left[\max_{1\leq k\leq m} \left| \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i \psi_k(X_i) \right| \right] \tag{6.41}$$

$$= \|\theta - \theta_0\|_1 \max_{1\leq k\leq m} \left| \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i \psi_k(X_i) \right| \tag{6.42}$$

Now, since $\mathrm{dist}(\theta,\ \theta_0) \leq M$, we have that $\|\theta - \theta_0\|_1 \leq \gamma(\theta_0, M)$ (recall definition 6.2.8), we can combine these results to give:

$$\mathrm{E}\, Z(M) \leq 4\gamma(\theta_0, M)\,\mathrm{E}\left\{ \max_{1\leq k\leq m} \left| \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i \psi_k(X_i) \right| \right\} \tag{6.43}$$

The next step is to bound this quantity by a constant factor depending only on n, m, and K_m. Let

$$\phi_k = \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i \psi_k(X_i), \quad k = 1, ..., m \tag{6.44}$$

An intermediate step in the proof of Bernstein's inequality (or applying Chebychev's inequality to the exponential) gives us that, for $\beta < n/K_m$,

$$\mathrm{E}\, \exp(\beta\phi_k) \leq \exp\left(\frac{\beta^2}{2(n - \beta K_m)} \right) \tag{6.45}$$

The same is true if we replace ϕ_k by $-\phi_k$. In this case,

$$\mathrm{E}\left(\max_{1\le k\le m}|\phi_k|\right) \le \frac{1}{\beta}\log\left[\mathrm{E}\exp\left(\beta\max_{1\le k\le m}\pm\phi_k\right)\right] \tag{6.46}$$

$$\le \frac{1}{\beta}\log\left(2m\exp\left(\frac{\beta^2}{2(n-\beta K)}\right)\right) \tag{6.47}$$

$$= \frac{\log(2m)}{\beta} + \frac{\beta}{2(n-\beta K)} \tag{6.48}$$

where the $2m$ in (6.47) comes from summing over both the positive and negative components to create a bound over the maximum of the absolute value. Now, we can take

$$\frac{n}{\beta} = K + \sqrt{\frac{n}{2\log 2m}} \tag{6.49}$$

which gives us a nice upper bound:

$$\mathrm{E}\left(\max_{1\le k\le m}|\phi_k|\right) = \mathrm{E}\left(\max_{1\le k\le m}\left|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\psi_k(X_i)\right|\right) \tag{6.50}$$

$$\le \sqrt{\frac{2\log(2m)}{n}} + \frac{\log(2m)}{n}K = a_n \tag{6.51}$$

Thus we have a bound on $\mathrm{E}\,Z(M)$ that we can use in conjunction with Bousquet's bound to get a bound on Z dependent only on M and constants defined in the rest of the problem. We have

$$\mathrm{E}\left[\frac{Z(M)}{\gamma(\theta_0,M)}\right] \le 4\left[\sqrt{\frac{2\log(2m)}{n}} + \frac{\log(2m)}{n}K\right] = 4a_n \tag{6.52}$$

Thus by Bousquet's inequality, we have $\forall t > 0$,

$$\mathrm{P}\left(Z(M) \ge \mathrm{E}\,Z(M) + \sqrt{\frac{2t'}{n}(1+2\,\mathrm{E}\,Z(M)K_m)} + \frac{t'}{3n}\right) \le e^{-t'} \tag{6.53}$$

$$\Rightarrow \mathrm{P}\left(Z(M) \ge 4\gamma(\theta_0,M)a_n\left(1+t\sqrt{2(1+8a_nK_m)} + \frac{8}{3}t^2a_nK_m\right)\right) \le e^{-16nt^2a_n^2} \tag{6.54}$$

$$\Rightarrow \mathrm{P}\left(Z(M) \ge 4Ma_n\left(1+t\sqrt{2(1+8a_n\Gamma_0K_m)} + \frac{8}{3}t^2a_n\Gamma_0K_m\right)\right) \le e^{-16nt^2a_n^2\Gamma^2(M)} \tag{6.55}$$

$$\Rightarrow \mathrm{P}(Z(M) \ge M\lambda_n(t)/2) \le \exp\left[-16nt^2a_n^2\Gamma^2(M)\right] \tag{6.56}$$

which concludes the proof. □

Step 2: Bounding the Difference from the Oracle

The first step of the proof is to bound θ around the oracle θ_n^* on the support of θ_n^*. The step after this will be to do the same on the step after that.

Lemma 6.2.10. *For all $\theta \in \Theta$ with $\operatorname{dist}(\theta,\ \theta_n^*) \leq d\zeta_n(t,\delta)$,*

$$2\lambda_n(t)\ \operatorname{dist}_{[\theta_n^*]}(\theta,\ \theta_n^*) \leq \delta\big(\mathcal{E}(f_\theta) + \mathcal{E}(f_{\theta_n^*})\big) + \nu_n(t,\delta,\theta_n^*) \tag{6.57}$$

$$= \delta\mathcal{E}(f_\theta) - \mathcal{E}(f_{\theta_n^*}) + \lambda_n(t)\zeta_n(t,\delta) \tag{6.58}$$

Proof. For conciseness, we drop the explicit dependence on δ and t for λ_n and ζ_n. Define

$$\phi_n(r) = \operatorname*{argmin}_{\theta\in\Theta,\ \operatorname{dist}(\theta,\ \theta_n^*)\leq r} \left[\delta\mathcal{E}f_\theta - 2\lambda_n\ \operatorname{dist}_{[\theta_n^*]}(\theta,\ \theta_n^*)\right]. \tag{6.59}$$

Then

$$2\lambda_n\ \operatorname{dist}_{[\theta_n^*]}(\theta,\ \theta_n^*) = 2\lambda_n \langle\theta \mid \theta_n^*\rangle\, \theta_n^* - \delta\mathcal{E}(f_\theta) + \delta\mathcal{E}(f_\theta) \tag{6.60}$$

$$\leq 2\lambda_n\ \operatorname{dist}(\phi_n(d\zeta_n),\ \theta)\ - \delta\mathcal{E}(f_{\phi_n(d'\zeta_n)}) + \delta\mathcal{E}(f_\theta) \tag{6.61}$$

$$\leq 2\lambda_n\sqrt{D_\theta}\left\|f_{\phi_n(d\zeta_n)} - f_{\theta_n^*}\right\|_2 - \delta\mathcal{E}(f_{\phi_n(d'\zeta_n)}) + \delta\mathcal{E}(f_\theta) \tag{6.62}$$

where the last step follows by assumption (A5) and we have abbreviated $D(\operatorname{support}(\theta))$ as D_θ. Thus

$$2\lambda_n\sqrt{D_\theta}\left\|f_{\phi_n(d\zeta_n)} - f_{\theta_n^*}\right\|_2 \leq 2\lambda_n\sqrt{D_\theta}\Big(\left\|f_{\phi_n(d\zeta_n)} - f^0\right\|_2 + \left\|f_{\theta_n^*} - f^0\right\|_2\Big) \tag{6.63}$$

Now by Fenchel's inequality [Boyd and Vandenberghe, 2004] and assumption (A4),

$$2\lambda_n\sqrt{D_\theta}\left\|f_\theta - f^0\right\|_2 \leq \delta\left[G(\left\|f_\theta - f^0\right\|_2) + G^\dagger\Big(2\lambda_n\sqrt{D_\theta}/\delta\Big)\right] \tag{6.64}$$

$$\leq \delta\mathcal{E}(f_\theta) + \nu_n(t,\delta,\theta)/2 \tag{6.65}$$

so

$$2\lambda_n \langle\theta \mid \theta_n^*\rangle\, \theta_n^* \leq \delta\mathcal{E}(f_\theta) + \delta\mathcal{E}(f_{\phi_n(d'\zeta_n)} + \nu_n - \delta\mathcal{E}(f_{\phi_n(d'\zeta_n)}) + \delta\mathcal{E}(f_\theta) \tag{6.66}$$

$$= \delta(\mathcal{E}(f_\theta) + \mathcal{E}(f_\theta)) + \nu_n. \tag{6.67}$$

Substituting the appropriate defined values completes the proof. □

Now we aim to bound the difference between a given θ and θ_n^* when the penalized risk of θ is less than or equal to θ_n^*.

Lemma 6.2.11. *Consider $\theta \in \Theta$ such that*

$$\mathcal{E}_n(f_\theta) + \lambda_n(t)\,\|\theta\|_1 \leq \mathcal{E}_n(f_{\theta_n^*}) + \lambda_n(t)\,\|\theta_n^*\|_1 \tag{6.68}$$

Furthermore, let $1 < \alpha \leq d$ and let N be a positive integer. Then,

$$\mathrm{P}(\,\mathrm{dist}(\theta,\ \theta_n^*)\ \leq \alpha\zeta_n(t,\delta)) \leq\ \mathrm{P}\bigg(\,\mathrm{dist}(\theta,\ \theta_n^*)\ \leq \left[1 + \frac{\alpha-1}{2^N}\right]\zeta_n(t,\delta)\bigg) + Ne^{-16na_n^2t^2\Gamma_0} \tag{6.69}$$

Proof. By the definition of $Z(M)$, we know that when $\mathrm{dist}(\theta,\ \theta_n^*)\ \leq \alpha\zeta_n$, we can move from the empirical solution to the true solution using lemma 6.2.9:

$$\mathcal{E}_n(f_\theta - f_{\theta_n^*}) \leq \lambda_n(\|\theta_n^*\|_1 - \|\theta\|_1) \tag{6.70}$$

$$\Rightarrow \mathcal{E}(f_\theta - f_{\theta_n^*}) \leq \lambda_n(\|\theta_n^*\|_1 - \|\theta\|_1) + Z(\alpha\zeta_n) \tag{6.71}$$

With probability at least $1 - \exp(-16na_n^2t^2\Gamma_0)$, $Z(\alpha\zeta_n) \leq \lambda_n\alpha\zeta_n/2$. Thus

$$\mathcal{E}(f_\theta) + \lambda_n\,\|\theta\|_1 \leq \frac{1}{2}\lambda_n\alpha\zeta_n + \mathcal{E}(f_{\theta_n^*}) + \lambda_n\,\|\theta_n^*\|_1 \tag{6.72}$$

Thus, when both this condition and $\mathrm{dist}(\theta,\ \theta_n^*)\ \leq \alpha\zeta_n$ are satisfied,

$$\mathcal{E}(f_\theta)+\lambda_n\,\|\theta\|_1 = \mathcal{E}(f_\theta) + \lambda_n\ \mathrm{dist}_{[\theta_n^*]}(\theta,\ 0)\ + \lambda_n\ \mathrm{dist}_{[[\theta_n^*]^c]}(\theta,\ 0) \tag{6.73}$$

$$\begin{aligned}\Rightarrow \mathcal{E}(f_\theta)+\lambda_n\ &\mathrm{dist}_{[[\theta_n^*]^c]}(\theta,\ 0)\\ &\leq \lambda_n\alpha\zeta_n + \mathcal{E}(f_{\theta_n^*}) + \lambda_n\big\{\,\mathrm{dist}_{[\theta_n^*]}(\theta_n^*,\ 0)\ -\ \mathrm{dist}_{[\theta_n^*]}(\theta,\ 0)\,\big\} \qquad (6.74)\\ &\leq \lambda_n\alpha\zeta_n + \mathcal{E}(f_{\theta_n^*}) + \lambda_n\ \mathrm{dist}_{[\theta_n^*]}(\theta,\ \theta_n^*) \qquad (6.75)\end{aligned}$$

as Cauchy-Schwartz implies that $\mathrm{dist}(\theta_1,\ 0)\ \leq\ \mathrm{dist}(\theta_1,\ \theta_2)\ +\ \mathrm{dist}(\theta_2,\ 0)$. Now, note that $\mathrm{dist}_{[[\theta_n^*]^c]}(\theta,\ \theta_n^*)\ =\ \mathrm{dist}_{[[\theta_n^*]^c]}(\theta,\ 0)$. Now, by assumption (M2), we have that

$$\begin{aligned}\mathcal{E}(f_\theta) + \lambda_n\ \mathrm{dist}(\theta,\ \theta_n^*)\ &\leq \mathcal{E}(f_\theta) + \lambda_n\big\{\,\mathrm{dist}_{[[\theta_n^*]^c]}(\theta,\ \theta_n^*)\ +\ \mathrm{dist}_{[\theta_n^*]}(\theta,\ \theta_n^*)\,\big\} \qquad (6.76)\\ &\leq \lambda_n\alpha\zeta_n + \mathcal{E}(f_{\theta_n^*}) + 2\lambda_n\,\langle\theta\mid\theta_n^*\rangle\,\theta_n^* \qquad (6.77)\end{aligned}$$

Now $\alpha \leq d$, so by lemma 6.2.10, we have that

$$\begin{aligned}\mathcal{E}(f_\theta) + \lambda_n\ \mathrm{dist}(\theta,\ \theta_n^*)\ &\leq \lambda_n\alpha\zeta_n + \mathcal{E}(f_{\theta_n^*}) + \delta(\mathcal{E}(f_\theta) + \mathcal{E}(f_{\theta_n^*})) + \nu_n \qquad (6.78)\\ &= \frac{\alpha+1}{2}\lambda_n\zeta_n + \delta\mathcal{E}(f_\theta) \qquad (6.79)\\ \Rightarrow \lambda_n\ \mathrm{dist}(\theta,\ \theta_n^*)\ &\leq \frac{\alpha+1}{2}\lambda_n\zeta_n \qquad (6.80)\\ \Rightarrow\ \mathrm{dist}(\theta,\ \theta_n^*)\ &\leq \frac{\alpha+1}{2}\zeta_n \qquad (6.81)\end{aligned}$$

as $\mathcal{E}(f_\theta) \geq 0$ and $0 < \delta < 1$. The lemma is attained by successively applying these steps N times. □

Lemma 6.2.12. *Let*

$$\theta_s = s\hat{\theta}_n + (1-s)\theta_n^* \tag{6.82}$$

and let $1 < \alpha \leq d$. *Then if*

$$s = \frac{\alpha\zeta_n(t,\delta)}{\alpha\zeta_n(t,\delta) + \operatorname{dist}\left(\hat{\theta}_n,\ \theta_n^*\right)} \tag{6.83}$$

The following holds for any positive integer N *with probability at least* $1 - N\exp\left[-16na_n^2t^2\Gamma_0\right]$*:*

$$\operatorname{dist}(\theta_s,\ \theta_n^*) \leq \left(1 + \frac{\alpha-1}{2^N}\right)\zeta_n(t,\delta) \tag{6.84}$$

Proof. The loss function and penalty are convex, with $\hat{\theta}_n$ minimizing $\mathcal{E}(f_\theta) + \lambda_n$, so

$$\mathcal{E}_n(f_{\theta_s}) + \lambda_n \left\|\theta_s\right\|_1 \leq \mathcal{E}_n(f_{\theta_n^*}) + \lambda_n(t)\left\|\theta_n^*\right\|_1 \tag{6.85}$$

Moreover,

$$\operatorname{dist}(\theta_s,\ \theta_n^*) \leq s\ \operatorname{dist}\left(\hat{\theta}_n,\ \theta_n^*\right) = \frac{\alpha\zeta_n\ \operatorname{dist}\left(\hat{\theta}_n,\ \theta_n^*\right)}{\alpha\zeta_n + \operatorname{dist}\left(\hat{\theta}_n,\ \theta_n^*\right)} \leq d\zeta_n \tag{6.86}$$

We can now apply lemma 6.2.12 with $\theta = \theta_s$ to finish the proof. □

Lemma 6.2.13. *Suppose* N_1 *and* N_2 *are positive integers, and let*

$$\Delta(N_1, N_2) = 1 + 2^{-N_2}\left[\frac{1 + (d_0^2 - 1)2^{-N_1}}{(d_0 - 1)(1 - 2^{-N_1})}\right] \tag{6.87}$$

Then with probability at least $1 - (N_1 + N_2)\exp\left[-16na_n^2t^2\Gamma_0\right]$,

$$\operatorname{dist}\left(\hat{\theta}_n,\ \theta_n^*\right) \leq \Delta(N_1, N_2)\zeta_n(t,\delta) \tag{6.88}$$

Proof. The proof to this exactly follows the proof given in [Van De Geer, 2008], lemma A7, with the L1 distance replaced by the given metric and the exponential bounds adjusted by Γ_0. We thus omit the proof for the sake of conciseness. □

Step 3: An Oracle Inequality

Theorem 6.2.14. *With probability at least*

$$1 - \left[2 + (N_1 + N_2)\log_2\left(\Delta(N_1, N_2)(\delta^{-1} - \delta) \vee 1\right)\right]\exp\left[-16na_n^2 t^2 \Gamma_0\right] \tag{6.89}$$

we have that

$$\mathcal{E}(f_{\hat{\theta}_n}) \leq \frac{\lambda_n(t)\zeta_n(t,\delta)}{2\delta} \tag{6.90}$$

$$\text{dist}\left(\hat{\theta}_n,\ \theta_n^*\right) \leq \Delta(N_1, N_2)\zeta_n(t,\delta) \tag{6.91}$$

Proof. The (tedious) proof to this exactly follows the proof given in [Van De Geer, 2008], theorem A4, with the same adjustments as in the previous lemma. We omit it here for the sake of conciseness. □

6.3 Conclusion

The final result given proves the consistency of results when solutions are allowed to be spatially ambiguous as defined by the pattern dictating the metric m. In particular, this allows us to control precisely how the solution can vary between the true solution and the recovered solution, allowing us to measure recovery in terms of spatial similarity.

Chapter 7

CONCLUSION

In this work, we have presented a collection of novel results for calculating a new class of sparse estimation problems. These problems arise when we are interest in penalizing the absolute differences between our parameters. While several approaches to do this were discovered independently of us, we extend the state of the art with several novel algorithms that incorporate other sparsity-inducing regularization terms into the problem. These present a powerful approach to these types of problems.

Our work provided solutions to solving three types of problems, given by the following equations. In chapters 2 and 3, we present and develop an algorithm to exactly solve the following problem. The algorithm to solve this relies on a novel bisection type of algorithm based on network flows.

$$\mathbf{u}^*(\lambda) = \underset{\mathbf{u}\in\mathbb{R}^n}{\operatorname{argmin}} \ \|\mathbf{u}-\mathbf{a}\|_2^2 + \lambda \left[\sum_i \xi_i(u_i) + \sum_{i,j} w_{ij}|u_i - u_j| \right], \tag{$\Re_{\mathrm{B}}$}$$

where $\mathbf{a} \in \mathbb{R}^n$ is given, λ is a non-negative regularization parameter, and $w_{ij} \geq 0$ controls the regularization of dependent variables. $\xi_i(u_i)$ is a convex piece-wise linear function, possibly the common L_1 penalty $\xi_i(u_i) = |u_i|$. In chapter 4, we then describe, algorithmically, the structure of the regularization path of this problem.

This algorithm is then extended to general loss functions using proximal operator methods. This allows us to solve the more general form of

$$\mathbf{u}^*(\lambda) = \underset{\mathbf{u}\in\mathbb{R}^n}{\operatorname{argmin}} \ \mathcal{L}(\mathbf{u},\mathbf{y}) + \lambda \left[\sum_i \xi_i(u_i) + \sum_{i,j} w_{ij}|u_i - u_j| \right], \tag{$\Re_{\mathrm{L}}$}$$

where $\mathbf{y}$ is a collection of observed response and $\mathcal{L}(\mathbf{u},\mathbf{y})$ is a smooth, convex log-likelihood term. Finally, this is extended to the case of total variation minimization in chapter 5:

$$u^* = \underset{\mathcal{F}(\Omega)}{\operatorname{argmin}} \ \|u-f\|_2^2 + \lambda\,\mathrm{TV}(u), \tag{$\Re_{\mathrm{TV}}$}$$

where now we seek to estimate a function using an approximation on a lattice. In this case, we give bounds on the approximation in terms of the size of the lattice.

7.1 List of Contributions

The following is a concise list of the novel significant theoretic and algorithmic contributions of this work.

Chapter 2. The Combinatorial Structure of Dependent Problems.

- A theoretical framework connecting recent results in submodular optimization to energy minimization on a graph structure.
- Concise proofs of the correspondence between the minimum norm vector and the solution sets for both network flows and general submodular functions. This result was discovered by Nagano et al. [2011]; we give a concise and simple proof.
- A novel and provably correct algorithm to find the minimizer to $\text{Argmin}_{\mathbf{u}\in\mathbb{R}^n} \|\mathbf{u}-\mathbf{a}\|_2^2 + \sum_{i,j} w_{ij}|u_i - u_j|$ by calculating the minimum norm vector on a specially constructed graph.

Chapter 3. Unary Regularizers and Non-Uniform Size Measures.

- A version of the minimum norm algorithm that gives the full solution path of submodular functions when biased by a non-uniform measure of the solution set.
- An extension of the algorithm in chapter 2 to calculate the weighted minimum norm vector exactly for network flow problems. This algorithm provides the solution to ($\Re_{\text{B}}$) and, using proximal methods, ($\Re_{\text{L}}$).

Chapter 4. Generalized Regularization Paths.

- A complete characterization of the solution path of the result in chapter 3 when the unary terms are arbitrary linear functions. This gives the regularization path over λ in ($\Re_{\text{B}}$).

Chapter 5. Total Variation Minimization.

- A method, with a rigorous bound, to approximate the infinite dimensional version of total variation problem in the form of ($\Re_{\text{B}}$).
- A new algorithm to calculate the complete regularization path of the discretized total variation problem.

Chapter 6. Consistency of Sparse Recovery with Dependent Variables.

- Bounds on the rate of convergence of the LASSO problem to the true optimum using oracle inequalities under a modified distance metric that incorporates correlations in the predictors.

BIBLIOGRAPHY

William K Allard. Total variation regularization for image denoising, i. geometric theory. *SIAM Journal on Mathematical Analysis*, 39(4):1150–1190, 2007.

William K Allard. Total variation regularization for image denoising, ii. examples. *SIAM Journal on Imaging Sciences*, 1(4):400–417, 2008.

William K Allard. Total variation regularization for image denoising, iii. examples. *SIAM Journal on Imaging Sciences*, 2(2):532–568, 2009.

Luigi Ambrosio and Simone Di Marino. Equivalent definitions of bv space and of total variation on metric measure spaces. *preprint*, 2012.

A. Andoni, P. Indyk, and R. Krauthgamer. Earth mover distance over high-dimensional spaces. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 343–352. Society for Industrial and Applied Mathematics, 2008.

Ben Appleton and Hugues Talbot. Globally minimal surfaces by continuous maximal flows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(1):106–118, 2006.

Francis Bach. Convex analysis and optimization with submodular functions: a tutorial. *arXiv preprint arXiv:1010.4207*, 2010a.

Francis Bach. Shaping level sets with submodular functions. *arXiv preprint arXiv:1012.1501*, 2010b.

Francis Bach. Learning with submodular functions: A convex optimization perspective. *arXiv preprint arXiv:1111.6453*, 2011.

Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *arXiv preprint arXiv:1108.0775*, 2011.

Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 2012.

Johnathan M Bardsley. An efficient computational method for total variation-penalized poisson likelihood estimation. *Inverse Problems and Imaging*, 2(2):167–185, 2008.

Johnathan M Bardsley and Aaron Luttman. Total variation-penalized poisson likelihood estimation for ill-posed problems. *Advances in Computational Mathematics*, 31(1-3):35–59, 2009.

Amir Beck and Marc Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *Image Processing, IEEE Transactions on*, 18(11):2419–2434, 2009.

Julien Bect, Laure Blanc-Féraud, Gilles Aubert, and Antonin Chambolle. An ℓ^1-unified variational framework for image restoration. In *Computer Vision-ECCV 2004*, pages 1–13. Springer, 2004.

Mikhail Belkin, Irina Matveeva, and Partha Niyogi. Regularization and semi-supervised learning on large graphs. In *Learning theory*, pages 624–638. Springer, 2004.

Giovanni Bellettini, Vicent Caselles, and Matteo Novaga. The total variation flow in $\mathbb{R}^n$. *Journal of Differential Equations*, 184(2):475–525, 2002.

Julian Besag, Peter Green, David Higdon, and Kerrie Mengersen. Bayesian computation and stochastic systems. *Statistical Science*, pages 3–41, 1995.

P.J. Bickel, B. Li, A.B. Tsybakov, S.A. van de Geer, B. Yu, T. Valdés, C. Rivero, J. Fan, and A. van der Vaart. Regularization in statistics. *Test*, 15(2):271–344, 2006.

C.M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Kevin Bleakley and Jean-Philippe Vert. The group fused lasso for multiple change-point detection. *arXiv preprint arXiv:1106.4199*, 2011.

O. Bousquet. A Bennett concentration inequality and its application to suprema of empirical processes. *Comptes Rendus Mathematique*, 334(6):495–500, 2002.

S.P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge Univ Pr, 2004.

Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 26–33. Ieee, 2003.

Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1124–1137, 2004.

Thomas Brox and Daniel Cremers. On the statistical interpretation of the piecewise smooth mumford-shah functional. In *Scale Space and Variational Methods in Computer Vision*, pages 203–213. Springer, 2007.

F. Bunea, A. Tsybakov, and M. Wegkamp. Aggregation and Sparsity Via 1 Penalized Least Squares. *Learning theory*, pages 379–391, 2006.

F. Bunea, A. Tsybakov, M.H. Wegkamp, et al. Sparsity oracle inequalities for the Lasso. *Electronic Journal of Statistics*, 1:169–194, 2007.

Vicent Caselles, Antonin Chambolle, and Matteo Novaga. Total variation in imaging. *Handbook of Mathematical Methods in Imaging*, pages 1016–1057, 2011.

A. Chambolle and J. Darbon. On total variation minimization and surface evolution using parametric maximum flows. *International journal of computer vision*, 84(3):288–307, 2009.

Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004.

Antonin Chambolle, Vicent Caselles, Daniel Cremers, Matteo Novaga, and Thomas Pock. An introduction to total variation for image analysis. *Theoretical foundations and numerical methods for sparse recovery*, 9:263–340, 2010.

Tony F Chan and Selim Esedoglu. Aspects of total variation regularized l^1 function approximation. *SIAM Journal on Applied Mathematics*, 65(5):1817–1837, 2005.

Tony F Chan and Luminita A Vese. Image segmentation using level sets and the piecewise-constant mumford-shah model. In *Tech. Rep. 0014, Computational Applied Math Group*. Citeseer, 2000.

Xi Chen, Seyoung Kim, Qihang Lin, Jaime G Carbonell, and Eric P Xing. Graph-structured multi-task regression and an efficient optimization method for general fused lasso. *arXiv preprint arXiv:1005.3579*, 2010.

Xi Chen, Qihang Lin, Seyoung Kim, Jaime G Carbonell, and Eric P Xing. Smoothing proximal gradient method for general structured sparse regression. *The Annals of Applied Statistics*, 6(2):719–752, 2012.

Haeran Cho and Piotr Fryzlewicz. Multiscale interpretation of taut string estimation and its connection to unbalanced haar wavelets. *Statistics and computing*, 21(4):671–681, 2011.

Vaek Chvtal. *Linear Programming*. W. H. Freeman and Company, New York, 1983.

T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms*. The MIT press, 2001.

Camille Couprie, Leo Grady, Hugues Talbot, and Laurent Najman. Combinatorial continuous maximum flow. *SIAM Journal on Imaging Sciences*, 4(3):905–930, 2011.

George Bernard Dantzig and Delbert R Fulkerson. On the max flow min cut theorem of networks. 1955.

Jérôme Darbon and Marc Sigelle. A fast and exact algorithm for total variation minimization. In *Pattern recognition and image analysis*, pages 351–359. Springer, 2005.

Jérôme Darbon and Marc 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.

PL Davies and Monika Meise. Approximating data with weighted smoothing splines. *Journal of Nonparametric Statistics*, 20(3):207–228, 2008.

Manfredo Perdigao Do Carmo and Manfredo Perdigao Do Carmo. *Differential geometry of curves and surfaces*, volume 2. Prentice-Hall Englewood Cliffs, 1976.

Lutz Dümbgen and Arne Kovac. Extensions of smoothing via taut strings. *Electronic Journal of Statistics*, 3:41–75, 2009.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–451, 2004.

N. El Zehiry, S. Xu, P. Sahoo, and A. Elmaghraby. Graph cut optimization for the mumford-shah model. In *The Seventh IASTED International Conference on Visualization, Imaging and Image Processing*, pages 182–187. ACTA Press, 2007.

Herbert Federer. Curvature measures. *Transactions of the American Mathematical Society*, 93(3):418–491, 1959.

Herbert Federer. Geometric measure theory, grundlehren der mathematishen wissenschaften 153, 1969.

Wendell H Fleming and Raymond Rishel. An integral formula for total gradient variation. *Archiv der Mathematik*, 11(1):218–222, 1960.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Department of Statistics, Stanford University, Tech. Rep*, 2008a.

Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008b.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

S. Fujishige. *Submodular functions and optimization*, volume 58. Elsevier Science, 2005.

Giorgio Gallo, Michael D Grigoriadis, and Robert E Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18(1):30–55, 1989.

Hao Gao, Hongkai Zhao, et al. Multilevel bioluminescence tomography based on radiative transfer equation part 2: total variation and l1 data fidelity. *Opt. Express*, 18(3):2894–2912, 2010.

E. Giné and V. Koltchinskii. Concentration inequalities and asymptotic results for ratio type empirical processes. *The Annals of Probability*, pages 1143–1216, 2006.

E Giusti. *Minimal surfaces and functions of bounded variation*, volume 80. Birkhauser, 1984.

Donald Goldfarb and Wotao Yin. Parametric maximum flow algorithms for fast total variation minimization. *SIAM Journal on Scientific Computing*, 31(5):3712–3743, 2009.

Bart Goris, Wouter Van den Broek, KJ Batenburg, Hamed Heidari Mezerji, and Sara Bals. Electron tomography based on a total variation minimization reconstruction technique. *Ultramicroscopy*, 113:120–130, 2012.

Martin Grötschel, László Lovász, and Lex Schrijver. Geometric algorithms and combinatorial optimization. *Algorithms and Combinatorics*, 2:1–362, 1993.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Linear Methods for Regression*. Springer, 2009.

Dorit S Hochbaum. The pseudoflow algorithm and the pseudoflow-based simplex for the maximum flow problem. In *Integer Programming and Combinatorial Optimization*, pages 325–337. Springer, 1998.

Dorit S Hochbaum. The pseudoflow algorithm: A new algorithm for the maximum-flow problem. *Operations research*, 56(4):992–1009, 2008.

Dorit S Hochbaum and Sung-Pil Hong. About strongly polynomial time algorithms for quadratic optimization over submodular constraints. *Mathematical programming*, 69(1-3):269–309, 1995.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM (JACM)*, 48(4): 761–777, 2001.

Rishabh Iyer, Stefanie Jegelka, and Jeff A. Bilmes. Fast semidifferential-based submodular function optimization. In *International Conference on Machine Learning (ICML)*, Atlanta, Georgia, 2013.

Stefanie Jegelka and Jeff Bilmes. Cooperative cuts for image segmentation. Technical Report UWEETR-2010-0003, University of Washington, Seattle, 2010.

Stefanie Jegelka and Jeff A. Bilmes. Multi-label cooperative cuts. In *CVPR 2011 Workshop on Inference in Graphical Models with Structured Potentials*, Colorado Springs, CO, June 2011. URL `http://users.cecs.anu.edu.au/~julianm/cvpr2011.html`.

Stefanie Jegelka, Hui Lin, and Jeff A. Bilmes. Fast approximate submodular minimization. In *Neural Information Processing Society (NIPS)*, Granada, Spain, December 2011.

H. Karloff, S. Khot, A. Mehta, and Y. Rabani. On earthmover distance, metric labeling, and 0-extension. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, page 556. ACM, 2006.

Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

Stephen L Keeling, Christian Clason, Michael Hintermüller, Florian Knoll, Antoine Laurain, and Gregory Von Winckel. An image space approach to cartesian based parallel mr imaging with total variation regularization. *Medical Image Analysis*, 16(1):189–200, 2012.

Seyoung Kim, Kyung-Ah Sohn, and Eric P Xing. A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics*, 25(12):i204–i212, 2009.

T. Klein and E. Rio. Concentration around the mean for maxima of empirical processes. *Annals of probability*, pages 1060–1077, 2005.

Roger Koenker and Gilbert Bassett Jr. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50, 1978.

V. Kolmogorov and Y. Boykov. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 564–571. IEEE, 2005.

V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

Vladimir Kolmogorov, Yuri Boykov, and Carsten Rother. Applications of parametric maxflow in computer vision. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

V. Koltchinskii. 2008 Saint Flour Lectures Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems. 2009.

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

M. Ledoux and M. Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer, 1991.

E. Levina and P. Bickel. The earth movers distance is the Mallows distance: Some insights from statistics. In *Proc. ICCV*, volume 2, pages 251–256. Citeseer, 2001.

Hui Lin and Jeff Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *North American chapter of the Association for Computational*

Linguistics/Human Language Technology Conference (NAACL/HLT-2010), Los Angeles, CA, June 2010.

Hui Lin and Jeff Bilmes. Learning mixtures of submodular shells with application to document summarization. In *Uncertainty in Artificial Intelligence (UAI)*, Catalina Island, USA, July 2012. AUAI.

Jun Liu, Lei Yuan, and Jieping Ye. An efficient algorithm for a class of fused lasso problems. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–332. ACM, 2010.

Julien Mairal, Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach. Network flow algorithms for structured sparsity. *arXiv preprint arXiv:1008.5209*, 2010.

Julien Mairal, Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach. Convex and network flow optimization for structured sparsity. *The Journal of Machine Learning Research*, 12:2681–2720, 2011.

P. Massart. About the constants in Talagrand's concentration inequalities for empirical processes. *Annals of Probability*, 28(2):863–884, 2000.

P. Massart. *Concentration inequalities and model selection*. Springer Verlag, 2007.

N. Meinshausen and B. Yu. Lasso-type recovery of sparse representations for high-dimensional data. *Annals of Statistics*, 37(1):246–270, 2009.

Vincent Michel, Alexandre Gramfort, Gaël Varoquaux, Evelyn Eger, and Bertrand Thirion. Total variation regularization for fmri-based prediction of behavior. *Medical Imaging, IEEE Transactions on*, 30(7):1328–1340, 2011.

David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5):577–685, 1989.

Kiyohito Nagano, Yoshinobu Kawahara, and Kazuyuki Aihara. Size-constrained submodular minimization through minimum norm base. In *Proc. ICML*, volume 23, 2011.

Mukund Narasimhan and Jeff Bilmes. Local search for balanced submodular clusterings. In *Twentieth International Joint Conference on Artificial Intelligence (IJCAI07)*, Hyderabad, India, January 2007.

Mukund Narasimhan, Nebojsa Jojic, and Jeff Bilmes. Q-clustering. In *Neural Information Processing Society (NIPS)*, Vancouver, Canada, December 2005.

Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

Yurii Nesterov. Gradient methods for minimizing composite objective function. Technical report, 2007.

James B Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118(2):237–251, 2009.

D. Panchenko. Symmetrization approach to concentration inequalities for empirical processes. *Annals of Probability*, 31(4):2068–2081, 2003.

Thomas Pock, Daniel Cremers, Horst Bischof, and Antonin Chambolle. An algorithm for minimizing the mumford-shah functional. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1133–1140. IEEE, 2009.

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

Wolfgang Ring. Structural properties of solutions to total variation regularization problems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 34(04):799–810, 2000.

Y. Rubner, C. Tomasi, and L.J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

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

Alex Sawatzky, Christoph Brune, Jahn Müller, and Martin Burger. Total variation processing of images with poisson statistics. In *Computer Analysis of Images and Patterns*, pages 533–540. Springer, 2009.

A. Schrijver. *Combinatorial optimization*, volume 24. Springer, 2003.

Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.

Dhruv B Sharma, Howard D Bondell, and Hao Helen Zhang. Consistent group identification and variable selection in regression with correlated predictors. *Journal of Computational and Graphical Statistics*, 22(2):319–340, 2013.

P. Stobbe and A. Krause. Efficient minimization of decomposable submodular functions. *Arxiv preprint arXiv:1010.5511*, 2010.

M. Talagrand. A new look at independence. *The Annals of probability*, 24(1):1–34, 1996.

B. Tarigan and S. van de Geer. Classifiers of support vector machine type with l1 complexity regularization. *Bernoulli*, 12(6):1045–1076, 2006.

Zhen Tian, Xun Jia, Kehong Yuan, Tinsu Pan, and Steve B Jiang. Low-dose ct reconstruction via edge-preserving total variation regularization. *Physics in medicine and biology*, 56(18):5949, 2011.

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

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

S.A. van de Geer. On non-asymptotic bounds for estimation in generalized linear models with highly correlated design. *Lecture Notes-Monograph Series*, pages 121–134, 2007.

S.A. Van De Geer. High-dimensional generalized linear models and the lasso. *Annals of Statistics*, 36(2):614, 2008.

AW Van der Vaart and J.A. Wellner. *Weak convergence and empirical processes*. Springer Verlag, 1996.

Roger J-B Wets. Lipschitz continuity of inf-projections. *Computational Optimization and Applications*, 25(1-3):269–282, 2003.

Thomas Wunderli. Total variation time flow with quantile regression for image restoration. *Journal of Mathematical Analysis and Applications*, 2013.

Sen Yang, Lei Yuan, Ying-Cheng Lai, Xiaotong Shen, Peter Wonka, and Jieping Ye. Feature grouping and selection over an undirected graph. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 922–930. ACM, 2012.

Gui-Bo Ye and Xiaohui Xie. Split bregman method for large scale fused lasso. *Computational Statistics & Data Analysis*, 55(4):1552–1569, 2011.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Hui Zou and Hao Helen Zhang. On the adaptive elastic-net with a diverging number of parameters. *Annals of statistics*, 37(4):1733, 2009.