UNIVERSITY OF CALIFORNIA

Los Angeles

Analysis and Application
# of Graph-Based Semi-Supervised Learning Methods

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mathematics

by

Xiyang Luo

2018

ABSTRACT OF THE DISSERTATION

Analysis and Application
of Graph-Based Semi-Supervised Learning Methods

by

Xiyang Luo
Doctor of Philosophy in Mathematics
University of California, Los Angeles, 2018
Professor Andrea Bertozzi, Chair

In recent years, the need for pattern recognition and data analysis has grown exponentially in various fields of scientific research. My research is centered around graph Laplacian based techniques for image processing and machine learning. Three papers pertaining to this theme will be presented in this thesis.

The first work is an application of graph Laplacian regularization to the problem of convolutional sparse coding. The additional regularization improves the robustness of the sparse representation with respect to noise, and has empirically shown to improve the performance of denoising on several well-known images. Efficient algorithms for computing the eigen-decomposition of the graph Laplacian were also incorporated to the solver for fast implementations of the method.

The second piece of work studies the convergence of the graph Allen-Cahn scheme. A technique inspired by the maximum principle for the heat equation is used to show stability of the convex-splitting numeric scheme. This coupled with techniques from convex optimization allows for a proof of convergence under an a-posteriori condition. The analysis is then generalized to handle spectral trunction, a common method to save computational cost, and also to the case of multi-class classification. In particular, the results for spectral trunction are drastically different from that of the original scheme in the worst case, but does not present itself in practical applications.

The third piece of work combines two fields of research, uncertainty quantification, and semi-supervised learning on graphs. The work presents a unified Bayesian framework that incorporates most previous methods for graph-based semi-supervised learning. A Bayesian framework allows for the computation of uncertainty for certain quantities under the posterior distribution. We show via solid numerical evidence that for a few carefully designed quantities, the expectations computed under the posterior yields meaningful notions of uncertainty for the classification problem. Efficient numerical methods were also devised to make possible the evaluation of these quantities for large scale graphs.

The dissertation of Xiyang Luo is approved.

Luminita Aura Vese

Christopher R. Anderson

Stanley J. Osher

Andrea Bertozzi, Committee Chair

University of California, Los Angeles

2018

To my family, mentors, friends, and collaborators.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor, Andrea Bertozzi. She has provided me with strong guidance and support throughout my Ph.D. career. Her wide range of expertise as well as her attitude towards research has truly been inspirational for me and other students in my group. I would also like to thank other members of my doctoral committee, Wotao Yin, Chris Anderson, and Luminita Vese, for being a part of my dissertation committee and for their guidance through my ph.D. life.

Second of all, I would like to express my gratitude to all my collaborators. I would especially like to thank Andrew Stuart from Caltech for his collaboration and guidance, without which the second part of my thesis would not have come to fruition. I am grateful to have worked with Brendt Wohlberg on my first paper during my graduate studies. I would also like to thank Bao Wang for working together on new branches of machine learning during the last part of my Ph.D. research.

Finally, I am grateful for all the support I have had from my friends and family during my graduate life. I would like to thank Zhaoyi Meng, Fangbo Zhang, Zach Boyd, Yuming Zhang, Chuyuan Fu for making my time in graduate school memorable. I would also like to thank my parents, Ancheng Luo and Yan Zhou for raising me to be the person I am. Finally, I would like to thank my loving wife Shuyun Chen. For without her love and companionship, the journey would not have been the same.

This thesis contains content from three publications listed on page xv. Chapter 3 includes the paper *Convolutional Laplacian Sparse Coding.* [LWum]. My contributions are algorithm formulation and numerical experiments. Chapter 4 includes the paper *Convergence of the Graph Allen-Cahn Equation* [LBed]. I contributed most of the proofs and numerical experiments. Chapter 5 contains the paper *Uncertainty Quantification in the Classification of High Dimensional Data* [BLSed]. My contribution includes the design of the model, designing and implementing the algorithm, and the numerical experiments.

This work is supported by NSF grants DMS-1417674, DMS-1045536, DMS-1118971, UC Lab Fees Research grant 12-LR-236660, ONR grant N00014-16-1-2119, UC Lab Fees Research grant 12-LR-236660, and the UCLA Dissertation Year Fellowship.

VITA

Education

2009 - 2013	B.S. Mathematics, Zhejiang University
2013 - 2018	Ph.D. Candidate, Department of Mathematics, UCLA

Experience

2015.6-2015.9	Visiting Researcher, LANL, Los Alamos
2017.6-2017.9	Intern, Google Research, Mountainview

PUBLICATIONS

Convolutional Laplacian Sparse Coding by X. Luo, B. Wohlberg; 2016 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI).

Convergence of the Graph Allen-Cahn Scheme by X.Luo, A. L. Bertozzi; Journal of Statistical Physics.

Uncertainty Quantification in Graph-Based Classification of High Dimensional Data by A. L. Bertozzi, X.Luo, A. M. Stuart, K. Zygalakis; SIAM journal on Uncertainty Quantification, in presss.

CHAPTER 1

Introduction

1.1 Convolutional Laplacian Sparse Coding

In Chapter 3, we present the Convolutional Laplacian Sparse Coding (CLSC) model. Convolutional sparse coding [ZKT10] is a variant of sparse coding. In classical sparse coding, a signal is decomposed into a sum of products between a set of real-valued coefficients and a corresponding set of dictionary filters. For convolutional sparse coding, we replace the coefficients by coefficient maps with the same size as the original signal, and convolve with the dictionary filters instead.

We introduce the concept of discrete convolution between a coefficient map $x \in \mathbb{R}^N$ and filter $d \in \mathbb{R}^d$.

$$[d * x]_i := \sum_k x_{i-k} d_k.$$

Define the vector $p-$ norm as

$$\|x\|_p = \left(\sum_i x_i^p \right)^{1/p}.$$

One of the most prominent formulations of this problem is Convolutional Basis Pursuit DeNoising (CBPDN)

$$\underset{\{\mathbf{x}_m\}}{\arg\min} \frac{1}{2} \Big\| \sum_m \mathbf{d}_m * \mathbf{x}_m - \mathbf{s} \Big\|_2^2 + \lambda \sum_m \|\mathbf{x}_m\|_1 \ , \tag{1.1}$$

where $\{\mathbf{d}_m\}$ is a set of M dictionary filters, $*$ denotes convolution, and $\{\mathbf{x}_m\}$ is a set of coefficient maps. Recent fast algorithms [Woh14] for solving this problem have begun to make it a viable approach for a wider variety of applications.

Convolutional sparse representations have a number of advantages over the standard approach of independently sparse coding of overlapping image patches. Convolutional representations are inherently invariant to translation. Moreover, CBPDN provides a single-valued representation that is optimal over the entire image instead of just locally within each patch. There are, however, also some challenges to using this representation, aside from the computational cost. One of these is the tendency for the set of coefficient maps to be sparse both down the stack of maps at each pixel location, as well as spatially within each map. This particular property of spatial sparsity is undesirable in some applications, including denoising of Gaussian white noise, where the spatial averaging of independent pixel estimates obtained from the standard patch-based method is beneficial, or in dictionary learning where high spatial sparsity reduces the number of patches in the training images that play a role in forming the dictionary.

The Convolutional Laplacian Sparse Coding (CLSC) model represents an attempt to remedy this weakness by incorporating a non-local regularization that reduces the spatial sparsity in an appropriate way, while retaining the local sparsity of the representation at each pixel location. Non-local methods have shown to improve white noise denoising among other tasks in many classical applications of dictionary learning. This method therefore could be seen as a natural extension of this approach to a convolutional setting.

In particular, this model can be considered as a convolutional variant of the previously-proposed Laplacian Sparse Coding method [GTC10], which has been applied to image classification tasks [GTC10, GTC13] as well as image restoration tasks [ZBC11, DLZ11]. The key difference between the proposed approach and the patch based Laplacian Sparse Coding in [GTC10] is that the sparse code is learned over *every* single patch and *jointly* over the entire image, due to the nature of the convolutional model. Thus unlike [GTC10, GTC13], there is no need to use the SIFT local descriptor to pre-define a set of patches to learn on, and there is no need for patch averaging to resolve the multi-valued estimation as in [ZBC11].

The proposed model is to augment Eq. (3.1) with the graph Dirichlet energy $\sum_m \langle \mathbf{x}_m, L\mathbf{x}_m \rangle$, where L is the graph Laplacian [Von07] of the image non-local graph [MBP09]. Each image patch corresponds to a vertex of the graph, and the weights w_{ij} between vertices represent the similarity between the corresponding image patches, typically computed as

$$w_{ij} = \exp\left(-d_{ij}^2/\tau\right) \ , \tag{1.2}$$

where d_{ij} is some metric (typical choices are Euclidean or Cosine) between an image patch centered at pixel i and that at pixel j , and τ controls the scaling of the metric. Given the weight matrix $W = (w_{ij})$, the graph Laplacian L is defined as $L = D - W$, where D is the diagonal matrix $D_{ii} = \sum_{i \neq j} w_{ij}$. The model is motivated by the non-local smoothing properties of the Dirichlet energy, which are apparent from the equation

$$\langle u, Lu \rangle = \sum_{\alpha,\beta \in V} w_{\alpha\beta}(u_\alpha - u_\beta)^2 \ . \tag{1.3}$$

Here α, β range through all vertices on the graph, and u is any real-valued function defined on the graph. Since $w_{\alpha\beta}$ is smaller if the vertices α, β are more similar, the Dirichlet energy will be small if similar vertices have similar u values.

In this context, the vertices α are image patches indexed by their spatial location (i, j), and the u corresponds to the sparse coefficients $\mathbf{x}_m$. Thus by the analysis above, the regularizer is an explicit penalty to force similar image patches to have similar sparse representations. In practice, we actually use the normalized Laplacian $L_s = I - D^{-1/2}WD^{-1/2}$ [Von07], since it handles outliers better and is the more common choice for non-local image graphs. However, the motivation remains the same and we will not make a distinction from here on.

Using this model, we are able to improve resistance to both white noise and salt and pepper noise compared to the standard convolutional sparse coding model. This is demonstrated via better PSNR scores on a selection of images, and also dictionary filters learned on the Flickr dataset [MST10].

1.2 Convergence of the Graph Allen-Cahn Equation

In Chapter 4, we present the study of the convergence of the graph Allen-Cahn scheme.

Diffuse interface model has been used widely in material science to model the free boundary of interfaces [TC94, CN94]. One of these models is the Allen-Cahn equation [CN94], the L^2 gradient flow of the Ginzburg-Landau functional which is defined as:

$$GL(u) = \frac{\epsilon}{2} \int |\nabla u|^2 + \frac{1}{\epsilon} \int W(u(x))dx. \tag{1.4}$$

Another commonly used model is the Cahn-Hilliard equation [BLO97, BHS09]. The diffuse interface models can often be used as a proxy for TV minimization since the Γ-limit of the Ginzburg-Landau functional is shown to be the TV semi-norm [KS89].

In [BF12], the Allen-Cahn equation has been generalized to weighted graphs, establishing a connection between the classical diffuse interface model and the graph cut problem. Following this line of work, a series of new algorithms were developed for semi-supervised and unsupervised classification problems on weighted graphs [MKB13, HLP13], applying techniques for TV minimization to the setting of weighted graphs.

The graph cut problem originated in computer science for the purpose of partitioning nodes on a graph [BVZ01]. It is tightly related to statistical physics due to its connections with Markov random fields, and spin systems. In particular, the maximum a posterior estimation of the Ising model can be formulated in terms of a graph cut problem [GPS89]. The results also generalizes to multiclass graph cut by extending to the generalized Potts model [BVZ98]. This idea has been applied to computer vision for the task of image segmentation, and image denoising [KT07]. Efficient solutions to the graph cut problem has also lead to efficient inference in certain types of Markov random fields [FGB11], in comparison to other techniques such as belief propagation [Yed11, Zha12], and semi-analytic methods [FD16].

The key observation linking the diffuse interface model with the graph cut problem is that the TV semi-norm, when suitably generalized to weighted graphs, coincides with the graph cut functional for discrete valued functions on graphs [GGO14]. Hence techniques for TV minimization can also be applied to solve the graph cut problem. This was made rigorous by the result that the graph Ginzburg-Landau functional Γ converges to the graph TV functional [GGO14].

In Chapter 4, we present the conditions for the discrete graph Allen-Cahn scheme to converge. The main conclusion presented is that the stepsize dt to ensure convergence is independent of the graph size N. This has the practical implication that the numerical scheme scales well with respect to the size of the graph.

The two central idea to this technique are the following. First, we observe that for the unnormalized graph Laplacian, the maximum principle for heat equation holds on the graph. This provides us with an L^∞ estimate on the iterates u^k. The second observation is that the semi-implicit discretization of the Allen-Cahn Equation coincides with the forward-backward splitting method [BV09] for minimizing a sum of convex functions. Combining existing technique for the foward-backward split together with the maximum principle gives as a convergence result for the scheme.

The same convergence analysis is also carried out for the scheme under spectral truncation using similar techniques. A surprising conclusion was that the stepsize needs to be scaled $O(N^{-1})$ in order to ensure convergence uniformly across *all* graphs. This estimate is proved to be sharp by a carefully constructed series of graph. The result is also generalized for multiclass classification.

In addition to the theoretical analysis, a variety of numerical examples are also presented to complement the theory.

1.3 Uncertainty Quantification for Graph-based Semi-supervised Learning

In Chapter 5, we present a work in which Uncertainty Quantification (UQ) is applied to the problem of graph-based semi-supervised learning. The starting point of the work presented revolves around the following objective function which incorporates many graph-based semi-supervised algorithms

$$\mathsf{J}(w) = \frac{1}{2}\langle w, Lw\rangle + \Phi(w; y),$$

where $\Phi(w; y)$ encompasses the information on the labelled nodes. Minimizing this objective function gives an approach towards solving the classification problem. However, in many classification problems, it is also important to assign a reliable uncertainty measure to the labels predicted by the model. Therefore, it is natural to consider a Bayesian interpretation of the optimization problem.

A unified Bayesian framework is presented in this chapter, which incorporates a variety of previous methods for graph semi-supervised learning. In addition, the value of the additional information gained is presented in detail. In particular, two metrics are presented that quantify 1) the uncertainty of the classification prediction for individual labels, 2) the aggregated uncertainty for the learning problem as a whole. Results from various numerical examples aim to demonstrate that the metrics provide us with additional insights to the classification problem.

From a computational point of view, it is essential to efficiently model the posterior densities $\mathbb{P}(w)$. Therefore in addition to the model, we present several efficient numerical algorithms to efficiently infer the posterior distribution. The first technique introduced is the pCN algorithm, an Markov Chain Monte Carlo (MCMC) method which, based on analogies with its use for PDE-based inverse problems [CRS13], has the potential to sample the posterior distribution in a number of steps which is independent of the number of graph nodes. Secondly, we introduce approximations exploiting the empirical properties of the spectrum of the graph Laplacian, generalizing methods used in the optimization context in [BF12], allowing for computations at each MCMC step which scale well with respect to the number of graph nodes.

CHAPTER 2

Background

2.1 Graph Semi-supervised Learning

Graphs are a powerful method of representing relational data [HN04], and could be applied to a variety of problems such as clustering [Von07], item recommendation [KSO16] and multivariate time series prediction [WLZ18]. Central to many graph-based learning methods is the graph Laplacian. In this section, we give a brief introduction to the graph Laplacian, and its application to semi-supervised learning.

Let G be a weighted graph on the set of nodes $Z = \{1, \dots, N\}$. The weights of the graph can be written in the form of a matrix $W = (w_{ij})$, where w_{ij} often characterizes the similarities or connections between two nodes i and j. Given a weight matrix W, one can construct three different kinds of graph Laplacians:

$$L^u = D - W \qquad \text{Unnormalized Laplacian,} \tag{2.1}$$

$$L^s = I - D^{-1/2} W D^{-1/2} \qquad \text{Symmetric Laplacian,} \tag{2.2}$$

$$L^{rw} = I - D^{-1} W \qquad \text{Random Walk Laplacian,} \tag{2.3}$$

where D is the diagonal matrix $d_{ii} = \sum_i w_{ij}$.

All three Laplacian matrices are commonly used in graph learning problems. In particular, the graph Dirichelet energy for the unnormalized graph Laplacian has the following property as shown in equation (2.4).

$$\frac{1}{2}\langle u, L^u u\rangle = \frac{1}{2}\sum_{ij} w_{ij}(u(i) - u(j))^2. \tag{2.4}$$

Similar to the classical Dirichelet energy, the graph Dirichelet energy penalizes similar nodes (i.e. pairs such that w_{ij} is large) from having different function values, bringing a notion of "smoothness" for functions defined on the graph.

One of the most common ways of obtaining a graph Laplacian is via a kernel distance function on a set of feature vectors in $\mathbb{R}^d$. Given a set of feature vectors

$$X = \{x(1), \ldots, x(j), \ldots, x(N)\},$$

where $x(j)$ is an element of $\mathbb{R}^d$, we can construct a weighted graph via the following

$$w_{ij} = h(\|x_i - x_j\|), \tag{2.5}$$

where h is some kernel such as the exponential function $\exp(-\frac{x^2}{\sigma})$. This representation effectively removes the dimensionality of the original data, while only keepin the similarity information between different instances of the feature set.

2.1.1 Continuum Limits of Graph Laplacian

The study of discrete objects/algorithm from a continuum point of view is an interesting. For example, the recent formulation of Deep Residual Networks (ResNets) [HZR16] to a dynamical system [CMH17] allows for more principled methods of designing networks and activation functions [WLL18]. The same line of thinking could be applied to the study of graph Laplacians. If we assume the features $\{x_i\}$ are sampled i.i.d. from some a priori distribution ρ is non-vanishing and compactly supported on a bounded domain $D \subset \mathbb{R}^N$, then for a fixed distance metric, the graph converges to a non-local operator on the Euclidean domain [VBB08]. Furthermore, if the distance metric is scaled in a proper manner with respect to the number of sample size, then the graph Laplacian converges to an elliptic diffusion operator in a sense made precise in [TS16]. We illustrate this effect in Figure 2.1.

Namely, Let $h_\epsilon(x) = \frac{1}{\epsilon^n} h\left(\frac{x}{\epsilon}\right)$. Define the sequence of graph weights w_{ij} by the following scaling,

$$w_{ij} = \frac{1}{N^2 \epsilon_N^2} h_{\epsilon_N}\left(\|x_i - x_j\|\right), \tag{2.6}$$

where ϵ_N satisfies $\epsilon_N \ll \frac{log(N)^p}{N^{1/m}}$, and $p = 1/d$ if $d \geq 3$, $p = 3/4$ if $d = 2$. Then the

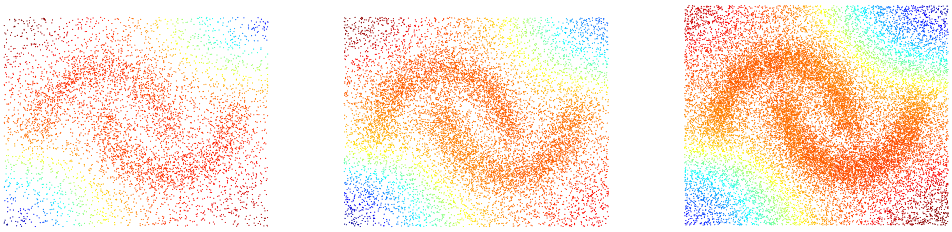


Figure 2.1: Fourth eigenvector of the unnormalized Graph Laplacian for graphs constructed from i.i.d. samples $\{x_i\}$ with $N = 6000$, $N = 12000$, $N = 24000$.

unnormalized graph Laplacian L_N converges to the continuum operator

$$\mathcal{L} = -\frac{1}{\rho^2}\mathrm{div}(\nabla u). \tag{2.7}$$

The convergence implies the following:

- For fixed k, the eigenvalues $\lambda_N^{(k)} \to \lambda^{(k)}$ almost surely as $N \to \infty$.
- For fixed k, the discrete eigenvectors are close to the continuous eigenvectors restricted to the sample points $\{x(i)\}$.

Numerically, this implies that under such a scaling, the eigenvectors and eigenvalues converges to its continuum limits under large N. Therefore, algorithms that are well-defined on the continuum problem have the potential to scale independently of dimension.

2.1.2 Semi-supervised Learning

There are numerous methods designed to leverage unlabelled data into the prediction model. The graph Laplacian-based methods aim to do exactly so by utilizing the similarity information between labelled and unlabelled instance. The fundamental assumption for all these models is that the signals to be predicted are "smooth" along the graph, i.e., similar nodes should have similar labels.

At the heart of many graph-based semi-supervised methods is the following functional

$$\mathsf{J}(u) = \frac{1}{2}\langle u, Lu\rangle + \Phi(u; y).$$

This is a typical fidelity + regularization framework where the term $\Phi(u; y)$ measures the fidelity of u with respect to observed label y, and $\frac{1}{2}\langle u, Lu\rangle$ provides the regularization. From an optimization point of view, the graph Laplacian provides an augmented loss function better suited to the semi-supervised learning task. This approach of designing loss function more suitable to the task at hand could be found in [BCM12, YFR07], and has more recently been applied to neural networks [MLE18, ESM17].

Below we also present a brief overview of the development of semi-supervised methods on graphs. The review [Zhu] provides an excellent overview of this topic up to 2007. A more recent review of graph semi-supervised methods could be found in [BF16]. Early graph-based learning were based on combinatorial approaches. Blum et al. posed the binary semi-supervised classification problem as a Markov random field (MRF) over the discrete state space of binary labels, the MAP estimation of which can be solved using a graph-cut algorithm in polynomial time [Zhu] . In general, inference for multi-label discrete MRFs is intractable [DJP92]. However, several approximate algorithms exist for the multi-label case [BVZ01, BVZ98, Mad10], and have been applied to many imaging tasks [BJ01, BKY96, Li12].

A different line of work is based on using the affinity function on the edges to define a real-valued function u on the nodes of the graph. The Dirichlet energy $\mathsf{J}_0(u) := \frac{1}{2}\langle u, Pu\rangle$, with P proportional to the graph Laplacian formed from the affinities on the edges, plays a central role. A key conceptual issue in the graph-based approach is then to connect the labels, which are discrete, to this real-valued function. Strategies to link the discrete and continuous data then constitute different modeling assumptions. The line of work initiated in [ZGL03] makes the assumption that the labels are also real-valued and take the real values ± 1, linking them directly to the real-valued function on the nodes of the graph. This may be thought of as a continuum relaxation of the discrete state space MRF in [BC01]. The basic method is to minimize $\mathsf{J}_0(u)$ subject to the hard constraint that u agrees with the label values. Alternatively this constraint may be relaxed to a soft additional penalty term added to $\mathsf{J}_0(w)$. These methods are a form of krigging, or Gaussian process regression [Wah90, WR96], on a graph. A Bayesian interpretation of the approach in [ZGL03] is given in [ZLG03] with further applications in hyper-parameter tuning given in [KQA05]. The Laplacian based approach has since been generalized in [ZBL04, BNS06, TC09, SB11, LP11]; in particular this line of work developed to study the transductive problem of assigning predictions to data points off the graph. A formal framework for graph-based regularization, using $\mathsf{J}_0(u)$, can be found in [BMN04, SBN06]. We also mention related methodologies such as the support vector machine (SVM) [Bis07] and robust convex minimization methods [AFS16a, AFS16b] which may be based around minimization of $\mathsf{J}_0(u)$ with an additional soft penalty term; however since these do not have a Bayesian interpretation we do not consider them here. Other forms of regularization have been considered such as the graph wavelet regularization [SFV11, HVG11].

The underlying assumption in much of the work described in the previous paragraph is that the labels are real-valued. An arguably more natural modelling assumption is that there is a link function, such as the sign function, connecting a real-valued function on the graph nodes with the labels via thresholding. This way of thinking underlies the probit approach [WR96] and the Bayesian level set method [ILS15, DIS16]. Lying between the approaches initiated by [ZGL03] and those based on thesholding are the methods based on optimization over real-valued variables which are penalized from taking values far from ± 1. This idea was introduced in the work of Bertozzi et al. [BF12, VB12]. It is based on a Ginzburg-Landau relaxation of the discrete Total Variation (TV) functional, which coincides with the graph cut energy. This was generalized to multiclass classification in [GMB14]. Following this line of work, several new algorithms were developed for semi-supervised and unsupervised classification problems on weighted graphs [HSB15, MKB13, OWO14, LB17, MBC16]. These methods have also lead to various practical applications [GHM17] such as network science [BBT17] and image segmentation [Men18, MMK17, MKH16].

2.1.3 Low Rank Approximations

For graphs with a large number of nodes N, it is sometimes prohibitively costly to directly perform computations on the graph Laplacian L, as is required in theory for in many of the algorithms presented in this thesis.

A method that is frequently used in classification tasks is to restrict the support of u to the eigenspace spanned by the first ℓ eigenvectors with the smallest non-zero eigenvalues of L. This is justified by the low rank properties of the weight matrix W for many of the graphs risen from practical applications. Moreover, the geometric information associated with the graph is mostly contained in the leading eigenvectors of the graph Laplacian. The second eigenvector, named the Fiedler vector, approximates the solution to the problem of the normalized cut on the graph. Spectral clustering [Von07] takes advantage of this phenomenon and produces a non-linear embedding of the graph to the components of its leading eigenvectors.

In many applications, the full weight matrix of the graph is so large that exact computations of the eigeivectors are computationally too expensive. This motivates the search for efficient methods of computing the leading eigevectors of the graph Laplacian L. A popular technique for approximating the leading eigenvectors of the graph Laplacian is the Nyström method [FBC04], and has been successfully applied to many classification problems on graphs. We describe the algorithm for Nyström extension in Algorithm 1 for the *symmetric* graph Laplacian as described in [BF16] below.

Another commonly used technique is to sparsify the graph via some approximate nearest neighbor search method [CFS09, HAS11]. There are many open source software, such as VLFeat [VF10] and PyFlann [ML09] that contains fast and robust implementations of these algorithms. Once the graph is constructed, efficient methods such as the Rayleigh-Chebyshev method [And10] designed for Hermitian sparse matrices could be used to compute the leading eigenvectors of the graph Laplacian.

2.2 Uncertainty Quantification

Uncertainty quantification (UQ) [Stu10] is a mathematical framework for inverse problems, which provides a tool for determining an unknown field from a set of finite measurements. In the most general of settings, let X, R be Banach spaces, and G be a forward map $G : X \to R$. The forward map typically takes $u \in X$, the input data to a particular Partial Differential

Algorithm 1 Nyström Extension for Symmetric Laplacians

```
Input: A set of features $\{x_i\}$.
Output: $K$ Approximate eigenvectors and eigenvalues $\{\phi_i\}$, $\lambda_i$.
Define: $\alpha(u, v) = \min\{1, \exp(\Phi(u; y) - \Phi(v; y)\}$.
Partition $Z = X \cup Y$, where $X$ contains $M$ randomly sampled nodes from the set $Z$.
Compute the adjacency matrix $W_{XX}$ and $W_{XY}$.
$d_X = W_{XX}\mathbf{1}_L + W_{XY}\mathbf{1}_{N-L}$.
$d_Y = W_{YX}\mathbf{1}_L + W_{YX}W_{XY}^{-1}\mathbf{1}_{N-L}$.
$s_X = \sqrt{d_X}$, and $s_Y = \sqrt{d_Y}$.
$W_{XX} = W_{XX}./(s_X s_X^T)$, and $W_{XY} = W_{XY}./(s_X s_Y^T)$.
Apply SVD to matrix $W_{XX}$, obtain $B_X \Gamma B_X^T = W_{XX}$, and compute $S = W_{XX}^{-1/2}$.
$Q = W_{XX} + SW_{XY}W_{YX}, S$ and compute SVD $A\Sigma A^T = Q.S$
Compute $\Phi = (B_X\Gamma^{1/2}, W_{YX}B_X\Gamma^{-1/2})^T B_X^T A\Sigma^{-1/2}$. The columns of $\Phi$ are the approximate eigenvectors.
$\lambda_i = 1 - \sigma_i$, where $\sigma_i$ is the diagonal entries of the matrix $\Sigma$.
```

Equation (PDE) problem, and outputs the solution $r \in R$. Observations (such as evaluation on a finite set of points) are often made on the output solution r, and we denote the map from the solution to the space of observation as $\mathcal{Q} : R \to Q$, with Q again being some Banach space. Bayesian inverse problems are concerned with the determination of the randomness in u from the observations in Q. To make this more precise, we are given a *prior* measure μ on X on the space of input functions. Let $F = Q \circ G$, and $y = F(u) + \eta$, where η is the observational noise. The fundamental quantity of interest is the posterior distribution $\mathbb{P}(u|y)$. Uncertainty quantification on a computational level involves with computing expectations under this specific posterior distribution.

Uncertainty quantification is most widely applied to subsurface geophysics and atmospheric and ocean sciences. The observational maps correspond to physical measurements of some quantity of interest, and the input data u is often some field of interest that is not directly measureable. More recently, Bayesian inverse problems has also been applied to image processing tasks such as de-noising and in-painting.

There is also a very natural way of associating the UQ framework with semi-supervised classification problems on graphs. Let G be a graph with nodes $Z = \{1, \dots, N\}$. We define $u \in \mathbb{R}^N$ as a real-valued function on u. We define the forward observational map via thresholding, i.e., $y = \text{Sign}(u + \eta)$, where η denotes the observational noise. The prior for u can be chosen as

$$\mu = N(0, C),$$

where $C = (\alpha + L)^{-1}$, $\alpha > 0$ and L is the graph Laplacian. This prior is analogous to the Gaussian prior measure defined on the continuous domain for a range of classical inverse problems. In fact, the connection can be made rigorous in the large graph limit as in Section 4.1. Other priors instead of the Laplacian can also be considered. For example, one could consider Besov type priors, or taking a power of the graph Laplacian.

The idea of low rank approximations could also be applied in the context of UQ. This is made clear by the Karhunen-Loeve (KL) expansion for any Gaussian prior. Namely, for any $u \sim \xi N(0, C)$

$$u = \sum_{i=1}^{N} \lambda_i^{1/2} \psi_i \xi_i, \tag{2.8}$$

where ψ_i, λ_i are the eigenvectors and eigenvalues of the covariance matrix C, and $\xi_i \sim N(0, 1)$ independently.

Through the KL expansion, a truncated version of the Gaussian prior could be obtained by simply truncating the terms in Eq.2.8, i.e., setting $y = \sum_{i=1}^{K} \lambda_i^{1/2} \psi_i \xi_i$. The truncation level K should be treated as another hyper-parameter of the model, and ideally tuned with respect to the data. Empirically, we have found that only very few eigenvectors compared to the number of nodes are needed, and that setting $K \ll N$ acts as a necessary regularizer for the model to perform well.

CHAPTER 3

Graph Laplacian Regularization on Convolutional Sparse Coding

3.1 Background

In this chapter, we study the application of graph-Laplacian based regularization to the problem of convolutional sparse coding. Below, we present the convlutional sparse coding model:

$$\underset{\{\mathbf{x}_m\}}{\arg\min} \frac{1}{2}\Big\|\sum_m \mathbf{d}_m * \mathbf{x}_m - \mathbf{s}\Big\|_2^2 + \lambda \sum_m \|\mathbf{x}_m\|_1 \ , \tag{3.1}$$

where $\{\mathbf{d}_m\}$ is a set of M dictionary filters, $*$ denotes convolution, and $\{\mathbf{x}_m\}$ is a set of coefficient maps. Recent fast algorithms [Woh14] for solving this problem have begun to make it a viable approach for a wider variety of applications.

While having multiple advantages, the method has many challenges aside from the computational cost. One such problem is the sensitivity to noise in the input. The extra representability obtained from the convolutional model results in a tendency to overfit to noise. Therefore, it is natural to consider additional regularizations to the original model. The method presented in this section adds a non-local Laplacian regularization to the original functional. The non-local functional encourages smoothing between similar image patches for the learned sparse coefficients, resulting in more robustness towards noise.

$$\begin{aligned}\underset{\{\mathbf{x}_m\}}{\arg\min} \frac{1}{2}\Big\|\sum_m \mathbf{d}_m * \mathbf{x}_m - \mathbf{s}\Big\|_2^2 &+ \lambda \sum_m \|\mathbf{x}_m\|_1 \\ &+ \frac{\mu}{2}\sum_m \langle \mathbf{x}_m, L\mathbf{x}_m\rangle \ . \end{aligned} \tag{3.2}$$

We also give a brief introduction to the classical Alternating Direction Method of Multipliers (ADMM) method [BPC10]. ADMM is an algorithm designed for optimizing objectives of the form

$$\min\{f(x)+g(z)\} \qquad \text{s.t. } Ax+Bz=c$$

The algorithm preserves the robustness of the dual gradient descent algorithm, while having superior convergence properties. The exact algorithm consists of alternatively minimizing the Augmented Lagrangian

$$L_\rho(x,z,y)=f(x)+g(z)+y^T(Ax+Bz-c)+(\rho/2)\|Ax+Bz-c\|^2.$$

The ADMM algorithm consists of iterations

$$\begin{aligned} x^{k+1} &= \arg\min_x L_\rho(x,z^k,y^k),\\ z^{k+1} &= \arg\min_z L_\rho(x^{k+1},z,y^k),\\ y^{k+1} &= y^k+\rho(Ax^{k+1}+Bz^{k+1}+c). \end{aligned}$$

3.2 Algorithm

We present two alternative algorithms for solving Eq. (3.2) are both based on the Alternating Direction Method of Multipliers (ADMM) [BPC10] framework. Their differences correspond to whether we perform an additional splitting in $\langle \mathbf{x}_m, L\mathbf{x}_m\rangle$, or include it in the ℓ^1 subproblem.

3.2.1 ADMM Double-Split

In this approach, we perform an additional splitting to give

$$\begin{aligned} &\arg\min_{\{\mathbf{x}_m\}} \frac{1}{2}\Big\|\sum_m \mathbf{d}_m * \mathbf{x}_m - \mathbf{s}\Big\|_2^2 + \lambda\sum_m \|\mathbf{y}_m\|_1 + \frac{\mu}{2}\sum_m \langle \mathbf{z}_m, L\mathbf{z}_m\rangle \\ &\qquad \text{s.t. } \mathbf{x}_m=\mathbf{y}_m, \quad \mathbf{x}_m=\mathbf{z}_m \ . \end{aligned} \tag{3.3}$$

The corresponding ADMM primal updates are

$$\begin{aligned}\{\mathbf{x}_m\}^{(j+1)} &= \underset{\{\mathbf{x}_m\}}{\arg\min}\, \frac{1}{2}\Big\|\sum_m \mathbf{d}_m * \mathbf{x}_m - \mathbf{s}\Big\|_2^2 + \\ &\rho\sum_m \Big\|\mathbf{x}_m - \frac{1}{2}\left(\mathbf{u}_m^{(j)} + \mathbf{y}_m^{(j)} + \mathbf{v}_m^{(j)} + \mathbf{z}_m^{(j)}\right)\Big\|_2^2, \end{aligned} \tag{3.4}$$

$$\begin{aligned}\{\mathbf{y}_m\}^{(j+1)} &= \underset{\{\mathbf{y}_m\}}{\arg\min}\, \lambda\sum_m \|\mathbf{y}_m\|_1 + \\ &\frac{\rho}{2}\sum_m \left\|\mathbf{x}_m^{(j+1)} - (\mathbf{y}_m + \mathbf{u}_m^{(j)})\right\|_2^2, \end{aligned} \tag{3.5}$$

$$\begin{aligned}\{\mathbf{z}_m\}^{(j+1)} &= \underset{\{\mathbf{z}_m\}}{\arg\min}\, \frac{\mu}{2}\sum_m \langle \mathbf{z}_m, L\mathbf{z}_m\rangle + \\ &\frac{\rho}{2}\sum_m \left\|\mathbf{z}_m - (\mathbf{x}_m^{(j+1)} + \mathbf{v}_m^{(j)})\right\|_2^2 \ . \end{aligned} \tag{3.6}$$

The $\mathbf{x}_m$ and $\mathbf{y}_m$ updates are the same as in the standard convolutional learning case, and can be efficiently solved in the Fourier domain and by soft thresholding respectively, as in [Woh14]. The $\mathbf{z}_m$ update involves solving a linear system.

It is worth emphasizing that, despite the double splitting, this algorithm can be expressed in the standard ADMM form if the two split variables are appropriately combined in block form by defining the matrix $\mathbf{A}$ mapping $\mathbf{x} \mapsto (\mathbf{x}, \mathbf{x})^T$ and $\mathbf{u} = (\mathbf{y}, \mathbf{z})^T$, and imposing the constraint $\mathbf{u}_m = \mathbf{A}\mathbf{x}_m$.

3.2.2 ADMM Single-Split

Instead of performing an additional splitting, we can also group the Laplacian term together with the ℓ^1 term and solve an $\ell^2 + \ell^1$ minimization as a sub-problem. The resulting iterations are

$$\begin{aligned}\{\mathbf{x}_m\}^{(j+1)} &= \underset{\{\mathbf{x}_m\}}{\arg\min} \frac{1}{2}\Big\|\sum_m \mathbf{d}_m * \mathbf{x}_m - \mathbf{s}\Big\|_2^2 +, \\ &\frac{\rho}{2}\sum_m \left\|\mathbf{x}_m - \mathbf{y}_m^{(j)} - \mathbf{u}_m^{(j)}\right\|_2^2, \end{aligned} \tag{3.7}$$

$$\begin{aligned}\{\mathbf{y}_m\}^{(j+1)} &= \underset{\{\mathbf{y}_m\}}{\arg\min}\, \lambda\sum_m \|\mathbf{y}_m\|_1 + \frac{\mu}{2}\sum_m \langle \mathbf{y}_m, L\mathbf{y}_m\rangle \\ &\frac{\rho}{2}\sum_m \left\|\mathbf{x}_m^{(j+1)} - \mathbf{y}_m - \mathbf{u}_m^{(j)}\right\|_2^2 \ . \end{aligned} \tag{3.8}$$

An efficient implementation of the algorithm is obtained by initializing the $\mathbf{y}_m$ sub-problem from the solution of the previous iterate. Moreover, each sub-problem can be solved inexactly with an adaptive tolerance ϵ_n compatible with the primal and dual residuals of the main ADMM iteration (we choose $\epsilon_k = \max\{r_k, s_k\}/10$). Finally, the $\mathbf{y}_m$ problem itself can be solved via a standard algorithms such as ADMM or FISTA [BT09].

3.2.3 Eigenspace Decomposition

A common trick when dealing with the graph Laplacian is to decompose L in the spectral domain, diagonalizing L using its eigenbasis $\{\mathbf{e}_k\}_{k\in\{1,\dots,n\}}$. Using this formulation, iterates involving L can be computed explicitly by computing inner products with eigenvectors. For example, the $\mathbf{z}$ update of Eq. (3.6) becomes

$$\mathbf{z}_m = \sum_k \frac{\rho}{\rho + \mu\lambda_k} \langle \mathbf{x}_m + \mathbf{v}_m, \mathbf{e}_k \rangle \mathbf{e}_k \ , \tag{3.9}$$

where λ_k is the k-th eigenvalue of L corresponding to e_k. This approach would still be infeasible if we were to compute all the eigenvectors of L, but for many non-local graphs derived from images, most of the larger eigenvalues are indeed close to unity if the graph Laplacian is normalized. Thus only the smallest few eigenvectors are needed to approximate the matrix L. This technique, called spectral truncation, has been successfully applied in graph cut algorithms for clustering [BF12, BF16, MBB15].

3.2.4 Speed of Algorithms

Here we compare the computational performance of various algorithm options. We have a choice of using eigenvectors or using the full matrix, and also using ADMM double-split or ADMM single-split for the main algorithm, giving a total of four combinations. We test each one on a set of problems of varying sizes, and plot the total convergence time relative to that of standard convolutional sparse coding (e.g. 2.0 means it takes twice as long to converge as the standard algorithm). The relative residual stopping tolerance [BPC10] is set to 10^{-3}. All algorithms are tested on the same image with the same parameters $\lambda = 0.1, \mu = 0.1$ except for the standard convolutional case, which is tested with $\lambda = 0.1$. As Figures 3.1 and 3.2 show, ADMM double-split is faster when using eigenvector truncation, and single-split is

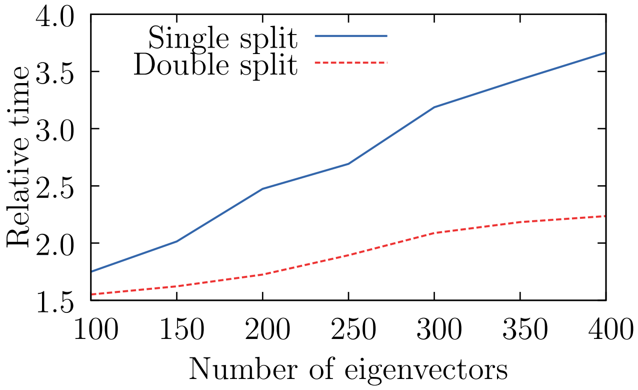


Figure 3.1: Eigenvector Time Test

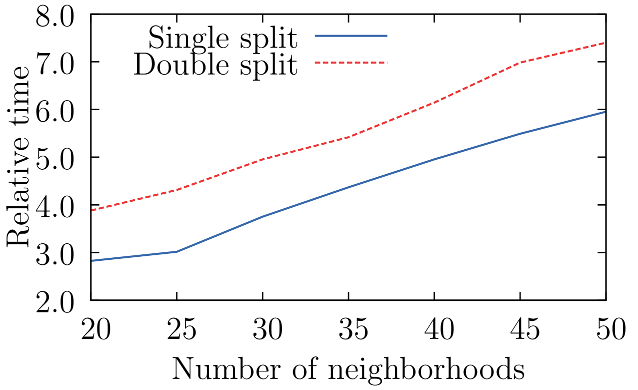


Figure 3.2: Full Matrix Time Test

faster when using the full matrix. This discrepancy is due to different implementations of the $\{\mathbf{z}_m\}$ update in ADMM double-split. In the full matrix case, $\{\mathbf{z}_m\}$ is updated by solving a symmetric linear system which can be costly, while in the eigenvector case the update only involves inner products with the eigenvectors.

3.2.5 Efficient Graph Computation

In general, it is too computationally expensive to generate the full non-local graph of the image. One way to deal with this is to use eigenvector decomposition as described in Sec. 3.2.3. Since only the first few eigenvectors are needed, it makes sense to use an algorithm that computes the eigenvectors without constructing the full graph. We use the Nystrom Extension [FBC04], a sampling strategy used to compute an approximation to the true eigenvectors. Error bounds for the Nystrom Extension have been studied in [Git11].

There are cases where too many eigenvectors are needed to accurately reflect the full graph Laplacian. In this case, we have to resort to using the full matrix L. A straightforward way to reduce cost is to sparsify the graph. Graph sparsification can be done via building a k-nearest neighbor graph or spatial localization, i.e., to restrict connections of pixels to only its spatial neighborhood. An interesting observation is that the case where too many eigenvectors are needed often occurs when the graph construction parameters have made the graph too disconnected, i.e., sparse. This suggests a guideline for choosing the best algorithm: if we intend the graph to be well connected, use eigenvector decomposition; otherwise, use a sparse Laplacian.

3.3 Numerical Results

3.3.1 Image Inpainting

Laplacian convolutional sparse coding can improve the performance of image inpainting compared to standard convolutional sparse coding. The model for inpainting using the standard convolutional sparse coding is

$$\begin{aligned}\mathop{\arg\min}_{\{\mathbf{x}_m,\mathbf{z}_1,\mathbf{z}_2\}} \frac{1}{2}\left\|\sum_m \mathbf{d}_m * \mathbf{x}_m + \mathbf{z}_1 + \mathbf{z}_2 - \mathbf{s}\right\|_2^2 + \lambda \sum_m \|\mathbf{x}_m\|_1 \\ + \sum_i \chi(i)\mathbf{z}_1(i) + \frac{\nu}{2}\|\nabla \mathbf{z}_2\|_2^2 \ , \end{aligned} \tag{3.10}$$

where $\chi(i) = 0$ if i is a missing pixel, and $+\infty$ otherwise. Here $\mathbf{s}$ will be the corrupted image, $\mathbf{z}_1$ will absorb the missing pixel values, $\mathbf{z}_2$ will be a low frequency component of the image[1], and the reconstruction will be $\mathbf{s}_{\text{rec}} = \sum_m \mathbf{d}_m * \mathbf{x}_m + \mathbf{z}_2$. The corresponding model for the Laplacian case is

$$\begin{aligned}\mathop{\arg\min}_{\{\mathbf{x}_m,\mathbf{z}_1,\mathbf{z}_2\}} \frac{1}{2}\left\|\sum_m \mathbf{d}_m * \mathbf{x}_m + \mathbf{z}_1 + \mathbf{z}_2 - \mathbf{s}\right\|_2^2 + \lambda \sum_m \|\mathbf{x}_m\|_1 \\ + \sum_i \chi(i)\mathbf{z}_1(i) + \frac{\mu}{2}\sum_m \langle \mathbf{x}_m, L\mathbf{x}_m\rangle + \frac{\nu}{2}\|\nabla \mathbf{z}_2\|_2^2 \, . \end{aligned} \tag{3.11}$$

We use the standard model Eq. (3.10) to inpaint the image first to construct the graph Laplacian L.

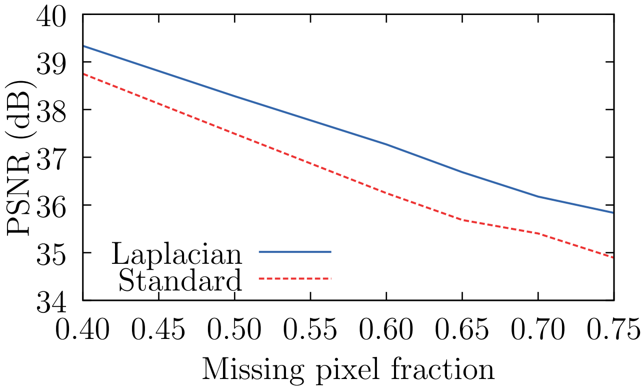


Figure 3.3: Lena Inpainting Comparison

Inpainting is tested on the 512×512 "Lena" image with missing pixel fraction ranging from 40% (PSNR 10.67dB) to 75% (PSNR 8.65dB), using a separately trained $12 \times 12 \times 36$ dictionary. A parameter search on λ and ν is done first to produce the best performance for the standard model. The same set of parameters is then used for the Laplacian model

[1] Employed here for similar reasons to the usual subtraction of the patch mean in patch based sparse coding.

with μ set to 0.1, which has proved empirically to be a good choice, and with a K-Nearest neighbor graph with $K = 40$, constructed using the cosine distance metric. A comparison of PSNR values for both cases is given in Fig. 3.3. The Laplacian model is consistently better than the standard convolutional model for all noise levels, with an average PSNR increase of around 0.85 dB.

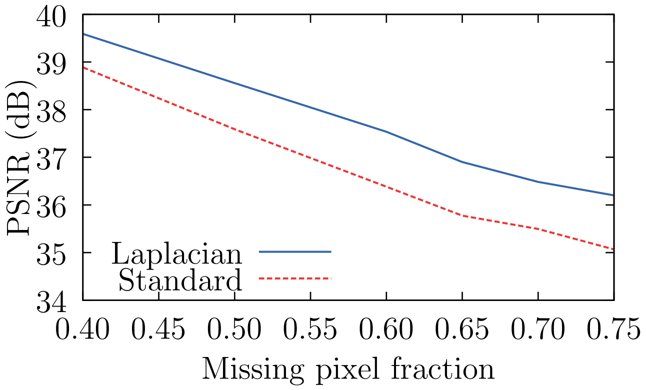


Figure 3.4: Straw Inpainting Comparison

As might be expected, the Laplacian model yields better performance for images with more structural similarity. If we repeat the same experiment on "Straw", a texture-rich image consisted of vertically aligned straws [str], the average PSNR increase is around 1 dB, as shown in Fig. 3.4. More importantly, the performance gap is wider for the "Straw" image when the corruption level is higher, showing that the model has better performance on images with more structural similarity.

3.3.2 Dictionary Learning

Dictionary learning with the graph Laplacian regularizer can be achieved by adding a constraint $\|\mathbf{d}_m\| \leq 1$ to Eq. (3.2) and updating $\mathbf{d}$ and $\mathbf{x}$ in a interleaved manner, as in [Woh14]. In some applications it is desirable to train dictionaries from images corrupted by Gaussian white noise. Convolutional dictionary learning has relatively poor resistance to noise in the training images due to the homogeneous treatment of sparsity in spatial and filter indices that is inherent in the ℓ^1 regularizer. This is substantially improved by incorporating the graph Laplacian regularization proposed here. This improvement is due to the nonzero coefficients of the Laplacian model having more spatial structure when given the same amount of sparsity as a result of the non-local smoothing effect of the graph Laplacian. A comparison using dictionaries trained on 5 randomly selected images from the MIRFlickr dataset [HL08] is presented in Fig. 3.5 and 3.6; note that the dictionary filters in Fig. 3.6 have substantially less noise in smooth regions.

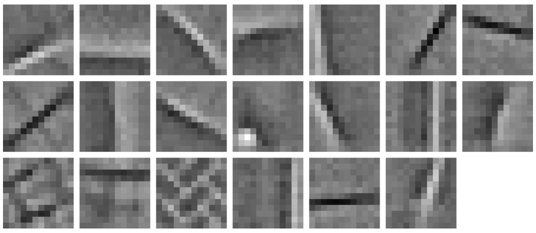


Figure 3.5: Best Standard Dictionary for $N = 20$

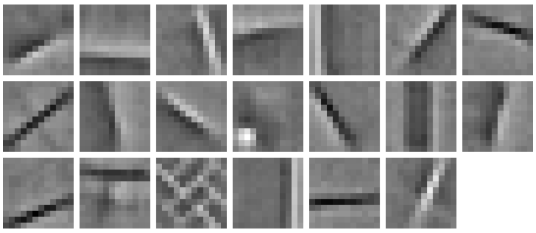


Figure 3.6: Best Laplacian Dictionary for $N = 20$

CHAPTER 4

Convergence of Graph Allen-Cahn Scheme

4.1 Background

In this section, we study the convergence of the discrete Allen-Cahn scheme on the graph, from both a theoretical and empirical point of view. The main results in this chapter are listed below:

- We prove that there exists a graph-independent upper-bound c such that for all $0 \leq dt \leq c$, the schemes (4.3), (4.5) are monotone and convergent in the Ginzburg-Landau energy, and that under an a posterior condition, the sequence $\{u^k\}$ is convergent.
- We generalize the results to incorporate spectral truncation and multiclass classification.
- We conduct a variety of numerical experiments to compare practical performance with theory.

We give a brief introduction of this scheme below. Define the Ginzburg-Landau energy on graphs by replacing the spatial Laplacian with the graph Laplacian L.

$$GL(u) = \frac{\epsilon}{2}\langle u, Lu\rangle + \frac{1}{\epsilon}\sum_i W(u(i)), \tag{4.1}$$

where W is the double well potential $W(x) = \frac{1}{4}(x^2-1)^2$. Let $\boldsymbol{W}(u) = \sum_i W(u(i))$. The Allen-Cahn equation on graphs is defined as the gradient flow of the graph Ginzburg-Landau functional.

$$u_t = -\nabla GL(u) = -\epsilon Lu - \frac{1}{\epsilon}\nabla \boldsymbol{W}(u). \tag{4.2}$$

The discrete graph Allen-Cahn scheme in [BF12] is a semi-implicit discretization of equation (4.2). The reason for being semi-implicit is to counter the ill-conditioning of the graph Laplacian

$$\frac{u^{k+1} - u^k}{dt} = -\epsilon L u^{k+1} - \frac{1}{\epsilon}\nabla \boldsymbol{W}(u^k). \tag{4.3}$$

To do semi-supervised learning, a quadratic fidelity term $\frac{1}{2}\eta\|u - \phi^0\|_\Lambda^2$ is added to the graph Ginzburg-Landau energy

$$F(u) = GL(u) + \frac{1}{2}\eta\|u - \phi^0\|_\Lambda^2, \tag{4.4}$$

where $\|u - \phi^0\|_\Lambda^2 := \langle u - \phi^0, \Lambda(u - \phi^0)\rangle$. Here Λ is a diagonal matrix where $\Lambda_{ii} = 1$ if i is in the fidelity set and 0 else. $\phi^0(i) \in \{1, -1\}$ corresponds to the known labels of the nodes in the fidelity set. In this setup, the value $u(i)$ can be interpreted as a continuous label assignment, and thresholding $u(i) > 0$ and $u(i) < 0$ gives a corresponding partition of the graph. Solving the gradient flow of $F(u)$ via the semi-implicit discretization, we have:

$$\frac{u^{k+1} - u^k}{dt} = -\epsilon L u^{k+1} - \frac{1}{\epsilon}\nabla \boldsymbol{W}(u^k) - dt * \eta\Lambda(u^k - \phi^0). \tag{4.5}$$

In later sections, we will study the scheme (4.3) first and then incorporate the fidelity term in the analysis.

Next, we introduce spectral truncation. Note in each iteration of (4.3) and (4.5), we need to solve a linear system of the form $(I + dtL)u = v$. In many applications, the number of nodes N on a graph is huge, and it is too costly to solve this equation directly. In [BF12, MKB13], a strategy proposed was to project u onto the m eigenvectors with the smallest eigenvalues. In practice, spectral truncation gives accurate segmentation results but is computationally much cheaper. The reason spectral truncation works is because the first few eigenvectors of the graph Laplacian contains most of the information needed to capture the geometry of the various clusters of the graph. In particular, the second eigenvector, named the Fiedler vector, approximates the solution to the normalized graph cut problem [Von07]. There are several methods for precomputing the eigenvectors including the random sampling Nyström method [FBC04] and the Raleigh-Chebyshev method [And10]. In practice, the selection of the stepsize dt is very important to the performance of the model, but is largely chosen empirically by trial and error.

We present some basic notations and definitions used in the rest of the chapter. First, we identify a vector $u \in \mathbb{R}^N$ with a function on the graph. We use $u(i)$ to denote the value of u on node i. We use a superscript u^k to denote the k-th iterate of some discrete scheme. We define the p-norm of a graph function u as the vector p-norm $\|u\|_p = (\sum_i |u(i)|^p)^{1/p}$. We also sometimes drop the subscript and write $\|u\|$ for 2-norms when there is no ambiguity. In the next few sections, we will frequently encounter functions with vector inputs of the form $\boldsymbol{F} : (u(1), \ldots u(N)) \mapsto (F_0(u(1)), \ldots F_N(u(N)))$. We denote such functions as a *diagonal map*, since $\boldsymbol{F}$ corresponds to a diagonal matrix when linear. We call the scalar functions F_k *components* of the diagonal map $\boldsymbol{F}$. In general, we use the same letter to denote components and the diagonal map. If the components are the same across dimensions, we omit the subscript and simply denote it as F. In general, we will use bold text for functions with vector input and plain text for scalar functions to disambiguate whenever there is a name clash. For example, we denote

$$\boldsymbol{W}(u) = \sum_i \boldsymbol{W}(u(i)) = \frac{1}{4}\sum_i (u(i)^2 - 1)^2 \tag{4.6}$$

to be the double-well functional, and

$$W(x) = \frac{1}{4}(x^2 - 1)^2 \tag{4.7}$$

to be the double-well function.

4.2 Maximum Principle-L^∞ Estimates

The main result for this section is the following:

Proposition 4.2.1 (A Priori Boundeness). *Define u^k by the semi-implicit graph Allen-Cahn scheme*

$$\frac{u^{k+1} - u^k}{dt} = -\epsilon L u^{k+1} - \frac{1}{\epsilon}\nabla \boldsymbol{W}(u^k), \tag{4.8}$$

where $\boldsymbol{W}$ is the double-well functional (4.6), and L is the unnormalized graph Laplacian . Assume $\|u^0\|_\infty \leq 1$. If $0 \leq dt \leq 0.5\epsilon$, then $\|u^k\|_\infty \leq 1$, $\forall k \geq 0$.

What is notable is that the stepsize restriction is independent of the graph. To see this, we split the discretization (4.3) into two parts.

$$\begin{cases} v^k = u^k - dt * \frac{1}{\epsilon}\nabla \boldsymbol{W}(u^k), \\ u^{k+1} = -dt * (\epsilon L^u u^{k+1}) + v^k. \end{cases} \tag{4.9}$$

We will prove that $\|u^{k+1}\|_\infty \leq \|v^k\|_\infty$ for all $dt > 0$ via the maximum principle, and show that the stepsize restriction essentially comes from the first line of (4.9). For future reference, we denote the first line of (4.9) as the *forward step* since it corresponds to a forward stepping scheme for the gradient flow, and the second line a backward step correspondingly.

4.2.1 Maximum Principle

The classical maximum principle argument relies on the fact that $\Delta u(x_0) \geq 0$ for x_0 a local minimizer. This fact is also true for graphs.

Proposition 4.2.2 (Second Order Condition on Graphs)**.** *Let u be a function defined on a graph, and L be either the unnormalized graph Laplacian or the random walk graph Laplacian . Suppose u achieves a local minimum at a vertex i, where a local minimum at vertex i is defined as $u(i) \leq u(j), \forall w_{ij} > 0$. Then we have $[Lu](i) \leq 0$.*

Proof. For both the random walk and the unnormalized Laplacian, we have the following:

$$\begin{cases} L_{ii} = -\sum_{j\neq i} L_{ij}, \\ L_{ij} \leq 0. \end{cases} \tag{4.10}$$

Let i be a local minimizer. Then

$$\begin{aligned} [Lu](i) &= L_{ii}u(i) + \sum_{j\neq i} L_{ij}u(j) \\ &= \sum_{j\neq i} L_{ij}(u(j) - u(i)) \leq 0 \quad \square \end{aligned} \tag{4.11}$$

The difference in sign for $[Lu](i)$ compared with the continuous case is due to the difference in convention for the graph Laplacian and the continuous Laplacian. Next, we prove a discrete analogue of the continuous time maximum principle, which states that the implicitly discretized scheme for the heat equation on graphs is decreasing in the L_∞ norm. This line of thought is inspired by the maximum principle for finite difference operators [Cia70].

Proposition 4.2.3 (Maximum Principle for Discrete Time)**.** *For any $dt \geq 0$, let u be a solution to*

$$u = -dt * (Lu) + v, \tag{4.12}$$

where L is either the unnormalized or the random walk Laplacian, then $\max_i u(i) \leq max_i v(i)$, and $\min_i u(i) \geq min_i v(i)$. Hence $\|u\|_\infty \leq \|v\|_\infty$.

Proof. Suppose $i = \arg\min_j u(j)$ is the node that attains the minimum for u. Then since $u(i) = dt * (-Lu)(i) + v(i)$ and $(-Lu)(i) \geq 0$ by Proposition 4.2.2, we have $min_u = u(i) \geq v(i) \geq min_v$. Arguing similarly with the maximum, we have that $\|u\|_\infty \leq \|v\|_\infty$. □

4.2.2 Proof of Boundedness

We can immediately show via the maximum principle that the stepsize restriction for the sequence u^k to be bounded depends only on the forward step. To be more precise, we have the following proposition.

Proposition 4.2.4. *Let u^k be defined via the semi-implicit scheme*

$$\begin{cases} v^k = u^k - ds * \mathbf{\Phi}(u^k), \\ u^{k+1} = -\dfrac{ds}{\sigma} * Lu^{k+1} + v^k. \end{cases} \tag{4.13}$$

where $\mathbf{\Phi}$ is the diagonal map $\mathbf{\Phi} : (u(1), \ldots, u(N)) \mapsto (\Phi^0(u(1)), \ldots, \Phi^N(u(N)))$, L is the unnormalized graph Laplacian, and σ some constant greater than 0. Define the forward map $\boldsymbol{\mathcal{F}}_{ds} : u \mapsto u - dt * \mathbf{\Phi}(u)$, and denote its components by $\mathcal{F}^i_{ds}$.

Suppose $\exists M > 0$ and some constant $c(M, \mathbf{\Phi})$ such that $\forall 0 \leq ds \leq c$, and $\forall i$, $\mathcal{F}^i_{ds}$ maps the interval $[-M, M]$ to itself. Then if $\|u^0\|_\infty \leq M$, we have $\|u^k\|_\infty \leq M$, $\forall k \geq 0$.

Proof. Suppose $\|u^k\|_\infty \leq M$. By induction and our assumption on $\mathcal{F}^i_{dt}$, $\|v^k\|_\infty \leq M$. By the maximum principle, $\|u^{k+1}\|_\infty \leq \|v^k\|_\infty \leq M$. □

We can now prove Proposition 4.2.1 by setting M and $\mathbf{\Phi}$ in Proposition 4.2.4 accordingly, and estimate the bound $c(M, \Phi)$.

Proof. We set $M = 1$ and $\mathbf{\Phi} = (W', \ldots, W')$, where W is the double-well function. By a change of variables $dt' = dt/\epsilon$ and $\sigma = \frac{1}{\epsilon^2}$, we can WLOG assume $\epsilon = 1$. Thus the component forward maps $\mathcal{F}^i_{dt}$ are

$$\mathcal{F}^i_{dt}(x) = x - dtW'(x) = x - dtx(x^2 - 1) := \mathcal{F}_{dt}(x). \tag{4.14}$$

The proposition is proved if we show $\mathcal{F}_{dt}$ maps $[-1, 1]$ to itself for $dt \leq 0.5$, which is shown in Lemma (4.2.5). □

Lemma 4.2.5. *Define $\mathcal{F}_{dt}$ as in (4.14). If $0 \leq dt \leq 0.5$, $\mathcal{F}_{dt}$ maps $[-1, 1]$ to itself.*

Proof. For a general M, we can estimate c by solving dt to satisfy (4.15)

$$\begin{cases} \max\limits_{x\in[-M,M]} \mathcal{F}_{dt}(x) \leq M \\ \\ \min\limits_{x\in[-M,M]} \mathcal{F}_{dt}(x) \geq -M \end{cases} \tag{4.15}$$

Since $\mathcal{F}_{dt}$ is cubic in x, (4.15) can be solved analytically via brute force calculation. Setting $M = 1$ and solving (4.15) for $dt \geq 0$ gives $0 \leq dt \leq 0.5$. □

Remark: The computation of $c(M, \mathbf{\Phi})$ by solving (4.15) involves only elementary calculations and we omit them for brevity.

The choice of the constant $M = 1$ is natural since the function value $u(i)$ corresponds to a *soft* prediction of the binary class label $\{-1, 1\}$. However, if we merely want to get boundedness without enforcing $\|u^k\|_\infty \leq 1$ we can get a larger stepsize bound by maximizing $c(M, W')$ with respect to M. By computer calculation, we found $\arg\max_M c(M, W') \approx 1.4$, and $c(1.4, W') \approx 2.1$. Namely,

Lemma 4.2.6. *For $0 \leq dt \leq 2.1$, $\mathcal{F}_{dt}$ maps $[-1.4, 1.4]$ to itself.*

The reason we are computing these constants explicitly is that we will compare them in Section 4.6 against results from real applications in a series of numerical experiments. For future reference, the $dt \leq 0.5$ bound will be called the "tight bound" where the $dt \leq 2.1$ bound will be called the "loose bound".

4.2.3 Generalizations of the scheme

In this section, we extend the previous result to the case where fidelity is added, and also to the case for symmetric graph Laplacians L^s.

We restate the the graph Allen-Cahn scheme with fidelity:

$$\begin{cases} v^k = & u^k - dt * (\frac{1}{\epsilon}\nabla \boldsymbol{W}(u^k) + \eta\Lambda(u^k - \phi^0)), \\ u^{k+1} = & - dt * (\epsilon L^u u^{k+1}) + v^k. \end{cases} \tag{4.16}$$

Λ is a diagonal matrix where $\Lambda_{ii} = 1$ if i is in the fidelity set and 0 else, and $\phi^0(i) \in \{1, -1\}$. We can use the same technique to estimate a graph-independent stepsize restriction c and prove boundedness.

Proposition 4.2.7 (Graph Allen-Cahn with fidelity)**.** *Define u^k by (4.16) and suppose $\|u^0\|_\infty \leq 1$. If dt satisfies $0 \leq dt \leq \frac{1}{2+\eta}\epsilon$, we have $\|u^k\|_\infty \leq 1$ for all k.*

Proof. Denote the forward map by $\boldsymbol{\mathcal{F}}_{dt}$. Since Λ is a diagonal matrix, $\boldsymbol{\mathcal{F}}_{dt}$ is a diagonal map. Note $\boldsymbol{\mathcal{F}}_{dt}$ has only three different component maps which we denote by $F^i_{dt}, i = 1, \ldots, 3$. Namely, $F^0_{dt}(u) = u - dt[\frac{1}{\epsilon}(u^2 - 1)u + \eta(u - 1)]$, $F^1_{dt}(u) = u - dt[\frac{1}{\epsilon}(u^2 - 1)u + \eta(u + 1)]$, $F^2_{dt}(u) = u - dt[\frac{1}{\epsilon}(u^2 - 1)u]$. By solving (4.15) with $M = 1$ for $F^m_{dt}, m = 1, \ldots, 3$ for non-negative dt, we get $0 \leq dt \leq \frac{1}{2+\eta}\epsilon$. □

The case for the symmetric graph Laplacian is a little different. Since L^s does not satisfy (4.10), we no longer have the maximum principle. However, we are still able to prove boundedness under the assumption that the graph satisfies a certain uniformity condition.

Proposition 4.2.8 (Symmetric graph Laplacian)**.** *Let $d_i = \sum_j w_{ij}$ be the degree of node i. Suppose $\rho \leq 4$ where ρ is defined below*

$$\rho = \frac{\max_i d_i}{\min_i d_i}. \tag{4.17}$$

Define u^k by the semi-implicit scheme (4.8) where L is the symmetric Laplacian L^s. Suppose $\|u^0\|_\infty \leq 1$. If $0 \leq dt \leq 0.25\epsilon$, we have $\|u^k\|_\infty \leq 2$, for all $k \geq 1$.

Proof. By definition of L^s and L^{rw}, we have the relation

$$L^s = D^{1/2} L^{rw} D^{-1/2} \tag{4.18}$$

Substituting (4.18) to the backward step, i.e., line 2 of (4.9), we have

$$D^{-1/2} u^{k+1} = -dt * L^{rw} D^{-1/2} u^{k+1} + D^{-1/2} v^k. \tag{4.19}$$

We will do a change of variables $\tilde{u}^k = \alpha D^{-1/2} u^k$, and $\tilde{v}^k = \alpha D^{-1/2} v^k$, where $\alpha = (\min_i d_i)^{1/2}$, and write the scheme in terms of $\tilde{u}^k$.

$$\begin{cases} \tilde{v}^k = \tilde{u}^k - dt * \dfrac{1}{\epsilon} \alpha D^{-1/2} \nabla \boldsymbol{W}(\dfrac{1}{\alpha} D^{1/2} \tilde{u}^k), \\ \tilde{u}^{k+1} = -\epsilon dt * L^{rw} \tilde{u}^{k+1} + \tilde{v}^k. \end{cases} \tag{4.20}$$

By the definition of α, we have $\|\tilde{u}^0\|_\infty \leq 1$. We will use the same technique as before to show $\|\tilde{u}^k\| \leq 1, \forall k$. By the maximum principle, $\|\tilde{u}^{k+1}\|_\infty \leq \|\tilde{v}^k\|_\infty$. Since D is diagonal, the forward map $\boldsymbol{\mathcal{F}}_{dt}$ of (4.20) is diagonal. Define $G_{dt}(c, x) = x - \frac{dt}{c} W'(cx) = x - \frac{dt}{c} x(c^2 x^2 - 1)$, the components of $\boldsymbol{\mathcal{F}}_{dt}$ are:

$$\tilde{v}^k(i) = \mathcal{F}^i_{dt}(\tilde{u}^k(i)) = G_{dt/\epsilon}(c_i, \tilde{u}^k(i)), \tag{4.21}$$

where $c_i = (\frac{d_i}{\min_j d_j})^{1/2} \in [1, 2]$. We can prove the theorem if we show $\mathcal{F}^i_{dt}$ maps $[-1, 1]$ to itself for all $i = 1, \ldots, N$. This is shown in the next lemma.

Lemma 4.2.9 (Uniform range)**.** *For any $0 \leq dt \leq 0.25$, and some fixed $c \in [1, 2]$, $G_{dt}(c, x)$ as a function of x maps $[-1, 1]$ to itself.*

The lemma is proved by solving $\max_{c\in[1,2],x\in[-1,1]} G_{dt}(c, x) \leq 1$ and $\min_{c\in[1,2],x\in[-1,1]} G_{dt}(c, x) \geq -1$ for $dt \geq 0$. □

Remark: The condition $\rho < 4$ is arbitrary and just chosen to simplify calculations for dt. The proposition here is weaker than Proposition 4.2.1 due to the loss of the maximum principle. We will see this again during the analysis of spectral truncation in Section 4.4.

4.3 Energy method-L^2 estimates

In this section, we derive estimates in terms of the L^2 norm. Our goal is to prove that the graph Allen-Cahn scheme is *monotone* under the stepsize restrictions in Section 4.2, and derive convergence results of the sequence $\{u^k\}$. We will drop the subscript for 2 norms in this section.

Our proof is loosely motivated by the analysis of convex concave splitting in [Eyr98, YRY02]. In [Eyr98], Eyre proved the following monotonicity result:

Proposition 4.3.1 (Eyre). *Let E_1, E_2 be C^1 functions on $\mathbb{R}^n$, where E_1 is convex and E_2 concave.Let $E = E_1 + E_2$. Then for any $dt > 0$, the semi-implicit scheme*

$$u^{k+1} = u^k - dt\nabla E_1(u^{k+1}) - dt\nabla E_2(u^k), \tag{4.22}$$

is monotone in E, namely,

$$E(u^{k+1}) \leq E(u^k), \quad \forall k \geq 0.$$

In our proof, we will set $E = GL(u)$, $E_1 = \frac{\epsilon}{2}\langle u, Lu\rangle$ and $E_2 = \frac{1}{\epsilon}\boldsymbol{W}(u)$. Since E_2 is not concave, we will have to generalize Proposition 4.3.1 for general E_2. But first, we digress a bit and establish the connection between the semi-implicit scheme (4.22) and the proximal gradient method, which simply assumes E_1 to be sub-differentiable. The reason for this generalization is to have a unified framework for dealing with E_1 taking extended real values, which is the case when we study spectral truncation in Section 4.4.

The proximal gradient iteration [BV09] is defined as

$$u^{k+1} = Prox_{dtE_1}(u^k - dt\nabla E_2(u^k)), \tag{4.23}$$

where the $Prox$ operator is defined as $Prox_{\gamma f}(x) = \arg\min_u\{f(u)+\frac{1}{2\gamma}\|u-x\|^2\}$. This scheme is in fact equivalent to the semi-implicit scheme (4.22) when E_1 is differentiable. This is clear from the implicit gradient interpretation of the proximal map. Namely, if $y = Prox_{\gamma f}(x)$,

$$y = x - \gamma\partial f(y). \tag{4.24}$$

∂f is the *subgradient* of f, which coincides with the gradient if f is differentiable.

Even though the subgradient can be multi-valued, the *Prox* operator is in fact well-defined if f is a closed proper convex functions taking extended real values. Classical results for convergence of proximal gradient method can be found in [BV09], but is not applicable here since it requires both E_1, E_2 to be convex. Instead, we will prove an energy estimate for proximal gradient methods when E_2 is a general function.

Proposition 4.3.2 (Energy Estimate)**.** *Let $E = E_1 + E_2$. Suppose E_1 is proper closed and convex, $E_2 \in C^2$. Define x^{k+1} by the proximal gradient scheme $x^{k+1} \in x^k - dt\partial E_1(x^{k+1}) - dt\nabla E_2(x^k)$. Suppose M satisfies*

$$M \geq \max_{\xi \in S} \|\nabla^2 E_2(\xi)\|, \tag{4.25}$$

where $S = \{\xi | \xi = tx^k + (1-t)x^{k+1}, t \in [0,1]\}$ is the line segment between x^k and x^{k+1}, we have

$$E(x^k) - E(x^{k+1}) \geq (\frac{1}{dt} - \frac{M}{2})\|x^{k+1} - x^k\|^2. \tag{4.26}$$

Proof.

$$\begin{aligned} E(x^k) - E(x^{k+1}) &= E_1(x^k) - E_1(x^{k+1}) + E_2(x^k) - E_2(x^{k+1}) \\ &\geq \langle \partial E_1(x^{k+1}), x^k - x^{k+1} \rangle + E_2(x^k) - E_2(x^{k+1}) \\ &= E_2(x^k) - E_2(x^{k+1}) - \langle \nabla E_2(x^k), x^k - x^{k+1} \rangle \\ &+ \frac{1}{dt}\|x^{k+1} - x^k\|^2 \\ &\geq \frac{1}{dt}\|x^{k+1} - x^k\|^2 - \frac{M}{2}\|x^{k+1} - x^k\|^2. \end{aligned}$$

The second line is by convexity of E_1 and the definition of subgradients, and $\partial E_1(x^{k+1})$ could be any vector in the subgradient set. The third line is by substituting the particular subgradient $\partial E_1(x^{k+1})$ in the the definition of x^{k+1}. The fourth line is obtained by one variable Taylor expansion of the function E_2 along the line segment between x^k and x^{k+1}. □

Next, we apply estimate (4.3.2) and the boundedness results in Section 4.2 to prove that the graph Allen-Cahn scheme is monotone in the Ginzburg-Landau energy under a graph-independent stepsize.

Proposition 4.3.3 (Monotonicity of the Graph Allen-Cahn Scheme). *Let u^k be the graph Allen-Cahn scheme with fidelity defined below:*

$$u^{k+1} = u^k - dt * (\epsilon L u^{k+1} + \frac{1}{\epsilon} W'(u^k) + \eta \Lambda(u^k - \phi^0)), \tag{4.27}$$

where L is the unnormalized Laplacian. If $\|u^0\|_\infty \leq 1$, then $\exists c$ independent of L such that $\forall 0 \leq dt \leq c$, we have the scheme is monotone under the Ginzburg-Landau energy with fidelity, namely, $E(u^k) = GL(u^k) + \frac{\eta}{2}\|u^k - \phi^0\|_\Lambda^2 \geq E(u^{k+1}) = GL(u^{k+1}) + \frac{\eta}{2}\|u^{k+1} - \phi^0\|_\Lambda^2$. Moreover, the scheme is convergent in function value. The result holds for symmetric Laplacians if we add the uniformity condition (4.17) on the graph.

Proof. From Proposition 4.2.1, $\exists c_1$ independent of L such that $0 \leq dt \leq c_1$, implies $\|u^k\|_\infty \leq 1, \forall k$. We set $E_2(u) = \frac{1}{\epsilon}\boldsymbol{W}(u) + \frac{\eta}{2}\|u^k - \phi^0\|_\Lambda^2$, and $E_1(u) = \frac{\epsilon}{2}\langle u, Lu\rangle$. Since (4.27) is equivalent to the proximal gradient scheme with E_1 and E_2 defined above, we can apply Proposition 4.3.2. Since the L^∞ unit ball is convex, line segments from u^k to u^{k+1} lie in the set $\{\|u\|_\infty \leq 1\}$, and we can estimate M by the inequality below

$$\max_{\|u\|_\infty \leq 1} \|\nabla^2 E_2(u)\|_2 \leq \max_{|x|\leq 1} |\frac{1}{\epsilon} W''(x) + \eta| = \frac{2}{\epsilon} + \eta.$$

Thus we can set $M = \frac{2}{\epsilon} + \eta$. Choose $c_2 \leq \frac{2}{M}$, and set $c = \min(c_1, c_2)$. We have $\forall 0 \leq dt \leq c$,

$$E(u^k) - E(u^{k+1}) \geq (\frac{1}{dt} - \frac{M}{2})\|u^{k+1} - u^k\|^2 \geq 0. \tag{4.28}$$

Hence u^k is monotone in E. Since E is bounded from below by 0, the sequence $\{E(u^k)\}$ is convergent. The case for the symmetric Laplacian is identical by a different estimate of $\max_{|x|_\infty \leq 1} \|\nabla^2 E_2\|$. □

Next, we discuss the convergence of the iterates $\{u^k\}$. First, we prove subsequence convergence of $\{u^k\}$ to a stationary point of $E(u)$.We first prove a lemma on the sequence $\{u^{k+1} - u^k\}$.

Lemma 4.3.4. *Let u^k, dt, be as in Proposition 4.3.3, then $\sum_{k=0}^\infty \|u^{k+1} - u^k\|^2 < \infty$. Hence $\lim\limits_{k\to\infty} \|u^{k+1} - u^k\| = 0$.*

Proof. Summing Equation (4.28), we have the following

$$E(u^0) - E(u^n) \geq (\frac{1}{dt} - M) \sum_{k=0}^{n-1} \|u^{k+1} - u^k\|^2, \tag{4.29}$$

holds for all n. Since $E(u) \geq 0$ and $dt \leq \frac{2}{M}$, we prove the lemma. □

Proposition 4.3.5. *(Subsequence convergence to stationary point) Let u^k, dt, be as in Proposition 4.3.3. Let S be the set of limit points of the set $\{u^k\}$. Then $\forall u^* \in S$, u^* is a critical point of E, i.e., $\nabla E(u^*) = 0$. Hence any convergent subsequence of u^k converges to a stationary point of E.*

Proof. By definition, $u^{k+1} = u^k - dt\nabla E_1(u^{k+1}) - dt\nabla E_2(u^k)$. Hence we have

$$\|\nabla E_1(u^k) + \nabla E_2(u^k)\| \leq (\|\nabla E_1(u^{k+1}) - \nabla E_1(u^k)\| + \frac{1}{dt}\|u^{k+1} - u^k\|). \tag{4.30}$$

Since $\{u^k\}$ is bounded and ∇E_1 is continuous, we have

$$\lim_{k\to\infty} \|\nabla E(u^k)\| = \lim_{k\to\infty} \|\nabla E_1(u^k) + \nabla E_2(u^k)\| = 0, \tag{4.31}$$

where we use $\lim_{k\to\infty} \|u^{k+1} - u^k\| = 0$. □

In general, we can not prove that the full sequence $\{u^k\}$ is convergent, since it is possible for the iterates $\{u^k\}$ to oscillate between several minimum. However we show that when the set of limit points is finite, we do have convergence. This is stated in the Lemma 4.3.6, which is proved in the appendix.

Lemma 4.3.6. *Let u^k be a bounded sequence in $\mathbb{R}^N$, and $\lim_{k\to\infty} \|u^{k+1} - u^k\| = 0$. Let S be the set of limit points of the set $\{u^k | k \geq 1\}$. If S has only finitely many points, then S contains only a single point u^*, and hence $\lim_{k\to\infty} u^k = u^*$.*

Finally, we provide an easy to check a posterior condition that guarantees convergence using the lemma above. The condition states that the iterates u^k must take values reasonably close to the double-well minimum -1 and 1. Empirically, we have observed that the values of u^k are usually around -1 and 1 near convergence, hence the condition is not that restrictive in practice.

Proposition 4.3.7. *(Convergence with A Posterior Condition) Let u^k, dt, be as in Proposition 4.3.3. Let $\delta > 0$ be any positive number. If for some K, we have $|u^k(i)| \geq \frac{1}{\sqrt{3}} + \delta$, for all $k \geq K$ and i, then we have $\lim\limits_{k\to\infty} u^k = u^*$, where u^* is some stationary point of the energy E.*

Proof. We only need to show that the set of stationary points of E on the domain $D = [\frac{1}{\sqrt{3}} + \delta, 1]^N$ is finite. Computing the Hessian of E, we have $\nabla^2 E(u) = \epsilon L + \frac{1}{\epsilon}(3u^2 - I) + \eta\Lambda$, where u^2 is the diagonal matrix whose entries are $u(i)^2$. Note that $\nabla^2 E(u)$ is positive definite on D since $\eta\Lambda$ and L are semi-positive definite, and $3u^2 - I$ is positive definite on D. Therefore, the stationary points are isolated on D. Since D is bounded, this implies finiteness. □

4.4 Analysis on Spectral Truncation

In this section, we study whether we are able to prove monotonicity and boundedness when using spectral truncation. First of all, we formally define the spectral truncated graph Allen-Cahn scheme. In this section, all conclusions hold for both the unnormalized Laplacian and the symmetric Laplacian. We will use the general notation L for both options.

Let $\{\phi^1, \phi^2, \ldots, \phi^m\}$ be eigenvectors of the graph Laplacian L ordered by eigenvalues in ascending order, i.e., $\lambda_1 \leq \lambda_2 \cdots \leq \lambda_N$. Define the m-th eigenspace as $V_m = span\{\phi^1, \phi^2, \ldots, \phi^m\}$, and P_m as the orthogonal projection operator onto the space V_m. Then the spectral truncated scheme is defined as

$$\begin{cases} v^k = u^k - dt * \frac{1}{\epsilon}\nabla \boldsymbol{W}(u^k), \\ u^{k+1} = P_m[-dt * (\epsilon L u^{k+1}) + v^k]. \end{cases} \tag{4.32}$$

Note that in practice, we do not directly solve the linear system on the second line of (4.32), but instead express u^{k+1} directly in terms of the eigenvectors as in (4.36). However, writing it in matrix is notationally more convenient in the subsequent analysis. We want to apply the energy estimates in Section 4.3 for spectral truncation. To do this, we first show that spectral truncation scheme (4.32) can be expressed as a proximal-gradient scheme for a suitably chose energy E_1 and E_2.

Proposition 4.4.1 (Reformulation of Spectral Truncation)**.** *The spectral truncated scheme (4.32) is equivalent to the proximal gradient scheme (4.23) with* $E_1 = \frac{\epsilon}{2}\langle u, Lu\rangle + I_{V_m}$*,* $E_2 = \frac{1}{\epsilon}\boldsymbol{W}(u)$*, where* I_{V_m} *is the indicator function of the m-th eigenspace, i.e.*

$$I_{V_m}(u) = \begin{cases} 0, & u \in V_m \\ +\infty, & \textit{else.} \end{cases} \tag{4.33}$$

Proof. Define u, u' as

$$u = \quad P_m[-dt * (\epsilon Lu) + v]. \tag{4.34}$$

$$u' = \quad \arg\min_y \frac{\epsilon}{2}\langle y, Ly\rangle + I_{V_m}(y) + \frac{1}{2dt}\|y - v\|^2. \tag{4.35}$$

We only have to show $u = u'$. Decomposing (4.34) in terms of the eigenbasis $\{\phi^1, \phi^2, \ldots, \phi^m\}$, we have

$$u = \sum_{j \le m} \frac{\langle v, \phi^j\rangle}{1 + dt\epsilon\lambda_j}\phi^j. \tag{4.36}$$

Since I_{V_m} is $+\infty$ outside V_m, we have $u' \in V_m$. Let $u' = \sum\limits_{i=1}^{m} c_i'\phi^i$, and $y = \sum\limits_{i=1}^{m} c_i\phi^i$ then the function in (4.35) becomes

$$\frac{\epsilon}{2}\langle y, Ly\rangle + \frac{1}{2dt}\|y - v\|^2 = \sum_{i=1}^{m}(\frac{\epsilon}{2}\lambda_i c_i^2 + \frac{1}{2dt}(c_i - \langle v, \phi^i\rangle)^2) + C. \tag{4.37}$$

And therefore

$$c_i' = \arg\min_c \frac{\epsilon}{2}\lambda_i c^2 + \frac{1}{2dt}(c - \langle v, \phi^i\rangle)^2 = \frac{\langle v, \phi^i\rangle}{1 + dt\epsilon\lambda_i}. \tag{4.38}$$

Hence we have $u = u'$. □

Since the orthogonal projection P_m is expansive in the l^∞ norm, i.e., $\|P_m u\|_\infty \le \|u\|_\infty$ does not always hold, we lose the maximum principle. However, we show that the energy estimate alone is enough to prove monotonicity and boundedness under a smaller stepsize.

Proposition 4.4.2. *Let L be either the symmetric or unnormalized graph Laplacian satisfying* $\rho_L = \max_i |\lambda_i| \le B$ *for some constant B. Set* $\epsilon = 1$ *and define* u^k *by the spectral truncation scheme (4.32). Suppose* $\|u^0\|_\infty \le 1$*, and* $u^0 \in V_m$*. Then there exists* $\delta > 0$ *dependent* *only on B such that $\forall 0 \leq dt \leq \delta N^{-1}$, The sequence $\{u^k\}$ is bounded and $GL(u^{k+1}) \leq GL(u^k)$, for all k. Here N is the dimension of u, i.e., number of vertices in the graph.*

The choice for $\epsilon = 1$ is only to avoid complicated dependencies on ϵ that obscures the proof. For the next two sections, we will assume $\epsilon = 1$ throughout. To prove the theorem, we first establish the following lemmas.

Lemma 4.4.3 (Inverse Bound)**.** *Let M be any positive constant. Set $\epsilon = 1$ in the GL functional. If $GL(u) \leq M$, then $\|u\|_2^2 \leq N + 2\sqrt{NM}$, where N is the dimension of u.*

Proof. By definition, $GL(u) = \frac{1}{4}\sum_i (u(i)^2 - 1)^2 + \frac{1}{2}\langle u, Lu\rangle \leq M$. Since $\frac{1}{2}\langle u, Lu\rangle \geq 0$, $\frac{1}{4}\sum_i (u(i)^2 - 1)^2 \leq M$. Then from the Cauchy-Schwarz inequality, $\sum_i (u(i)^2 - 1) \leq 2\sqrt{NM}$, and hence $\|u\|_2^2 \leq N + 2\sqrt{NM}$. □

Lemma 4.4.4. *Let u^k and u^{k+1} be successive iterates defined in (4.32). Then the following inequality holds:*

$$\|u^{k+1}\|_2 \leq (1 + dt)\|u^k\|_2 + dt\|u^k\|_2^3. \tag{4.39}$$

Proof. Since L is symmetric semi-positive definite, and the orthogonal projection P_m is non-expansive in the l^2 norm, we have $\|u^{k+1}\|_2 \leq \|v^k\|_2$. Since $v^k(i) = u^k(i) - dt * [u^k(i)(u^k(i)^2 - 1)]$, let $g(i) = (u(i))^3$, then

$$\begin{aligned} \|v^k\|_2 &\leq (1 + dt)\|u^k\|_2 + dt\|g\|_2 = (1 + dt)\|u^k\|_2 + dt\|u^k\|_6^3 \\ &\leq (1 + dt)\|u^k\|_2 + dt\|u^k\|_2^3, \end{aligned} \tag{4.40}$$

where $\|u^k\|_6 \leq \|u^k\|_2$ is by the norm equivalence equation stated in Lemma 4.7.1 in the Appendix. □

Next, we prove the main proposition. The idea is to choose dt small enough such that monotonicity in GL is satisfied, and then apply Lemma 4.4.3 to have a bound on u^k.

Proposition 4.4.2. Let $E_1(u) = \frac{\epsilon}{2}\langle u, Lu\rangle + I_{V_m}$, $E_2(u) = \frac{1}{\epsilon}\boldsymbol{W}(u)$, and $E = E_1 + E_2 = GL(u) + I_{V_m}$. By Proposition 4.4.1, (4.32) is equivalent to the proximal gradient scheme for the splitting $E = E_1 + E_2$. We also have $\forall k \geq 0$, $E(u^k) = GL(u^k)$. $u^0 \in V_m$ is by our assumption, and $u^k \in V_m, k \geq 1$ is because u^k is the image of the projection map P_m. Therefore, we will denote $E(u^k)$ and $GL(u^k)$ interchangeably.

We claim that there exists constants $\delta > 0$, independent of N such that $\forall 0 \leq dt \leq \delta N^{-1}$, equation (4.41) holds for all k.

$$\begin{aligned} GL(u^k) \leq& GL(u^0) \leq C_0 N, \\ \|u^k\|_2 \leq& C_1\sqrt{N}, \end{aligned} \tag{4.41}$$

where $C_0 = (1 + B)$, and $C_1 = \sqrt{(1 + 2\sqrt{1 + B})N}$.

We argue by induction. For the case $k = 0$, since $\|u^0\|_\infty \leq 1$, we have $\|u^0\|_2 \leq \sqrt{N} < C_1\sqrt{N}$. We also have $GL(u^0) \leq \rho_L\|u^0\|_2^2 + \sum_{i\leq N} 1 \leq C_0 N$, since $\|u^0\|^2 \leq N$, and $\rho_L \leq B$.

For the induction step, suppose (4.41) is satisfied for iteration k. Since $\|u^k\|_2 \leq C_1\sqrt{N}$, we apply Lemma 4.4.4 and get $\|u^{k+1}\|_2 \leq \frac{A_1}{2}(1+dt)N^{1/2} + \frac{A_1}{2}dtN^{3/2}$ for some A_1 independent of N. Therefore, we can choose δ_1 independent of N such that $\forall 0 \leq dt \leq \delta_1 N^{-1}$, $\|u^{k+1}\|_2 \leq A_1 N^{1/2}$.

Next, we apply Proposition 4.3.2 and show $E(u^k) \geq E(u^{k+1})$. Since $\|u^{k+1}\|_\infty \leq \|u^{k+1}\|_2 \leq A_1 N^{1/2}$. We can set M in Proposition 4.3.2 by the estimate below:

$$\max_{\|\xi\|_\infty \leq A_1\sqrt{N}} \|\nabla^2(W)(\xi)\|_2 \leq \max_{|\xi| \leq A_1\sqrt{N}} |W''(\xi)| = \max_{|\xi| \leq A_1\sqrt{N}} |(3\xi^2 - 1)| \leq A_2 N,$$

where A_2 independent of N, and we can set $M = A_2 N$. Let $\delta_2 = \frac{2}{A_2}$, and $\delta = \min(\delta_1, \delta_2)$, we have $GL(u^{k+1}) \leq GL(u^k) \leq C_0 N$ for all $0 \leq dt \leq \delta N$. This proves the second line of the induction in (4.41).

To prove the first line of (4.41), note that since $GL(u^{k+1}) \leq C_0 N$, we can apply the inverse bound Lemma 4.4.3 and get $\|u^{k+1}\|_2 \leq C_1\sqrt{N}$. This completes the induction step. □

In Proposition 4.4.2, we assumed the initial condition u^0 to be in the feasible set V_m. This in general is not done in practice, as u^0 is usually chosen to have binary values $\{-1, 1\}$. The corollary below gives a monotonicity result for u^0 not in the feasible set.

Corollary 4.4.5. *Let u^k be as defined in Proposition 4.4.2. Let u^0 be any vector satisfying $\|u^0\|_\infty \leq 1$. Then exists δ independent of N such that $\forall dt < \delta N^{-3/2}$, $\{u^k\}$ is bounded and $GL(u^k) \leq GL(u^{k+1})$ for $k \geq 1$.*

Proof. Since u^0 is not in the feasible set V_m, $E(u^0) = +\infty \neq GL(u^0)$. However, since $u^1 \in V_m$, we can start the induction from $k = 1$. Since $\|u^1\|_2 \leq \|v^0\|_2 \leq \sqrt{N}$, we can estimate $GL(u^1) \leq B\|u^1\|_2^2 + \sum_{i=1}^N (u^2 - 1)^2 = O(N^2)$. By Lemma 4.4.3, $GL(u) = O(N^2)$ implies $\|u\|_2 = O(N^{3/4})$, hence we can set the induction as below.

$$\begin{cases} GL(u^1) \leq & C_0 N^2, \\ \|u^1\|_2 \leq & C_1 N^{\frac{3}{4}}. \end{cases} \tag{4.42}$$

To prove (4.42), we apply Lemma 4.4.4 and choose $0 \leq dt \leq \delta_1 N^{-3/2}$ so that $\|v^k\|_2 \leq A_1 N^{3/4}$. We then apply Proposition 4.3.2 and estimate

$$\max_{|\xi| \leq A_1 \leq N^{3/4}} |W''(\xi)| \leq A_2 N^{3/2} := M,$$

and set $\delta_2 = \frac{2}{A2}$. By choosing $\delta = \min(\delta_1, \delta_2)$, we prove monotonicity for $0 \leq dt \leq \delta N^{-3/2}$. □

4.4.1 A Counter Example for Graph-Independent Stepsize Restriction

We proved that the spectral truncated scheme is monotone under stepsize range $0 \leq dt \leq \delta = O(N^{-1})$. One would hope to achieve a graph-free stepsize rule as in the case of the original scheme without spectral truncation (4.3). However, as we show in our example below, a constant stepsize to guarantee monotonicity over all graph Laplacian of all sizes is not possible.

Proposition 4.4.6 (Graph Size Dependent Stepsize Restriction)**.** *Define u^k as in (4.32), with $\epsilon = 1$. For any $\delta > 0$ and $dt = \delta N^{-\alpha}, 0 \leq \alpha < 1$, we can always find an unnormalized graph Laplacian $L_{N \times N}$ and some initial condition $\|u^0\|_\infty = 1$ such that the scheme in (4.32) with truncation number $m = 3$ is not monotone in the Ginzburg-Landau energy.*

Remark: $\alpha = 0$ is the case for graph-independent stepsize. However, this result is stronger and claims that dt has to be at least $O(N^{-1})$ for monotonicity to hold for all graphs.

To prove Proposition 4.4.6, we explicitly construct a collection of weighted graphs that require increasingly small stepsizes to guarantee monotonicity as the graph size N increases. The graph is defined in Definition 4.4.7, and illustrated in Fig 4.1.

To give the idea behind the construction, we note that the reason why the maximum principle fails for spectral truncation is because a general orthogonal projection P is expansive in the l^∞ norm. Namely, for some vector $\|v\|_\infty \leq 1$, we have in the worst case $\|P(v)\|_\infty = O(\sqrt{N})$. Our strategy is to explicitly construct a graph such that projection onto one of its eigenspaces P_m attains this worst case l^∞ norm expansion. This is made precise in Proposition 4.4.8.

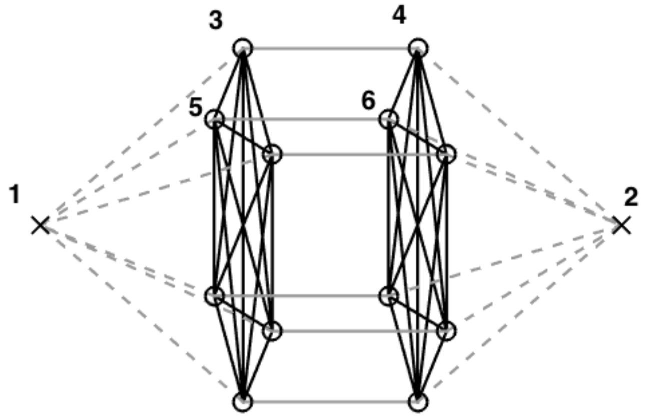


Figure 4.1: Illustration of Worst Case Graph with $N = 7$

Definition 4.4.7. *(Counter Example Graph)*

1. Indexing*: Nodes on the left are indexed by odd numbers and nodes on the right even. The first and the second node(in x) correspond to the left most and right most node respectively. We assume there are N nodes in circles on each side, and hence the graph has a total of* $2N + 2$ *nodes.*

2. Edge Weights*: Connect all nodes to each other within clusters and set edge weights to 10 (black solid edges). Connect the inter cluster nodes in a pairwise fashion and set weights to 1 (gray solid edges). Finally, connect the outlier node with the clusters and set edge weights to* $\frac{\gamma}{N}$, *where* $\gamma = \frac{(1-\frac{1}{\sqrt{N}})(2+\frac{1}{N}-\frac{1}{\sqrt{N}})}{1-\frac{3}{\sqrt{N}}} = 2 + o(1)$ *(gray dashed edges).*

Namely,

$$w_{ij} = \begin{cases} 10, & i,j \textit{ of same parity and} \neq \textit{1,2} \\ 1, & (i,j) = (2k-1,2k) \textit{ or } (2k-1,2k), k \geq 2 \\ \dfrac{\gamma}{N}, & i = 1, j \neq 2 \textit{ or } j = 1, i \neq 2 \end{cases} \tag{4.43}$$

Proposition 4.4.8. *Under the setup above, the second and third eigenvectors of the graph Laplacian are*

$$\begin{aligned} \phi^2 &= (a, a, -\frac{a}{N}, -\frac{a}{N}, \ldots, -\frac{a}{N}, -\frac{a}{N}), \\ \phi^3 &= \left(\frac{1}{2}, -\frac{1}{2}, \frac{1}{2\sqrt{N}}, -\frac{1}{2\sqrt{N}}, \ldots, \frac{1}{2\sqrt{N}}, -\frac{1}{2\sqrt{N}}\right), \end{aligned} \tag{4.44}$$

where $a = \sqrt{\frac{N}{2(N+1)}}$*. Moreover, let* $u^0 = Sign(\phi^3) = (-1, 1, \ldots, -1, 1)$*. Then the projection of* u^0 *onto the eigenspace* $V_3 = span\{\phi^1, \phi^2, \phi^3\}$ *satisfies* $P_3(u^0) = C\sqrt{N}\phi^3$*. Hence* $\|P_3(u^0)\|_\infty = O(\sqrt{N})\|u^0\|_\infty$*.*

We refer the reader to the appendix for the proof of this proposition. Next, we give a proof of Proposition 4.4.6. The idea is that after the first two iterations, the values on u^k, with $k = 2$ on the outlier nodes are arbitrarily higher that that of u^1, and thus the scheme cannot be monotone in the Ginzburg-Landau energy.

Proposition 4.4.6. Define u^k by the spectral truncated scheme (4.32) with $u^0 = Sgn(\phi^3)$ and $dt = \delta N^{-\alpha}$ for some $\delta > 0$ and $0 < \alpha < 1$.

By basic calculations, we have $u^1 = C_0\sqrt{N}\phi^3$, and $u^2 = C_1 N^{\theta/2}\phi^3 + C_2 N^{\frac{\theta-1}{2}}$, where $\theta = 3/2 - \alpha > 1/2$, hence $u^2(1)$ is asymptotically larger than $u^1(1)$ with respect to N. Hence $GL(u^2) > GL(u^1)$ for N large, and the scheme is not monotone in GL for large N. □

4.4.2 Heuristic Explanation for Good Typical Behavior

Despite the pathological behavior of the example given above, the stepsize for spectral truncation does not depend badly on the size N in practice. In this section, we attempt to give a heuristic explanation of this from two viewpoints.

The first view is to analyze the projection operator P_m in the L^∞ norm. The reason why the maximum principle fails is because P_m is expansive in the L^∞ norm. Namely, for some vector $\|v\|_\infty \leq 1$, we have $\|P_m(v)\|_\infty = O(\sqrt{N})$ in the worst case. However, an easy analysis shows the probability of attaining such an $O(\sqrt{N})$ bound decays exponentially as N grows large, as shown in a simplified analysis in Proposition 4.7.3 of the Appendix. Thus in practice, it is very rare that adding P_m would violate the maximum principle"too much".

The second view is to restrict our attention to data that come from a random sample. Namely, we assume our data points x^i are sampled i.i.d. from a probability distribution p, and that the graph Laplacian is computed from the Euclidean distance $\|x^i - x^j\|$. In [VBB08], it is proven that under very general assumptions, the discrete eigenfunctions, eigenvalues converges to continuous limits almost surely. Moreover, the projection operators P_k converges compactly almost surely to their continuous limits. Moreover, results for continuous limits of graph-cut problems can be found in [TSB14]. Under this set up, we can define the Allen-Cahn scheme on the continuous domain and discuss its properties on suitable function spaces. The spectral truncated scheme *still* would not satisfy the maximum principle, but at least it evolves in a sample-size independent fashion. Of course a rigorous proof would require heavy functional analysis.

4.5 Results for Multiclass Classification

The previous analysis can be carried over in a straight forward fashion to the multiclass case.

Multiclass diffuse interface algorithm on graphs can be found in [GFP15,HSB15,MKB13]. We state some basic notations. Let K be the number of classes, and N the number of nodes on the graph. We define $\boldsymbol{u}$ to be an $N \times K$ matrix, where each entry $\boldsymbol{u_{ij}}$ represents the "confidence" of the ith node belonging to the jth class. We think of $\boldsymbol{u}$ as a vector valued function on the graph, and denote its rows by $\boldsymbol{u}(i)$.

The Ginzburg Landau functional for multiclass is defined as

$$GL(\boldsymbol{u}) = \frac{\epsilon}{2} tr(\boldsymbol{u}L\boldsymbol{u}) + \frac{1}{\epsilon}\sum_{i=1}^{N} W(\boldsymbol{u}(i)). \tag{4.45}$$

where $e_k = (0, 0, \ldots, 1, \ldots, 0)^t$, and W is the l^2 "multi-well".

$$W(x) = (\prod_{k=1}^{K} \|\boldsymbol{x} - e^k\|_2^2), \tag{4.46}$$

In [GMB14], a different well function is used using L^1 norms instead of L^2. However, the algorithm in [GMB14] uses a subgradient descent followed by a projection onto the Gibbs simplex. Since the Gibbs simplex itself is already bounded, this renders the boundedness result trivial, and therefore we will only prove the results for the L^2 well. Define $\boldsymbol{W}(\boldsymbol{u}) = \sum_{i=1}^{N} W(\boldsymbol{u}(i))$. We minimize GL by the semi-implicit scheme below

$$\begin{cases} \boldsymbol{v^k} = \boldsymbol{u^k} - dt * \dfrac{1}{2\epsilon}\nabla \boldsymbol{W}(\boldsymbol{u^k}), \\ \boldsymbol{u^{k+1}} = -dt * (\epsilon L\boldsymbol{u^{k+1}}) + \boldsymbol{v^k}. \end{cases} \tag{4.47}$$

The main proposition we prove is this.

Proposition 4.5.1. *Let L be the unnormalized graph Laplacian. Suppose $\boldsymbol{u^0} \in [0,1]^{N\times K}$, and define $\boldsymbol{u^k}$ by the equation (4.47). Then $\exists c$ dependent only on K such that if $0 \le dt \le c$, we have $\boldsymbol{u^k} \in [0,1]^{N\times K}$ for all $k \ge 0$.*

Remark: The choice for $u^k \in [0,1]^{N\times K}$ instead of an l^∞ bound is natural in the multiclass algorithm since we want the final results to have components close to $\{0,1\}$ instead of $\{-1,1\}$.

Proof. Suppose $\boldsymbol{u^k} \in [0,1]^{N\times K}$. Since line 2 of (4.47) is decoupled in columns of $\boldsymbol{u^{k+1}}$, we can apply maximum principle to each column and have $\max \boldsymbol{u^{k+1}} \le \max \boldsymbol{v^k}$, and $\min \boldsymbol{u^{k+1}} \ge \min \boldsymbol{v^k}$. Hence we only have to show $\boldsymbol{v^k} \in [0,1]^{N\times K}$. Since the rows in line 1 of (4.47) are decoupled, we only have to show that the forward map maps each row of $\boldsymbol{v^k}$ to $[0,1]^K$ with $0 \le dt \le c$. This is proven in the lemma below. □

Lemma 4.5.2. *Define $F_{dt} : \mathbb{R}^K \to \mathbb{R}^K$ as $F(x) = x - dt\nabla W(x)$, where W is the multi-well $W(x) = (\prod_{k=1}^K \|x - e^k\|_2^2)$. Then $\exists$ c dependent only on K such that $\forall 0 \leq dt \leq c$, F_{dt} maps $[0,1]^K$ to itself.*

Proof. Given $x \in [0,1]^K$, we denote components of x by x_i. Let $y = F_{dt}(x)$. For each i, $y_i = (1 - 2dt\sum_j G_j(x))x_i + 2dtG_i(x)$, where $G_j(x) = \prod_{k\neq j} \|\boldsymbol{x} - e^k\|_2^2$. We set $\frac{1}{2c} = \max_{x\in[0,1]^K} \sum_j G_j(x)$. Then $\forall 0 \leq dt \leq c$, we have $1 \geq (1 - 2dt\sum_j G_j(x)) \geq 0$. We then prove $y_i \in [0,1]$. For one direction, since $x_i \geq 0$, $y_i \geq 2dtG_j(x) \geq 0$. In the other direction, $y_i \leq 1 - 2dt\sum_j G_j(x) + 2dtG_i(x) \leq 1$. □

Remark: Using the same argument as in previous sections, we can extend the result to incorporate fidelity and also show monotonicity. We omit these discussions for the sake of brevity.

4.6 Numerical Results

In this section, we construct various numerical experiments of increasingly larger scales. This helps demonstrate our theory, and also have some implication on the real world performance of the schemes.

4.6.1 Two Moons

The two moons data was used by Buhler et al [BH09] in exploring spectral clustering with p-Laplacians. It is constructed from sampling from two half circles of radius one on $\mathbb{R}^2$, centered at (0,0) and (1,0.5). Gaussian noise of standard deviation 0.02 in $\mathbb{R}^{100}$ is then added to each of the points. The weight matrix is constructed using Zelnik-Manor and Perona's procedure [ZP04]. Namely, set $w_{ij} = e^{-d(i,j)/\sqrt{\tau_i\tau_j}}$, where τ_i is the Mth closest distance to i. W is further symmetrized by taking the max between two symmetric entries.

Fig. 4.2 is an illustration of the data set of three different sizes being segmented perfectly under a uniform stepsize with 5% fidelity points. The parameters for the experiment is

$dt = 0.5, \epsilon = 1$, which is exactly the tight bound.

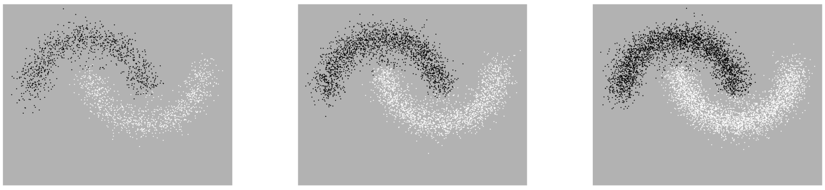


Figure 4.2: Segmentation results under the same stepsize for Two Moons with sample sizes 1000, 2000, 3000 respectively.

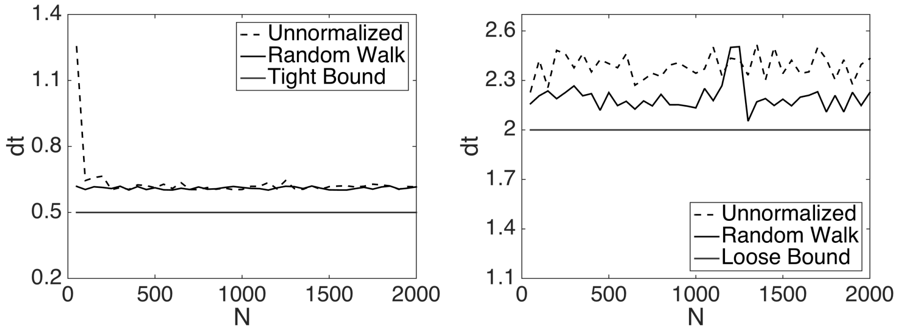


Figure 4.3: Two Moons Segmentation Problem. Left: Maximum stepsize satisfying $\|u^k\|_\infty \leq 1$. Right: Left: Maximum stepsize satisfying $\|u^k\|_\infty \leq 10$. N is the number of nodes.

To test the theory, we compute several "maximum stepsizes" that ensures some criterion (e.g. bounded after 500 iterations, etc.), and compare this with the stepsize predicted by the theory. Bisection with 1e-5 accuracy is used to determine the maximum stepsize that satisfies the criterion given.

Fig 4.3 plots the maximum stepsize for the scheme (4.3) to be bounded by 1.0005, 10 respectively. Random $-1, 1$ initial conditions are chosen. No fidelity terms are added and the diffuse parameter $\epsilon = 1$. We also compute results for the random walk Laplacian and the unnormalized Laplacian as comparison. The actual results are independent of graph size, and also match the tight and loose bound nicely.

In the next experiment, we switch our criterion from boundedness to monotonicity in the function value. Namely, we compute the stepsizes for which the scheme is monotone in 500 iterations.

Fig.4.4 (left) plots the maximum stepsize for the scheme to maintain monotonicity for the three different types of Laplacians. As we can see, the typical maximum stepsize is between the tight and loose bound. Fig.4.4 (right) fixes $N = 2000$ and varies ϵ to plot the relation between dt and ϵ. They are almost linear as predicted by the 0.5ϵ bound.

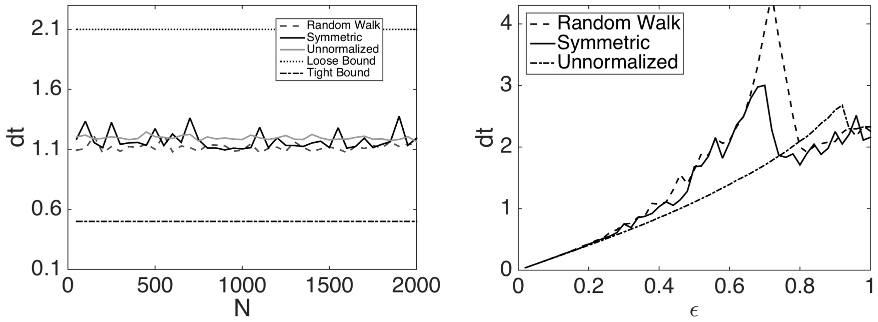


Figure 4.4: Left: Maximum stepsize for monotonicity, fixing $\epsilon = 1$ varying N. Right: Maximum stepsize for monotonicity, fixing $N = 2000$ varying ϵ, ϵ is the interface scale parameter. N is the number of nodes on the graph.

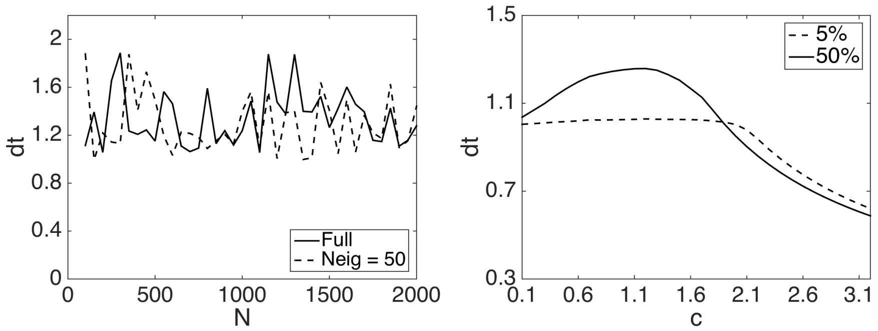


Figure 4.5: Left: Maximum stepsize for monotonicity comparing spectral truncation vs full scheme. Right: Maximum stepsize for monotonicity for scheme with fidelity.N the number of nodes. c is the fidelity strength.

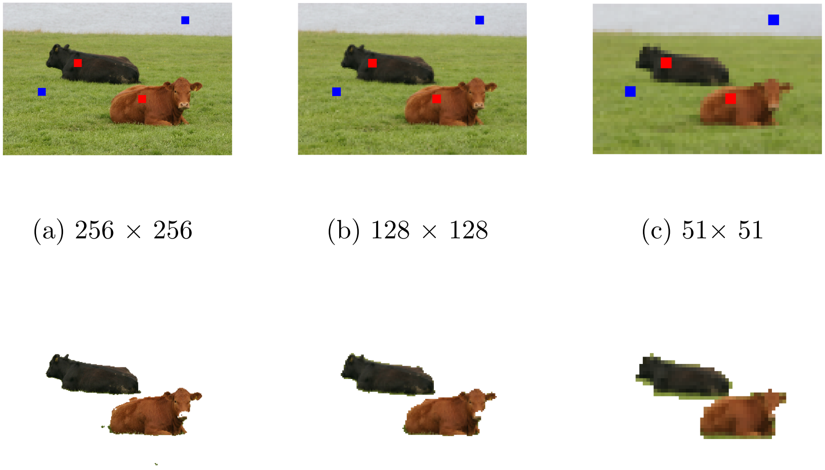
Fig.4.5 (left) plots maximum stepsize for monotonicity for the scheme with spectral truncation. The results are compared with the original scheme without spectral truncation, and we see that the maximum stepsizes are roughly in the same range. Fig.4.5 (right) plots the effects of adding a quadratic fidelity term with power c while keeping $\epsilon = 1$ fixed. As we can see from the result, the fidelity term does constitute an additional restriction when c is large. However, stepsizes remain roughly the same for small c. It is hard to analyze the exact effect when c and ϵ are comparable.

4.6.2 Two Cows

The point of this experiment is to test the effects of Nyström extension on the stepsize and overall performance of the algorithm. Nyström extension is a sampling technique used to approximate eigenvectors without explicitly computing the graph Laplacian [BFC02,FBC04, FBM01].This is very useful because for large dense graphs such as non-local graphs from images, it is often computationally impractical and inefficient to compute the entire weight matrix, and Nyström extension gives a solution to this problem. However, Nyström extension is only approximate, and the following examples show that this imprecision does not impose a great restriction on the stepsize selection.

The images of the two cows are from the Microsoft Database. From the original 312×280 image, we generate 10 images with successily lower resolution of $(312/k) \times (280/k)$, $k = 1, \ldots 10$. A non-local graph constructed from feature windows of size 7×7 is used, and weights are constructed by the standard Gaussian Kernel $w_{ij} = e^{-d_{ij}/\sigma^2}$. The eigenvectors are constructed by using Nyström extension, the details of which could be found in [BF12].

Fig.4.6 illustrates three images with 1,1/2,1/5 times original resolution being segmented under the uniform condition $dt = 2$, $\epsilon = 4$. The blue and red box corresponds to fidelity points of the two classes, the constant in front of the fidelity are $c_1 = 1$ and $c_2 = 0.4$ for the cows and the background respectively. Fig.4.7 is a plot of N vs dt. To ensure segmentation quality, smaller epsilon had to be chosen for images of lower resolution, and the final result is displayed in terms of the dt/ϵ ratio.



(a) 256 × 256

(b) 128 × 128

(c) 51× 51

(d) 256 × 256

(e) 128 × 128

(f) 51× 51

Figure 4.6: Images of different resolution segmented under the same stepsize

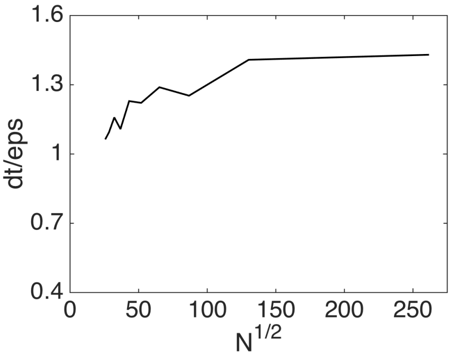


Figure 4.7: Maximum Stepsize for Monotonicity for the Two Cows Under a Series of Different Resolution. N is the number of nodes in the graph, which equals $A \times B$ with A, B the height and width of an image.

4.6.3 MNIST

This experiment is used to demonstrate the case of multiclass clustering by the L^2 multiclass Ginzburg-Landau functional.

The MNIST database [LC98], found at http://yann.lecun.com/exdb/mnist/, is a data set of 70000 28 × 28 images of handwritten digits from 0-9. The graph is constructed by first doing a PCA dimension reduction and again using the same Zelnik-Manor and Perona's procedure with 50 nearest neighbors. For our purpose, we focus on clusters of size three. Table 4.1 shows the limit stepsizes of various tuples, and the error rate when segmented under a uniform stepsize. 5% fidelity points are used, and $\epsilon = 1$. The scheme is projected onto the first 100 eigenvectors. It is shown here that they are still segmented around the same stepsize.

Tuples	{4,6,7}	{3,5,8}	{1,0,9}	{0,6,1}	{2,7,1}
Max dt	0.5823	0.5914	0.5716	0.5701	0.5755
Correct (dt=0.5)	97.98%	97.58%	96.00%	96.36%	98.22%

Table 4.1: Clustering results of MNIST. For each digit, $N \approx 6000$. First Row: triplets of digits to be classified. Second Row: Maximum stepsize for monotonicity. Third Row: Error rate with a fixed dt that is close to the maximum stepsize.

4.7 Supplementary Proofs

Lemma 4.7.1 (Norm Conversions). *Let $1 \leq p < q \leq +\infty$, and $\|u\|_p$ be the vector p-norm $(\sum_i (u(i)^p))^{1/p}$. Then we have the following inequality between different norms:*

$$\|u\|_q \leq \|u\|_p \leq \|u\|_q N^{1/p-1/q}.$$

Proof. The right hand side is by the generalized Holder's inequality. The left hand side is by simple power expansion of multinomials. □

Lemma 4.3.6. Let $S = \{u_0^*, \ldots, u_n^*\}$ be the set of finite limit points for the set $\{u^k\}$. Since S is finite, choose ϵ such that the epsilon balls of the points u_i^* do not overlap. Choose N such that for any $k \geq N$, we have$\|u^{k+1} - u^k\| < \frac{\epsilon}{4}$. By the definition of a limit point, there exists $n' > n > N$ such that $u^n \in B(u_0^*, \epsilon/2)$ and $u^{n'} \in B(u_1^*, \epsilon/2)$. Since $\|u^{k+1} - u^k\| < \frac{\epsilon}{4}$, $\exists n < k < n'$ such that u^k is outside an $\epsilon/2$ neighborhood of S. Since there should be infinitely many such pairs n and n', there are infinitely many points outside the $\epsilon/2$ neighborhood of S, contradicting to S being the only limit points of the set $\{u^k\}$. □

Proposition 4.4.8. Recall that when the graph G is connected, the eigenspace of eigenvalue 0 is spanned by the constant vector $e = (1, 1, \dots, 1)$ [Von07]. To prove Proposition 4.4.8, we first establish a lemma that characterizes the non-constant eigenvectors using symmetries of the graph.

Lemma 4.7.2. *Let L be the unnormalized graph Laplacian defined in Definition 4.4.7. For any eigenvalue $\lambda > 0$ of L, we can always find an eigenvector ϕ with λ as its eigenvalue such that ϕ is in one of the forms below.*

1. $(a, -a, b, -b, \dots, b, -b), a \neq 0$
2. $(a, a, -\frac{a}{N}, -\frac{a}{N}, \dots, -\frac{a}{N}, -\frac{a}{N})$, $a = \sqrt{\frac{N}{2(N+1)}}$.
3. $(0, 0, a, -a, \dots, a, -a), a = \frac{1}{\sqrt{2N}}$
4. $(0, 0, a, -a, -\frac{a}{N-1}, \frac{a}{N-1}, \dots, -\frac{a}{N-1}, \frac{a}{N-1})$, $a = \sqrt{\frac{N-1}{2N}}$.
5. $(0, 0, a, a, -\frac{a}{N-1}, \cdots - \frac{a}{N-1})$, $a = \sqrt{\frac{N-1}{2N}}$.

Proof. Suppose $\phi = (a_0, \bar{a}_0, b_1, \bar{b}_1, \dots, b_N, \bar{b}_N)$ with eigenvalue $\lambda > 0$. Since $e = (1, 1, \dots, 1)$ is an eigenvector of L with eigenvalue 0, we have $\langle \phi, e \rangle = 0$, i.e.

$$\sum_i \phi(i) = 0.$$

Define the eigenspace of engenvalue λ as V_λ. Since the graph is invariant under reflection and permutation of the nodes in the cluster, V_λ is also invariant under these actions. Namely, define

$$R(\phi) = \quad (\bar{a}_0, a_0, \bar{b}_1, b_1, \dots, \bar{b}_N, b_N), \tag{4.48}$$

$$\sigma(\phi) = \quad (a_0, \bar{a}_0, b_{\sigma(1)}, \bar{b}_{\sigma(1)}, \dots, b_{\sigma(N)}, \bar{b}_{\sigma(N)}), \tag{4.49}$$

where σ is any permutation of $1, \dots, N$, then $R(\phi)$ and $\sigma(\phi)$ are also eigenvectors of L with eigenvalue λ. □

Let

$$\xi_0 = \frac{1}{N} \sum_{\sigma \in C(1,N)} \sigma(\phi) = (a_0, \bar{a}_0, b_*, \bar{b}_*, \dots, b_*, \bar{b}_*),$$

where $C(1, N)$ is the cyclic permutation group of index $1, \dots N$, and $b_* = \sum(b_i)/N$. Then either $\xi_0 \neq 0 \in V_\lambda$, or $\xi_0 = (0, 0, \dots 0)$. We discuss each case seperately.

Case 1:($\xi_0 \neq 0$) Denote $\xi_0 = (a, \bar{a}, b, \bar{b}, \dots, b, \bar{b})$. Define $\xi_1 = \frac{1}{2}(\xi_0 + R(\xi_0))$. By the same reasoning, either $\xi_1 = 0$ or $\xi_1 \neq 0 \in V_\lambda$. $\xi_1 = 0$ implies $a = -\bar{a}$, $b = -\bar{b}$, and ξ_0 is of the form 1. If $\xi_1 \neq 0 \in V_\lambda$, ξ_1 is of the form $(a, a, b, b, \dots, b, b)$. Eliminating b the equation $\sum_i \xi_1(i) = 0$ and normalizing, ξ_1 is of form 2.

Case 2:($\xi_0 = 0$) Since $a_0 = 0, \bar{a}_0 = 0$, $\phi = (0, 0, b_1, \bar{b}_1, \dots, b_N, \bar{b}_N)$. Since $\phi_0 \neq 0$, we can WLOG assume b_1 or $\bar{b}_1 \neq 0$. Let

$$\xi_1 = \frac{1}{N-1} \sum_{\sigma \in C(2,N)} \sigma(\phi) := (0, 0, a, \bar{a}, b, \bar{b}, \dots, b, \bar{b}),$$

where $C(2, N)$ is the cyclic permutation group from $2, \dots, N$. $\xi_1 \neq 0$ since $b_1, \bar{b}_1$ are not all zero. Let $\xi_2 = \frac{1}{2}(\xi_1 + R(\xi_1))$. If $\xi_2 = 0$, $a = -\bar{a}$, $b = -\bar{b}$. Define

$$\xi_3 = \frac{1}{N} \sum_{\sigma \in C(1,N)} \sigma(\xi_1) = (0, 0, \frac{a + (N-1)b}{N}, -\frac{a + (N-1)b}{N}, \dots) \tag{4.50}$$

Then $\xi_3 \neq 0$ gives form 3 with $a = 0$, and $\xi_3 = 0$ implies ξ_1 is of form 4. Finally, $\xi_2 \neq 0$ and $\langle e, \xi_2 \rangle = 0$ gives ξ_2 is of form 5.

To prove Proposition 4.4.8, we will show that for the particular weights we have chosen, the minimum Dirichelet energy $\frac{1}{2}\langle u, Lu \rangle$ for the vector forms 1-5 are ordered by $2 < 1 < 3 < 4, 5$, and justify that the vector in form 2 and some vector in form 1 are the second and third engenvectors respectively.

Define γ as in Definition 4.4.7. Recall the variational formulation of the second eigenvector

$$\arg\min_u Dir(u) = \langle u, Lu \rangle \quad s.t. \quad \langle u, e \rangle = 0, \|u\|_2 = 1. \tag{4.51}$$

First, we define χ_*^1 to be the minimizer of (4.51) under the additional constraint $\chi_*^1 = (a, -a, b, -b, \dots, b, -b)$. We are not claiming that χ_*^1 is an eigenvector under this definition now, but will show this afterwards. Writing in terms of a and b, and using the relation

$$\frac{1}{2}\langle u, Lu \rangle = w_{ij}(u(i) - u(j))^2,$$

we have (4.51) is equivalent with

$$\begin{aligned}\min_{a,b} F(a,b) =&(b-a)^2 + 2N\gamma b^2, \\ s.t. \quad & a^2 + Nb^2 = 1/2.\end{aligned} \tag{4.52}$$

Let k be the Lagrange multiplier, the optimality condition is

$$\begin{cases} a = (1 + 2N\gamma + kN)b, \\ b = (1+k)a, \\ 1/2 = a^2 + Nb^2 \end{cases} \tag{4.53}$$

$$k^2 + (\frac{1}{N} + 3\gamma + 1)k + 2\gamma = 0. \tag{4.54}$$

Solving k for $\gamma = \frac{(1-1/\sqrt{N})(2+1/N-1/\sqrt{N})}{1-3\sqrt{N}}$, we have $k = \frac{1}{\sqrt{N}} - 1$, and $a = \sqrt{N}b$. Hence ξ^1_* is equal to the vector ϕ^3 defined on (4.44). Let the $\chi^1 = \chi^1_*$, and $\chi^i, i \geq 2$ be the vectors 2-5 in Lemma 4.7.2. We will find the second eigenvector ϕ^2 by evaluating the Dirichelet energy $\langle \chi^i, L\chi^i \rangle$ for χ^1 to χ^5. Note that since ϕ^2 is an eigenvector, we can WLOG assume ϕ^2 to be in one of χ^i. Computing the Dirichelet energy, we have $Dir(\chi^1) = 1.5 + o(1)$, $Dir(\chi^2) = 1 + o(1)$, $Dir(\chi^3) = 5 + o(1)$, $Dir(\chi^4) = 50 + o(1)$, $Dir(\chi^5) = 50 + o(1)$, and that all the vectors χ^i are in the feasible set $\langle \chi^i, e \rangle = 0$.

This implies χ^2 is the unique second eigenvector of L for N large. Since $\langle \chi^1_*, \chi^2 \rangle = 0$, χ^1_* is the third eigenvector of L since it has the next smallest Dirichelet energy. □

Proposition 4.7.3. *Define the set*

$$M = \{\, u \in \mathbb{R}^N \mid \|u\|_\infty \leq 1, \max_{P_m} \|P_m u\|_\infty \geq C\sqrt{N} \,\},$$

where P_m is any projection operator onto a subspace, and $0 < C < 1$. Then the volume(with respect to the standard L^2 metric in R^N) of the set M decreases exponentially with respect to the number of dimensions N.

The proposition shows that if u were sampled uniformly from a unit cube, then the probability of some projection P_m expanding the max norm by a factor of $O(\sqrt{N})$ is exponentially decreasing.

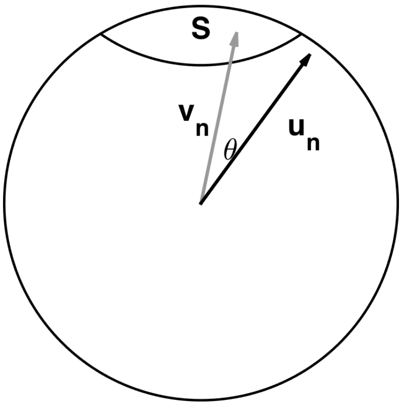


Figure 4.8: Illustration of Proposition 4.7.3. S is one of the "caps" that v_n resides in. u_n and v_n have angle less than θ.

Proof. Let $u \in M$. Then by definition of the set M, $\exists$ some projection P_m such that $\|P_m u\|_\infty \geq C\sqrt{N}$. Define $v := P_m u$ and $v_n := \frac{v}{\|v\|_2}$. Define $u_n := \frac{u}{\|u\|_2}$. Since v_n is the projected direction of u, $P_m u = \langle u, v_n \rangle v_n$. Then we have

$$C\sqrt{N} \leq \|P_m u\|_\infty = \langle u, v_n \rangle \|v_n\|_\infty = \|u\|_2 \|v_n\|_\infty \langle u_n, v_n \rangle.$$

Since $\|u\|_2 \leq \sqrt{N}$, we have

$$\|v_n\|_\infty \langle u_n, v_n \rangle \geq C. \tag{4.55}$$

Since $\langle u_n, v_n \rangle \leq 1$, the projected direction v_n must be in the set $S = \{v \mid \|v\|_2 = 1, \|v\|_\infty \geq C\}$. However, the set S contains the N "caps" of a unit sphere (see Fig.4.8), and hence is exponentially decreasing in volume with respect to the sphere. On the other hand, since $\|v_n\|_\infty \leq 1$, by (4.55) we have $\langle u_n, v_n \rangle \geq C$, and thus u lies in a cone $K(v_n)$ with angle $cos(\theta) \geq C$. Hence $u \in K_v + N$, and since cones K_v have volume exponentially decreasing with respect to N as well, we have $Vol(M)$ is exponentially decreasing with respect to N.

□

CHAPTER 5

Graph-based Uncertainty Quantification

5.1 The Bayesian framework

In this chapter, we introduce a Bayesian framework for quantifying uncertainty on graph-based learning, and develop efficient algorithms for inference. Note that most graph-based learning methods could be written as an optimization of the form

$$\min_{w} \mathsf{J}(w) = \frac{1}{2}\langle w, Pw\rangle + \Phi(w),$$

where P is a power of the graph Laplacian L. There are various choices of the graph Laplacian, as discussed in the introduction. One can refer to [BF12, Von07] for a full discussion. We will work exclusively with the normalized Laplacian, defined as

$$L = I - D^{-1/2}AD^{-1/2}, \tag{5.1}$$

We define $\{\lambda_j\}$ to the the eigenvalues of the Laplacian L in sorted order, and $\{q_j\}$ be the corresponding eigenvectors, i.e.,

$$\lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{N-1} \leq \lambda_{\max} < \infty, \quad \langle q_j, q_k\rangle = \delta_{jk}. \tag{5.2}$$

The eigenvector corresponding to $\lambda_0 = 0$ is $q_0 = D^{\frac{1}{2}}\mathbb{1}$ and $\lambda_1 > 0$, assuming a fully connected graph. Then $L = Q\Lambda Q^*$ where Q has columns $\{q_k\}_{k=0}^{N-1}$ and Λ is a diagonal matrix with entries $\{\lambda_k\}_{k=0}^{N-1}$. Using these eigenpairs the graph Dirichlet energy can be written as

$$\frac{1}{2}\langle u, Lu\rangle = \frac{1}{2}\sum_{j=1}^{N-1} \lambda_j(\langle u, q_j\rangle)^2; \tag{5.3}$$

this is analogous to decomposing the classical Dirichlet energy using Fourier analysis. The classical Dirichlet energy penalizes non-smooth functions over the continuous domain, and similar conclusions also hold on the graph domain as well. In the next section, we will define a Gaussian prior measure based on the graph Dirichlet energy that is biased towards smoother functions on the graph.

5.1.1 Gaussian Prior Measure

We now show how to build a Gaussian distribution with negative log density proportional to $J_0(u)$. Such a Gaussian prefers functions that have larger components on the first few eigenvectors of the graph Laplacian, where the eigenvalues of L are smaller. The corresponding eigenvectors carry rich geometric information about the weighted graph. For example, the second eigenvector of L is the *Fiedler vector* and solves a relaxed normalized min-cut problem [Von07, HK04]. The Gaussian distribution thereby connects geometric intuition embedded within the graph Laplacian to a natural probabilistic picture.

To make this connection concrete we define diagonal matrix Σ with entries defined by the vector

$$(0, \lambda_1^{-1}, \cdots, \lambda_{N-1}^{-1})$$

and define the positive semi-definite covariance matrix $C = cQ\Sigma Q^*$; choice of the scaling c will be discussed below. We let $\mu_0 := \mathcal{N}(0, C)$. Note that the covariance matrix is that of a Gaussian with variance proportional to λ_j^{-1} in direction q_j thereby leading to structures which are more likely to favour the Fiedler vector ($j = 1$), and lower values of j in general, than it does for higher values. The fact that the first eigenvalue of C is zero ensures that any draw from μ_0 changes sign, because it will be orthogonal to q_0.[1] To make this intuition explicit we recall the Karhunen-Loeve expansion which constructs a sample u from the Gaussian μ_0 according to the random sum

$$u = c^{\frac{1}{2}} \sum_{j=1}^{N-1} \lambda_j^{-\frac{1}{2}} q_j z_j, \tag{5.4}$$

where the $\{z_j\}$ are i.i.d. $\mathcal{N}(0, 1)$. Equation (5.2) thus implies that $\langle u, q_0 \rangle = 0$.

We choose the constant of proportionality c as a rescaling which enforces the property

[1] Other treatments of the first eigenvalue are possible and may be useful but for simplicity of exposition we do not consider them in the scope of this chapter.

$\mathbb{E}|u|^2 = N$ for $u \sim \mu_0 := \mathcal{N}(0, C)$; in words the per-node variance is 1. Note that, using the orthogonality of the $\{q_j\}$,

$$\mathbb{E}|u|^2 = c\sum_{j=1}^{N-1} \lambda_j^{-1}\mathbb{E}z_j^2 = c\sum_{j=1}^{N-1}\lambda_j^{-1} \quad \Longrightarrow \quad c = N\Big(\sum_{j=1}^{N-1}\lambda_j^{-1}\Big)^{-1}. \tag{5.5}$$

We reiterate that the support of the measure μ_0 is the space $U := q_0^{\perp} = \operatorname{span}\{q_1, \cdots, q_{N-1}\}$ and that, on this space, the probability density function is proportional to

$$\exp\Big(-c^{-1}J_0(u)\Big) = \exp\Big(-\frac{1}{2c}\langle u, Lu\rangle\Big),$$

so that the *precision matrix* of the Gaussian is $P = c^{-1}L$. In what follows the sign of u will be related to the classification; since all the entries of q_0 are positive, working on the space U ensures a sign change in u, and hence a non-trivial classification.

5.1.2 Extension to the Ginzburg-Landau Prior

The Gaussian measure discussed in the section above can be extended to non-Gaussian cases as well. We discuss one such possibility by introducing the Ginzburg-Landau prior, a measure that pushes values close to the binary labels $\{-1, 1\}$.

Let μ_0 be the prior Gaussian measure defined in the section above. We define a measure ν_0 such that

$$\frac{d\nu_0}{d\mu_0}(v) \propto e^{-\sum_{j\in Z} W_\epsilon(v(j))}. \tag{5.6}$$

We name ν_0 the Ginzburg-Landau measure, since the negative log density function of ν_0 is the graph Ginzburg-Landau functional

$$\mathsf{GL}(v) := \frac{1}{2c}\langle v, Lv\rangle + \sum_{j\in Z} W_\epsilon(v(j)). \tag{5.7}$$

The motivation for considering such a measure is the following: For the models considered in this chapter, the label space of the problem is discrete while the latent variable u through which we will capture the correlations amongst nodes of the graph, encoded in the feature vectors, is real-valued. Ginzburg-Landau allows for a smooth relaxation of thresholding that is tighter than relaxing the domain of u to the entire real line. We make precise this connection in the paragraphs below.

Define the (signum) function $S : \mathbb{R} \mapsto \{-1, 1\}$ by

$$S(u) = 1,\ u \geq 0 \quad \text{and} \quad S(u) = -1,\ u < 0.$$

The function S may be relaxed by defining $S_\epsilon(u) = v|_{t=1}$ where v solves the gradient flow

$$\dot{v} = -\nabla W_\epsilon(v), \quad v|_{t=0} = u \qquad \text{for potential} \qquad W_\epsilon(v) = \frac{1}{4\epsilon}(v^2 - 1)^2.$$

Note that $S_\epsilon(\cdot) \to S(\cdot)$, pointwise, as $\epsilon \to 0$, on $\mathbb{R}\backslash\{0\}$. This reflects the fact that the gradient flow minimizes W_ϵ, asymptotically as $t \to \infty$, whenever started on $\mathbb{R}\backslash\{0\}$.

We have introduced a Gaussian measure μ_0 on the latent variable u which lies in $U \subset \mathbb{R}^N$; we now want to introduce two ways of constructing non-Gaussian measures on the label space $\{-1, 1\}^N$, or on real-valued relaxations of label space, building on the measure μ_0. The first is to consider the push-forward of measure μ_0 under the map S: $S^\sharp\mu_0$. When applied to a sequence $l : Z \mapsto \{-1, 1\}^N$ this gives

$$\big(S^\sharp\mu_0\big)(l) = \mu_0\Big(\{u|S(u(j)) = l(j), \forall 1 \leq j \leq N\}\Big),$$

recalling that N is the cardinality of Z. The definition is readily extended to components of l defined only on subsets of Z. Thus $S^\sharp\mu_0$ is a measure on the label space $\{-1, 1\}^N$. The second approach is to work with a change of measure from the Gaussian μ_0 in such a way that the probability mass on $U \subset \mathbb{R}^N$ concentrates close to the label space $\{-1, 1\}^N$. We may achieve this by defining the measure ν_0 via its Radon-Nykodim derivative

$$\frac{d\nu_0}{d\mu_0}(v) \propto e^{-\sum_{j\in Z} W_\epsilon(v(j))}, \tag{5.8}$$

which is exactly the Ginzburg-Landau measure. The Ginzburg-Landau distribution defined by ν_0 can be interpreted as a non-convex ground relaxation of the discrete MRF model [Zhu], in contrast to the convex relaxation which is the Gaussian Field [ZGL03]. Since the double well has minima at the label values $\{-1, 1\}$, the probability mass of ν_0 is concentrated near the modes ±1, and ϵ controls this concentration effect.

5.1.3 Likelihood Function and Model

In any Bayesian framework, a likelihood function must be chosen to link the data y to the prior. The choice of which Bayesian likelihood to use is related to the data itself, and making this choice is beyond the scope of our discussion. Currently the choice must be addressed on a case by case basis, as is done when choosing an optimization method for classification. Nonetheless we will demonstrate that the shared structure of many of these likelihood means that a common algorithmic framework can be adopted and we will make some conclusions about the relative costs of applying this framework to the models.

In Figure 5.1 we plot the component of the negative log likelihood at a labelled node j, as a function of the latent variable $u = u(j)$ with data $y = y(j)$ fixed, for the probit and Bayesian level-set models. The log likelihood for the Ginzburg-Landau formulation is not directly comparable as it is a function of the relaxed label variable $v(j)$, with respect to which it is quadratic with minimum at the data point $y(j)$.

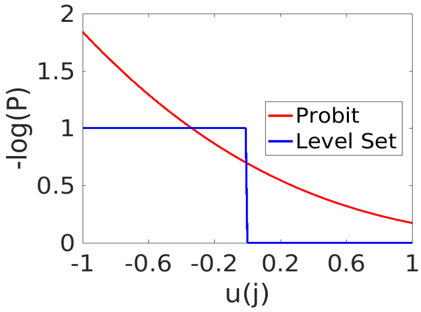


Figure 5.1: Plot of a component of the negative log likelihood for a fixed node j. We set $\gamma = 1/\sqrt{2}$ for probit and Bayesian level-set. Since $\Phi(u(j); 1) = \Phi(-u(j); -1)$ for probit and Bayesian level-set, we omit the plot for $y(j) = -1$.

We denote the latent variable by $u(j)$, $j \in Z$, the thresholded value of $u(j)$ by $l(j) = S(u(j))$ which is interpreted as the label assignment at each node j, and noisy observations of the binary labels by $y(j)$, $j \in Z'$. The variable $v(j)$ will be used to denote the real-valued relaxation of $l(j)$ used for the Ginzburg-Landau model. Recall Bayes formula which transforms a prior density $\mathbb{P}(u)$ on a random variable u into a posterior density $\mathbb{P}(u|y)$ on the conditional random variable $u|y$:

$$\mathbb{P}(u|y) = \frac{1}{\mathbb{P}(y)}\mathbb{P}(y|u)\mathbb{P}(u).$$

We will now apply this formula to condition our graph latent variable u, whose thresholded values correspond to labels, on the noisy label data y given at Z'. As prior on u we will always use $\mathbb{P}(u)du = \mu_0(du)$; we will describe two different likelihoods. We will also apply the formula to condition relaxed label variable v, on the same label data y, via the formula

$$\mathbb{P}(v|y) = \frac{1}{\mathbb{P}(y)}\mathbb{P}(y|v)\mathbb{P}(v).$$

We will use as prior the non-Gaussian $\mathbb{P}(v)dv = \nu_0(dv)$. The probit and Bayesian level-set models lead to posterior distributions μ (with different subscripts) in latent variable space, and pushforwards under S, denoted ν (also with different subscripts), in label space. The Ginzburg-Landau formulation leads to a measure ν in (relaxed) label space.

In the sections below, we will state the prior, likelihood, and MAP estimator for three Bayesian models considered in this chapter.

5.1.3.1 Probit

The probit method is designed for classification and is described in [WR96]. In that context Gaussian process priors are used and, unlike the graph Laplacian construction used here, do not depend on the unlabel data. Combining Gaussian process priors and graph Laplacian priors was suggested and studied in [BNS06,SBN06,LP11]. A recent fully Bayesian treatment of the methodology using unweighted graph Laplacians may be found in the paper [HZ16]. In detail our model is as follows.

Prior We take as prior on u the Gaussian μ_0. Thus

$$\mathbb{P}(u) \propto \exp\Big(-\frac{1}{2}\langle u, Pu\rangle\Big).$$

Likelihood For any $j \in Z'$

$$y(j) = S\Big(u(j) + \eta(j)\Big)$$

with the $\eta(j)$ drawn i.i.d from $\mathcal{N}(0,\gamma^2)$. We let

$$\Psi(v;\gamma) = \frac{1}{\sqrt{2\pi\gamma^2}}\int_{-\infty}^{v} \exp\big(-t^2/2\gamma^2\big)dt,$$

the cumulative distribution function (cdf) of $\mathcal{N}(0,\gamma^2)$, and note that then

$$\mathbb{P}\big(y(j) = 1|u(j)\big) = \mathbb{P}\big(\mathcal{N}(0,\gamma^2) > -u(j)\big) = \Psi(u(j);\gamma) = \Psi(y(j)u(j);\gamma);$$

similarly

$$\mathbb{P}\big(y(j)=-1|u(j)\big)=\mathbb{P}\big(\mathcal{N}(0,\gamma^2)<-u(j)\big)=\Psi(-u(j);\gamma)=\Psi(y(j)u(j);\gamma).$$

Posterior Bayes' Theorem gives posterior μ_{p} with probability density function (pdf)

$$\mathbb{P}_{\mathrm{p}}(u|y)\propto\exp\Big(-\frac{1}{2}\langle u,Pu\rangle-\Phi_{\mathrm{p}}(u;y)\Big)$$

where

$$\Phi_{\mathrm{p}}(u;y):=-\sum_{j\in Z'}\log\big(\Psi(y(j)u(j);\gamma)\big).$$

We let ν_{p} denote the push-forward under S of $\mu_{\mathrm{p}}:\nu_{\mathrm{p}}=S^\sharp\mu_{\mathrm{p}}$.

MAP Estimator This is the minimizer of the negative of the log posterior. Thus we minimize the following objective function over U:

$$\mathsf{J}_{\mathrm{p}}(u)=\frac{1}{2}\langle u,Pu\rangle-\sum_{j\in Z'}\log\Big(\Psi(y(j)u(j);\gamma)\Big).$$

This is a convex function, a fact which is well-known in related contexts, but which we state and prove in Proposition 1 Section 2 of the supplementary materials for the sake of completeness. In view of the close relationship between this problem and the level-set formulation described next, for which there are no minimizers, we expect that minimization may not be entirely straightforward in the $\gamma\ll 1$ limit. This is manifested in the presence of near-flat regions in the probit log likelihood function when $\gamma\ll 1$.

Our variant on the probit methodology differs from that in [HZ16] in several ways: (i) our prior Gaussian is scaled to have per-node variance one, whilst in [HZ16] the per node variance is a hyper-parameter to be determined; (ii) our prior is supported on $U=q_0^\perp$ whilst in [HZ16] the prior precision is found by shifting L and taking a possibly fractional power of the resulting matrix, resulting in support on the whole of $\mathbb{R}^N$; (iii) we allow for a scale parameter γ in the observational noise, whilst in [HZ16] the parameter $\gamma=1$.

5.1.3.2 Level-Set

This method is designed for problems considerably more general than classification on a graph [ILS15]. For the current application, this model is exactly the same as probit except for the order in which the noise $\eta(j)$ and the thresholding function $S(u)$ is applied in the definition of the data. Thus we again take as **Prior** for u, the Gaussian μ_0. Then we have:

Likelihood For any $j \in Z'$

$$y(j) = S\big(u(j)\big) + \eta(j)$$

with the $\eta(j)$ drawn i.i.d from $\mathcal{N}(0, \gamma^2)$. Then

$$\mathbb{P}\Big(y(j)|u(j)\Big) \propto \exp\Big(-\frac{1}{2\gamma^2}|y(j) - S\big((u(j)\big)|^2\Big).$$

Posterior Bayes' Theorem gives posterior $\mu_{\rm ls}$ with pdf

$$\mathbb{P}_{\rm ls}(u|y) \propto \exp\Big(-\frac{1}{2}\langle u, Pu\rangle - \Phi_{\rm ls}(u;y)\Big)$$

where

$$\Phi_{\rm ls}(u;y) = \sum_{j\in Z'}\Big(\frac{1}{2\gamma^2}|y(j) - S\big(u(j)\big)|^2\Big).$$

We let $\nu_{\rm ls}$ denote the pushforward under S of $\mu_{\rm ls}$: $\nu_{\rm ls} = S^\sharp \mu_{\rm ls}$.

MAP Estimator Functional The negative of the log posterior is, in this case, given by

$$\mathsf{J}_{\rm ls}(u) = \frac{1}{2}\langle u, Pu\rangle + \Phi_{\rm ls}(u;y).$$

However, unlike the probit model, the Bayesian level-set method has no MAP estimator – the infimum of $\mathsf{J}_{\rm ls}$ is not attained and this may be seen by noting that, if the infumum was attained at any non-zero point $u^\star$ then $\epsilon u^\star$ would reduce the objective function for any $\epsilon \in (0,1)$; however the point $u^\star = 0$ does not attain the infimum. This proof is detailed in [ILS15] for a closely related PDE based model, and the proof is easily adapted.

5.1.3.3 Ginzburg-Landau

For this model, we take as prior the Ginzburg-Landau measure ν_0 defined by (5.6), and employ a Gaussian likelihood for the observed labels. This construction gives the Bayesian posterior whose MAP estimator is the objective function introduced and studied in [BF12].

Prior We define prior on v to be the Ginzburg-Landau measure ν_0 given by (5.6) with density

$$\mathbb{P}(v) \propto e^{-\mathsf{GL}(v)}.$$

Likelihood For any $j \in Z'$

$$y(j) = v(j) + \eta(j)$$

with the $\eta(j)$ drawn i.i.d from $\mathcal{N}(0, \gamma^2)$. Then

$$\mathbb{P}\Big(y(j)|v(j)\Big) \propto \exp\Big(-\frac{1}{2\gamma^2}|y(j) - v(j)|^2\Big).$$

Posterior Recalling that $P = c^{-1}L$ we see that Bayes' Theorem gives posterior $\nu_{\rm gl}$ with pdf

$$\begin{aligned} \mathbb{P}_{\rm gl}(v|y) &\propto \exp\Big(-\frac{1}{2}\langle v, Pv\rangle - \Phi_{\rm gl}(v; y)\Big), \\ \Phi_{\rm gl}(v; y) &:= \sum_{j\in Z} W_\epsilon\big(v(j)\big) + \sum_{j\in Z'}\Big(\frac{1}{2\gamma^2}|y(j) - v(j)|^2\Big)\Big). \end{aligned}$$

MAP Estimator This is the minimizer of the negative of the log posterior. Thus we minimize the following objective function over U:

$$\mathsf{J}_{\rm gl}(v) = \frac{1}{2}\langle v, Pv\rangle + \Phi_{\rm gl}(v; y).$$

This objective function was introduced in [BF12] as a relaxation of the min-cut problem, penalized by data; the relationship to min-cut was studied rigorously in [VB12]. The minimization problem for $\mathsf{J}_{\rm gl}$ is non-convex and has multiple minimizers, reflecting the combinatorial character of the min-cut problem of which it is a relaxation.

5.1.4 Uncertainty Quantification for Graph Based Learning

Uncertainty quantification for semi-supervised learning is concerned with completely characterizing these posterior distributions. In practice this may be achieved by sampling using MCMC methods. In this section we will numerically study four measures of uncertainty:

- The empirical pdfs of the latent and label variables at certain nodes;
- The posterior mean of the label variables at certain nodes;
- The posterior variance of the label variables averaged over all nodes;
- The posterior mean or variance to order nodes into those whose classifications are most uncertain and those which are most certain.

For the probit and Bayesian level-set models we interpret the thresholded variable $l = S(u)$ as the binary label assignments corresponding to a real-valued configuration u; for Ginzburg-Landau we may simply take $l = v$ as the model is posed on (relaxed) label space. The node-wise posterior mean of l can be used as a useful confidence score of the class assignment of each node. The node-wise posterior mean s^l_j is defined as

$$s^l_j := \mathbb{E}_\nu(l(j)), \tag{5.9}$$

with respect to any of the posterior measures ν in label space. Note that for probit and Bayesian level set $l(j)$ is a binary random variable taking values in $\{\pm 1\}$ and we have $s^l_j \in [-1, 1]$. In this case if $q = \nu(l(j) = 1)$ then $q = \frac{1}{2}(1 + s^l_j)$. Furthermore

$$\mathrm{Var}_\nu(l(j)) = 4q(1 - q) = 1 - (s^l_j)^2.$$

Later we will find it useful to consider the variance averaged over all nodes and hence define[2]

$$\mathrm{Var}(l) = \frac{1}{N}\sum_{j=1}^{N} \mathrm{Var}_\nu(l(j)). \tag{5.10}$$

Note that the maximum value obtained by Var(l) is 1. This maximum value is attained under the Gaussian prior μ_0 that we use in this chapter. The deviation from this maximum under the posterior is a measure of the information content of the labelled data. Note, however, that the prior does contain information about classifications, in the form of correlations between vertices; this is not captured in (5.10).

5.2 MCMC Algorithms

From Section 5.1.3, we see that for all of the models considered, the posterior $\mathbb{P}(w|y)$ has the form

$$\mathbb{P}(w|y) \propto \exp\bigl(-\mathsf{J}(w)\bigr), \quad \mathsf{J}(w) = \frac{1}{2}\langle w, Pw\rangle + \Phi(w))$$

for some function Φ, different for each of the three models (acknowledging that in the Ginzburg-Landau case the independent variable is $w = v$, real-valued relaxation of label

[2] Strictly speaking $\mathrm{Var}(l) = N^{-1}\mathrm{Tr}\bigl(\mathrm{Cov}(l)\bigr)$.

space, whereas for the other models $w = u$ an underlying latent variable which may be thresholded by $S(\cdot)$ into label space.) The sampler we employ does not use information about the gradient of Φ; the MAP estimation algorithm does, but is only employed on the Ginzburg-Landau and probit models. Both sampling and optimization algorithms use spectral properties of the precision matrix P, which is proportional to the graph Laplacian L.

Broadly speaking there are two strong competitors as samplers for this problem: Metropolis-Hastings based methods, and Gibbs based samplers. In this chapter we focus entirely on Metropolis-Hastings methods as they may be used on all three models considered here. In order to induce scalability with respect to size of Z we use the preconditioned Crank-Nicolson (pCN) method described in [CRS13] and introduced in the context of diffusions by Beskos et. al. in [BRS08] and by Neal in the context of machine learning [Nea]. The method is also robust with respect to the small noise limit $\gamma \to 0$ in which the label data is perfect. The pCN based approach is compared with Gibbs like methods for probit, to which they both apply, in [A 18]; both large data sets $N \to \infty$ and small noise $\gamma \to 0$ limits are considered.

5.2.1 pCN

The standard random walk Metropolis (RWM) algorithm suffers from the fact that the optimal proposal variance or stepsize scales inverse proportionally to the dimension of the state space [RGG97], which is the graph size N in this case. The pCN method is designed so that the proposal variance required to obtain a given acceptance probability scales independently of the dimension of the state space (here the number of graph nodes N), hence in practice giving faster convergence of the MCMC when compared with RWM [BRS09]. We restate the pCN method as Algorithm 2, and then follow with various variants on it in Algorithms 3 and 4. In all three algorithms $\beta \in [0, 1]$ is the key parameter which determines the efficiency of the MCMC method: small β leads to high acceptance probability but small moves; large β leads to low acceptance probability and large moves. Somewhere between these extremes is an optimal choice of β which minimizes the asymptotic variance of the algorithm when applied to compute a given expectation.

Algorithm 2 pCN Algorithm

```
Input: $L$. $\Phi(u)$. $u^{(0)} \in U$.
Output: $M$ Approximate samples from the posterior distribution
Define: $\alpha(u,w) = \min\{1, \exp(\Phi(u) - \Phi(w)\}$.
while $k < M$ do
    $w^{(k)} = \sqrt{1-\beta^2}u^{(k)} + \beta\xi^{(k)}$, where $\xi^{(k)} \sim \mathcal{N}(0,C)$ via equation (5.11).
    Calculate acceptance probability $\alpha(u^{(k)}, w^{(k)})$.
    Accept $w^{(k)}$ as $u^{(k+1)}$ with probability $\alpha(u^{(k)}, w^{(k)})$, otherwise $u^{(k+1)} = u^{(k)}$.
end while
```

The value $\xi^{(k)}$ is a sample from the prior μ_0. If the eigenvalues and eigenvectors of L are all known then the Karhunen-Loeve expansion (5.11) gives

$$\xi^{(k)} = c^{\frac{1}{2}} \sum_{j=1}^{N-1} \lambda_j^{-\frac{1}{2}} q_j z_j, \tag{5.11}$$

where c is given by (5.5), the $z_j, j = 1 \ldots N-1$ are i.i.d centred unit Gaussians and the equality is in law.

5.2.2 Low Rank Approximations of the Graph Laplacian

For graphs with a large number of nodes N, it is prohibitively costly to directly sample from the distribution μ_0, since doing so involves knowledge of a complete eigen-decomposition of L, in order to employ (5.11). We discuss two strategies below to lower the computational overhead of computing the entire spectral decomposition of the graph Laplacian L.

5.2.2.1 Spectral Truncation

A method that is frequently used in classification tasks is to restrict the support of u to the eigenspace spanned by the first ℓ eigenvectors with the smallest non-zero eigenvalues of L (hence largest precision) and this idea may be used to approximate the pCN method; this leads to a low rank approximation. In particular we approximate samples from μ_0 by

$$\xi_\ell^{(k)} = c_\ell^{\frac{1}{2}} \sum_{j=1}^{\ell-1} \lambda_j^{-\frac{1}{2}} q_j z_j, \tag{5.12}$$

where c_ℓ is given by (5.5) truncated after $j = \ell - 1$, the z_j are i.i.d centred unit Gaussians and the equality is in law. This is a sample from $\mathcal{N}(0, C_\ell)$ where $C_\ell = c_\ell Q \Sigma_\ell Q^*$ and the diagonal entries of Σ_ℓ are set to zero for the entries after ℓ. In practice, to implement this algorithm, it is only necessary to compute the first ℓ eigenvectors of the graph Laplacian L. This gives Algorithm 3.

Algorithm 3 pCN Algorithm With Spectral Projection

```
Input: $L$. $\Phi(u)$. $u^{(0)} \in U$.
Output: $M$ Approximate samples from the posterior distribution
Define: $\alpha(u, w) = \min\{1, \exp(\Phi(u) - \Phi(w)\}$.
**while** $k < M$ **do**
    $w^{(k)} = \sqrt{1-\beta^2} u^{(k)} + \beta \xi_\ell^{(k)}$, where $\xi_\ell^{(k)} \sim \mathcal{N}(0, C_\ell)$ via equation (5.12).
    Calculate acceptance probability $\alpha(u^{(k)}, w^{(k)})$.
    Accept $w^{(k)}$ as $u^{(k+1)}$ with probability $\alpha(u^{(k)}, w^{(k)})$, otherwise $u^{(k+1)} = u^{(k)}$.
**end while**
```

5.2.2.2 Spectral Approximation

Spectral projection often leads to good classification results, but may lead to reduced posterior variance and a posterior distribution that is overly smooth on the graph domain. We propose an improvement on the method that preserves the variability of the posterior distribution but still only involves calculating the first ℓ eigenvectors of L. This is based on the empirical observation that in many applications the spectrum of L saturates and satisfies, for $j \geq \ell$, $\lambda_j \approx \bar{\lambda}$ for some $\bar{\lambda}$. Such behaviour may be observed in b), c) and d) of Figure 5.2; in particular note that in the hyperspectal case $\ell \ll N$. We assume such behaviour in deriving the low rank approximation used in this subsection. (See supplementary materials for a detailed discussion of the graph Laplacian spectrum.) We define $\Sigma_{\ell,o}$ by overwriting the diagonal entries of Σ from ℓ to $N-1$ with $\bar{\lambda}^{-1}$. We then set $C_{\ell,o} = c_{\ell,o} Q \Sigma_{\ell,o} Q^*$, and generate samples from $\mathcal{N}(0, C_{\ell,o})$ (which are approximate samples from μ_0) by setting

$$\xi_{\ell,o}^{(k)} = c_{\ell,o}^{\frac{1}{2}} \sum_{j=1}^{\ell-1} \lambda_j^{-\frac{1}{2}} q_j z_j + c_{\ell,o}^{\frac{1}{2}} \bar{\lambda}^{-\frac{1}{2}} \sum_{j=\ell}^{N-1} q_j z_j, \tag{5.13}$$

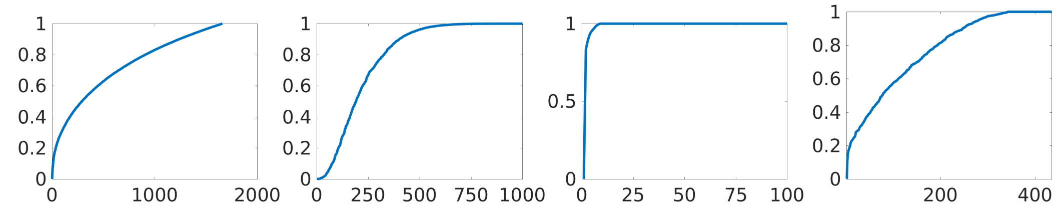
where $c_{\ell,o}$ is given by (5.5) with λ_j replaced by $\bar{\lambda}$ for $j \geq \ell$, the $\{z_j\}$ are centred unit Gaussians, and the equality is in law. Importantly samples according to (5.13) can be computed very efficiently. In particular there is no need to compute q_j for $j \geq \ell$, and the quantity $\sum_{j=\ell}^{N-1} q_j z_j$ can be computed by first taking a sample $\bar{z} \sim \mathcal{N}(0, I_N)$, and then projecting $\bar{z}$ onto $U_\ell := \operatorname{span}(q_\ell, \dots, q_{N-1})$. Moreover, projection onto U_ℓ can be computed only using $\{q_1, \dots, q_{\ell-1}\}$, since the vectors span the orthogonal complement of U_ℓ. Concretely, we have

$$\sum_{j=\ell}^{N-1} q_j z_j = \bar{z} - \sum_{j=1}^{\ell-1} q_j \langle q_j, \bar{z} \rangle,$$

where $\bar{z} \sim \mathcal{N}(0, I_N)$ and equality is in law. Hence the samples $\xi_{\ell,o}^{(k)}$ can be computed by

$$\xi_{\ell,o}^{(k)} = c_{\ell,o}^{\frac{1}{2}} \sum_{j=1}^{\ell-1} \lambda_j^{-\frac{1}{2}} q_j z_j + c_{\ell,o}^{\frac{1}{2}} \bar{\lambda}^{-\frac{1}{2}} \Big(\bar{z} - \sum_{j=1}^{\ell-1} q_j \langle q_j, \bar{z} \rangle \Big). \tag{5.14}$$

The vector $\xi_{\ell,o}^{(k)}$ is a sample from $\mathcal{N}(0, C_{\ell,o})$ and results in Algorithm 4. Under the stated empirical properties of the graph Laplacian, we expect this to be a better approximation of the prior covariance structure than the approximation of the previous subsection.



(a) MNIST49 (b) Two Moons (c) Hyperspectral (d) Voting Records

Figure 5.2: Spectra of graph Laplacian of various datasets. See Sec.5.3 for the description of the datsets and graph construction parameters. The $y-$axis are the eigenvalues and the $x-$axis the index of ordering

Algorithm 4 pCN Algorithm With Spectral Approximation

```
Input: $L$. $\Phi(u)$. $u^{(0)} \in U$.
Output: $M$ Approximate samples from the posterior distribution
Define: $\alpha(u, w) = \min\{1, \exp(\Phi(u) - \Phi(w)\}$.
while $k < M$ do
    $w^{(k)} = \sqrt{1-\beta^2}u^{(k)} + \beta\xi^{(k)}_{\ell,o}$, where $\xi^{(k)}_{\ell,o} \sim \mathcal{N}(0, C_{\ell,o})$ via equation (5.14).
    Calculate acceptance probability $\alpha(u^{(k)}, w^{(k)})$.
    Accept $w^{(k)}$ as $u^{(k+1)}$ with probability $\alpha(u^{(k)}, w^{(k)})$, otherwise $u^{(k+1)} = u^{(k)}$.
end while
```

5.3 Numerical Experiments

In this section we conduct a series of numerical experiments on four different data sets that are representative of the field of graph semi-supervised learning. There are three main purposes for the experiments. First we perform uncertainty quantification, as explained in subsection 5.1.4. Secondly, we study the spectral approximation and projection variants on pCN sampling as these scale well to massive graphs. Finally we make some observations about the cost and practical implementation details of these methods, for the different Bayesian models we adopt; these will help guide the reader in making choices about which algorithm to use. We present the results for MAP estimation in Section 2 of the supplementary materials, alongside the proof of convexity of the probit MAP estimator.

The quality of the graph constructed from the feature vectors is central to the performance of any graph learning algorithms. In the experiments below, we follow the graph construction procedures used in the previous works. [BF12, HSB15, MKB13] which applied graph semi-supervised learning to all of the datasets that we consider in this chapter. Moreover, we have verified that for all the reported experiments below, the graph parameters are in a range such that spectral clustering [Von07] (an unsupervised learning method) gives a reasonable performance. The methods we employ lead to refinements over spectral clustering (improved classification) and, of course, to uncertainty quantification (which spectral clustering does not address).

5.3.1 Data Sets

We introduce the data sets and describe the graph construction for each data set. In all cases we numerically construct the weight matrix A, and then the graph Laplacian L.[3]

5.3.1.1 Two Moons

The two moons artificial data set is constructed to give noisy data which lies near a nonlinear low dimensional manifold embedded in a high dimensional space [BH09]. The data set is constructed by sampling N data points uniformly from two semi-circles centered at $(0,0)$ and $(1,0.5)$ with radius 1, embedding the data in $\mathbb{R}^d$, and adding Gaussian noise with standard deviation σ. We set $N = 2,000$ and $d = 100$ in this chapter; recall that the graph size is N and each feature vector has length d. We will conduct a variety of experiments with different labelled data size J, and in particular study variation with J. The default value, when not varied, is J at 3% of N, with the labelled points chosen at random.

We take each data point as a node on the graph, and construct a fully connected graph using the self-tuning weights of Zelnik-Manor and Perona [ZP04], with $K = 10$. Specifically we let x_i, x_j be the coordinates of the data points i and j. Then weight a_{ij} between nodes i and j is defined by

$$a_{ij} = \exp\Big(-\frac{\|x_i - x_j\|^2}{2\tau_i\tau_j}\Big), \tag{5.15}$$

where τ_j is the distance of the K-th closest point to the node j.

5.3.1.2 House Voting Records from 1984

This dataset contains the voting records of 435 U.S. House of Representatives; for details see [BF12] and the references therein. The votes were recorded in 1984 from the 98^{th} United States Congress, 2^{nd} session. The votes for each individual is vectorized by mapping a yes vote to 1, a no vote to -1, and an abstention/no-show to 0. The data set contains 16 votes

[3] The weight matrix A is symmetric in theory; in practice we find that symmetrizing via the map $A \mapsto \frac{1}{2}A + \frac{1}{2}A^*$ is helpful.

that are believed to be well-correlated with partisanship, and we use only these votes as feature vectors for constructing the graph. Thus the graph size is $N = 435$, and feature vectors have length $d = 16$. The goal is to predict the party affiliation of each individual, given a small number of known affiliations (labels). We pick 3 Democrats and 2 Republicans at random to use as the observed class labels; thus $J = 5$ corresponding to less than 1.2% of fidelity (i.e. labelled) points. We construct a fully connected graph with weights given by (5.15) with $\tau_j = \tau = 1.25$ for all nodes j.

5.3.1.3 MNIST

The MNIST database consists of $70,000$ images of size 28×28 pixels containing the handwritten digits 0 through 9; see [LC98] for details. Since in this chapter we focus on binary classification, we only consider pairs of digits. To speed up calculations, we subsample randomly $2,000$ images from each digit to form a graph with $N = 4,000$ nodes; we use this for all our experiments except in subsection 5.3.4 where we use the full data set of size $N = O(10^4)$ for digit pair $(4, 9)$ to benchmark computational cost. The nodes of the graph are the images and as feature vectors we project the images onto the leading 50 principal components given by PCA; thus the feature vectors at each node have length $d = 50$. We construct a K-nearest neighbor graph with $K = 20$ for each pair of digits considered. Namely, the weights a_{ij} are non-zero if and only if one of i or j is in the K nearest neighbors of the other. The non-zero weights are set using (5.15) with $K = 20$.

We choose the four pairs $(5, 7)$, $(0, 6)$, $(3, 8)$ and $(4, 9)$. These four pairs exhibit increasing levels of difficulty for classification. This fact is demonstrated in Figures 5.3a - 5.3d, where we visualize the datasets by projecting the dataset onto the second and third eigenvector of the graph Laplacian. Namely, each node i is mapped to the point $(Q(2, i), Q(3, i)) \in \mathbb{R}^2$, where $L = Q\Lambda Q^*$.

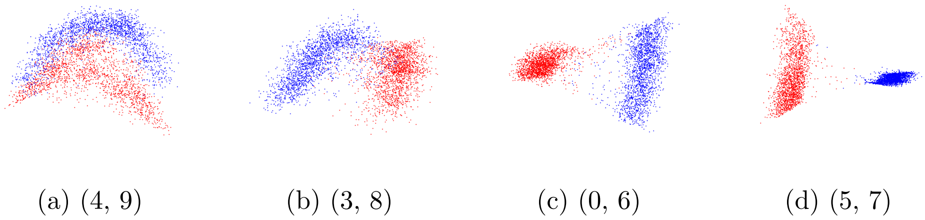

(a) (4, 9) (b) (3, 8) (c) (0, 6) (d) (5, 7)

Figure 5.3: Visualization of data by projection onto 2^{nd} and 3^{rd} eigenfuctions of the graph Laplacian for the MNIST data set, where the vertical dimension is the 3^{rd} eigenvector and the horizontal dimension the 2^{nd}. Each subfigure represents a different pair of digits. We construct a 20 nearest neighbour graph under the Zelnik-Manor and Perona scaling [ZP04] as in (5.15) with $K = 20$.

5.3.1.4 HyperSpectral Image

The hyperspectral data set analysed for this project was provided by the Applied Physics Laboratory at Johns Hopkins University; see [BLC11] for details. It consists of a series of video sequences recording the release of chemical plumes taken at the Dugway Proving Ground. Each layer in the spectral dimension depicts a particular frequency starting at $7,830$ nm and ending with $11,700$ nm, with a channel spacing of 30 nm, giving 129 channels; thus the feature vector has length $d = 129$. The spatial dimension of each frame is 128×320 pixels. We select 7 frames from the video sequence as the input data, and consider each spatial pixel as a node on the graph. Thus the graph size is $N = 128 \times 320 \times 7 = 286,720$. Note that time-ordering of the data is ignored. The classification problem is to classify pixels that represent the chemical plumes against pixels that are the background.

We construct a fully connected graph with weights given by the cosine distance:

$$w_{ij} = \frac{\langle x_i, x_j \rangle}{\|x_i\| \|x_j\|}.$$

This distance is small for vectors that point in the same direction, and is insensitive to their magnitude. We consider the normalized Laplacian defined in (5.1). Because it is computationally prohibitive to compute eigenvectors of a Laplacian of this size, we apply the Nyström extension [WS00, FBC04] to obtain an approximation to the true eigenvectors and eigenvalues; see [BF12] for details pertinent to the set-up here. We emphasize that each pixel in the 7 frames is a node on the graph and that, in particular, pixels across the 7 time-frames are also connected. Since we have no ground truth labels for this dataset, we generate known labels by setting the segmentation results from spectral clustering as ground truth. The default value of J is $8,000$, and labels are chosen at random. This corresponds to labelling around 2.8% of the points. We only plot results for the last 3 frames of the video sequence in order to ensure that the information in the figures it not overwhelmingly large.

5.3.2 Uncertainty Quantification

In this subsection we demonstrate both the feasibility, and value, of uncertainty quantification in graph classification methods. We employ the probit and the Bayesian level-set model for most of the experiments in this subsection; we also employ the Ginzburg-Landau model but since this can be slow to converge, due to the presence of local minima, it is only demonstrated on the voting records dataset. The pCN method is used for sampling on various datasets to demonstrate properties and interpretations of the posterior. In all experiments, all statistics on the label l are computed under the push-forward posterior measure onto label space, ν.

5.3.2.1 Posterior Mean as Confidence Scores

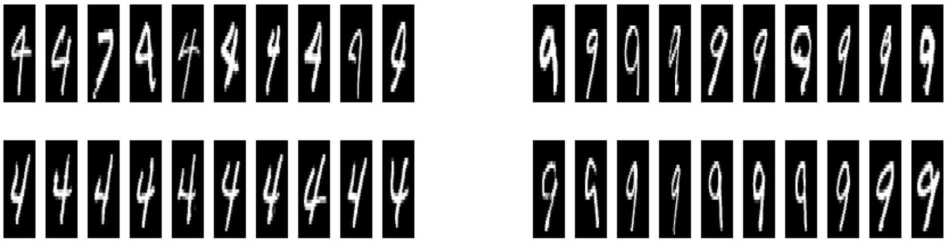
We construct the graph from the MNIST $(4, 9)$ dataset following subsection 5.3.1. The noise variance γ is set to 0.1, and 4% of fidelity points are chosen randomly from each class. The probit posterior is used to compute (5.9). In Figure 5.4 we demonstrate that nodes with scores s_j^l closer to the binary ground truth labels ± 1 look visually more uniform than nodes with s_j^l far from those labels. This shows that the posterior mean contains useful information which differentiates between outliers and inliers that align with human perception. The scores s_j^l are computed as follows: we let $\{u^{(k)}\}_{k=1}^M$ be a set of samples of the posterior measure obtained from the pCN algorithm. The probability $\mathbb{P}(S(u(j) = l(j))$

is approximated by

$$\mathbb{P}\big(S(u(j) = l(j)\big) \approx \frac{1}{M}\sum_{k=1}^{M} \mathbf{1}_{u^{(k)}(j)>0}$$

for each j. Finally the score

$$s_j^l = 2\mathbb{P}\big(S(u(j) = l(j)\big) - 1.$$



(a) Fours in MNIST

(b) Nines in MNIST

Figure 5.4: "Hard to classify" vs "easy to classify" nodes in the MNIST $(4, 9)$ dataset under the probit model. Here the digit "4" is labeled +1 and "9" is labeled -1. The top (bottom) row of the left column corresponds to images that have the lowest (highest) values of s_j^l defined in (5.9) among images that have ground truth labels "4". The right column is organized in the same way for images with ground truth labels 9 except the top row now corresponds to the highest values of s_j^l. Higher s_j^l indicates higher confidence that image j is a 4 and not a "9", hence the top row could be interpreted as images that are "hard to classify" by the current model, and vice versa for the bottom row. The graph is constructed as in Section 5.3, and $\gamma = 0.1$, $\beta = 0.3$.

5.3.2.2 Posterior Variance as Uncertainty Measure

In this set of experiments, we show that the posterior distribution of the label variable $l = S(u)$ captures the uncertainty of the classification problem. We use the posterior variance of l, averaged over all nodes, as a measure of the model variance; specifically formula (5.10). We study the behaviour of this quantity as we vary the level of uncertainty within certain inputs to the problem. We demonstrate empirically that the posterior variance is approximately monotonic with respect to variations in the levels of uncertainty in the input data, as it should be; and thus that the posterior variance contains useful information about the classification. We select quantities that reflect the separability of the classes in the feature space.

Figure 5.5 plots the posterior variance $\mathrm{Var}(l)$ against the standard deviation σ of the noise appearing in the feature vectors for the two moons dataset; thus points generated on the two semi-circles overlap more as σ increases. We employ a sequence of posterior computations, using probit and Bayesian level-set, for $\sigma = 0.02 : 0.01 : 0.12$. Recall that $N = 2,000$ and we choose 3% of the nodes to have the ground truth labels as observed data. Within both models, γ is fixed at 0.1. A total of 1×10^4 samples are taken, and the proposal variance β is set to 0.3. We see that the mean posterior variance increases with σ, as is intuitively reasonable. Furthermore, because γ is small, probit and Bayesian level-set are very similar models and this is reflected in the similar quantitative values for uncertainty.

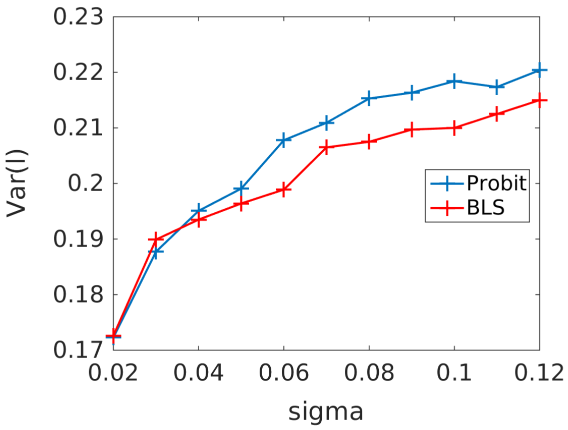


Figure 5.5: Mean Posterior Variance defined in (5.10) versus feature noise σ for the probit model and the BLS model applied to the Two Moons Dataset with $N = 2,000$. For each trial, a realization of the two moons dataset under the given parameter σ is generated, where σ is the Gaussian noise on the features defined in Section 5.3.1.1 , and 3% of nodes are randomly chosen as fidelity. We run 20 trials for each value of σ, and average the mean posterior variance across the 20 trials in the figure. We set $\gamma = 0.1$ and $\beta = 0.3$ for both models.

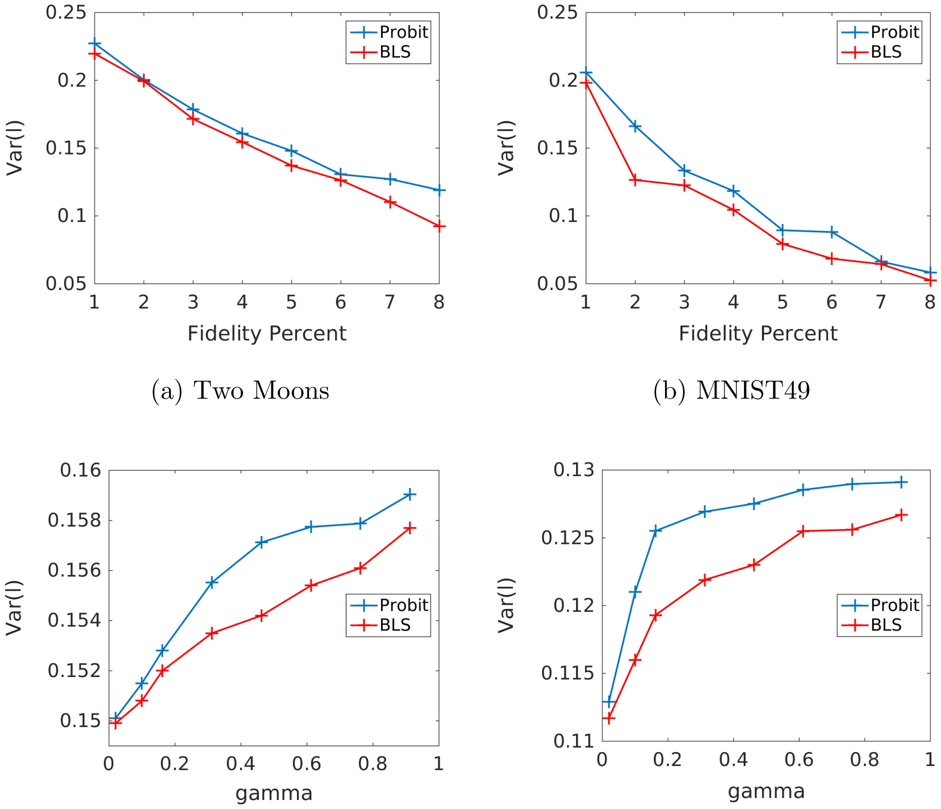
A similar experiment studies the posterior label variance $\mathrm{Var}(l)$ as a function of the pair of digits classified within the MNIST data set. We choose 4% of the nodes as labelled data, and set $\gamma = 0.1$. The number of samples employed is 1×10^4 and the proposal variance β is set to be 0.3. Table 5.3.2.2 shows the posterior label variance. Recall that Figures 5.3a - 5.3d suggest that the pairs $(4, 9), (3, 8), (0, 6), (5, 7)$ are increasingly easy to separate, and this is reflected in the decrease of the posterior label variance shown in Table 5.3.2.2.

Digits	(4, 9)	(3, 8)	(0, 6)	(5, 7)
probit	0.1485	0.1005	0.0429	0.0084
BLS	0.1280	0.1018	0.0489	0.0121

Table 5.1: Mean Posterior Variance of different digit pairs for the probit model and the BLS model applied to the MNIST Dataset. The pairs are organized from left to right according to the separability of the two classes as shown in Fig.5.3a - 5.3d. For each trial, we randomly select 4% of nodes as fidelity. We run 10 trials for each pairs of digits and average the mean posterior variance across trials. We set $\gamma = 0.1$ and $\beta = 0.3$ for both models.

The previous two experiments in this subsection have studied posterior label variance $\mathrm{Var}(l)$ as a function of variation in the prior data. We now turn and study how posterior variance changes as a function of varying the likelihood information, again for both two moons and MNIST data sets. In Figures 5.6a and 5.6b, we plot the posterior label variance against the percentage of nodes observed. We observe that the observational variance decreases as the amount of labelled data increases. Figures 5.6c and 5.6d show that the posterior label variance increases almost monotonically as observational noise γ increases. Furthermore the level set and probit formulations produce similar answers for γ small, reflecting the close similarity between those methods when γ is small – when $\gamma = 0$ their likelihoods coincide.

In summary of this subsection, the label posterior variance $\mathrm{Var}(l)$ behaves intuitively as expected as a function of varying the prior and likelihood information that specify the statistical probit model and the Bayesian level-set model. The uncertainty quantification thus provides useful, and consistent, information that can be used to inform decisions made on the basis of classifications.



(a) Two Moons

(b) MNIST49

(c) Two Moons

(d) MNIST

Figure 5.6: Mean Posterior Variance as in (5.10) versus percentage of labelled points and noise level γ for the probit model and the BLS model applied to the Two Moons dataset and the 4-9 MNIST dataset. For two moons, we fix $N = 2,000$ and $\sigma = 0.06$. For each trial, we generate a realization of the two moons dataset while the MNIST dataset is fixed. For a), b) γ is fixed at 0.1, and a certain percentage of nodes are selected at random as labelled. For c), d), the proportion of labelled points is fixed at 4%, and γ is varied across a range. Results are averaged over 20 trials.

5.3.2.3 Visualization of Marginal Posterior Density

In this subsection, we contrast the posterior distribution $\mathbb{P}(v|y)$ of the Ginzburg-Landau model with that of the probit and Bayesian level-set (BLS) models. The graph is constructed from the voting records data with the fidelity points chosen as described in subsection 5.3.1. In Figure 5.7 we plot the histograms of the empirical marginal posterior distribution on

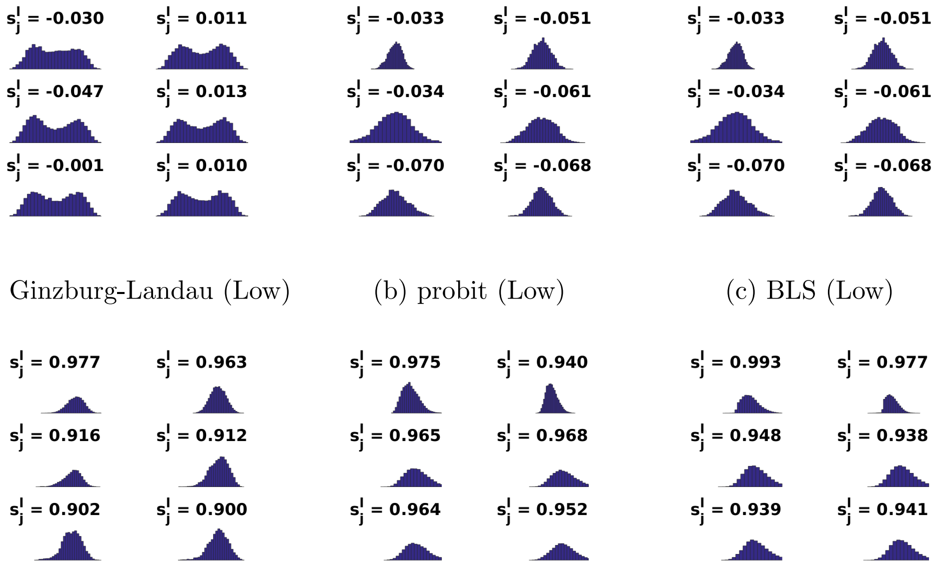
$\mathbb{P}(v(i)|y)$ and $\mathbb{P}(u(i)|y)$ for a selection of nodes on the graph. For the top row of Figure 5.7, we select 6 nodes with "low confidence" predictions, and plot the empirical marginal distribution of u for probit and BLS, and that of v for the Ginzburg-Landau model. Note that the same set of nodes is chosen for different models. The plots in this row demonstrate the multi-modal nature of the Ginzburg-Landau distribution in contrast to the uni-modal nature of the probit posterior; this uni-modality is a consequence of the log-concavity of the probit likelihood. For the bottom row, we plot the same empirical distributions for 6 nodes with "high confidence" predictions. In contrast with the top row, the Ginzburg-Landau marginal for high confidence nodes is essentially uni-modal since most samples of v evaluated on these nodes have a fixed sign.

5.3.3 Spectral Approximation and Projection Methods

Here we discuss Algorithms 3 and 4, designed to approximate the full (but expensive on large graphs) Algorithm 2.

First, we examine the quality of the approximation by applying the algorithms to the voting records dataset, a small enough problem where sampling using the full graph Laplacian is feasible. To quantify the quality of approximation, we compute the posterior mean of the thresholded variable s_j^l for both Algorithm 3 and Algorithm 4, and compare the mean absolute difference $\frac{1}{N}|s_j^l - s_j^{l*}|$ where s_j^{l*} is the "ground truth" value computed using the full Laplacian. Using $\gamma = 0.1$, $\beta = 0.3$, and a truncation level of $\ell = 150$, we observe that the mean absolute difference for spectral projection is **0.1577**, and **0.0261** for spectral approximation. In general, we set $\bar{\lambda}$ to be $\max_{j \le \ell} \lambda_j$ where ℓ is the truncation level.

Next we apply the spectral projection/approximation algorithms with the Bayesian level-set likelihood to the hyperspectral image dataset; the results for probit are similar (when we use small γ) but have greater cost per step, because of the cdf evaluations required for probit. The first two rows in Fig.5.8 show that the posterior mean s_j^l is able to differentiate between different concentrations of the plume gas. We have also coloured pixels with $|s_j^l| < 0.4$ in red to highlight the regions with greater levels of uncertainty. We observe that the red pixels



(a) Ginzburg-Landau (Low) (b) probit (Low) (c) BLS (Low)

(d) Ginzburg-Landau (High) (e) probit (High) (f) BLS (High)

Figure 5.7: Visualization of marginal posterior density for low and high confidence predictions across different models. Each image plots the empirical marginal posterior density of a certain node i, obtained from the histogram of 1×10^5 approximate samples using pCN. Columns in the figure (e.g. a) and d)) are grouped by model. From left to right, the models are Ginzburg-Landau, probit, and Bayesian level-set respectively. From the top down, the rows in the figure (e.g. a)-c)) denote the low confidence and high confidence predictions respectively. For the top row, we select 6 nodes with the lowest absolute value of the posterior mean s_j^l, defined in equation (5.9), averaged across three models. For the bottom row, we select nodes with the highest average posterior mean s_j^l. We show the posterior mean s_j^l on top of the histograms for reference. The experiment parameters are: $\epsilon = 10.0$, $\gamma = 0.6$, $\beta = 0.1$ for the Ginburg-Landau model, and $\gamma = 0.5$, $\beta = 0.2$ for the probit and BLS model.

mainly lie in the edges of the gas plume, which conforms with human intuition. As in the voting records example in the previous subsection, the spectral approximation method has greater posterior uncertainty, demonstrated by the greater number of red pixels in the second row of Fig.5.8 compared to the first row. We conjecture that the spectral approximation is closer to what would be obtained by sampling the full distribution, but we have not verified this as the full problem is too large to readily sample. The bottom row of Fig.5.8 shows the result of using optimization based classification, using the Gibzburg-Landau method. This is shown simply to demonstrate consistency with the full UQ approach shown in the other two rows, in terms of hard classification.

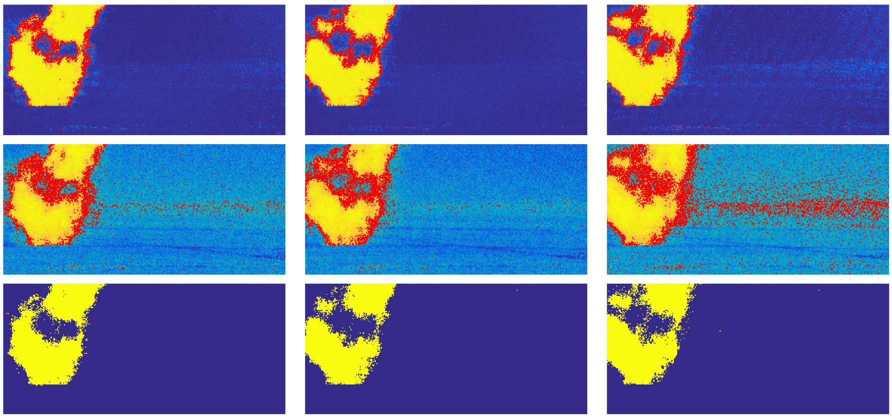


Figure 5.8: Inference results on hyperspectral image dataset using spectral projection (top row), spectral approximation (middle row), and Ginzburg-Landau classification (bottom row). For the top two rows, the values of s_j^l are plotted on a $[-1,1]$ color scale on each pixel location. In addition, we highlight the regions of uncertain classification by coloring the pixels with $|s_j^l| < 0.4$ in red. The bottom row is the classification result from the Ginzburg-Landau model, shown here as a comparison. The truncation level $\ell = 40$, and for the spectral approximation algorithm, $\bar{\lambda} = 1$. We set $\gamma = 0.1$, $\beta = 0.08$ and use $M = 2 \times 10^4$ MCMC samples. We create the label data by subsampling $8,000$ pixels ($\approx 2.8\%$ of the total) from the labellings obtained by spectral clustering.

5.3.4 Comparitive Remarks About The Different Models

At a high level we have shown the following concerning the three models based on probit, level-set and Ginzburg-Landau:

- Bayesian level set is considerably cheaper to implement than probit in Matlab because the norm cdf evaluations required for probit are expensive.

- Probit and Bayesian level-set behave similarly, for posterior sampling, especially for small γ, since they formally coincide when $\gamma = 0$.

- Probit and Bayesian level-set are superior to Ginzburg-Landau for posterior sampling; this is because probit has log-concave posterior, whilst Ginzburg-Landau is multi-modal.

- Ginzburg-Landau provides the best hard classifiers, when used as an optimizer (MAP estimator), and provided it is initialized well. However it behaves poorly when not initialized carefully because of multi-modal behaviour. In constrast probit provides almost indistinguihsable classifiers, comparable or marginally worse in terms of accuracy, and has a convex objective function and hence a unique minimizer. (See supplementary materials for details of the relevant experiments.)

We expand on the details of these conclusions by studying run times of the algorithms. All experiments are done on a 1.5GHz machine with Intel Core i7. In Table 5.2, we compare the running time of the MCMC for different models on various datasets. We use an a posteriori condition on the samples $u^{(k)}$ to empirically determine the sample size M needed for the MCMC to converge. Note that this condition is by no means a replacement for a rigorous analysis of convergence using auto-correlation, but is designed to provide a ballpark estimate of the speed of these algorithms on real applications. We now define the a posteriori condition used. Let the approximate samples be $\{u^{(k)}\}$. We define the cumulative average as $\tilde{u}^{(k)} = \frac{1}{k}\sum_{j=1}^{k} u^{(j)}$, and find the first k such that

$$\|\tilde{u}^{(kT)} - \tilde{u}^{((k-1)T)}\| \le \text{tol}, \tag{5.16}$$

where tol is the tolerance and T is the number of iterations skipped. We set $T = 5000$, and also tune the stepsize parameter β such that the average acceptance probability of the MCMC is over 50%. We choose the model parameters according to the experiments in the sections above so that the posterior mean gives a reasonable classification result.

We note that the number of iterations needed for the Ginzburg-Landau model is much higher compared to probit and the Bayesian level-set (BLS) method; this is caused by the

[4] According to the reporting in [MSB14].

Data	Voting Records	MNIST49	Hyperspectral
(Tol)	$tol = 1 \times 10^{-3}$	$tol = 1.5 \times 10^{-3}$	$tol = 2 \times 10^{-2}$
(N)	$N = 435$	$N \approx 1.1 \times 10^4$	$N \approx 2.9 \times 10^5$
(Neig)	$Neig = 435$	$Neig = 300$	$Neig = 50$
(J)	$J = 5$	$J = 440$	$J = 8000$
Preprocessing	$t = 0.7s$	$t = 50.8s$	$t < 60s^4$
probit	$t = 8.9s$,	$t = 176.4s$,	$t = 5410.3s$,
	$M = 10^4$	$M = 1.5 \times 10^4$	$M = 1.5 \times 10^4$
BLS	$t = 2.7s$,	$t = 149.1s$,	$t = 970.8s$,
	$M = 10^4$	$M = 1.5 \times 10^4$	$M = 1.5 \times 10^4$
GL	$t = 161.4s$	-	-
	$M = 1.8 \times 10^5$	-	-

Table 5.2: Timing for MCMC methods. We report both the number of samples M and the running time of the algorithm t. The time for GL on MNIST and Hyperspectral is omitted due to running time being too slow. J denotes the number of fidelity points used. For the voting records, we set $\gamma = 0.2, \beta = 0.4$ for probit and BLS, and $\gamma = 1$, $\beta = 0.1$ for Ginzburg-Landau. For MNIST, we set $\gamma = 0.1, \beta = 0.4$. For Hyperspctral, we set $\gamma = 1.0$, and $\beta = 0.1$.

presence of multiple local minima in Ginzburg-Landau, in contrast to the log concavity of probit. probit is slower than BLS due to the fact that evaluations of the cdf function for Gaussians is slow.

5.4 Further Directions

Some future directions for UQ on graphs include improvement of the current inference method, connections between the different models presented in this chapter, and generalization to multiclass classification, for example by vectorizing the latent variable (as in existing non-Bayesian multiclass methods [GMB14, MKB13]), and applying multi-dimensional analogues of the likelihood functions used in this chapter. Hierarchical methods could also be applied to account for the uncertainty in the various hyperparameters such as the label noise γ, or the length scale ϵ in the Ginzburg-Landau model. Finally, we could study in more detail the effects of either the spectral projection or the approximation method, either analytically on some tractable toy examples, or empirically on a suite of representative problems.

Studying the modeling assumptions themselves, guided by data, provides a research direction of long term value. Such questions have not been much studied. For example the choice of the signum function to relate the latent variable to the categorial data could be questioned, and other models employed; or the level value of 0 chosen in the level set approach could be chosen differently, or as a hyper-parameter. Furthermore the form of the prior on the latent variable u could be questioned. We use a Gaussian prior which encodes first and second order statistical information about the unlabelled data. This Gaussian could contain hyper-parameters, of Whittle-Matern type, which could be learnt from the data; and more generally other non-Gaussian priors could and should be considered. For instance, in image data, it is often useful to model feature vectors as lying on submanifolds embedded in a higher dimensional space; such structure could be exploited. More generally, addressing the question of which generative models are appropriate for which types of data is an interesting and potentially fruitful research direction.

REFERENCES

[A 18] O.Papaspiliopoulo A.Stuart A. Bertozzi, X. Luo. "Scalabe and robust sampling methods for Bayesian graph-based semi-supervised learning." *In preparation.*, 2018.

[AFS16a] Carlos M Alaíz, Michaël Fanuel, and Johan AK Suykens. "Convex Formulation for Kernel PCA and its Use in Semi-Supervised Learning." *arXiv preprint arXiv:1610.06811*, 2016.

[AFS16b] Carlos M Alaíz, Michaël Fanuel, and Johan AK Suykens. "Robust Classification of Graph-Based Data." *arXiv preprint arXiv:1612.07141*, 2016.

[And10] Christopher R Anderson. "A Rayleigh–Chebyshev procedure for finding the smallest eigenvalues and associated eigenvectors of large sparse Hermitian matrices." *Journal of Computational Physics*, **229**(19):7477–7487, 2010.

[BBT17] Zachary Boyd, Egil Bae, Xue-Cheng Tai, and Andrea L Bertozzi. "Simplified Energy Landscape for Modularity Using Total Variation." *arXiv preprint arXiv:1707.09285*, 2017.

[BC01] Avrim Blum and Shuchi Chawla. "Learning from labeled and unlabeled data using graph mincuts." 2001.

[BCM12] Stephen Boyd, Corinna Cortes, Mehryar Mohri, and Ana Radovanovic. "Accuracy at the top." In *Advances in neural information processing systems*, pp. 953–961, 2012.

[BF12] Andrea L Bertozzi and Arjuna Flenner. "Diffuse interface models on graphs for classification of high dimensional data." *Multiscale Modeling & Simulation*, **10**(3):1090–1118, 2012.

[BF16] Andrea L Bertozzi and Arjuna Flenner. "Diffuse interface models on graphs for classification of high dimensional data." *SIAM Review*, **58**(2):293–328, 2016.

[BFC02] Serge Belongie, Charless Fowlkes, Fan Chung, and Jitendra Malik. "Spectral partitioning with indefinite kernels using the Nyström extension." In *Computer Vision ECCV 2002*, pp. 531–542. Springer, 2002.

[BH09] Thomas Bühler and Matthias Hein. "Spectral clustering based on the graph p-Laplacian." In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 81–88. ACM, 2009.

[BHS09] Martin Burger, Lin He, and Carola-Bibiane Schönlieb. "Cahn-Hilliard inpainting and a generalization for grayvalue images." *SIAM Journal on Imaging Sciences*, **2**(4):1129–1167, 2009.

[Bis07] C Bishop. "Pattern Recognition and Machine Learning (Information Science and Statistics), 1st edn. 2006. corr. 2nd printing edn." *Springer, New York*, 2007.

[BJ01] Yuri Y Boykov and M-P Jolly. "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images." In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pp. 105–112. IEEE, 2001.

[BKY96] Marc Berthod, Zoltan Kato, Shan Yu, and Josiane Zerubia. "Bayesian image classification using Markov random fields." *Image and Vision Computing*, **14**(4):285–295, 1996.

[BLC11] Joshua B Broadwater, Diane Limsui, and Alison K Carr. "A primer for chemical plume detection using LWIR sensors." *Technical Paper, National Security Technology Department, Las Vegas, NV*, 2011.

[BLO97] L Bertini, C Landim, and S Olla. "Derivation of Cahn-Hilliard equations from Ginzburg-Landau models." *Journal of Statistical Physics*, **88**(1-2):365–381, 1997.

[BLSed] Andrea L Bertozzi, Xiyang Luo, Andrew M Stuart, and Konstantinos C Zygalakis. "Uncertainty Quantification in Graph-Based Classification of High Dimensional Data." *SIAM/ASA Journal on Uncertainty Quantification*, **6**(2):568–595, 2018, Copyright ©2018 Society for Industrial and Applied Mathematics. Reprinted with permission. All rights reserved.

[BMN04] Mikhail Belkin, Irina Matveeva, and Partha Niyogi. "Regularization and semi-supervised learning on large graphs." In *International Conference on Computational Learning Theory*, pp. 624–638. Springer, 2004.

[BNS06] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples." *Journal of Machine Learning Research*, **7**(Nov):2399–2434, 2006.

[BPC10] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. "Distributed optimization and statistical learning via the alternating direction method of multipliers." *Foundations and Trends in Machine Learning*, **3**(1):1–122, 2010.

[BRS08] A. Beskos, G. Roberts, A. M. Stuart, and J. Voss. "MCMC methods for diffusion bridges." *Stochastics and Dynamics*, **8**(03):319–350, 2008.

[BRS09] Alexandros Beskos, Gareth Roberts, and Andrew Stuart. "Optimal scalings for local Metropolis-Hastings chains on nonproduct targets in high dimensions." *The Annals of Applied Probability*, pp. 863–898, 2009.

[BT09] Amir Beck and Marc Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems." *SIAM journal on imaging sciences*, **2**(1):183–202, 2009.

[BV09] Stephen Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge university press, 2009.

[BVZ98] Yuri Boykov, Olga Veksler, and Ramin Zabih. "Markov random fields with efficient approximations." In *Computer vision and pattern recognition, 1998. Proceedings. 1998 IEEE computer society conference on*, pp. 648–655. IEEE, 1998.

[BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. "Fast approximate energy minimization via graph cuts." *IEEE Transactions on pattern analysis and machine intelligence*, **23**(11):1222–1239, 2001.

[CFS09] Jie Chen, Haw-ren Fang, and Yousef Saad. "Fast approximate kNN graph construction for high dimensional data via recursive Lanczos bisection." *Journal of Machine Learning Research*, **10**(Sep):1989–2012, 2009.

[Cia70] Philippe G Ciarlet. "Discrete maximum principle for finite-difference operators." *Aequationes mathematicae*, **4**(3):338–352, 1970.

[CMH17] Bo Chang, Lili Meng, Eldad Haber, Frederick Tung, and David Begert. "Multilevel Residual Networks from Dynamical Systems View." *arXiv preprint arXiv:1710.10348*, 2017.

[CN94] JW Cahn and A Novick-Cohen. "Evolution equations for phase separation and ordering in binary alloys." *Journal of statistical physics*, **76**(3-4):877–909, 1994.

[CRS13] S. L. Cotter, G. O. Roberts, A. M. Stuart, and D. White. "MCMC methods for functions: modifying old algorithms to make them faster." *Statistical Science*, **28**(3):424–446, 2013.

[DIS16] Matthew M Dunlop, Marco A Iglesias, and Andrew M Stuart. "Hierarchical Bayesian Level Set Inversion." *arXiv preprint arXiv:1601.03605*, 2016.

[DJP92] Elias Dahlhaus, David S Johnson, Christos H Papadimitriou, Paul D Seymour, and Mihalis Yannakakis. "The complexity of multiway cuts." In *Proceedings of the twenty-fourth annual ACM symposium on theory of computing*, pp. 241–251. ACM, 1992.

[DLZ11] Weisheng Dong, Xin Li, Lei Zhang, and Guangming Shi. "Sparsity-based image denoising via dictionary learning and structural clustering." In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 457–464, 2011.

[ESM17] Elad Eban, Mariano Schain, Alan Mackey, Ariel Gordon, Ryan Rifkin, and Gal Elidan. "Scalable Learning of Non-Decomposable Objectives." In *Artificial Intelligence and Statistics*, pp. 832–840, 2017.

[Eyr98] David J Eyre. "An unconditionally stable one-step scheme for gradient systems. (https://www.math.utah.edu/ eyre/research/methods/stable.ps)." *Unpublished article*, 1998.

[FBC04] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. "Spectral grouping using the Nystrom method." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**(2):214–225, 2004.

[FBM01] Charless Fowlkes, Serge Belongie, and Jitendra Malik. "Efficient spatiotemporal grouping using the Nyström method." In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pp. I–231. IEEE, 2001.

[FD16] Cyril Furtlehner and Aurélien Decelle. "Cycle-Based Cluster Variational Method for Direct and Inverse Inference." *Journal of Statistical Physics*, **164**(3):531–574, 2016.

[FGB11] Alexander Fix, Aritanan Gruber, Endre Boros, and Ramin Zabih. "A graph cut algorithm for higher-order Markov random fields." In *2011 International Conference on Computer Vision*, pp. 1020–1027. IEEE, 2011.

[GFP15] Cristina Garcia-Cardona, Arjuna Flenner, and Allon G Percus. "Multiclass Semi-Supervised Learning on Graphs using Ginzburg-Landau Functional Minimization." In *Pattern Recognition Applications and Methods*, pp. 119–135. Springer, 2015.

[GGO14] Yves van Gennip, Nestor Guillen, Braxton Osting, and Andrea L Bertozzi. "Mean curvature, threshold dynamics, and phase field theory on finite graphs." *Milan Journal of Mathematics*, **82**(1):3–65, 2014.

[GHM17] Yves van Gennip, Blake Hunter, Anna Ma, Daniel Moyer, Ryan de Vera, and Andrea L Bertozzi. "Unsupervised record matching with noisy and incomplete data." *arXiv preprint arXiv:1704.02955*, 2017.

[Git11] Alex Gittens. "The spectral norm error of the naive Nystrom extension." *arXiv preprint arXiv:1110.5305*, 2011.

[GMB14] Cristina Garcia-Cardona, Ekaterina Merkurjev, Andrea L Bertozzi, Arjuna Flenner, and Allon G Percus. "Multiclass data segmentation using diffuse interface methods on graphs." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **36**(8):1600–1613, 2014.

[GPS89] Dorothy M Greig, Bruce T Porteous, and Allan H Seheult. "Exact maximum a posteriori estimation for binary images." *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 271–279, 1989.

[GTC10] Shenghua Gao, Ivor Wai-Hung Tsang, Liang-Tien Chia, and Peilin Zhao. "Local features are not lonely – Laplacian sparse coding for image classification." In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 3555–3561, 2010.

[GTC13] Shenghua Gao, Ivor Wai-Hung Tsang, and Liang-Tien Chia. "Laplacian sparse coding, hypergraph Laplacian sparse coding, and applications." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(1):92–104, 2013.

[HAS11] Kiana Hajebi, Yasin Abbasi-Yadkori, Hossein Shahbazi, and Hong Zhang. "Fast approximate nearest-neighbor search with k-nearest neighbor graph." In *IJCAI*

Proceedings-International Joint Conference on Artificial Intelligence, volume 22, p. 1312, 2011.

[HK04] Desmond J Higham and Milla Kibble. "A unified view of spectral clustering." *University of Strathclyde mathematics research report*, **2**, 2004.

[HL08] Mark J Huiskes and Michael S Lew. "The MIR Flickr retrieval evaluation." In *Proc. 1st ACM Intl. Conf. on Multimedia Information Retrieval*, pp. 39–43, 2008.

[HLP13] Huiyi Hu, Thomas Laurent, Mason A Porter, and Andrea L Bertozzi. "A method based on total variation for network modularity optimization using the MBO scheme." *SIAM Journal on Applied Mathematics*, **73**(6):2224–2246, 2013.

[HN04] Simon Haykin and Neural Network. "A comprehensive foundation." *Neural networks*, **2**(2004):41, 2004.

[HSB15] Huiyi Hu, Justin Sunu, and Andrea L Bertozzi. "Multi-class Graph Mumford-Shah Model for Plume Detection Using the MBO scheme." In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 209–222. Springer, 2015.

[HVG11] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. "Wavelets on graphs via spectral graph theory." *Applied and Computational Harmonic Analysis*, **30**(2):129–150, 2011.

[HZ16] Jarno Hartog and Harry van Zanten. "Nonparametric Bayesian label prediction on a graph." *arXiv preprint arXiv:1612.01930*, 2016.

[HZR16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[ILS15] Marco A Iglesias, Yulong Lu, and Andrew M Stuart. "A Bayesian Level Set Method for Geometric Inverse Problems." *Interfaces and Free Boundary Problems, arXiv preprint arXiv:1504.00313*, 2015.

[KQA05] Ashish Kapoor, Yuan Qi, Hyungil Ahn, and Rosalind Picard. "Hyperparameter and kernel learning for graph based semi-supervised classification." In *NIPS*, pp. 627–634, 2005.

[KS89] Robert V Kohn and Peter Sternberg. "Local minimisers and singular perturbations." *Proceedings of the Royal Society of Edinburgh: Section A Mathematics*, **111**(1-2):69–84, 1989.

[KSO16] Da Kuang, Zuoqiang Shi, Stanley Osher, and Andrea Bertozzi. "A harmonic extension approach for collaborative ranking." *arXiv preprint arXiv:1602.05127*, 2016.

[KT07] Pushmeet Kohli and Philip HS Torr. "Dynamic graph cuts for efficient inference in markov random fields." *IEEE transactions on pattern analysis and machine intelligence*, **29**(12):2079–2088, 2007.

[LB17] Xiyang Luo and Andrea L. Bertozzi. "Convergence of the Graph Allen–Cahn Scheme." *Journal of Statistical Physics*, **167**(3):934–958, May 2017.

[LBed] Xiyang Luo and Andrea L Bertozzi. "Convergence of the graph Allen–Cahn scheme." *Journal of Statistical Physics*, **167**(3-4):934–958, 2017. Copyright ©2011 by American Physical Society. All rights reserved.

[LC98] Yann LeCun and Corinna Cortes. "The MNIST database of handwritten digits, http://yann.lecun.com/exdb/mnist/.", 1998.

[Li12] Stan Z Li. *Markov random field modeling in computer vision.* Springer Science & Business Media, 2012.

[LP11] Shuai Lu and Sergei V Pereverzev. "Multi-parameter regularization and its numerical realization." *Numerische Mathematik*, **118**(1):1–31, 2011.

[LWum] Xiyang Luo and Brendt Wohlberg. "Convolutional Laplacian sparse coding." In *Image Analysis and Interpretation (SSIAI), 2016 IEEE Southwest Symposium on*, pp. 133–136. IEEE., 2016. ©2016 IEEE. Reprinted, with permission, from Xiyang Luo, Brendt Wohlberg, Convolutional Laplacian sparse coding, Image Analysis and Interpretation (SSIAI), 2016 IEEE Southwest Symposium.

[Mad10] Aleksander Madry. "Fast approximation algorithms for cut-based problems in undirected graphs." In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pp. 245–254. IEEE, 2010.

[MBB15] Ekaterina Merkurjev, Egil Bae, Andrea L Bertozzi, and Xue-Cheng Tai. "Global Binary Optimization on Graphs for Classification of High-Dimensional Data." *Journal of Mathematical Imaging and Vision*, **52**(3):414–435, 2015.

[MBC16] Ekatherina Merkurjev, Andrea L Bertozzi, and Fan Chung. "A semi-supervised heat kernel pagerank mbo algorithm for data classification." Technical report, University of California, Los Angeles Los Angeles United States, 2016.

[MBP09] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. "Non-local sparse models for image restoration." In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 2272–2279, 2009.

[Men18] Zhaoyi Meng. *High Performance Computing and Real Time Software for High Dimensional Data Classification.* PhD thesis, University of California, Los Angeles, 2018.

[MKB13] Ekaterina Merkurjev, Tijana Kostic, and Andrea L Bertozzi. "An MBO scheme on graphs for classification and image processing." *SIAM Journal on Imaging Sciences*, **6**(4):1903–1930, 2013.

[MKH16] Zhaoyi Meng, Alice Koniges, Yun Helen He, Samuel Williams, Thorsten Kurth, Brandon Cook, Jack Deslippe, and Andrea L Bertozzi. "OpenMP parallelization and optimization of graph-based machine learning algorithms." In *International Workshop on OpenMP*, pp. 17–31. Springer, 2016.

[ML09] Marius Muja and David Lowe. "Flann-fast library for approximate nearest neighbors user manual." *Computer Science Department, University of British Columbia, Vancouver, BC, Canada*, 2009.

[MLE18] Alan Mackey, Xiyang Luo, and Elad Eban. "Constrained Classification and Ranking via Quantiles." *arXiv preprint arXiv:1803.00067*, 2018.

[MMK17] Zhaoyi Meng, Ekaterina Merkurjev, Alice Koniges, and Andrea L Bertozzi. "Hyperspectral image classification using graph clustering methods." *Image Processing On Line*, **7**:218–245, 2017.

[MSB14] Ekaterina Merkurjev, Justin Sunu, and Andrea L Bertozzi. "Graph MBO method for multiclass segmentation of hyperspectral stand-off detection video." In *Image Processing (ICIP), 2014 IEEE International Conference on*, pp. 689–693. IEEE, 2014.

[MST10] J. C. Mattingly, A. M. Stuart, and M. Tretyakov. "Convergence of numerical time-averaging and stationary measures via the Poisson equation." *SIAM Journal of Numerical Analysis*, **48**:552–577, 2010.

[Nea] R. Neal. "Regression and classification using Gaussian process priors." *Bayesian Statistics*, **6**:475. Available at http://www.cs.toronto. edu/ radford/valencia.abstract.html.

[OWO14] Braxton Osting, Chris D White, and Édouard Oudet. "Minimal Dirichlet energy partitions for graphs." *SIAM Journal on Scientific Computing*, **36**(4):A1635–A1651, 2014.

[RGG97] Gareth O Roberts, Andrew Gelman, Walter R Gilks, et al. "Weak convergence and optimal scaling of random walk Metropolis algorithms." *The Annals of Applied Probability*, **7**(1):110–120, 1997.

[SB11] Amarnag Subramanya and Jeff Bilmes. "Semi-supervised learning with measure propagation." *Journal of Machine Learning Research*, **12**(Nov):3311–3370, 2011.

[SBN06] Vikas Sindhwani, Misha Belkin, and Partha Niyogi. "The Geometric Basis of Semi-supervised Learning." 2006.

[SFV11] David I Shuman, Mohammadjavad Faraji, and Pierre Vandergheynst. "Semi-supervised learning with spectral graph wavelets." In *Proceedings of the International Conference on Sampling Theory and Applications (SampTA)*, number EPFL-CONF-164765, 2011.

[str] "Image of Vertical Straw Texture." `http://texturee.deviantart.com/art/Straw-Texture-260793536` (Nov. 2015).

[Stu10] Andrew M Stuart. "Inverse problems: a Bayesian perspective." *Acta Numerica*, **19**:451–559, 2010.

[TC94] Jean E Taylor and John W Cahn. "Linking anisotropic sharp and diffuse surface motion laws via gradient flows." *Journal of Statistical Physics*, **77**(1-2):183–197, 1994.

[TC09] Partha Talukdar and Koby Crammer. "New regularized algorithms for transductive learning." *Machine Learning and Knowledge Discovery in Databases*, pp. 442–457, 2009.

[TS16] Nicolás Garcia Trillos and Dejan Slepčev. "Continuum limit of total variation on point clouds." *Archive for rational mechanics and analysis*, **220**(1):193–241, 2016.

[TSB14] Nicolas Garcia Trillos, Dejan Slepcev, James von Brecht, Thomas Laurent, and Xavier Bresson. "Consistency of Cheeger and Ratio Graph Cuts." *arXiv preprint arXiv:1411.6590*, 2014.

[VB12] Yves Van Gennip, Andrea L Bertozzi, et al. "Γ-convergence of graph Ginzburg-Landau functionals." *Advances in Differential Equations*, **17**(11/12):1115–1180, 2012.

[VBB08] Ulrike Von Luxburg, Mikhail Belkin, and Olivier Bousquet. "Consistency of spectral clustering." *The Annals of Statistics*, pp. 555–586, 2008.

[VF10] Andrea Vedaldi and Brian Fulkerson. "VLFeat: An open and portable library of computer vision algorithms." In *Proceedings of the 18th ACM international conference on Multimedia*, pp. 1469–1472. ACM, 2010.

[Von07] Ulrike Von Luxburg. "A tutorial on spectral clustering." *Statistics and computing*, **17**(4):395–416, 2007.

[Wah90] Grace Wahba. *Spline models for observational data.* SIAM, 1990.

[WLL18] Bao Wang, Xiyang Luo, Zhen Li, Wei Zhu, Zuoqiang Shi, and Stanley J Osher. "Deep Learning with Data Dependent Implicit Activation Function." *arXiv preprint arXiv:1802.00168*, 2018.

[WLZ18] Bao Wang, Xiyang Luo, Fangbo Zhang, Baichuan Yuan, Andrea L Bertozzi, and P Jeffrey Brantingham. "Graph-Based Deep Modeling and Real Time Forecasting of Sparse Spatio-Temporal Data." *arXiv preprint arXiv:1804.00684*, 2018.

[Woh14] Brendt Wohlberg. "Efficient Convolutional Sparse Coding." In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 7173–7177, May 2014.

[WR96] Christopher KI Williams and Carl Edward Rasmussen. "Gaussian Processes for Regression." 1996.

[WS00] Christopher KI Williams and Matthias Seeger. "Using the Nyström method to speed up kernel machines." In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, pp. 661–667. MIT press, 2000.

[Yed11] Jonathan S Yedidia. "Message-passing algorithms for inference and optimization." *Journal of Statistical Physics*, **145**(4):860–890, 2011.

[YFR07] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. "A support vector method for optimizing average precision." In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 271–278. ACM, 2007.

[YRY02] Alan L Yuille, Anand Rangarajan, and AL Yuille. "The concave-convex procedure (CCCP)." *Advances in neural information processing systems*, **2**:1033–1040, 2002.

[ZBC11] Miao Zheng, Jiajun Bu, Chun Chen, Can Wang, Lijun Zhang, Guang Qiu, and Deng Cai. "Graph regularized sparse coding for image representation." *IEEE Transactions on Image Processing*, **20**(5):1327–1336, 2011.

[ZBL04] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. "Learning with local and global consistency." *Advances in neural information processing systems*, **16**(16):321–328, 2004.

[ZGL03] Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al. "Semi-supervised learning using Gaussian fields and harmonic functions." In *ICML*, volume 3, pp. 912–919, 2003.

[Zha12] Pan Zhang. "Inference of kinetic Ising model on sparse graphs." *Journal of Statistical Physics*, **148**(3):502–512, 2012.

[Zhu] Xiaojin Zhu. "Semi-supervised learning literature survey." *Technical Report TR1530.*

[ZKT10] Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and Rob Fergus. "Deconvolutional networks." In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 2528–2535, June 2010.

[ZLG03] Xiaojin Zhu, John D Lafferty, and Zoubin Ghahramani. "Semi-supervised learning: From Gaussian fields to Gaussian processes." 2003.

[ZP04] Lihi Zelnik-Manor and Pietro Perona. "Self-tuning spectral clustering." In *Advances in neural information processing systems*, pp. 1601–1608, 2004.