Title
Graph-Based Learning and Data Analysis

Author
Li, Hao

Publication Date
2020

UNIVERSITY OF CALIFORNIA

Los Angeles

Graph-Based Learning and Data Analysis

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

by

Hao Li

2020

ABSTRACT OF THE DISSERTATION

Graph-Based Learning and Data Analysis

by

Hao Li
Doctor of Philosophy in Mathematics
University of California, Los Angeles, 2020
Professor Andrea Bertozzi, Chair

We present several results on the subject of graph-based semi-supervised learning and a novel application of network analysis to analyzing complex spatiotemporal data. The first piece of work showcases a specific graph-based semi-supervised learning algorithm in the application to ego-activity classification in body-worn video. The classification method is inspired by three interrelated processes: the Allen–Cahn equation, the Merriman–Bence–Osher scheme, and mean curvature flow. We present results on real-world body-worn videos and demonstrate our method's comparable performance to supervised methods. The second piece of work presents semi-supervised learning problem in the framework of Bayesian inverse problems; we prove posterior consistency and elucidate how hyperparameter choices in the Bayesian model combine to affect the contraction rates of the posterior. The third piece of work presents a method of uncertainty quantification in the aforementioned framework; we also develop the foundations for a system with a human in the loop who serves to provide additional class labels based on the uncertainty quantification. The fourth piece of work further extends the Bayesian inverse problem framework to the active learning problem. We introduce an adaptation of non-Gaussian Bayesian models to allow efficient calculations previously done only on Gaussian models and a novel way of choosing new training data. The last piece of work presents a multivariate point-process model that infers latent relationships from complex spatiotemporal data.

The dissertation of Hao Li is approved.

Stanley J. Osher

Mason Alexander Porter

Luminita Aura Vese

Andrea Bertozzi, Committee Chair

University of California, Los Angeles

2020

To my family, mentors, friends, and collaborators.

Table of Contents

List of Figures

Acknowledgments

First of all, I would like to thank my advisor, Andrea Bertozzi. She has provided me with invaluable advice, and guided me in all aspects of my graduate student life. I would also like to thank other members of my doctoral committee, Mason Porter, Stanley Osher, and Luminita Vese, for being a part of my dissertation committee and for their guidance.

Second of all, I would like to thank all my collaborators. I especially would like to thank Mason Porter, Andrew Stuart, and Jeff Brantingham; through our collaborations, they taught me how to approach a research problem in applied mathematics and how to present scientific results to a multidisciplinary audience. I am also grateful for my many collaborations with Kevin Miller, with whom I had countless fruitful discussions. I would like to thank Bamdad Hosseini who helped me understand the posterior consistency problem and Baichuan Yuan who helped me learn the details of point-process models. I would like to thank Matt Haberland for our collaboration on the body-worn video project, during which he showed me engineers' approach to problem solving. Moreover, I would like to thank Honglin Chen, Alexander Song, Osman Akar, Adam Dhillon, Tiankuang Zhou, Hui Jin, Yanghui Wang, Thien Nguyen, Dominic Yang and Yurun Ge for our collaborations during the summer REU programs.

Lastly, I am grateful of all the support I have had from my friends and family during the last five years of my life. I would like to thank Qi Guo, Xuchen Han, Chuyuan Fu, and Zhuoran Tong for making my time in graduate school memorable. I would also like to thank my parents, Xiaoping Li and Fen Wang, for supporting my many decisions. Last but not the least, I would like to thank Xie He for her unconditional love and support.

This thesis contains content from three publications and two preprints. Chapter 2 is a version of *PDEs on Graphs for Semi-Supervised Learning Applied to First-Person Activity Recognition in Body-Worn Video* [87] that is currently under review by Discrete and Continuous Dynamical Systems. This work was done in collaboration with Matt Haberland; together we supervised undergraduate researchers Honglin Chen, Alexander Song, Osman

Akar, Adam Dhillon, and Tiankuang Zhou. I appreciate their help with labeling body-worn video footage and conducting preliminary experiments. Chapter 3 is a version of *Posterior Consistency of Semi-Supervised Regression on Graphs* [18] which is currently in revision. This work was done in collaboration with Kevin Miller and Bamdad Hosseini under the supervision of Andrew Stuart and Andrea Bertozzi. Bamdad Hosseini and Andrew Stuart introduced Kevin Miller and I to the posterior consistency problem that we study and background knowledge of Bayesian inverse problems. Chapter 4 is a version of *Uncertainty Quantification for Semi-Supervised Multi-Class Classification in Image Processing and Ego-Motion Analysis of Body-Worn Videos* [126]. This work was done in collaboration with undergraduate researchers Yiling Qiao, Change Shi, and Chenjian Wang, whom I supervised together with Matt Haberland. I thank them for their help with conducting preliminary experiments and I thank Xiyang Luo and Andrew Stuart for an introduction to statistical sampling and uncertainty quantification. The whole project was supervised by Andrea Bertozzi who gave me helpful comments and suggestions on the experimental design and presentation. Chapter 5 is a version of *Efficient Graph-Based Active Learning with Probit Likelihood via Gaussian Approximations* [103]. This work was done in collaboration with Kevin Miller under the supervision of Andrea Bertozzi. Kevin Miller brought up the idea of combining our Bayesian inverse problem framework with active learning and proposed a novel active learning strategy. Chapter 6 is a version of *Multivariate Spatiotemporal Hawkes Processes and Network Reconstruction* [168]. This work is done in collaboration with Baichuan Yuan under the supervision of Andrea Bertozzi, Jeffrey Brantingham, and Mason Porter. I thank Baichuan Yuan for contributing the algorithm and helping me learn the details of point-process models. I also apprecaite the collaborators' input on the experimental design and presentation.

The work in this dissertation was supported by NIJ grant 2014-R2-CX-0101, NSF grant DMS-1737770, NSF grant DMS-1952339, NSF grant DMS-1818977, NSF grant DMS-1417674, NSF grant DMS-2027277, and DARPA grant FA8750-18-2-0066.

Curriculum Vitae

2012 – 2016	B.S., Mathematics of Computation, University of California, Los Angeles (UCLA).
2014 – 2016	M.A., Applied Mathematics, University of California, Los Angeles (UCLA).

Publications and Preprints

Andrea L. Bertozzi, Bamdad Hosseini, Hao Li, Kevin Miller, Andrew M. Stuart, “Posterior Consistency of Semi-Supervised Regression on Graphs,” (2020),
https://arxiv.org/abs/2007.12809.

Hao Li, Honglin Chen, Alexander Song, Matt Haberland, Osman Akar, Adam Dhillon, Tiankuang Zhou, Andrea L. Bertozzi, and P. Jeffrey Brantingham, “PDEs on Graphs for Semi-Supervised Learning Applied to First-Person Activity Recognition in Body-Worn Video,” (2020),
https://arxiv.org/abs/1904.09062.

Kevin Miller, Hao Li, and Andrea L. Bertozzi, “Efficient Graph-Based Active Learning with Probit Likelihood via Gaussian Approximations,” (2020), Workshop on Real World Experiment Design and Active Learning, at the International Conference on Machine Learning (ICML), July 18, 2020.

Hui Jin, Xie He, Yanghui Wang, Hao Li, and Andrea L. Bertozzi, “Noisy Subgraph Isomorphisms on Multiplex Networks,” (2019), IEEE International Conference on Big Data (Big Data), pp. 4899–4905, DOI: 10.1109/BigData47090.2019.9005645.

Thien Nguyen, Dominic Yang, Yurun Ge, Hao Li, and Andrea Bertozzi, “Applications of Structural Equivalence to Subgraph Isomorphism on Multichannel Multigraphs,” (2019), IEEE International Conference on Big Data (Big Data), pp. 4913–4920,
DOI: 10.1109/BigData47090.2019.9006538.

Yiling Qiao, Chang Shi, Chenjian Wang, Hao Li, Matt Haberland, Xiyang Luo, Andrew M. Stuart, and Andrea L. Bertozzi. “Uncertainty Quantification for Semi-Supervised Multi-Class Classification in Image Processing and Ego-Motion Analysis of Body-Worn Videos,” (2019), Electronic Imaging, Image Processing: Algorithms and Systems XVII, pp. 264-1–264-7(7),
DOI: https://doi.org/10.2352/ISSN.2470-1173.2019.11.IPAS-264.

Baichuan Yuan, Hao Li, Andrea L. Bertozzi, P. Jeffrey Brantingham, and Mason Porter, “Multivariate Spatiotemporal Hawkes Processes and Network Reconstruction,” (2019), SIAM J. Mathematics of Data Science, 1(2), pp. 356–382.

CHAPTER 1

Introduction

We present several results on the subject of graph-based semi-supervised learning (SSL) in Chapters 2–5 and an application of network analysis to complex spatiotemporal data via point-process models in Chapter 6. Semi-supervised learning is the problem of labeling all points within a dataset (the *unlabeled data*) by utilizing knowledge of a subset of noisy observed labels (the *labeled data*). This is done by exploiting correlations and geometric information present in the dataset combined with label information [77,171]. Semi-supervised learning has been studied extensively in the past two decades and has been successfully applied to, for instance, hyperspectral images [101] and body-worn videos [99]. We focus on graph-based methods that utilize similarity graphs; a similarity is measured for each pair of nodes (i.e. data points) and label information is spread across the similarity graph from a small set of labeled fidelity points. The similarity information is often leveraged via graph Laplacians, which have been used in a myriad of machine-learning methods (see, for instance, [161, 165, 169, 173]). The analogy between the graph Laplacian and the classical Laplacian operator inspires a number of PDE-based classification methods, e.g. [17, 52]; this also motivates the recent development in uncertainty quantification to the machine learning community. In their recent work [19], the authors used an efficient sampling method that was originally developed for PDE-based inverse problems [31] to perform uncertainty quantification for the binary classification problem.

Chapter 2 showcases a specific graph-based semi-supervised learning algorithm in the context of ego-activity classification in body-worn video. The proposed method quantifies the similarities between pairs of short segments of video according to motion-based features. Then, it spreads the label information from a small set of manually labeled video segments to unlabeled data. This process is inspired by three interrelated dynamical processes on graphs: the Allen-Cahn equation [7], the Merriman-Bence-Osher scheme [102], and the mean curvature flow [17]. With the aid of the Nyström extension [47], the proposed algorithm can be scaled to handle the enormous size of body-worn video datasets. We present results on real-world body-worn videos and demonstrate its comparable performance to supervised methods.

In Chapter 3, we study SSL problem in the framework of Bayesian inverse problems (BIPs). In this context the Bayesian formulation has a novel structure in which the unlabeled data defines the prior distribution and the labeled data defines the likelihood. We study posterior consistency; that is, the contraction of the resulting Bayesian posterior distribution onto the ground truth solution in certain parametric limits related to parameters underlying our model. We adopt ideas from spectral clustering [161] in unsupervised learning to construct and analyze the prior arising from a similarity graph constructed from the unlabeled data. This prior information interacts with the labeled data via the likelihood. When the prior information (from the unlabeled data) and the likelihood (from the labeled data) complement each other, then a form of Bayesian posterior consistency can be achieved and the posterior measure on the predicted labels contracts around the ground truth. Furthermore our analysis elucidates how hyperparameter choices in the prior and quantitative measures of clustering in the dataset and the noise in labels combine to affect the contraction rates of the posterior.

Chapter 4 concludes the theoretical discussion and presents a method of uncertainty quantification (UQ) in the aforementioned framework. This work is inspired by [19] of which the authors proposed to pair binary classification problem with UQ. Besides a label assigned to each data point, UQ seeks to estimate a measure of uncertainty; the uncertainty measure helps identify hard-to-classify data points that require further investigation. We push the previously binary UQ methodology to a multi-class setting. We extend the binary graphical probit method to a multi-class version and develop a Gibbs sampler that draws samples from the posterior distribution. We propose a confidence measure for each data point that we find correlates with the classification performance; we observe that data points with higher confidence scores are more likely to be classified correctly. Along with the new methodology and the empirical observations, we develop the foundations for a system with a human in the loop who serves to provide additional class labels based on the confidence scores; our UQ method identifies hard-to-classify data points and the human in the loop assigns ground truth to them, leading to improved classification performance.

Chapter 5 further extends the BIP framework to active learning problem, in which an active learner intelligently select the training data to optimize the overall classification performance. We focus on the *pool-based* active learning paradigm; that is, the active learner has access to a fixed pool of unlabeled data points from which it can decide the next training point. We provide a unifying framework for active learning in many graph-based SSL models based on the BIP framework discussed in Chapter 3. We also introduce an adaptation of non-Gaussian Bayesian models to allow efficient calculations previously done only on Gaussian models and a novel model-change active learning acquisition function built around our adaptation.

Lastly, Chapter 6 presents a multivariate point-process model that enables us to infer latent relationships from complex spatiotemporal data. The inferred relationships provide considerable insights into the structure and dynamics of complex spatiotemporal data [13] via network analysis [111]. In our model, each node in a network is associated with a spatiotemporal point process. The nodes can "trigger" each other, so events that are associated with one node increase the probability that there will be events associated with the other nodes. Such triggering should decrease with both distance and time according to some spatial and temporal kernels. We adopt a nonparametric approach [97] to learn both spatial and temporal kernels from data using an expectation-maximization-type (EM-type) algorithm [160] in the absence of the knowledge of the actual triggering mechanism.

1.1 Notation for Graph-Based Semi-Supervised Learning

In this section, we summarize relevant notation common to Chapters 2–5; notation specific to each individual chapter will be presented in the chapter.

Consider a set of nodes $Z = \{1, \cdots, N\}$ and an associated set of feature vectors $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$. Each feature vector $\mathbf{x}_j$ is assumed to be a point in $\mathbb{R}^d$. We may view X as a function $X : Z \to \mathbb{R}^d$ or as a matrix in $\mathbb{R}^{d\times N}$ with columns given by $\mathbf{x}_j$. We refer to X as the unlabeled data. We assume that every element of Z belongs to one of M classes.

Now let $Z' \subseteq Z$ be a subset of $J \leq N$ nodes and define a function $Y : Z' \to \mathbb{R}^M$, noting that this may also be viewed as a matrix $Y \in \mathbb{R}^{J\times M}$. The rows of Y are denoted by $\{Y(1), \cdots, Y(J)\}$ and comprise a collection of noisy observed labels on Z'; in practice, we use $Y(j) \in \{\mathbf{e}_1, \cdots, \mathbf{e}_M\}$ where the $\mathbf{e}_j \in \mathbb{R}^M$ are the standard coordinate vectors. We refer to Y as the labeled data. We further define $H \in \mathbb{R}^{J\times N}$ to be the matrix obtained by removing the $Z \setminus Z'$ rows of the identity matrix $I_N \in \mathbb{R}^{N\times N}$.

We construct the similarity graph $G = \{Z, W\}$ with vertices Z and self-adjoint weighted adjacency matrix $W = (w_{ij})$. The weights $w_{ij} \geq 0$ reflect the affinity of data pairs $(\mathbf{x}_i, \mathbf{x}_j) \in X \times X$, the edge set of the graph. For example, we may construct W using a kernel $\kappa : \mathbb{R}_+ \to \mathbb{R}_+$ by setting

$$w_{ij} = \kappa(|\mathbf{x}_i - \mathbf{x}_j|). \tag{1.1}$$

The kernel κ is assumed to be positive, non-increasing, and with bounded variance; a natural example is the Gaussian kernel $\kappa(t) = \exp\left(-|t|^2/r^2\right)$, or the indicator function of the interval $[0, r]$, both with bandwidth $r \in \mathbb{R}^+$. Note that (1.1) implies that W is symmetric and the suggested weight constructions lead to w_{ij} which encode the pairwise similarities between the points in X.

Given a weight matrix W with the properties illustrated by this explicit construction, we introduce a *graph Laplacian* operator on G of the form

$$L = D^{-p}(D - W)D^{-p}, \tag{1.2}$$

where $D = \text{diag}\{d_i\}$ with entries $d_i := \sum_{j \in Z} w_{ij}$ is the diagonal degree matrix and $p \in \mathbb{R}$ is a user-defined parameter. Taking $p = 0$ gives the unnormalized Laplacian while $p = 1/2$ gives the normalized Laplacian. Other normalizations of L are also possible and can result in non-symmetric operators; see [60, Sec. 5.1] for a detailed discussion.

We let $\langle \cdot, \cdot \rangle$ denote the Euclidean inner product and $|\cdot|$ the Euclidean norm; we use $\|\cdot\|_2$ to denote the induced operator Euclidean norm on matrices. Recall that $\|\cdot\|_F$ denotes the Frobenius norm on matrices and define $\langle A, B \rangle_F := \text{Tr}\left(A^T B\right)$, the inner-product which induces this norm. We use $\otimes$ to denote the Kronecker product between matrices. Occasionally we use $|S|$ to denote the cardinality of a set S; confusion with the Euclidean distance should not arise as we will clarify the notation based on the context.

CHAPTER 2

Application to Body-Worn Video Classification

2.1 Background

In this chapter, we discuss an application of graph-based semi-supervised learning to the problem of ego-activity recognition in body-worn video; we classify camera-wearer's activities when the videos were recorded. This application takes advantage of both PDE-based image processing of the video using classical optical flow techniques and discrete graph clustering of the video frames according to their ego-activity. This is a version of [87]. This work was done in collaboration with Matt Haberland, with whom I supervised undergraduate researchers Honglin Chen, Alexander Song, Osman Akar, Adam Dhillon, and Tiankuang Zhou. They helped label body-worn video footage and conduct preliminary experiments. I devised and implemented the algorithm and designed and conducted the final set of experiments of which results are shown in this chapter.

With the development of body-worn camera technology, it is now possible and convenient to record continuously for a long period of time, enabling video capture of entire days. Research in summarizing and segmenting egocentric videos recorded by body-worn cameras dates back to the early 2000s [5]. Since then, this has been an active research area due to the advancement in computer vision and machine learning [37]; classifying ego-activities in body-worn video footage is well-studied in the context of sports videos [74] and lifelog videos [44, 120, 122, 135]. The task of activity recognition in body-worn video can be categorized into three lines of research: (1) one relies on object-hand interactions and video content (i.e. what objects and people are in the video), (2) one uses the motion of the camera, and (3) ones uses a combination of the previous two.

Works following the first approach rely on object detection and tracking to classify the camera-wearer's activities, for instance, [43, 44, 88, 120, 139]. Popular benchmark data sets used to validate methods focusing on hand-object interactions are the GTEA (Georgia Tech Egocentric Activity) and GTEA Gaze+ data sets, provided by [44], and ADL-short (Activities of Daily Living) in [135] and ADL-long in [120]. The GTEA and GTEA Gaze+ data sets were recorded by Tobii eye-tracking glasses when wearers are cooking in a natural setting, so these two data sets contain eye-gaze direction information not typically available in other body-worn video data sets. Both ADL data set are recorded with a chest-mounted camera when the wearers are performing various daily tasks indoors.

The second line of research is to recognize activities based on motion analysis. A wide variety of motion features have been proposed in the literature. The authors of [74] used a histogram-based motion feature to classify sports activities in videos recorded by head-mounted GoPro cameras. Ryoo and Matthies [128] proposed a motion descriptor that inspired our feature selection method. In [99], the authors used inferred camera movement signals and their dominant frequencies. Many ways of incorporating temporal information in motion analysis were proposed; for instance, the authors of [129] proposed to apply multiple temporal pooling operators to per-frame motion descriptors. Deep convolutional neural networks were also used to extract motion features; for instance, Abebe and Cavallaro [1] proposed to learn a motion representation by using two-dimensional convolution neural network on stacked spectrograms and a Long Short-Term Memory (LSTM) network. With multiple available features extracted, the authors of [117] proposed a multiple kernel learning method to combine local and global motion features. A benchmark data set for this line of research is the HUJI EgoSeg data set provided by the authors of [121], which was recorded when the wearer is performing a variety of activities in both indoor and outdoor settings.

For the third line of research, methods that utilize both appearance (i.e. object recognition and tracking) and motion cues are often combined with deep learning. Both [122] and [93] use a two-stream deep convolution neural network, one stream for images and another stream for optical flow fields, to discover long-term activities in body-worn video.

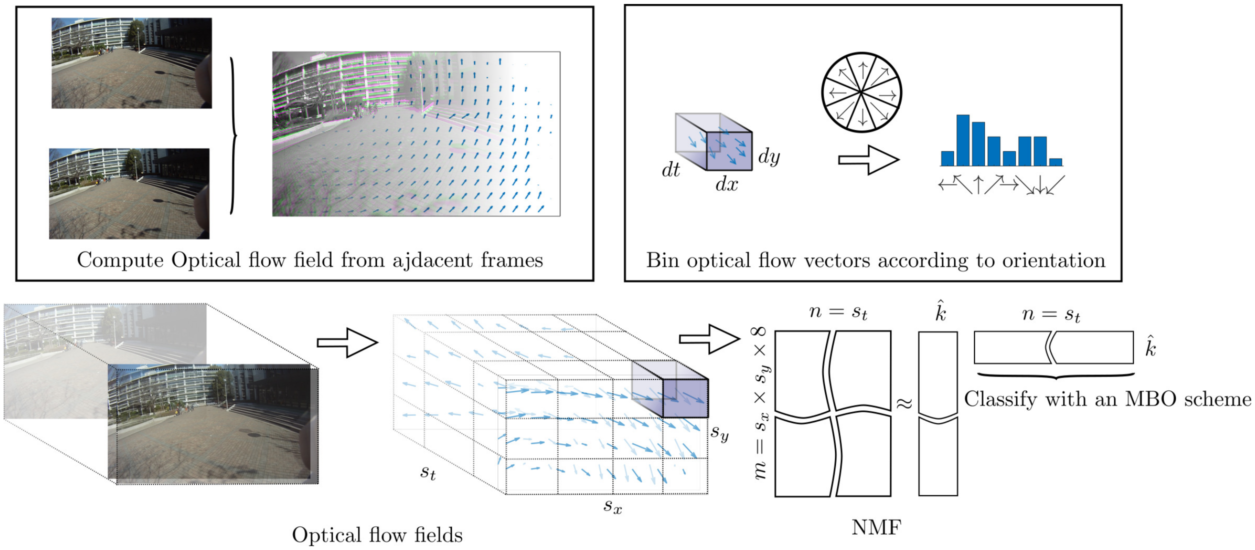

Compute Optical flow field from ajdacent frames
Bin optical flow vectors according to orientation
dt
dx
dy
Optical flow fields
s_t
s_x
s_y
$n = s_t$
$m = s_x \times s_y \times 8$
$\approx$
$\hat{k}$
$n = s_t$
$\hat{k}$
Classify with an MBO scheme
NMF

Figure 2.1: A summary of the proposed method. First, we compute a dense optical flow field for each pair of consecutive frames. We then divide each optical flow field into $s_x \times s_y$ spatial regions, where each region consists of $\mathrm{d}x \times \mathrm{d}y$ pixels, and divide the video into s_t temporal segments, where each segment consists of $\mathrm{d}t$ frames. For each $\mathrm{d}x \times \mathrm{d}y \times \mathrm{d}t$ cuboid, we count the number of flow vectors with direction lying within in each octant, yielding a $s_x \times s_y$ histogram for each segment of video. We reshape and concatenate each histogram into a single feature vector of dimension $s_x \times s_y \times 8$ describing the motion that occurs within the video segment. The dimension of the feature vectors is reduced with NMF and we smooth them with a moving-window average operator. Finally, we classify the smoothed features with a semi-supervised MBO scheme.

Both [20] and [164] use an auto-encoder network to extract motion and appearance features in an unsupervised fashion.

2.2 Method

We start with extracting features based on motion cues from the video. The extracted motion features are potentially high-dimensional, so they are compressed to a lower-dimensional (50 dimensions in our experiments) representation to alleviate computational burden. Finally, we classify the video footage with the low-dimensional representation using a PDE-based semi-supervised learning method with only 10% training data from each class of activity. The flowchart in Figure 2.1 summarizes the proposed system, which we detail below.

2.2.1 Motion Descriptor

Our motion descriptor is similar to the one presented in [128] except for the final dimension reduction step: [128] uses principle component analysis (PCA) whereas we choose non-negative matrix factorization (NMF) because the features are inherently non-negative. Before we compute any feature, we resize all video frames to have a resolution of 576×1024 and hence an aspect ratio of $16:9$, allowing us to choose a uniform set of video parameters across all data sets.

2.2.1.1 Dense Optical Flow Fields

Dense optical flow fields [12, 41, 64, 91], which describe relative motion between objects in the scene and the camera, form the basis of our motion analysis. Optical flow fields are fields of two-dimensional vectors (u, v) defined on the two-dimensional domain of images. In the discrete setting, an optical flow field associates each pixel in an image with an optical flow vector which consists of a horizontal and vertical component. An optical flow field is calculated from a pair of consecutive frames under the assumption that pixels displaced according to the optical flow field should preserve their intensities after the displacement. Formally, let $x(t), y(t)$ be the pixel location of a particular pixel that is displaced according to the optical flow field,

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} u(x(t), y(t)) \\ v(x(t), y(t)) \end{bmatrix}.$$

Then the intensity constancy assumption can be formulated by

$$\frac{\mathrm{d}}{\mathrm{d}t} I\left(x(t), y(t), t\right) = 0, \tag{2.1}$$

which yields the following identity [64],

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0.$$

The well-known Horn-Schunk method then seeks the optical flow field (u, v) by minimizing

$$\iint \left(\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t}\right)^2 + \lambda \left(|\nabla u|^2 + |\nabla v|_2^2\right) \mathrm{d}x \, \mathrm{d}y. \tag{2.2}$$

Note that the first term of (2.2) encourages the flow fields to satisfy the intensity constancy assumption (2.1) while the second term regularizes this ill-posed problem by promoting smooth vector fields. In their original paper, the authors of [64] solve the Euler-Lagrange equation of (2.2) to determine the optical flow fields; myriads of optical flow methods have been proposed in the past three decades and we refer readers to the survey paper [45] for this subject.

Assuming that the objects recorded in a pair of frames are static, the optical flow field encodes the movement of the camera and hence the movement of the camera-wearer. Although this assumption does not necessarily hold perfectly for real-world body-worn video footage, static background objects often cover the majority of frames, and thus we can use optical flow fields to estimate the movement of the camera-wearer. Even when this assumption is not true, we have found that optical flow fields induced by the movement of objects instead of the camera-wearer are still helpful in certain situations. For instance, they characterize driving a car by the static interior of the vehicle and the movement in the windshield region. This is also observed in the experiments conducted by [122]; the authors find distinctive patterns of optical flow fields in the windshield region that correspond well to the camera-wearer driving a car.

2.2.1.2 Histograms on Dense Optical Flow Fields

Using optical flow fields is common in classifying ego-activities. Different motion features are effectively different ways of aggregating them. For instance, the authors of [74, 128, 129] binned optical flow vectors to construct features in the form of concatenated histograms, the authors of [20, 122, 164] aggregated them via convolution kernels, and the authors of [99, 121] inferred camera movement using unaggregated optical flow fields as input. In our case, we compute the motion descriptors, proposed in [128], as histograms of extracted dense optical flow vectors. We bin the vectors according to their locations in the frames and orientations, and then count the number of vectors in each bin. Note that we lose magnitude information in this process because the bins only correspond with locations and orientations. The features proposed in [74] retain magnitude information by further grouping optical flow vectors according to their magnitudes, but in our experiments we observe comparable performance using the simplified features.

To compute the motion descriptors from the optical flow fields, we consider a video as a 3D volume with frames (optical flow fields) stacked along the time axis. We spatially divide each frame into s_x by s_y rectangular regions of fixed width $\mathrm{d}x$ and height $\mathrm{d}y$ pixels; the choice of $\mathrm{d}x$ and $\mathrm{d}y$ determines the spatial resolution of the final features. We have found that choosing $\mathrm{d}x$ and $\mathrm{d}y$ that are divisible by the total number of pixels in length and height, respectively — yielding $s_x = 16$ and $s_y = 9$ — gives good performance on all data sets tested. We also divide the video into s_t video segments, each with a fixed time duration ΔT with $\mathrm{d}t$ frames. We choose ΔT depending on the time scale of the ego-activities that we wish to classify. For instance, we choose $\Delta T = 0.2$ second for videos containing a mix of long term and short term ego-activities, whereas we choose $\Delta T = 4$ seconds if we wish to classify relatively long-term activities. The choice of ΔT also determines the computation cost of the subsequent analysis. A finer time resolution, i.e. a smaller ΔT, yields more video segments for a given video and hence results in more computations.

Consider the optical flow vectors in each $\mathrm{d}x \times \mathrm{d}y \times \mathrm{d}t$ volume. We place each of them into one of the pre-defined eight histogram bins based on its orientation. Formally, a vector is placed in a bin depending on its phase. Repeating the above steps for every $\mathrm{d}x \times dy \times \mathrm{d}t$ volumes in each video segment of duration ΔT, we obtain a feature vector with a dimension of $s_x \times s_y \times 8$ for each segment, which we reshape into a single column vector. By repeating the above procedures for every video segments of length ΔT and stacking obtained feature vectors, we obtain a data matrix $\mathcal{D}$ with the number of columns equal to the number of segments in the video. A detailed description of this procedure is presented in Algorithm 3.

2.2.1.3 Non-Negative Matrix Factorization

The concatenated histograms for each video segment can have $9 \times 16 \times 8 = 1152$ entries, which can potentially be expensive to compute with. To alleviate this problem, we employ dimension reduction techniques. In [128], the authors use the principal component analysis (PCA) to perform dimension reduction. However, we use non-negative matrix factorization (NMF) [85] because the concatenated histograms are inherently non-negative. NMF is widely used in the context of topic modeling, where users want to learn topics, a collection of words that often co-occur in textual documents, each of which is represented by a histogram of words. In our case, each video segment is represented by a histogram of "motion words"; each motion word is the movement of a specific orientation in a specific region of the frame. Analogously, a topic — a collection of motion words — describes a global movement pattern. We then model the concatenated histogram of motion words of each video segment as a non-negative linear combination of the topics.

NMF factorizes a non-negative $d' \times N$ matrix $\mathcal{D}$ (in our case, $d' = s_x \times s_y \times 8$ and $N = s_t$) into the product of two low-rank non-negative $d' \times d$ and $d \times N$ matrices X' and X. The number d is chosen by the users according to their computation resources.We choose $d = 50$ for all considered data sets. Formally, this is achieved by solving the following constrained minimization problem,

$$\min_{X',X} \|\mathcal{D} - X'X\|_F^2, \text{subject to } X' \geq 0, X \geq 0. \tag{2.3}$$

Each column in X' represents a basis vector (a topic), and each entry in X represents the non-negative linear combination coefficients. Each column in the matrix X is the feature vector for a single video segment, which will be passed into our classification algorithm after a post-processing step (detailed in Section 2.2.1.4).

2.2.1.4 Post-Processing

We assume a certain degree of temporal regularity of the extracted features: the duration of activities is typically much longer than transitions between them, and so transitions are relatively rare. We note that none of our previous feature extraction procedures take advantage of this temporal regularity. Each optical flow field is computed from only two adjacent frames, motion descriptors are aggregated within non-overlapping video segments, and NMF treats columns in the data matrix $\mathcal{D}$ (motion descriptors of video segments) independently. We apply a moving-window average operator on each row of X and then pass these averaged features to the classification method.

2.2.2 Classification Method

In this section, we outline a graph-based semi-supervised learning method based on minimizing the graph total variation (TV), which has been studied in [17, 52, 100]. We consider each video segment as a node in a weighted graph. The edge weight between a pair of nodes i and j is chosen to be the similarity

$$w_{ij} = \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{r_{ij}}\right), \tag{2.4}$$

where r_{ij} are scaling constants. Here $\mathbf{x}_i$ is the ith column of matrix X obtained from NMF. The scaling constants can either be the same chosen to be r for all pairs of i and j, or chosen locally for each individual pair [169]. We choose the local scaling constants $r_{ij} = r_i r_j$ where r_i is the distance between i and its kth nearest neighbor.

We aim to partition N nodes into M classes (i.e. ego-activities) such that

1. similar nodes between which edge weights are large (i.e. w_{ij} are close to 1) should be in the same class, and

2. fidelity nodes (i.e. manually labeled nodes) should be classified according to their labels.

To achieve (1), we optimize the graph total variation defined as follows. Let U be an $\{0,1\}^M$-valued assignment function on the set of nodes, that is $U(i) = \mathbf{e}_m$ meaning we assign the ith data point to class m. We can then define the graph total variation

$$|U|_{TV} = \frac{1}{2} \sum_{i,j\in Z} w_{ij} \|U(i) - U(j)\|_1, \tag{2.5}$$

which is referred to as anisotropic total variation by [54]. We observe that (2.5) admits a trivial minimizer that is constant across all nodes. To avoid this problem and to incorporate the training data, we introduce a least-squares data fidelity term

$$\Phi(U;Y) = \frac{1}{2\gamma^2} \sum_{i\in Z'} |U(i) - Y(i)|^2\,; \tag{2.6}$$

recall from Section 1.1 that $Z' \subseteq Z$ is a subset of $J \leq N$ nodes that are labeled and $Y : Z' \to \mathbb{R}^M$ comprises a collection of observed labels on Z'. We weight the fidelity term by a positive parameter $1/\gamma^2$ to balance the graph TV term and the fidelity term in the objective function,

$$\frac{1}{2}|U|_{TV} + \Phi(U;Y). \tag{2.7}$$

Instead of minimizing (2.7) directly, we solve the Ginzburg-Landau relaxation [17] for $U(i) \in \mathbb{R}^M$. Namely, we replace the graph total variation $|U|_{TV}$ with

$$GL_\varepsilon(U) = \frac{1}{4} \sum_{i,j\in Z} w_{ij} |U(i) - U(j)|^2 + \frac{1}{\varepsilon} \sum_{i\in Z} \mathcal{P}\left(U(i)\right), \tag{2.8}$$

where ε is a small positive constant, and $\mathcal{P}$ is a multi-well potential with minima at the corners of the unit simplex, for instance

$$\mathcal{P}\left(U(i)\right) = \prod_{m=1}^{M} \frac{1}{4} \|U(i) - \mathbf{e}_m\|_1^2. \tag{2.9}$$

The authors of [157] prove the following Γ-convergence;

$$GL_\varepsilon(U) \xrightarrow{\Gamma} \begin{cases} |U|_{TV}, & \text{if } U \text{ is binary} \\ +\infty, & \text{otherwise.} \end{cases} \tag{2.10}$$

as $\varepsilon \to 0$ in the case of $M = 2$. The Γ-convergence ensures that the minimizers of $GL_\varepsilon(U)$ approach the minimizers of $|U|_{TV}$ and hence justifies the Ginzburg-Landau relaxation of the Total Variation. After the Ginzburg-Landau relaxation, we arrive at the objective function

$$GL_\varepsilon(U) + \Phi(U;Y), \tag{2.11}$$

which we minimize with respect to U.

To formulate (2.11) in terms of matrices, we first identify U by a $N \times M$ matrix and Y by a $J \times M$ matrix of which the ith row of U and Y is given by $U(i)$ and $Y(i)$, respectively. We can write (2.11) in the matrix form

$$\frac{1}{2}\left\langle U, LU \right\rangle_F + \frac{1}{\varepsilon}\sum_{i\in Z}\mathcal{P}\left(U(i)\right) + \frac{1}{2\gamma^2}\|HU - Y\|_F^2, \tag{2.12}$$

where L is the graph Laplacian matrix with $p = 0$ in (1.2). In practice, we choose L to be the symmetrically normalized Laplacian with $p = 1/2$.

2.2.2.1 Optimization Scheme

Minimizing (2.12) using the standard gradient descent method yields

$$\frac{\partial U}{\partial t} = -LU - \frac{1}{\varepsilon}\nabla\hat{\mathcal{P}}\left(U\right) - \frac{1}{\gamma^2}H^T(HU - Y), \tag{2.13}$$

where $\hat{\mathcal{P}}(U) = \sum_{i\in Z}\mathcal{P}\left(U(i)\right)$. This is known as the graph Allen–Cahn equation. In the continuum, the Allen–Cahn equation converges to the mean curvature flow and an analogous convergence for the graph case has been established in [92]. We follow [98] to use a variant of the Merriman–Bence–Osher (MBO) scheme to approximate and solve (2.13). We note that, in the continuum, the MBO scheme is known to approximate the mean curvature flow, just as the Allen–Cahn equation. An explicit connection between the graph Allen–Cahn equation and the MBO scheme has been explored in the recent artical [26]. In short, we first randomly initialize U^0, which we use as the initial condition for (2.13). We then alternate between the following two steps:

1. *Diffusion*: for given U^j, we obtain $U^{j+\frac{1}{2}}$ by solving a force-driven heat equation

$$\frac{\partial U}{\partial t} = -LU - \frac{1}{\gamma^2}H^T(HU - Y), \tag{2.14}$$

for $t_j \leq t \leq t_j + \frac{1}{2}\Delta t$, where Δt is a parameter.

2. *Threshold*: we threshold $U^{j+\frac{1}{2}}$ to obtain U^{j+1}, i.e.

$$U^{j+1}(i) = \mathbf{e}_{\hat{m}} \text{ , where } \hat{m} = \arg\max_m U_m^{j+\frac{1}{2}}(i). \tag{2.15}$$

   For a small ε, this approximates solving

$$\frac{\partial U}{\partial t} = -\frac{1}{\varepsilon}\nabla\hat{\mathcal{P}}(u) \tag{2.16}$$

   for $t_j + \frac{1}{2}\Delta t \leq t \leq t_{j+1} = t_j + \Delta t$.

Choosing Δt is delicate. If it is too small, $U^{j+1} = U^j$ after thresholding, whereas if it is too large, U converges to the steady-state solution of (2.14),

$$\left(L + \frac{1}{\gamma^2}H^T H\right)^{-1} H^T Y,$$

in one diffusion step, independent of the initial condition U^j. Either way, extreme Δt results in a "freezing" scheme. In [158], the authors give guidance on how to choose Δt in the case of unnormalized graph Laplacian and $M = 2$ (i.e. binary classification). Currently, there is no analogous result for normalized graph Laplacian and multi-class classification. We have found, however, that $\Delta t = 0.1$ gives nontrivial dynamics (i.e. convergent and not "freezing") on all data sets used in testing.

2.2.2.2 Numerical Methods

We follow [17, 52] to employ a semi-implicit ordinary differential equation solver to solve (2.14), and use a pseudo-spectral method coupled with the Nyström extension to make the ordinary differential equation solver efficient. We note that the graph Laplacian matrix L is large, with N^2 entries where N is the number of data points; it is also not inherently sparse, which makes approximation techniques such as the Nyström extension necessary.

For the ordinary differential equation solver, we take N_{step} time steps to reach $U^{j+\frac{1}{2}}$ from U^j, where N_{step} is a parameter to choose. Formally, we let $U^{j,s}$, $s = 0, 1, \cdots, N_{\text{steps}}$ denote the numerical solutions of (2.14) at intermediate time $t_j + s\delta t$, where $\delta t = \Delta t / 2N_{\text{step}}$. We solve

$$\frac{U^{j,s+1} - U^{j,s}}{\delta t} = -LU^{j,s+1} - \frac{1}{\gamma^2}(HU^{j,s} - Y) \tag{2.17}$$

for $U^{j,s+1}$. We use $N_{\text{step}} = 10$ to ensure convergence of the ordinary differential equation solver when $\eta < 500$ and $\Delta t = 0.1$.

We use a pseudo-spectral method to solve (2.17). We project the solution U onto an orthonormal eigenbasis of the graph Laplacian L, or an eigen-subbasis that consists of N_{eig} eigenvectors corresponding to the smallest N_{eig} eigenvalues. We detail how we compute the spectrum of L with the Nyström extension in Section 2.2.3. Choosing an $N_{\text{eig}} \ll N$ will greatly improve the efficiency of the algorithm because solving (2.17) only requires $O(NN_{\text{eig}})$ operations if the eigenvectors and eigenvalues of L are provided. Suppose $\boldsymbol{\phi}$ is an $N \times N_{\text{eig}}$ eigenvector matrix, of which the ith column $\boldsymbol{\phi}_i$ is the eigenvector of L corresponding to the ith smallest eigenvalue λ_i, and Λ is the diagonal matrix containing all N_{eig} smallest eigenvalues λ_j. We let a denote the coordinates we obtain by projecting columns of U onto the eigen-subspace spanned by columns of $\boldsymbol{\phi}$, i.e. $a = \boldsymbol{\phi}^T U$. Solving (2.17) in the eigen-subspace $\boldsymbol{\phi}$ leads to an explicit update rule for a and U:

$$\begin{aligned} a^{j,s+1} &= (I + \delta t\Lambda)^{-1} a^{j,s} - \delta t \frac{1}{\gamma^2} \boldsymbol{\phi}^T H^T (HU^{j,s} - Y)\,, \\ U^{j,s+1} &= \boldsymbol{\phi} a^{j,s+1}\,. \end{aligned}$$

2.2.3 Nyström Extension

We employ the Nyström extension [47], which approximates the eigenvectors and eigenvalues of L with $O\left(NN_{\text{eig}}^3\right)$ computation complexity and $O(NN_{\text{eig}})$ memory requirement. With $N_{\text{eig}} \ll N$, the computation complexity and memory scales linearly with respect to the number of data points. The idea of the Nyström extension is to uniformly randomly sample a smaller set of data points $A \subset Z$ with $|A| = N_{\text{sample}} \ll N$, perform spectral decomposition on an $N_{\text{sample}} \times N_{\text{sample}}$ system calculated from the set of data points A, and then interpolate

Algorithm 1 Graph MBO scheme [17]

```
**Inputs:** $\boldsymbol{\phi}, \Lambda, H, Y, \gamma$, and initial guess $U^0$.
**Outputs:** $U$.
**Initialize** $U^{0,0} = u^0, a^{0,0} = \boldsymbol{\phi}^T U^0$.
**for** $k = 1, 2, \cdots$, MaxIter or $U^j$ has converged **do**
    a. Diffusion:
    **for** $s = 0, 1, \cdots, N_{\text{step}} - 1$ **do**
        $a^{j,s+1} = (I + \delta t\Lambda)^{-1} a^{j,s} - \delta t \frac{1}{\gamma^2} \boldsymbol{\phi}^T H^T (HU^{j,s} - Y)$.
        $U^{j,s+1} = \boldsymbol{\phi} a^{j,s+1}$.
    **end for**
    b. Threshold $U^{j+1/2} = U^{j,N_{\text{step}}}$ :
    **for** $i = 1, 2, \cdots, N$ **do**
        $U^{j+1,0}(i) = \mathbf{e}_{\hat{m}}$, where $\hat{m} = \arg\max_m U_m^{j,N_{\text{step}}}(i)$
    **end for**
**end for**
```

the result to obtain an approximation to the spectral decomposition of the entirety of L. Let A^c be the complement of A. Let W_{AA} denote the weights associated with nodes in set A, and similarly, let $W_{AA^c} = W_{A^cA}^T$ denote weights between nodes in set A and A^c. If we reorder the nodes so that $A = \{1, 2, \cdots, N_{\text{sample}}\}$ and $A^c = \{N_{\text{sample}} + 1, N_{\text{sample}} + 2, \cdots, N\}$, we can rewrite

$$W = \begin{bmatrix} W_{AA} & W_{AA^c} \\ W_{A^cA} & W_{A^cA^c} \end{bmatrix}. \tag{2.18}$$

It can be shown [47] that the matrix $W_{A^cA^c}$ can be approximated by $W_{A^cA^c} \approx W_{A^cA} W_{AA}^{-1} W_{AA^c}$ in the context of approximating the spectral decomposition. The Nyström extension uses this property to approximate the spectrum of W, and henceforth L. We summarize the Nyström extension algorithm to approximate the spectrum of symmetric graph Laplacian in Algorithm 2. An analogous algorithm for unnormalized graph Laplacian can be found in [17]. In Algorithm 2, $\mathbf{1}$ denotes a vector of one's that is used to compute the strength of each nodes, i.e. the sum of weights, and let $\cdot./\cdot$ denote component-wise division between two matrices of the same size. We let $\sqrt{\cdot}$ denote the non-negative square root of each component of a non-negative matrix.

2.3 Experiments

We apply our method on two publicly available data sets, the Quad data set [74], and the HUJI (Hebrew University of Jerusalem) EgoSeg data set [121], and compare our results to those reported in [74, 121, 122, 129, 150]. We also apply both our method and the one proposed in [99][1] on a police body-worn video data set provided by the LAPD. The goal of our research on police body-worn video is to help promote transparency and accountability of law enforcement. Our experimental procedures and parameters are summarized in Table 2.1. The measures of success that we use are precision

$$\frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \times 100\%$$

[1] The implementation of method proposed in [99] was kindly provided by the authors.

Algorithm 2 Nyström extension for symmetrically normalized graph Laplacian [17] [47]

```
**Inputs:** $\{\mathbf{x}_i\}_{i=1}^N$ and $\{r_{ij}\}_{i,j=1}^N$.
**Outputs:** $\boldsymbol{\phi}, \{\lambda_j\}_{j=1}^{N_{\text{eig}}}$.
Uniformly sample $A \subset Z = \{1, 2, \cdots, N\}$ with $|A| = N_{\text{sample}} \geq N_{\text{eig}}$ at random.
Compute $W_{AA}$ and $W_{AA^c}$ using (2.4).
Compute the strength of nodes in $A$, $d_A = W_{AA}\mathbf{1}$.
Approximate the strength of nodes in $B$, $d_B = W_{A^cA}\mathbf{1} + W_{A^cA}W_{AA}^{-1}W_{AA^c}\mathbf{1}$.
Normalize $W_{AA} = W_{AA}./\sqrt{d_A d_A^T}$.
Normalize $W_{AA^c} = W_{AA^c}./\sqrt{d_A d_{A^c}^T}$.
Perform spectral decomposition on $W_{AA} + W_{AA}^{-1/2}W_{AA^c}W_{AA^c}^T W_{AA}^{-1/2}$ to obtain the $N_{\text{eig}}$ largest eigenvalues $\{\omega_i\}_{i=1}^{N_{\text{eig}}}$ and the corresponding eigenvectors $\{\psi_i\}_{i=1}^{N_{\text{eig}}}$. We let $\Psi$ denote the matrix of the eigenvectors and $\Omega$ be a diagonal matrix with $\omega_i$'s on the diagonal.
Output $\lambda_i = 1 - \omega_i$, and $\boldsymbol{\phi} = \begin{bmatrix} W_{AA}^{1/2} \\ W_{A^cA}W_{AA}^{-1/2} \end{bmatrix} \Psi\Omega^{-1/2}$.
```

and recall

$$\frac{\text{True Positive}}{\text{True Positive + False Negative}} \times 100\%,$$

within each class, mean precision and recall directly averaged over all classes, and the overall accuracy, i.e. the percentage of correctly classified data points.

The feature extraction is done on an offline machine to ensure the security of the LAPD video. Subsequent analysis, including the Nyström extension and the graph MBO scheme, is performed on a 2.3GHz machine with Intel Core i7 and 4 GB of memory. Both experiments on the Quad data set and the HUJI EgoSeg data set can be finished within a minute after extracting features; each batch of the LAPD body-worn video data set (see Section 2.3.2 for details) takes around two minutes.

Table 2.1: Setup of experiments on body-worn videos

	Motion feature				NMF	Spectrum of the Graph Laplacian				MBO		
	ΔT (sec)	FPS	Number of segments	Window size (segment)	d	N_{eig}	r_{ij}	N_{sample}	Batch size (segment)	$1/\gamma^2$	Δt	N_{step}
Quad	1/60	60	14,399	-	50	500	$\tau = 1$	1000	-	300	0.1	10
LAPD	1/5	30	274,443	5	50	2000	$k = 100$	2000	30000	400	0.1	10
LAPD [99]	1/5	30	274,443	-	-	2000	$k = 100$	2000	30000	400	0.1	10
HUJI	4	15	36,421	20	50	400	$k = 40$	400	-	300	0.1	10

2.3.1 Quad Data Set

The authors of [74] choreographed and made public the Quad data set, which is about four minutes long and filmed at 60 frames per second. The footage was recorded with a head-mounted Go-Pro Camera while the camera-wearer was undergoing nine ego-activities (reported in Table 2.2), such as walking, jumping, and climbing up stairs[2]. The authors of [74] and [99] tested their ego-activity classification methods on this data set; we follow the same experimental protocol as [99]. Each video "segment" is chosen to be an individual frame and we uniformly sample 10% segments within each category as fidelity in agreement with the protocol employed in [99]. This choice of one frame per segment yields 14,399 segments.

In Table 2.2, we report precision within each category and the mean precision, averaged over nine classes; the authors of [74] have also reported the mean precision and the authors of [99] provided detailed precision per class. Both our method and the method in [99] use 10% of the video, sampled uniformly, as fidelity. The method in [74] is unsupervised and the reported mean precision is calculated after matching the discovered ego-activity categories to the given labels in a way that the best match gives the highest harmonic mean of the precision and recall (i.e. the best F-measure). Our result is overall an improvement upon [99] in terms of precision.

The Quad data set only consists of a short choreographed video, in which activities of

[2] The reported categories of ego-activities are the same ones used in [99] but are different from [74].

Table 2.2: Class proportion and precision of the Quad data set

		Precision		
Class	**Proportion**	[74]	[99]	**Ours**
Jump	14%	-	92%	99%
Stand	13%	-	87%	87%
Walk	12%	-	84%	98%
Step	12%	-	93%	98%
Turn Left	11%	-	89%	96%
Turn Right	10%	-	92%	96%
Run	9%	-	92%	96%
Look Up	8%	-	80%	90%
Look Down	7%	-	84%	89%
Mean	11%	95%	88%	94%

interest have a relatively balanced proportion, and the challenges we observe in the field data sets are absent. However, the experiment on the choreographed data set validates the ability of our method in recognizing ego-activities in body-worn videos. We further test our method and showcase the applicability of our method to data sets that consist of multiple videos of different lengths that are not choreographed and recorded in a variety conditions.

2.3.2 LAPD BWV Data Set

The LAPD body-worn video data set consists of 100 videos with a total length of 15.25 hours recorded at 30 frames per second. The video footage is recorded by cameras mounted on police officers' chests when they are performing a variety of law enforcement activities. The data set consists of videos recorded both inside vehicles and outdoors and under a variety of illumination conditions. We manually annotated each frame of all 100 videos with one of 14 class labels. Although we train on and classify video footage in all 14 categories, we exclude five classes, "exit driver seat", "exit passenger seat", "enter passenger seat", "enter driver seat" and"obscured camera", from performance evaluations of the ego-activity recognition algorithms as they are transitioning activities. We report activity proportions of the selected classes in Table 2.3 and, for completeness, all 14 classes in Table 2.5.

We apply the method in [99] with the provided implementation on the LAPD body-worn video data set. [99] computes a feature vector per frame instead of per short video segment, which consists of 6 frames (0.2 seconds). The average of the frame-wise features over a segment is used as the feature vector of the segment. By doing so, the numbers of video segments to classify in both methods are the same. We apply a moving window average operator with a window size of one second (five segments) to our features. The features of [99] inherently incorporate temporal information, so we use the mean per-frame features averaged over each video segment.

We divide the 274,443 segments into 9 disjoint batches, each of which consists of approximately 30,000 segments. As each segment has a duration of 0.2 seconds, each batch therefore consists of 100 minutes of footage spanning multiple videos. We perform the classification on each batch independently and concatenate the classification results. We note that both our method and the method proposed in [99] make use of the Nyström extension and the MBO scheme described in Section 2.2.3 and 2.2.2, respectively, so they share the same set of parameters. We choose $N_{\text{sample}} = 2000$ and $N_{\text{eig}} = 2000$ to be the same for both methods for each batch so that they share the same computation cost and both give good performance relatively to other choices of parameters. We tested parameters $1/\gamma^2$ ranging from 0.01 to 1000 on both methods and found that $1/\gamma^2 = 400$ and r selected automatically according to [169] with $k = 100$ work well for both methods.

With regards to sampling fidelity points, we use the same protocol as the one used in [99] where we uniformly sample 10% segments within each class. Consequently, we have many more samples of common activities than rare activities.

In Table 2.3, we report the precision and recall within each class and their respective means averaged over the selected nine classes. We refer readers to Table 2.5 for a full table of all 14 classes as well as the overall accuracy, which is the proportion of video segments

Table 2.3: Class proportion, precision, and recall of the selected nine classes in the LAPD body-worn video data set

		Precision		Recall	
Class	**Proportion**	[99]	**Ours**	[99]	**Ours**
Stand still	62%	73%	89%	85%	95%
In stationary car	16%	41%	93%	43%	89%
Walk	9%	38%	70%	19%	59%
In moving car	5%	70%	91%	25%	84%
At car window	0.64%	17%	71%	10%	45%
At car trunk	0.58%	73%	71%	11%	51%
Run	0.33%	96%	75%	11%	53%
Bike	0.33%	85%	86%	14%	75%
Motorcycle	0.08%	100%	92%	10%	71%
Mean	10%	66%	82%	25%	69%

that are correctly classified. We also present a sample of the color-coded classification results in Figure 2.2 and the confusion matrices in Figure 2.3. We report the classification results of the entire 14 ego-activity categories in the LAPD body-worn video data set in Table 2.5 as well as the full confusion matrices in Figure 2.6.

Our method outperforms [99] in most of the categories in terms of precision and is a major improvement according to recall. We theorize that the features proposed in [99] are too simple to distinguish among the increased variety of ego-activities in the larger LAPD body-worn video data set. The features they propose do not make use of the locality of motion within each frame, which we consider crucial in order to differentiate, for instance, driving a car and walking forward. Both activities feature forward motion, but the motion is localized within the windshield region only in the former case.

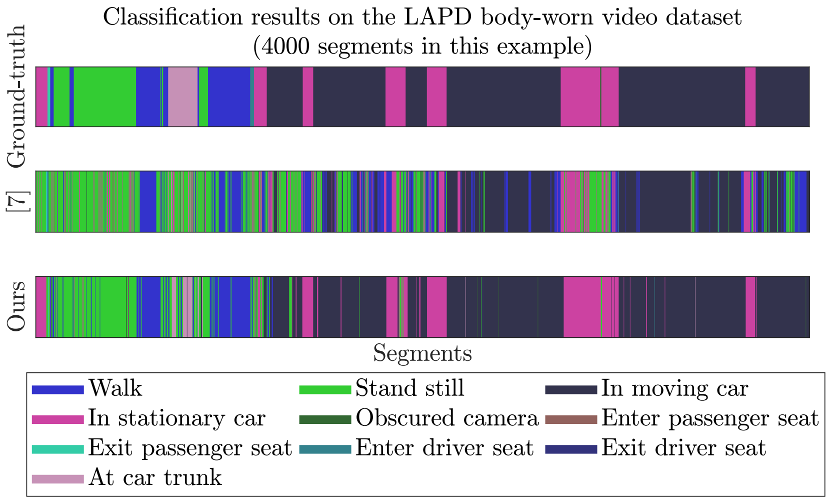
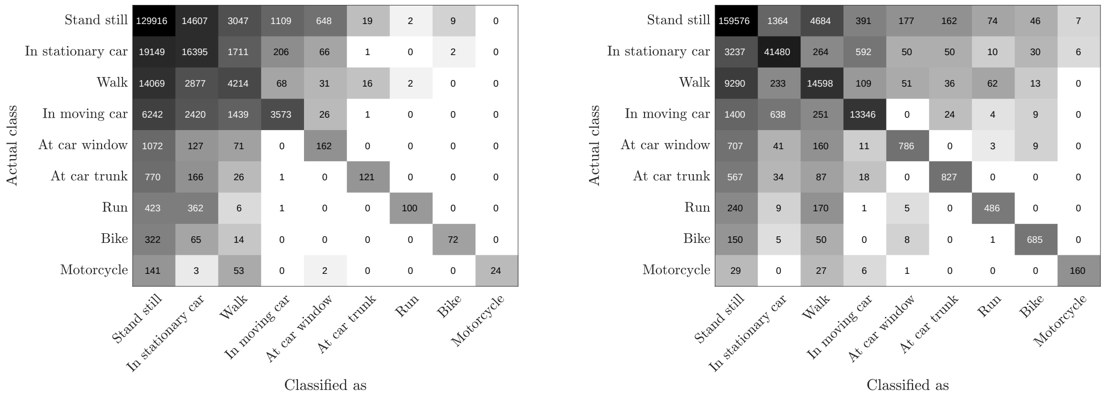


Figure 2.2: Classification results on a contiguous sample of 4000 segments (approximately 13 minutes) from the LAPD body-worn video data set. The results are obtained by running both methods with the parameters described in Section 2.3.2.



Actual class \ Classified as	Stand still	In stationary car	Walk	In moving car	At car window	At car trunk	Run	Bike	Motorcycle
Stand still	129916	14607	3047	1109	648	19	2	9	0
In stationary car	19149	16395	1711	206	66	1	0	2	0
Walk	14069	2877	4214	68	31	16	2	0	0
In moving car	6242	2420	1439	3573	26	1	0	0	0
At car window	1072	127	71	0	162	0	0	0	0
At car trunk	770	166	26	1	0	121	0	0	0
Run	423	362	6	1	0	0	100	0	0
Bike	322	65	14	0	0	0	0	72	0
Motorcycle	141	3	53	0	2	0	0	0	24

(a) Method proposed in [99]

Actual class \ Classified as	Stand still	In stationary car	Walk	In moving car	At car window	At car trunk	Run	Bike	Motorcycle
Stand still	159576	1364	4684	391	177	162	74	46	7
In stationary car	3237	41480	264	592	50	50	10	30	6
Walk	9290	233	14598	109	51	36	62	13	0
In moving car	1400	638	251	13346	0	24	4	9	0
At car window	707	41	160	11	786	0	3	9	0
At car trunk	567	34	87	18	0	827	0	0	0
Run	240	9	170	1	5	0	486	0	0
Bike	150	5	50	0	8	0	1	685	0
Motorcycle	29	0	27	6	1	0	0	0	160

(b) Ours

Figure 2.3: Confusion matrices for the LAPD body-worn video data set. The background intensity in cell (m, m') corresponds to the number of data points in class m that are classified as class m' by the algorithm.

2.3.3 HUJI EgoSeg Data Set

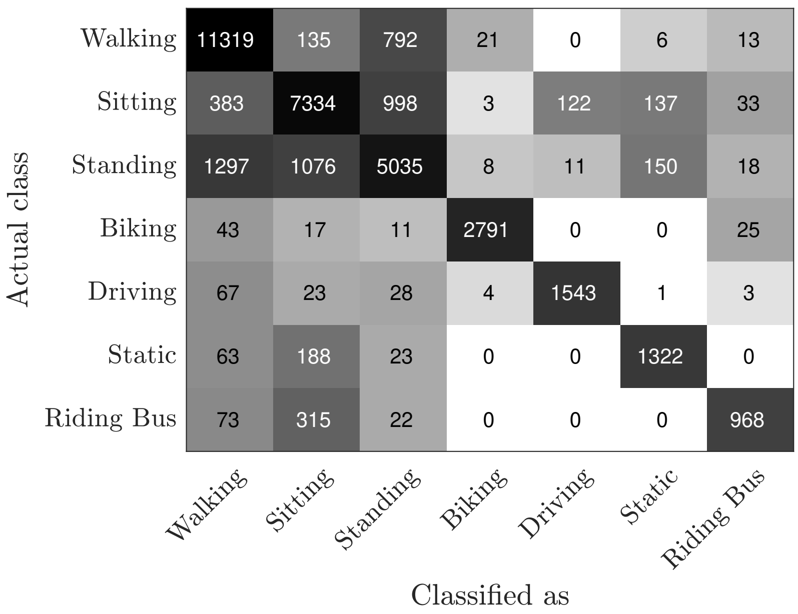
We also evaluate the performance of our method on the HUJI EgoSeg data set [121] [122]. This data set contains 65 hours of egocentric videos including 44 videos shot using a head-mounted GoPro Hero3+, the Disney data set [42] and other YouTube videos[3]. The data set contains 7 ego-action categories: *Walking, Driving, Riding Bus, Biking, Standing, Sitting, and Static*. We normalize the frame rate of each video to 15 frames per second to match with the normalized frame rate in [122]. We divide each video sequence into segments of 4 seconds ($\Delta T = 4$ seconds, 60 frames), which also matches the length of each video segment in [122]. The activities in the HUJI EgoSeg data set are all relatively long-term activities compared to the LAPD body-worn video data set, so using longer video segments reduces the number of data points without the risk of missing short-term activities. With our choice of ΔT, we have 36,421 segments. For the Nyström extension and the MBO scheme, we choose the combination of $N_{\text{sample}} = 400$, $N_{\text{eig}} = 400$, $1/\gamma^2 = 300$, and $k = 40$.

We follow the same experimental protocol of [121, 122] to divide the entire data set into a training set and a testing set. We randomly pick video sequences until we have 1300 segments (approximately 90 minutes of video) per class as the training set, and we uniformly sample 10% of the training set as fidelity points, which is about 10% of the training data used in [122][4]. In this experiment, we use recall to evaluate the performance since it is the common measure of success in [121, 122, 129, 150]. Table 2.4 details the classification results on the testing set. The classification performance of methods other than ours are reported in [122]. We also report the confusion matrix in Figure 2.4 and a color-coded sample of the classification result in Figure 2.5.

We observe that the recalls of “sitting”, “standing”, and “riding bus” are typically lower than other activities across all five methods, so we believe that these activities are inher-

[3] The HUJI EgoSeg data set can be downloaded at http://www.vision.huji.ac.il/egoseg/videos/dataset.html.

[4] The authors of [122] do not explicitly mention the fidelity percentage; we estimate the percentage according to their released code at http://www.vision.huji.ac.il/egoseg/.



	Walking	Sitting	Standing	Biking	Driving	Static	Riding Bus
Walking	11319	135	792	21	0	6	13
Sitting	383	7334	998	3	122	137	33
Standing	1297	1076	5035	8	11	150	18
Biking	43	17	11	2791	0	0	25
Driving	67	23	28	4	1543	1	3
Static	63	188	23	0	0	1322	0
Riding Bus	73	315	22	0	0	0	968

Figure 2.4: Confusion matrix for the HUJI EgoSeg data set. The background intensity in cell (m, m') corresponds to the number of data points in class m that are classified as class m' by the algorithm.

ently difficult to recognize with motion-based features. According to Table 2.4, our method outperforms other methods that use handcrafted motion and/or appearance features with or without deep convolution neural networks, with the exception of [122]. We emphasize that our method uses around one-tenth of the training data of the supervised methods and still achieves comparable results. When we use the entire training set as fidelity, the mean recall only increases slightly.

2.4 Conclusion

In this chapter, we study an application of graph-base semi-supervised learning method to ego-activity recognition in first-person video. We propose a system for classifying ego-activities in body-worn video footage using handcrafted features based on motion cues. Our experiments illustrate that the features are able to differentiate a variety of ego-activities

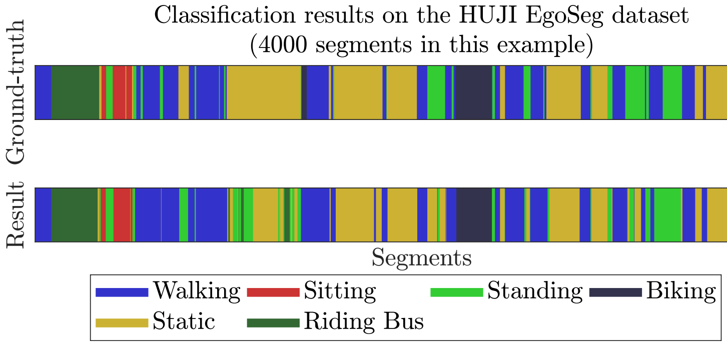


Figure 2.5: Classification results on a contiguous sample of 4000 segments (approximately 4 hours) from the testing set of HUJI EgoSeg data set. The recall of the same experiment is reported in Table 2.4.

Table 2.4: Class proportion and recall of the HUJI EgoSeg data set

		Recall				
Class	**Proportion**	[121]	[129]	[150]	[122]	**Ours**
Walking	34%	83%	91%	79%	89%	91%
Sitting	25%	62%	70%	62%	84%	71%
Standing	21%	47%	44%	62%	79%	47%
Biking	8%	86%	34%	36%	91%	88%
Driving	5%	74%	82%	92%	100%	95%
Static	4%	97%	61%	100%	98%	96%
Riding Bus	4%	43%	37%	58%	82%	84%
Mean	14%	70%	60%	70%	89%	82%
Training		∼60%	∼60%	∼60%	∼60%	6%

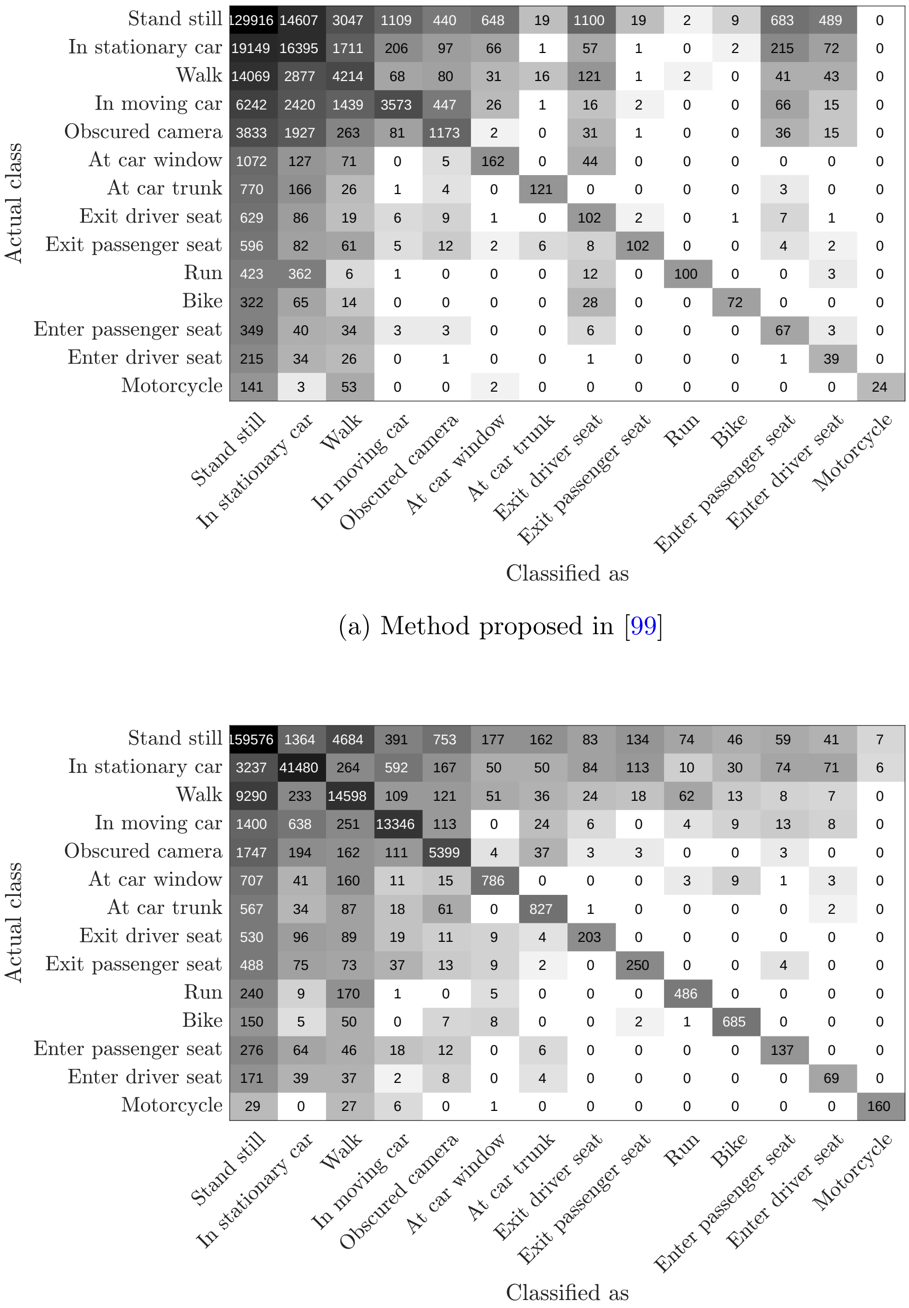
and yield better classification results than an earlier work [99]. The semi-supervised classification method addresses the challenge of insufficient training data; it achieves comparable performance to supervised methods on two publicly available benchmark data sets using only 10% of training data. The proposed system also demonstrates promising results on field data from body-worn cameras used by the Los Angeles Police Department.

Table 2.5: Class proportion, precision , recall, and accuracy on the LAPD body-worn video data set

		Precision		**Recall**	
Class	**Proportion**	[99]	**Ours**	[99]	**Ours**
Stand still	62%	73%	89%	85%	95%
In stationary car	16%	41%	93%	43%	89%
Walk	9%	38%	70%	19%	59%
In moving car	5%	70%	91%	25%	84%
Obscured camera	2%	51%	80%	15%	70%
At car window	0.64%	17%	71%	10%	45%
At car trunk	0.58%	73%	71%	11%	51%
Exit driver	0.35%	6%	50%	11%	21%
Exit passenger	0.34%	79%	48%	11%	26%
Run	0.33%	96%	75%	11%	53%
Bike	0.33%	85%	86%	14%	75%
Enter passenger	0.20%	5%	45%	13%	24%
Enter driver	0.12%	5%	34%	12%	20%
Motorcycle	0.08%	100%	92%	10%	71%
Mean	7%	53%	71%	21%	56%
Accuracy		65%	88%		

Algorithm 3 Global Motion Descriptor

```
**Inputs:** Optical flow fields matrix $O \in \mathbb{R}^{n_f \times n_x \times n_y \times 2}$
**Outputs:** Matrix $X \in \mathbb{R}^{s_t \times (s_x \cdot s_y \cdot 8)}$
**Initialize** $\mathrm{d}t = 60, \mathrm{d}x = \mathrm{d}y = 64, s_x = n_x/\mathrm{d}x, s_y = n_y/\mathrm{d}y,$
    $s_t = \lfloor n_f/dt \rfloor$, histogram count matrix $\mathcal{D}' \in \mathbb{R}^{s_t \times s_x \times s_y \times 8}$
**for** $i = 0 : s_t$ **do**
    **for** $j = 0 : s_x$ **do**
        **for** $k = 0 : s_y$ **do**
            % Step 1. Partition:
            cuboid $= O[i\mathrm{d}t : (i{+}1)\mathrm{d}t, j\mathrm{d}x : (j{+}1)\mathrm{d}x,$
                        $kdy : (k{+}1)dy, :]$
            % reshape: $\mathbb{R}^{\mathrm{d}t \times \mathrm{d}x \times \mathrm{d}y \times 2} \mapsto \mathbb{R}^{(\mathrm{d}t \cdot \mathrm{d}x \cdot \mathrm{d}y) \times 2}$
            cuboid = reshape(cuboid)
            % Step 2. Histogram count:
            **for** $l = 0, 1, \cdots, (\mathrm{d}t \cdot \mathrm{d}x \cdot \mathrm{d}y)$ **do**
                $bin = \lfloor \text{phase}(\text{cuboid}[l, :]) / \frac{\pi}{4} \rfloor$
                $\mathcal{D}'[i, j, k, bin] = \mathcal{D}'[i, j, k, bin] + 1$
            **end for**
        **end for**
    **end for**
**end for**
% reshape: $\mathbb{R}^{s_t \times s_x \times s_y \times 8} \mapsto \mathbb{R}^{(s_x \cdot s_y \cdot 8) \times s_t}$
$\mathcal{D} = \text{reshape}(\mathcal{D}')$
```



Actual class \ Classified as	Stand still	In stationary car	Walk	In moving car	Obscured camera	At car window	At car trunk	Exit driver seat	Exit passenger seat	Run	Bike	Enter passenger seat	Enter driver seat	Motorcycle
Stand still	129916	14607	3047	1109	440	648	19	1100	19	2	9	683	489	0
In stationary car	19149	16395	1711	206	97	66	1	57	1	0	2	215	72	0
Walk	14069	2877	4214	68	80	31	16	121	1	2	0	41	43	0
In moving car	6242	2420	1439	3573	447	26	1	16	2	0	0	66	15	0
Obscured camera	3833	1927	263	81	1173	2	0	31	1	0	0	36	15	0
At car window	1072	127	71	0	5	162	0	44	0	0	0	0	0	0
At car trunk	770	166	26	1	4	0	121	0	0	0	0	3	0	0
Exit driver seat	629	86	19	6	9	1	0	102	2	0	1	7	1	0
Exit passenger seat	596	82	61	5	12	2	6	8	102	0	0	4	2	0
Run	423	362	6	1	0	0	0	12	0	100	0	0	3	0
Bike	322	65	14	0	0	0	0	28	0	0	72	0	0	0
Enter passenger seat	349	40	34	3	3	0	0	6	0	0	0	67	3	0
Enter driver seat	215	34	26	0	1	0	0	1	0	0	0	1	39	0
Motorcycle	141	3	53	0	0	2	0	0	0	0	0	0	0	24

(a) Method proposed in [99]

Actual class \ Classified as	Stand still	In stationary car	Walk	In moving car	Obscured camera	At car window	At car trunk	Exit driver seat	Exit passenger seat	Run	Bike	Enter passenger seat	Enter driver seat	Motorcycle
Stand still	159576	1364	4684	391	753	177	162	83	134	74	46	59	41	7
In stationary car	3237	41480	264	592	167	50	50	84	113	10	30	74	71	6
Walk	9290	233	14598	109	121	51	36	24	18	62	13	8	7	0
In moving car	1400	638	251	13346	113	0	24	6	0	4	9	13	8	0
Obscured camera	1747	194	162	111	5399	4	37	3	3	0	0	3	0	0
At car window	707	41	160	11	15	786	0	0	0	3	9	1	3	0
At car trunk	567	34	87	18	61	0	827	1	0	0	0	0	2	0
Exit driver seat	530	96	89	19	11	9	4	203	0	0	0	0	0	0
Exit passenger seat	488	75	73	37	13	9	2	0	250	0	0	4	0	0
Run	240	9	170	1	0	5	0	0	0	486	0	0	0	0
Bike	150	5	50	0	7	8	0	0	2	1	685	0	0	0
Enter passenger seat	276	64	46	18	12	0	6	0	0	0	0	137	0	0
Enter driver seat	171	39	37	2	8	0	4	0	0	0	0	0	69	0
Motorcycle	29	0	27	6	0	1	0	0	0	0	0	0	0	160

(b) Ours

Figure 2.6: Confusion matrices for the LAPD police body-worn video data set. The background intensity of cell (m, m') corresponds to the number of data points in class m that are classified as class m' by the algorithm.

CHAPTER 3

Bayesian Framework and Posterior Consistency

3.1 Background

In this chapter, we study SSL problem in Bayesian inverse problems (BIPs) framework, building on a widely adopted semi-supervised regression (SSR) approach to SSL developed in the machine-learning community. In this context, the Bayesian formulation has a novel structure in which the unlabeled data defines the prior distribution and the labeled data defines the likelihood. This chapter is a version of [18]. This work was done in collaboration with Kevin Miller and Bamdad Hosseini under the supervision of Andrew Stuart and Andrea Bertozzi. Bamdad Hosseini and Andrew Stuart introduced me to the posterior consistency problem that we study and background knowledge of Bayesian inverse problems. I contributed most of the proofs and designing numerical experiments to illustrate the theoretical results.

The goal of this chapter is to study posterior consistency; that is, the contraction of the resulting Bayesian posterior distribution onto the ground-truth solution in certain parametric limits related to parameters underlying our model. We adopt ideas from spectral clustering in unsupervised learning to construct and analyze the prior arising from a similarity graph constructed from the unlabeled data. This prior information is combined with the labeled data via the likelihood. When the prior information (from the unlabeled data) and the likelihood (from the labeled data) complement each other, then a form of Bayesian posterior consistency can be achieved and the posterior measure on the predicted labels contracts around the ground truth. Furthermore our analysis elucidates how hyperparameter choices in the prior, quantitative measures of clustering in the data set and the noise in labels combine to affect the contraction rates of the posterior. In the following three subsections, we review relevant literature, formulate the problem mathematically and describe our contributions.

3.1.1 Relevant Literature

Many approaches to SSL and SSR have been developed in the literature and a detailed discussion of all of them is outside the scope of this chapter. We refer the reader to the review articles [171] and [77] for, respectively, the state-of-the-art in 2005 and a more recent appraisal of the field.

The consistency of supervised learning and regression is well-developed; see [147] for a literature review, as well as the preceding work in [141,142,167] which establish the problem in the framework of [159]. All of this work on supervised classification focuses on the large data/large number of features setting, and often considers only linearly separable data; that is, a straight line can separate the data correctly according to the ground-truth labels. Therefore, these previous works do not leverage the power of graph-based techniques to extract geometric information in large unlabeled data sets, a primary feature of the SSR problems studied in this work.

Graph-based techniques are widely used in unsupervised learning [15,161], a subject that has seen significant analysis in relation to consistency. The papers [137,138] perform a careful analysis of the spectral gaps of graph Laplacians resulting from clustered data, studying recursive methods for multi-class clustering. The paper [112] introduced an approach for the analysis of multi-class unsupervised learning based on perturbations of a perfectly clustered case. The paper [162] introduced the idea of studying the consistency of spectral clustering in the limit of large data sets in which the graph Laplacians converge to a limiting integral operator. The articles [152, 153] took this idea further by proving the convergence of graph Laplacian operators to differential operators by controlling the local connectivity of the graph as a function of the number of nodes.

In this chapter, our focus is on SSL [77] in the framework of the influential papers [173,174] where the categorical labels $\{1, \dots, M\}$ are embedded in $\mathbb{R}^M$ and the SSR approach to SSL is adopted. Bertozzi and Flenner [17] introduced an interesting relaxation of this assumption, by means of a Ginzburg-Landau penalty term which favors real-values close to ±1 but does not enforce the categorical values ±1 exactly. In contrast to these relaxations, the probit approach to classification, described in the classic text on Gaussian process regression [127] and analyzed in [61] in the context of SSL, works directly with the categorical labels and does not rely on the embedding step.

The idea of regularization by graph Laplacians for SSL was developed in different contexts such as manifold regularization [16], Tikhonov regularization [14] and local learning regularization [166] as well as more recent articles focusing on large data settings [58, 59]. However, while graph regularization methods are widely applied in practice the rigorous analysis of their properties, and in particular asymptotic consistency and posterior contraction rates, are not well-developed within the context of SSL and SSR. Indeed, to the best of our knowledge the Bayesian consistency of SSR has not been analyzed. Studying SSL/SSR in a Bayesian setting introduces new challenges that require careful consideration about assumptions regarding graph structure and statistical properties of the resulting model [73]. We build on the spectral analysis of the graph Laplacian introduced in [112] to study unsupervised learning, and refined in [61] to study the consistency of optimization-based approaches to binary and one-hot SSL.

The subject of Bayesian posterior consistency is aimed at reconciling the large data limits of frequentist and Bayesian approaches to statistical inference problems. Early influential works in this field concentrated on negative results concerning the Bayesian nonparametric setting where the prior and likelihood were inconsistent [38]. Subsequent work in this area concentrated on positive results, demonstrating that minimax rates of convergence can be obtained within the Bayesian setting [53, 155] by studying posterior measure concentration through Bernstein-Von Mises-type theorems [50, 155] provided that priors are constructed carefully. The celebrated paper [25] demonstrates how large data and small noise limits (i.e. in the limit of noise variance going to zero) are intimately related, and this link underpins subsequent studies of inverse problems from the perspective of Bayesian posterior consistency. This line of work was initiated in the paper [156] where the small noise limit of linear inverse problems was studied. A number of papers in this area followed [3, 107] and it is currently an active research area, particularly in relation to nonlinear inverse problems [55].

In some problems, optimization approaches other than fully Bayesian approaches are adopted, and the study of consistency for inverse problems in this setting is overviewed in [40]. Linking this to maximum a posteriori (MAP) estimators for inverse problems was a subject developed in [35] and the study of consistency for MAP estimators in semi-supervised learning, and in particular use of the probit likelihood model, is undertaken in [61].

3.1.2 Problem Setup

Underlying this analysis is the assumption that the labeled data is determined by a generative model of the form

$$Y = HU^\dagger + \gamma\eta, \tag{3.1}$$

here $U^\dagger \in \mathbb{R}^{N\times M}$ is the ground-truth latent variable that gives the true labels of all of the nodes in Z and $\eta \in \mathbb{R}^{J\times M}$ is a matrix with independent standard Gaussian entries, i.e., $\eta_{jm} \overset{iid}{\sim} \mathcal{N}(0,1)$. The parameter $\gamma > 0$ is the standard deviation of the observation noise. It is instructive to think of the rows of $U^\dagger$ as being chosen from $\{\mathbf{e}_1, \cdots, \mathbf{e}_M\}$, although generalizations of this setting are possible.

The model (3.1) casts the SSL problem of inferring the true labels on Z as the SSR problem of finding $U^\dagger$, adopting the terminology of [77]: our modeling assumption makes the observations Y real-valued, rather than categorical as in classification, and therefore is considered a regression problem. The SSR problem is ill-posed, requiring the learning of NM parameters from JM noisy data points, since we typically have far fewer labels than the total number of unlabeled data points, i.e. $J \ll N$. The labeled data may be viewed as providing prior information that renders this ill-posed problem tractable. To this end, we formulate SSR in the framework of Bayesian inversion [28, 36, 70].

The main goal is to analyze the consistency of the Bayesian SSR problem by identifying the conditions under which the posterior measure μ^Y (defined in (3.7) below) contracts around the ground-truth matrix $U^\dagger$ in (3.1). Formally, we define the following functional as a measure of posterior contraction

$$\mathcal{I} := \mathbb{E}_{Y|U^\dagger}\mathbb{E}_{\mu^Y}\left\|U - U^\dagger\right\|_F^2 , \tag{3.2}$$

where the inner expectation is with respect to the posterior measure μ^Y on U while the outer expectation is with respect to the law of $Y|U^\dagger$ following (3.1). With this notation, our aim is to solve the following problem:

Problem 1 (Posterior consistency of Bayesian SSR)**.** *Under what conditions on the graph G, the labeled set Z', the ground-truth $U^\dagger$ and the hyperparameters τ, α entering the definition of the prior can we ensure that $\mathcal{I} \downarrow 0$ as the noise-level γ in the unlabeled data, and some measure ϵ of closeness to perfect clustering in the labeled data, tend to zero.*

Indeed we will find explicit bounds on $\mathcal{I}$ which give consistency in the limit $(\epsilon, \gamma) \to 0$ and reveal the role of parameter choices for τ, α in the form of the contraction rate. Our bounds are applicable for small values of γ, τ, ϵ (the explicit condition under which the bounds hold will be presented) and not just in the asymptotic regimes where $(\gamma, \tau, \epsilon) \to 0$.

3.1.3 Main Results

We study posterior contraction, as measured by the quantity $\mathcal{I}$. In the theory we develop, the quantity of labeled data and unlabeled data will be fixed. The prior that we use is a discrete analog of the Matérn prior with graph Laplacian used in place of the continuum Laplacian in the differential operator formulation popularized in [90]. In this interpretation τ is an inverse length-scale and α controls the regularity of the prior; details are given in the next section. The parameter γ is the noise standard deviation in (3.1) and the parameter ϵ is defined formally through the notion of a weakly connected graph as introduced in [112] and used in [61]:

Definition 1 (Weakly connected graph)**.** *Let $0 < \epsilon \ll 1$, then a graph $G = \{Z, W\}$ is weakly connected with K clusters if it consists of pathwise connected components $\widetilde{G}_k = \{\widetilde{Z}_k, \widetilde{W}_k\}$ for*

$k = 1, \ldots, K$ so that the edge weights between elements in different $\widetilde{G}_k$ are $\mathcal{O}(\epsilon)$. In other words, up to a reordering of Z, the matrix W is an $\mathcal{O}(\epsilon)$ perturbation of a block diagonal weight matrix, and the graph Laplacian associated with each block has a one-dimensional null-space.

The following informal theorem is stated with precise conditions as Corollary 1 which itself follows from Theorem 2, both stated and proved in Section 3.3.

Main Theorem. Let $G = \{Z, W\}$ be weakly connected with K components $\widetilde{G}_k$ and perturbation parameter $0 < \epsilon \ll 1$ as in Definition 1. Suppose that the columns of the ground-truth matrix $U^\dagger \in \mathbb{R}^{N\times M}$ belong to the span of the indicator functions of the $\widetilde{G}_k$ and fix $\alpha > 0$ and fix τ so that

$$\epsilon = \epsilon_0 \tau^{\max\{2,2\alpha\}}.$$

Then, for appropriately chosen ϵ_0, there exists $\Xi > 0$, independent of ϵ and γ, so that

$$\mathcal{I} \leq \Xi \max\left\{\gamma^2, \epsilon^{\min\{1,\alpha\}}\right\}.$$

After stating the theorem, let us give insight into it. The parameters ϵ and γ are inherent to the specific SSR problem and the data set at hand. Broadly speaking ϵ is a geometric property of the point cloud X of unlabeled data and γ is the noise standard deviation of the labels. Hence these parameters are fixed, although they are generally unknown. Then the theorem implies the following:

- If $\epsilon^{\min\{1,\alpha\}} \leq \gamma^2$, then the measurement noise dominates over the measure of closeness to perfect clustering and posterior contraction is controlled by the γ parameter.
- If $\gamma^2 < \epsilon^{\min\{1,\alpha\}}$, then the measure of closeness to perfect clustering is dominant in comparison to the measurement noise, and posterior contraction is controlled by the ϵ parameter.
- In the latter case, we also observe that choosing $\alpha < 1$ gives a sublinear contraction rate in ϵ while a linear rate is achieved if $\alpha \geq 1$. Thus it is preferable to tune (α, τ^2)

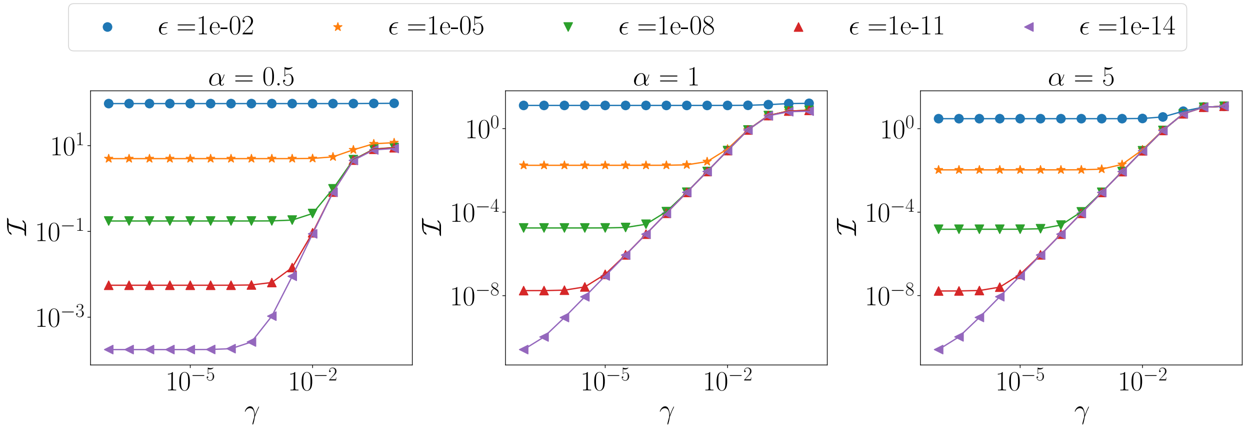


Figure 3.1: A numerical demonstration of Theorem 2 on a synthetic data set (detailed in Subsection 3.4.1). Details of this experiment are described in Section 3.4. The value of $\mathcal{I}$ reduces with γ up to the point where $\gamma^2 \approx \epsilon^{\min\{1,\alpha\}}$ where the errors saturate as predicted by the upper bound in the main theorem. Smaller values of ϵ result in smaller values of $\mathcal{I}$ that indicates higher concentration of posterior probability mass around the ground-truth $U^\dagger$.

so that $\alpha \geq 1$ and $\tau^2 = \mathcal{O}(\epsilon^{1/\alpha})$. For reasons related to the large data limit $N \to \infty$, it is natural to choose $\alpha > \frac{d}{2}$ and since d is typically larger than 2, this enforces $\alpha > 1$; see [60].

These insights are also supported by our numerical experiments in Section 3.4; furthermore these experiments also verify the sharpness of the upper bound in Theorem 2. As a prelude to these detailed experiments, Figure 3.1 contains the results of a computational example which illustrates our main theorem on a synthetic data set. We postpone details of this experimental set-up to Section 3.4, but studying the figure at this point already gives useful insight: for fixed values of ϵ the value of $\mathcal{I}$ goes to zero at a rate proportional to γ^2 until an inflection point, around $\gamma^2 \approx \epsilon^{\min\{1,\alpha\}}$, after which the error saturates; the saturation levels themselves go to zero like $\epsilon^{\min\{1,\alpha\}}$. These facts are exactly as predicted by our theory.

The rest of this chapter is structured as follows. We outline the details of the Bayesian SSR problem in Section 3.2, introducing the likelihood and the prior in Subsections 3.2.1 and 3.2.2 followed by an analytic expression for the posterior measure in Subsection 3.2.3.

Section 3.3 is dedicated to our consistency analysis and presents detailed versions of our primary results that are summarized in the Main Theorem. We first analyze the disconnected graph case in Subsection 3.3.1 to gain some insight into the behavior of the posterior. We then study the weakly connected graph setting in Subsection 3.3.2. We present the proofs of these results, relying on lemmata that are stated in Section 3.3, but deferring their proof to Section 3.5. We collect numerical experiments in Section 3.4 that demonstrate the sharpness of the contraction rates and bounds obtained in Section 3.3. We present experiments which illustrate situations in which the label noise dominates the closeness to clustering, and vice versa. Section 3.5 contains the detailed proofs of the lemmata that support the main theoretical results developed in Section 3.3; these are also illustrated by numerical results presented in Subsections 3.6.1 and 3.6.2.

3.2 Bayesian Formulation Of SSR

In this section we outline the Bayesian formulation of the SSR problem in detail. We derive the likelihood potential Φ in Subsection 3.2.1 and construct the prior measure in Subsection 3.2.2. An analytic expression for the posterior measure is given in Subsection 3.2.3.

3.2.1 The Likelihood

Based on the generative model (3.1) for the labeled data $Y \in \mathbb{R}^{J\times M}$, we define the likelihood distribution $\mathbb{P}(Y|U)$ with density proportional to

$$\exp\left(-\frac{1}{2\gamma^2}\left\|HU - Y\right\|_F^2\right). \tag{3.3}$$

It is therefore convenient to define the likelihood potential

$$\Phi : \mathbb{R}^{N\times M} \times \mathbb{R}^{J\times M} \to \mathbb{R}^+, \qquad \Phi(U;Y) := \frac{1}{2\gamma^2}\|HU - Y\|_F^2. \tag{3.4}$$

Remark 1. *We note that if the entries in the noise matrix η are not independent but rather correlated, then the expression (3.4) needs to be modified by weighting the $\|\cdot\|_F$ norm by the inverse square root of the covariance operator of η. This will make no significant difference* *to what follows and we work with i.i.d. noise only to simplify the exposition.*

3.2.2 The Prior

We now detail the Gaussian prior measure construction and demonstrate how it expresses the geometric information in the unlabeled data X. Recall the definition of the graph Laplacian in Section 1.1; we define the prior covariance matrix $C_\tau \in \mathbb{R}^{N\times N}$ with hyperparameters $\tau^2, \alpha > 0$ to be

$$C_\tau := \tau^{2\alpha}(L + \tau^2 I_N)^{-\alpha}. \tag{3.5}$$

Graph Laplacian operators are positive semi-definite (see [161, Prop. 1]); the matrix C_τ is therefore strictly positive definite thanks to the shift by $\tau^2 I_N$. The normalization by $\tau^{2\alpha}$ ensures that the largest eigenvalue of C_τ is 1, while $\alpha > 0$ controls the rate of decay of the rest of the eigenvalues of C_τ; when the graph Laplacian is constructed from nearly clustered data, C_τ will exhibit a spectral gap and the eigenvectors associated with eigenvalues near one will contain geometric information about the clusters; we refer to this phenomenon as the *smoothing effect* of C_τ.

With C_τ at hand, we conclude our definition of the prior on the unknown U, the Gaussian measure $\mu_0(\mathrm{d}U) = \mathcal{N}(0, I_M \otimes C_\tau)$ with Lebesgue density

$$\mu_0(\mathrm{d}U) := \frac{1}{[(2\pi)^N \det(C_\tau)]^{\frac{M}{2}}} \exp\left(-\frac{1}{2}\langle U, C_\tau^{-1}U\rangle_F\right) \mathrm{d}U. \tag{3.6}$$

If we introduce the columns $\{\mathbf{u}_1, \cdots, \mathbf{u}_M\}$ of U, then we note the prior can be written as

$$\mu_0(\mathrm{d}U) = \frac{1}{[(2\pi)^N \det(C_\tau)]^{\frac{M}{2}}} \prod_{m=1}^{M} \exp\left(-\frac{1}{2}\left\langle \mathbf{u}_m, C_\tau^{-1}\mathbf{u}_m\right\rangle\right) \mathrm{d}\mathbf{u}_m.$$

The above expression reveals that, a priori, each column of U has the same distribution, and is independent of the others, and that this distribution on columns favours structure across Z which reflects the eigenvectors of the largest eigenvalues of C_τ. The matrix C_τ is chosen so that this eigenstructure reflects clustering present in the unlabeled data, for appropriately chosen τ, determined through the analysis in this chapter.

Remark 2. *The prior covariance C_τ defined in (3.5) depends on the unlabeled data X through the matrix L and the weight matrix W. This perspective differs significantly from standard BIPs, where the data only appears in the likelihood and the prior is constructed independent of the data (other than, perhaps, a noise-dependent scaling). In our formulation of SSR, the labeled data appear in the likelihood potential Φ while the unlabeled data are used to construct the prior measure μ_0.*

3.2.3 The Posterior

Using Bayes' rule, we can determine the posterior μ^Y from the likelihood $\mathbb{P}(Y|U)$ and prior μ_0 defined through the Radon-Nikodym derivative

$$\frac{\mathrm{d}\mu^Y}{\mathrm{d}\mu_0}(U) = \frac{1}{\vartheta(Y)} \exp\Big(- \Phi(U;Y)\Big). \tag{3.7}$$

The posterior measure μ^Y is the Gaussian defined by

$$\mu^Y(\mathrm{d}U) = \frac{1}{\vartheta(Y)} \exp\left(-\frac{1}{2\gamma^2} \|HU - Y\|_F^2 - \frac{1}{2}\left\langle U, C_\tau^{-1} U\right\rangle_F \right) \mathrm{d}U. \tag{3.8}$$

It is well-known that linear inverse problems with additive Gaussian noise and a Gaussian prior result in Gaussian posteriors [165]; this is due to the conjugacy of the prior and the likelihood. In this case, we have the additional property that the independence of the columns $\mathbf{u}_\ell$ of U under the prior μ_0 is preserved under the posterior μ^Y. To see this, we introduce the columns $\{\mathbf{y}_1, \cdots, \mathbf{y}_M\}$ of Y and note that we may write

$$\mu^Y(\mathrm{d}U) \propto \exp\left[-\frac{1}{2}\sum_{m=1}^{M} \frac{1}{\gamma^2} \left| H\mathbf{u}_m - \mathbf{y}_m \right|^2 + \left\langle \mathbf{u}_m, C_\tau^{-1}\mathbf{u}_m \right\rangle \right].$$

Using this structure as the product of i.i.d. Gaussians in each of the M columns of U, Proposition 1 shows that $\mu^Y = \mathcal{N}(U^*, I \otimes C^*)$, where $U^* \in \mathbb{R}^{N\times M}$ is the matrix with columns

$$\mathbf{u}_m^* = \frac{1}{\gamma^2} C^* H^T \mathbf{y}_m, \qquad m = 1, \ldots, M,$$

and C^* is the covariance matrix

$$C^* = \left(C_\tau^{-1} + \frac{1}{\gamma^2} H^T H \right)^{-1}.$$

3.3 Consistency Of Bayesian SSR

In this section, we prove consistency of the posterior μ^Y. We study consistency with respect to two small parameters: γ, which measures noise in the the labeled data Y, and ϵ, which measures the closeness to perfectly clustered unlabeled data X. Recall from the main theorem that our goal is to show that the measure of contraction $\mathcal{I}$ (defined in (3.2)) is controlled with the noise standard deviation γ or the geometric perturbation parameter ϵ, whenever the prior hyperparameters τ, α satisfy certain conditions that will be presented in the following sections. We will show that letting $\gamma \to 0$ results in posterior contraction, until a floor is reached that is determined by ϵ. Furthermore the analysis will reveal guidance about the choice of the hyperparameters τ and α in the prior. In Section 3.3.1 we consider the case of a disconnected graph with $\epsilon = 0$ and obtain contraction rates with respect to γ. In Section 3.3.2 we build on the disconnected case to obtain our desired results for weakly connected graphs with ϵ small.

3.3.1 Disconnected Graph

Consider a weighted graph $G_0 = \{Z, W_0\}$ consisting of $K < N$ connected components $\widetilde{G}_k$, i.e., the subgraphs $\widetilde{G}_k$ are pathwise connected — any two nodes can be joined by a path within $\widetilde{G}_k$ — but the weight of edges that connect two distinct components $\widetilde{G}_i, \widetilde{G}_\ell$ are zero. Without loss of generality, we assume that the nodes in Z are ordered so that $Z = \{\widetilde{Z}_1, \widetilde{Z}_2, \cdots, \widetilde{Z}_K\}$ with the $\widetilde{Z}_k$ denoting the index set of nodes in subgraph $\widetilde{G}_k$. We refer to $\widetilde{Z}_k$ as the clusters and let $\widetilde{N}_k = |\widetilde{Z}_k|$ denote the number of nodes in the k-th cluster. We make the following assumptions on the graph G_0.

Assumption 1. *The graph $G_0 = \{Z, W_0\}$ satisfies the following conditions:*

(a) The weighted matrix $W_0 \in \mathbb{R}^{N \times N}$ is block diagonal

$$W_0 = \mathrm{diag}(\widetilde{W}_1, \widetilde{W}_2, \cdots, \widetilde{W}_K),$$

with $\widetilde{W}_k \in \mathbb{R}^{\widetilde{N}_k \times \widetilde{N}_k}$ denoting the weight matrices of the subgraphs $\widetilde{G}_k$.

(b) Let $\widetilde{L}_k$ be the graph Laplacian matrices of the subgraphs $\widetilde{G}_k$, i.e.,

$$\widetilde{L}_k := \widetilde{D}_k^{-p}(\widetilde{D}_k - \widetilde{W}_k)\widetilde{D}_k^{-p}$$

with $\widetilde{D}_k$ denoting the degree matrix of $\widetilde{W}_k$. There exists a uniform constant $\theta > 0$ so that for $k = 1, \cdots, K$ the submatrices $\widetilde{L}_k$ satisfy

$$\langle \mathbf{v}, \widetilde{L}_k \mathbf{v} \rangle \geq \theta \langle \mathbf{v}, \mathbf{v} \rangle, \tag{3.9}$$

for all vectors $\mathbf{v} \in \mathbb{R}^{\widetilde{N}_k}$ and $\mathbf{v} \perp \widetilde{D}_k^p \mathbf{1}$ with $\mathbf{1} \in \mathbb{R}^{\widetilde{N}_k}$ denoting the vector of ones. In other words, the $\widetilde{L}_k$ have a uniform spectral gap.

Remark 3. *The existence of such θ as in (3.9) follows from [161, Props. 2 and 3], which states that 0 is an eigenvalue of $\widetilde{L}_k$ with multiplicity 1 and that the corresponding eigenvector is $\widetilde{D}_k^p \mathbf{1}$, under the pathwise connected assumption.*

With a disconnected graph G_0 as above, we proceed as in Section 3.2.2 and consider graph Laplacian and covariance matrices of the form

$$L_0 := D_0^{-p}(D_0 - W_0)D_0^{-p} \quad \text{and} \quad C_{\tau,0} := \tau^{2\alpha}(L_0 + \tau^2 I_N)^{-\alpha}, \tag{3.10}$$

with D_0 denoting the diagonal degree matrix of W_0 and parameters $\tau, \alpha > 0$. Note that

$$L_0 = \operatorname{diag}(\widetilde{L}_1, \widetilde{L}_2, \cdots, \widetilde{L}_K),$$

and that $C_{\tau,0}$ inherits a similar block-diagonal structure. We use the covariance matrix $C_{\tau,0}$ to define prior measures μ_0 of the form (3.6). In order to show posterior contraction with such a prior, we also need to make some assumptions on the index set of labeled data Z' and the ground-truth matrix $U^\dagger$; these encode the idea that the labels are coherent with the geometric structure implied by the perfect clustering of the data.

Assumption 2. *At least one label is observed in each cluster $\widetilde{Z}_k$; that is,*

$$|Z' \cap \widetilde{Z}_k| > 0 \qquad \forall k = 1, \ldots, K.$$

Assumption 3. *Let* $(\mathbf{u}_m^\dagger)^T$ *for* $m = 1, \ldots, M$ *denote the columns of* $U^\dagger$. *Then* $\mathbf{u}_m^\dagger \in \operatorname{span}\{\bar{\boldsymbol{\chi}}_1, \ldots, \bar{\boldsymbol{\chi}}_K\}$, *where the weighted set functions are defined by*

$$\bar{\boldsymbol{\chi}}_k := \frac{D_0^p \mathbf{1}_k}{|D_0^p \mathbf{1}_k|}, \tag{3.11}$$

with $\mathbf{1}_k \in \mathbb{R}^N$ *denoting indicator of the cluster* $\widetilde{Z}_k$.

Remark 4. *We note here that our current exposition does not address posterior contraction when Assumption 3 is violated. While this is an interesting and practically pertinent question, we delay it for future study. We conjecture that as long as the ground-truth variable* $U^\dagger$ *is consistent with the observed labeling and the true underlying clustering structure of the unlabeled data* X, *then posterior contraction will occur around the projection of* $U^\dagger$ *onto* $\operatorname{span}\{\bar{\boldsymbol{\chi}}_1, \ldots, \bar{\boldsymbol{\chi}}_K\}$.

With the above assumptions in hand, we are ready to present our first posterior contraction result in the case of disconnected graphs.

Theorem 1. *Suppose that Assumptions 1, 2 and 3 are satisfied in turn by the disconnected graph* G_0, *the labeled set* Z' *and the ground-truth matrix* $U^\dagger$. *Consider the label model* (3.1), *the prior measure* $\mu_0(\mathrm{d}U) = \mathcal{N}(0, C_{\tau,0})$ *as in* (3.6), *and the resulting posterior measure* $\mu^Y(\mathrm{d}U)$ *as in* (3.8). *Then there is a constant* $\Xi > 0$ *so that, for every fixed* $\gamma, \tau, \alpha > 0$, *we have*

$$\mathcal{I}(\gamma, \alpha, \tau) \leq \Xi \max\left\{\gamma^2, \tau^{2\alpha}\right\} \left(1 + \max\left\{\gamma^2, \tau^{2\alpha}\right\} \|U^\dagger\|_F^2\right).$$

We prove this theorem in Section 3.3.1.1; here we discuss the intuition behind it. If $U \sim \mu_0$ as above then $\mathbf{u}_m \overset{iid}{\sim} \mathcal{N}(0, C_{\tau,0})$ where we recall $\mathbf{u}_m$ are the columns of U. Thus by the Karhunen-Loéve (KL) theorem,

$$\mathbf{u}_m \overset{d}{=} \sum_{j=1}^{N} \frac{1}{\sqrt{\lambda_{j,0}}} \xi_{mj} \boldsymbol{\phi}_{j,0},$$

with $\{(\lambda_{j,0}, \boldsymbol{\phi}_{j,0})\}_{j=1}^N$ denoting the eigenpairs of $C_{\tau,0}$ and $\xi_{mj} \overset{iid}{\sim} \mathcal{N}(0,1)$. The matrix L_0 has a K dimensional null-space spanned by the $\bar{\boldsymbol{\chi}}_k$ and this null-space is associated to the eigenvalue 1 for $C_{\tau,0}$. Furthermore, when τ^2 is small the remaining eigenvalues of $C_{\tau,0}$ are also small. These ideas are made rigorous in [61, Lemm. A.2 and Prop. A.3]. From those results it follows that

$$\mathbf{u}_m \overset{d}{=} \sum_{j=1}^{K} \xi_{mj} \bar{\boldsymbol{\chi}}_j + \mathcal{O}(\tau^{2\alpha}), \tag{3.12}$$

meaning that the prior is concentrated on $\operatorname{span}\{\bar{\boldsymbol{\chi}}_1, \ldots, \bar{\boldsymbol{\chi}}_K\}$. The posterior μ^Y also decouples along the columns $\mathbf{u}_m$ following Proposition 1 and so the SSR problem can be viewed as M separate BIPs for each column of $\mathbf{u}_m$, all with the same structure. In the limit of $\tau \to 0$, the prior mass concentrates on the K dimensional subspace spanned by the set-functions $\bar{\boldsymbol{\chi}}_k$. Since the posterior is absolutely continuous with respect to the prior, the posterior mass will also concentrate on the same subspace. The assumptions on the ground-truth $U^\dagger$ ensure that the data is consistent with the columns $\mathbf{u}_m$ lying in this subspace and give information on assignation of labels, corresponding to weights on the $\bar{\boldsymbol{\chi}}_m$. Hence, letting $\gamma \to 0$ yields concentration of the posterior around the ground-truth matrix $U^\dagger$ under Assumptions 2 and 3.

Remark 5. *Theorem 1 suggests that, in this perfectly clustered setting, choosing τ to achieve $\tau^{2\alpha} = \gamma^2$ is optimal, since it balances the two sources of error in the contraction rate. However, in the next section, we study the case that the unlabeled data is not perfectly clustered, where we measure the proximity of it to being perfectly clustered with the parameter ϵ. We state our theorems in a setting in which τ scales as a power of ϵ, rather than γ. We make this choice because τ and ϵ are linked intrinsically through the unsupervised learning task encapsulated in the prior measure, based on the unlabeled data, whilst γ enters separately through the likelihood, which captures the labeled data. In a broader picture, these considerations about the choice of τ suggest the importance of choosing this hyperparameter according to the data and the importance of using hierarchical Bayesian methods to learn such parameters.*

3.3.1.1 Proof of Theorem 1

We first bound the expectation with respect to the posterior distribution in (3.2), which is the mean square error of the estimator $U|Y$. We define the matrix C_0^* to be the posterior covariance obtained by substituting the prior covariance $C_{\tau,0}$ from (3.10) into (3.24), i.e.,

$$C_0^* := \left(C_{\tau,0}^{-1} + \frac{1}{\gamma^2}B\right)^{-1}. \tag{3.13}$$

For brevity we suppress the dependence of C_0^* on τ, α, and γ and we let $B := H^T H$. We then have

$$\mathbb{E}_{U|Y}\|U - U^\dagger\|_F^2 = \sum_{m=1}^M \mathbb{E}_{\mathbf{u}_m|\mathbf{y}_m}\left|\mathbf{u}_m - \mathbf{u}_m^\dagger\right|^2 = M\mathrm{Tr}(C_0^*) + \sum_{m=1}^M \left|\frac{1}{\gamma^2}C_0^* H^T \mathbf{y}_m - \mathbf{u}_m^\dagger\right|^2.$$

The first identity relies on the independence of the columns $\mathbf{u}_m$ of U under the posterior distribution, as established in Proposition 1. The second identity comes from the fact that the mean square error is the sum of the variance and squared bias of the estimator of each column.

We may now apply the outer expectation in definition of $\mathcal{I}$ with respect to the data $Y|U^\dagger$, and since $\mathrm{Tr}(C_0^*)$ does not depend on Y, we may pull it out of the outer expectation and write

$$\mathcal{I}(\gamma, \alpha, \tau) = M\mathrm{Tr}(C_0^*) + \mathbb{E}_{Y|U^\dagger}\left(\sum_{m=1}^M \left|\frac{1}{\gamma^2}C_0^* H^T \mathbf{y}_m - \mathbf{u}_m^\dagger\right|^2\right). \tag{3.14}$$

Since we assumed

$$\mathbf{y}_m|\mathbf{u}_m^\dagger \sim \mathcal{N}(H\mathbf{u}_m^\dagger, \gamma^2 I_J) \tag{3.15}$$

and the columns $\left\{\mathbf{y}_m^T\right\}_{m=1}^M$ are independent conditional on $U^\dagger$, we can write

$$\mathbb{E}_{Y|U^\dagger}\left|\frac{1}{\gamma^2}C_0^* H^T \mathbf{y}_m - \mathbf{u}_m^\dagger\right|^2 = \mathbb{E}_{\mathbf{y}_m|\mathbf{u}_m^\dagger}\left|\frac{1}{\gamma^2}C_0^* H^T \mathbf{y}_m - \mathbf{u}_m^\dagger\right|^2.$$

This expectation is the mean square error of the posterior mean estimator of $\mathbf{u}_m^\dagger$, which can be decomposed into a variance and a squared bias term:

$$\begin{aligned}\mathbb{E}_{\mathbf{y}_m|\mathbf{u}_m^\dagger}\left|\frac{1}{\gamma^2}C_0^* H^T \mathbf{y}_m - \mathbf{u}_m^\dagger\right|^2 = \mathrm{Tr}\left(\mathrm{Cov}\left(\frac{1}{\gamma^2}C_0^* H^T \mathbf{y}_m\right)\right) + \\ \left|\mathbb{E}_{\mathbf{y}_m|\mathbf{u}_m^\dagger}\left(\frac{1}{\gamma^2}C_0^* H^T \mathbf{y}_m\right) - \mathbf{u}_m^\dagger\right|^2,\end{aligned}$$

where $\mathrm{Cov}(\cdot)$ denotes the covariance matrix of a random vector. We compute the variance term using (3.15):

$$\mathrm{Cov}\left(\frac{1}{\gamma^2}C_0^* H^T \mathbf{y}_m\right) = \frac{1}{\gamma^2}C_0^* H^T \mathrm{Cov}\left(\mathbf{y}_m\right)\frac{1}{\gamma^2}H(C_0^*)^T = \frac{1}{\gamma^2}C_0^* B C_0^*,$$

where we used the fact that $\mathrm{Cov}(\mathbf{y}_m) = \gamma^2 I_J$ and $B = H^T H \in \mathbb{R}^{N\times N}$. For the bias term, we can write

$$\mathbb{E}_{\mathbf{y}_m|\mathbf{u}_m^\dagger}\left(\frac{1}{\gamma^2}C_0^* H^T \mathbf{y}_m\right) = \frac{1}{\gamma^2}C_0^* H^T H \mathbf{u}_m^\dagger = \frac{1}{\gamma^2}C_0^* B \mathbf{u}_m^\dagger.$$

Putting these terms together yields

$$\mathbb{E}_{\mathbf{y}_m|\mathbf{u}_m^\dagger}\left|\frac{1}{\gamma^2}C_0^* H^T \mathbf{y}_m - \mathbf{u}_m^\dagger\right|^2 = \frac{1}{\gamma^2}\mathrm{Tr}\left(C_0^* B C_0^*\right) + \left|\frac{1}{\gamma^2}C_0^* B \mathbf{u}_m^\dagger - \mathbf{u}_m^\dagger\right|^2.$$

Substituting this identity back into (3.14) yields

$$\mathcal{I}(\gamma,\alpha,\tau) = M\mathrm{Tr}(C_0^*) + \frac{M}{\gamma^2}\mathrm{Tr}(C_0^* B C_0^*) + \sum_{m=1}^{M}\left|\frac{1}{\gamma^2}C_0^* B \mathbf{u}_m^\dagger - \mathbf{u}_m^\dagger\right|^2. \tag{3.16}$$

The desired bound now follows from Lemmata 1, 2, and 3 that in turn bound the first, second, and third term in the right hand side of (3.16). These Lemmata are proved in Section 3.5.2.

Lemma 1. *Suppose Assumptions 1 and 2 are satisfied by the disconnected graph G_0 and the labeled set Z', respectively. Then there exists a constant $\Xi > 0$, such that for any $\gamma, \tau, \alpha > 0$, we have*

$$\mathrm{Tr}(C_0^*) \leq \Xi \max\{\gamma^2, \tau^{2\alpha}\}, \tag{3.17}$$

where C_0^* is the posterior covariance matrix in (3.13).

Lemma 2. *Suppose Lemma 1 is satisfied. Then for any $\gamma, \tau, \alpha > 0$, we have*

$$\frac{1}{\gamma^2}\mathrm{Tr}(C_0^* B C_0^*) \leq \Xi \max\left\{\gamma^2, \tau^{2\alpha}\right\},$$

with the same constant $\Xi > 0$ as in (3.17).

Lemma 3. *Suppose Assumptions 1, 2, and 3 are in turn satisfied by the disconnected graph G_0, the labeled set Z', and the ground-truth function $U^\dagger$. Then for any $\gamma, \tau, \alpha > 0$ and $m = 1, \dots, M$, we have*

$$\left|\frac{1}{\gamma^2}C_0^* B \mathbf{u}_m^\dagger - \mathbf{u}_m^\dagger\right| \leq \Xi \max\{\gamma^2, \tau^{2\alpha}\},$$

where $\Xi > 0$ is the same constant as in (3.17).

3.3.2 Weakly Connected Graph

We now consider a generalization of the setting in the previous subsection, in which the disconnected graph $G_0 = \{Z, W_0\}$ is perturbed, and the perturbation results in a connected graph $G_\epsilon = \{Z, W_\epsilon\}$. Following [61] we summarize the following set of assumptions on this perturbed graph G_ϵ.

Assumption 4. *The graph $G_\epsilon = \{Z, W_\epsilon\}$ satisfies the following three conditions.*

(a) The weight matrix W_ϵ can be expanded in the form

$$W_\epsilon = W_0 + \sum_{h=1}^{\infty} \epsilon^h W^{(h)}, \tag{3.18}$$

where W_0 is the weight matrix of a disconnected graph G_0.

(b) The matrices $W^{(h)}$ are self-adjoint and $\{\|W^{(h)}\|_2\}_{h=1}^{\infty} \in \ell^\infty$.

(c) Let $w_{ij}^{(0)}$ and $w_{ij}^{(h)}$ denote the entries of W_0 and $W^{(h)}$ respectively. Then, for $h \geq 1$, we assume

$$\begin{cases} w_{ij}^{(h)} \geq 0, \quad \textit{if} \quad w_{ij}^{(0)} = 0 \quad \textit{for} \quad i, j \in Z, i \neq j \\ w_{ii}^{(h)} = 0. \end{cases} \tag{3.19}$$

The assumptions (b) and (c) above ensure that W_ϵ is a well-defined adjacency matrix. Also note that (c) allows for $w_{ij}^{(h)}$, $h \geq 1$, to be negative whenever $w_{ij}^{(0)} > 0$. With the above assumptions identified we can proceed analogously to Section 3.2.2 to define Laplacian and covariance matrices

$$L_\epsilon := D_\epsilon^{-p}(D_\epsilon - W_\epsilon)D_\epsilon^{-p} \quad \text{and} \quad C_{\tau,\epsilon} := \tau^{2\alpha}(L_\epsilon + \tau^2 I_N)^{-\alpha}, \tag{3.20}$$

with D_ϵ denoting the degree matrix of W_ϵ and parameters $\tau, \alpha > 0$. We then use the covariance matrix $C_{\tau,\epsilon}$ to define a prior measure μ_0 of the form (3.6) on the weakly connected graph G_ϵ. With the assumptions made about the disconnected set-up in Subsection 3.3.1 and the above new assumptions on the weakly connected set-up, we can now present our main posterior contraction result, the analogue of Theorem 1, for weakly connected graphs G_ϵ.

Theorem 2. *Suppose Assumptions 1, 2, 3 and 4 are satisfied by the disconnected graph G_0, the labeled set Z', the ground-truth matrix $U^\dagger$ and the weakly connected graph G_ϵ. Fix $\alpha > 0$. Then there exist constants $\epsilon_0 \in (0,1)$ and $\Xi, \Xi_1 > 0$ such that for any sequence $(\epsilon, \tau, \gamma) \to 0$, the following holds uniformly:*

$$\mathcal{I}(\gamma, \alpha, \tau, \epsilon) \leq \Xi \max\left\{\gamma^2, \left(\frac{\tau^2}{1 - \Xi_1 \epsilon/\tau^2}\right)^\alpha\right\} \times \left(1 + \max\left\{\gamma^2, \left(\frac{\tau^2}{1 - \Xi_1 \epsilon/\tau^2}\right)^\alpha\right\} \left[\epsilon + \frac{\epsilon}{\tau^{2\alpha}} + \left(1 + \frac{\epsilon}{\tau^2}\right)^\alpha\right]^2 \|U^\dagger\|^2\right).$$

The intuition behind the proof is that we use the same ideas which underlie Theorem 1, which concerns the case $\epsilon = 0$, coupled with new arguments which control perturbations to the spectrum of $C_{\tau,\epsilon}$ with respect to that of $C_{\tau,0}$. Specifically $C_{\tau,\epsilon}$ now has a one-dimensional null-space associated with the eigenvalue 1, but has an additional $K-1$ eigenvalues of size $1 - \mathcal{O}(\epsilon/\tau^2)$. The remaining eigenvalues are small, of $\mathcal{O}(\tau^{2\alpha})$, if an appropriate relationship between ϵ and τ is imposed. The eigenfunctions associated with the K eigenvalues at, or near, 1, nearly span the same space as the N weighted set-functions $\{\bar{\boldsymbol{\chi}}_k\}_{k=1}^K$. Let $(\mathbf{u}_m)^T$ denote the columns of U that is drawn from the prior distriubtion μ_0. Then it follows from [61, A.10] that these columns concentrate on the span of the $\bar{\boldsymbol{\chi}}_k$ with errors of the form $\mathcal{O}\left(\epsilon^2\tau^{-4} + \tau^{4\alpha} + \epsilon^2\right)$ when $\epsilon = o(\tau^2)$ and of the form $\mathcal{O}\left(\tau^{4\alpha} + \epsilon^2\right)$ when $\epsilon = \Theta(\tau^2)$. These approximation results for the columns $\mathbf{u}_m$ under the prior underlie the proof. The rest of the proof follows in the footsteps of Theorem 1. First, we decouple the posterior on the columns of U using Proposition 3.23 to obtain M independent BIPs. In each BIP, the prior concentration on the span of $\bar{\boldsymbol{\chi}}_k$ results in posterior concentration along the same subspace, at which point, the noise standard deviation γ in the likelihood potential Φ controls the contraction of the posterior around the ground-truth matrix $U^\dagger$ under Assumptions 2 and 3.

3.3.2.1 Proof of Theorem 2

Let us define the perturbed posterior covariance matrix

$$C_\epsilon^* := \left(C_{\tau,\epsilon}^{-1} + \frac{1}{\gamma^2} B\right)^{-1}, \tag{3.21}$$

following (3.20) with the prior covariance matrix $C_{\tau,\epsilon}$. Observe that the arguments leading up to the upper bound (3.3.1.1) hold with C_0^* replaced with C_ϵ^*. Thus we immediately obtain the identity

$$\mathcal{I}(\gamma,\alpha,\tau,\epsilon) = M\mathrm{Tr}(C_\epsilon^*) + \frac{M}{\gamma^2}\mathrm{Tr}(C_\epsilon^* B C_\epsilon^*) + \sum_{m=1}^{M}\left|\frac{1}{\gamma^2}C_\epsilon^* B\mathbf{u}_m^\dagger - \mathbf{u}_m^\dagger\right|^2. \tag{3.22}$$

Similarly to Section 3.3.1.1, we prove Theorem 2 by bounding each term in the right-hand side of (3.22) in the Lemmata 4, 5, and 6 below. The proofs are collected in Section 3.5.3.

Lemma 4. *Suppose Assumptions 1, 2, and 4 are satisfied by the disconnected graph G_0, the labeled set Z', and the weakly connected graph G_ϵ. Fix $\alpha > 0$. Then there exist constants $\epsilon_0 \in (0,1)$ and $\Xi_0, \Xi_1 > 0$ such that along any sequence $(\epsilon,\tau,\gamma) \to 0$, the following holds uniformly:*

$$\mathrm{Tr}(C_\epsilon^*) \le \Xi_0 \max\left\{\gamma^2, \left(\frac{\tau^2}{1-\Xi_1\epsilon/\tau^2}\right)^\alpha\right\},$$

with C_ϵ^* as in (3.21).

Lemma 5. *Suppose that the conditions of Lemma 4 are satisfied and fix $\alpha > 0$. Then there exist constants $\epsilon_0 \in (0,1)$ and $\Xi_0, \Xi_1 > 0$ such that for any sequence $(\epsilon,\tau,\gamma) \to 0$, the following holds uniformly:*

$$\frac{1}{\gamma^2}\mathrm{Tr}(C_\epsilon^* B C_\epsilon^*) \le \Xi_0 \max\left\{\gamma^2, \left(\frac{\tau^2}{1-\Xi_1\epsilon/\tau^2}\right)^\alpha\right\},$$

where $\epsilon_0 \in (0,1)$ and $\Xi_0, \Xi_1 > 0$ are the same constants as in Lemma 4.

Lemma 6. *Suppose Assumptions 1, 2, 3, and 4 are satisfied by the disconnected graph G_0, the labeled set Z', the ground-truth matrix $U^\dagger$ and the weakly connected graph G_ϵ respectively and fix $\alpha > 0$. Then there exist constants $\epsilon_0 \in (0,1)$ and $\Xi_1, \Xi_2 > 0$ such that for any sequence $(\epsilon,\tau,\gamma) \to 0$, the following holds uniformly:*

$$\left|\frac{1}{\gamma^2}C_\epsilon^* B\mathbf{u}_m^\dagger - \mathbf{u}_m^\dagger\right| \le \Xi_2 \max\left\{\gamma^2, \left(\frac{\tau^2}{1-\Xi_1\epsilon/\tau^2}\right)^\alpha\right\}\left[\epsilon + \frac{\epsilon}{\tau^{2\alpha}} + \left(1+\frac{\epsilon}{\tau^2}\right)^\alpha\right]|\mathbf{u}_m^\dagger|.$$

3.3.3 Corollary

We now present a corollary of Theorem 2 that is the precisely stated version of our informal main theorem from Section 3.1.

Corollary 1. *Suppose that the conditions of Theorem 2 are satisfied and that for a fixed $\alpha > 0$, the hyperparameters (ϵ, τ) are chosen to satisfy*

$$2\Xi_1 \epsilon = \tau^{\max\{2,2\alpha\}}.$$

Then there exists $\Xi_2 > 0$ depending on α and the constants Ξ, Ξ_1 from Theorem 2 but independent of ϵ and γ, so that

$$\mathcal{I} \leq \Xi_2 \max\left\{\gamma^2, \epsilon^{\min\{1,\alpha\}}\right\}.$$

Remark 6. *The reader is encouraged to study the discussion following the informal main theorem for an interpretation of this result in terms of asymptotic consistency. We also note that an application of Markov's inequality can immediately extend the bound in Corollary 1 to a bound on the expected probabilities of posterior samples being found far from the ground-truth $U^\dagger$. More precisely,for any $\delta > 0$ we have that*

$$\mathbb{E}_{Y|U^\dagger}\left\{\mu^Y\left(\left\|U - U^\dagger\right\|_F > \delta\right)\right\} \leq \frac{\mathcal{I}}{\delta^2}.$$

3.4 Numerical Experiments

In this section, we provide numerical experiments that elucidate our main theoretical results and in particular examine the convergence rate of the contraction functional $\mathcal{I}$ with respect to both the ϵ and γ parameters. We use a synthetic example in Subsection 3.4.1 as well as the MNIST database of handwritten digits [84] in Subsection 3.4.2. In both examples, we compute $\mathcal{I}$ via the decomposition given in (3.16), which provides us with an explicit formula to numerically compute the contraction measure. We then vary ϵ and γ parameters while choosing $\tau = \epsilon^{1/\max\{2,2\alpha\}}$. We numerically differentiate $\log(\mathcal{I})$ with respect to $\log(\epsilon)$ and $\log(\gamma)$ to estimate the rate of convergence with respect to these two parameters. A surface plot of these derivatives is then presented in Figures 3.2 and 3.3, for the two respective data sets, in which the color encodes the estimated rate of convergence in terms of the respective variables. The dark blue colors in these plots indicate a rate of convergence of $\mathcal{I}$ that is close to zero, while bright yellow colors indicate larger convergence rates of $\mathcal{I}$. Further numerical results are presented in Subsections 3.6.1 and 3.6.2, taking a closer look at the rates of convergence of different bias and variance terms that contribute to $\mathcal{I}$.

3.4.1 Synthetic Data

We construct a synthetic weakly connected graph consisting of three clusters of 100 nodes each, where each cluster represents a different class. We obtain the weight matrix W_ϵ following (3.18); we truncate the expansion at the ϵ^3 level. Each entry of weight matrices W_0 and $W^{(h)}$, $h = 1, 2, 3$ are drawn independently from a uniform distribution on $[0, 1]$. The matrices W_0 and $W^{(h)}$, $h = 1, 2, 3$ are fixed once sampled and are used to construct W_ϵ for different ϵ values. Each W_ϵ is then symmetrized via the transformation $W_\epsilon \mapsto (W_\epsilon + W_\epsilon^T)/2$. We pick one node from each cluster to be labeled and choose ground-truth $U^\dagger = [\bar{\boldsymbol{\chi}}_1, \bar{\boldsymbol{\chi}}_2, \bar{\boldsymbol{\chi}}_3]^T$. We vary ϵ values from 10^{-1} to 10^{-15} and γ ranging from 10^{-1} to $10^{-7.5}$; τ is taken to be $\epsilon^{1/\max\{2,2\alpha\}}$.

In Figure 3.1, we demonstrate the convergence of $\mathcal{I}$ in the limit of the noise standard deviation γ going to zero, for different values of α and ϵ. We see posterior contraction with respect to γ until a floor is reached; this floor depends on $\epsilon$and is smaller for smaller ϵ.

In Figure 3.2, we study this phenomenon in more detail. Let us define

$$c_\epsilon := \partial \log(\mathcal{I})/\partial \log(\epsilon) \geq 0 \text{ and } c_\gamma := \partial \log(\mathcal{I})/\partial \log(\gamma) \geq 0,$$

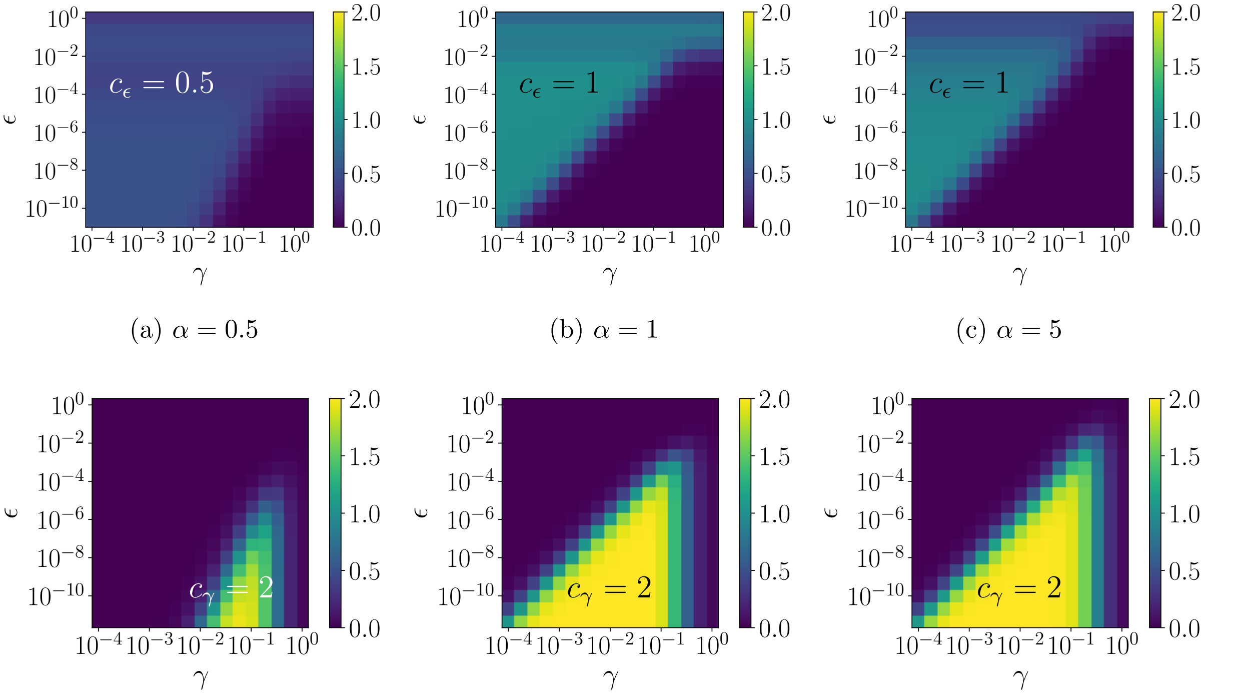
which correspond to contraction rates of $\mathcal{O}(\epsilon^{c_\epsilon})$ and $\mathcal{O}(\gamma^{c_\gamma})$ respectively. We present surface plots in Figure 3.2 of c_ϵ (top row) and c_γ (bottom row) as functions of ϵ, γ for various values of α. Darker (lighter) regions correspond to smaller (larger) values of the logarithmic slopes c_ϵ, c_γ. In regions with lighter values (i.e. $c_\epsilon, c_\gamma > 0$), we observe posterior contraction because the logarithmic slopes are nonzero. The darker regions correspond to instances where the contraction has ceased as indicated by the logarithmic slopes being zero. This is the phenomenon that is displayed in Figure 3.1, where the value of $\mathcal{I}$ reduces with respect to γ up to the point where the errors saturate at an ϵ-dependent value as predicted by the bounds in Theorem 2.

In the bottom row of Figure 3.2, horizontal "slices" of the plot correspond to a fixed value of ϵ which is how Figure 3.1 can be obtained. Going from right to left, we observe that the contraction rate is on the order of γ^2, until the point that $\gamma^2 \approx \epsilon^{\min\{1,\alpha\}}$ when our theory predicts that the $\mathcal{I}$ will saturate and contraction has stopped, i.e., $c = 0$. These plots illustrate the sharpness of our theoretical bounds of Theorem 2 for the posterior contraction measure $\mathcal{I}$. Similar results, with the roles of ϵ and γ swapped, are seen in the top row of Figure 3.2.

3.4.2 MNIST Data

In this subsection, we use the MNIST data set [84] to test our theory on an empirical data set. MNIST is a data set of 70,000 grayscale 28×28 pixel images of handwritten digits (0–9), of which we use only the digits 1, 4, and 7. Each image is represented by a vector $\mathbf{x}_i \in \mathbb{R}^{784}$ and we normalize the pixel values to range from 0 to 1. To confirm our theory in practice presents the issue of determining how to control the parameter ϵ that is inherent to the clustering structure of a given fixed unlabeled data set X given in application. However, in this example, we may use the fact that every image is labeled and so the clustering structure of the data set is known. Using this we may devise an ϵ-dependent parameter set to observe what happens in the $\epsilon \to 0$ limit.

First, we create a similarity graph G based on the unlabeled data X of reshaped images $\mathbf{x}_i \in \mathbb{R}^{784}$. Given the known clustering (i.e. class memberships) of the points in the MNIST data set, we can identify the inter-cluster edges, those edges that connect nodes of different clusters corresponding to different digits. If the original weight matrix is given by W, with



(a) $\alpha = 0.5$

(b) $\alpha = 1$

(c) $\alpha = 5$

(d) $\alpha = 0.5$

(e) $\alpha = 1$

(f) $\alpha = 5$

Figure 3.2: A numerical demonstration of Theorem 2 on a synthetic data set. The top panels showcase numerical estimates of $c_\epsilon = \frac{\partial \log(\mathcal{I})}{\partial \log(\epsilon)}$ for different α values and the bottom panels showcase the numerical estimates of $c_\gamma = \frac{\partial \log(\mathcal{I})}{\partial \log(\gamma)}$. In the dark blue regions, $c_\epsilon, c_\gamma \approx 0$, indicating that $\mathcal{I}$ stays approximately flat with respect to the respective variable ϵ or γ and so contraction has approximately ceased; the slope of the brighter regions is annoated in each panel and implies posterior contraction. The transition between the dark and bright regions occurs approximately at $\epsilon = \gamma^{2/\min\{1,\alpha\}}$.

entries w_{ij}, then we scale the inter-cluster edges by ϵ to obtain W_ϵ as:

$$[W_\epsilon]_{ij} = \begin{cases} w_{ij}, & \text{if } i, j \in \tilde{Z}_k \\ \epsilon w_{ij}, & \text{if } i \in \tilde{Z}_k, j \in \tilde{Z}_\ell, \text{ with } k \neq \ell. \end{cases}$$

Sending $\epsilon \to 0$ then results in a disconnected graph, where each cluster represents a different digit.

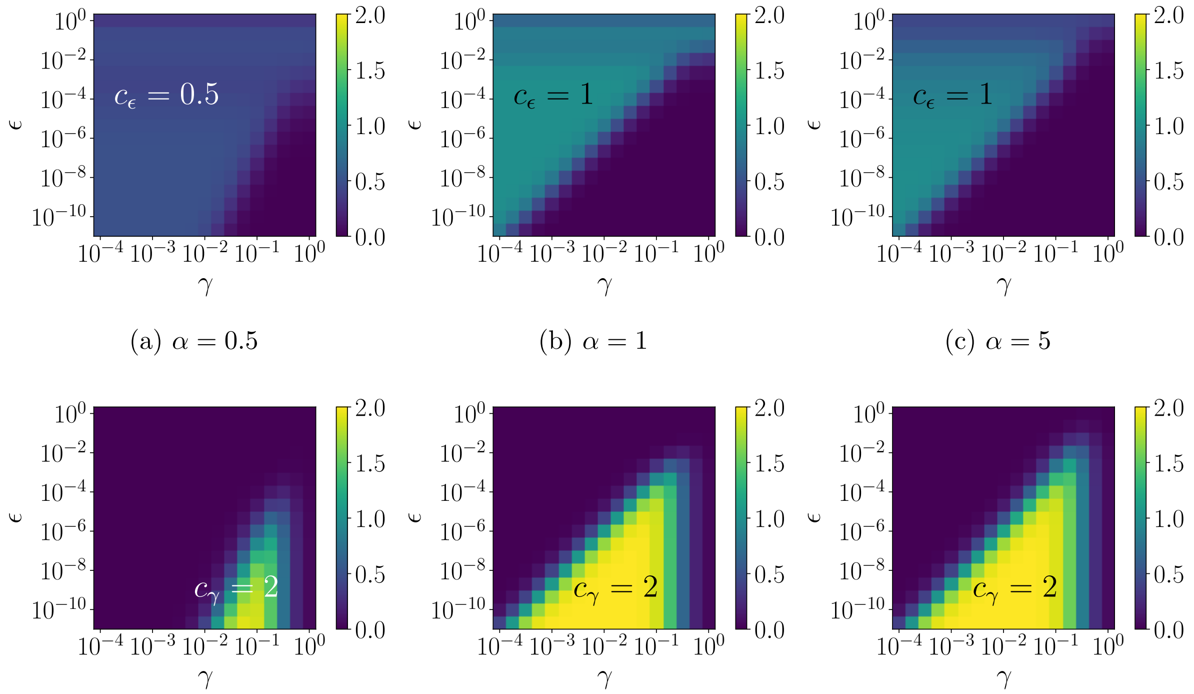
For our experiment, we sample 100 images uniformly at random from the digits 1, 4, and 7. The similarity graph $W = (w_{ij})$ is constructed via the Gaussian kernel and the Zelnik-Perona scaling [169], $w_{ij} = \exp(-|\mathbf{x}_i - \mathbf{x}_j|^2/r_i r_j)$, where r_i is the Euclidean distance between data point i and its 15th nearest neighbor. Following the same procedure as the synthetic data, we pick one node from each digit to be labeled and choose the ground-truth $U^\dagger = [\bar{\boldsymbol{\chi}}_1, \bar{\boldsymbol{\chi}}_2, \bar{\boldsymbol{\chi}}_3]^T$. We evaluate the contraction measurement $\mathcal{I}$ for a range of ϵ and γ. We present the results in Figure 3.3. It is clear that Figure 3.3 is nearly identical to Figure 3.2, demonstrating that the behavior on this MNIST data set is close to that observed in the synthetic case; the two sets of experiments together attest to the sharpness of our contraction rate estimates in Theorem 2.

3.5 Proof of Lemmata

In this section, we start by discussing useful properties of the posterior measure in Subsection 3.5.1; in particular, we show that the posterior is Gaussian and give closed-form expressions for its mean and covariance. In Subsections 3.5.2 and 3.5.3, we present detailed proofs of the lemmata used to prove our main results, Theorems 1 and 2. Numerical experiments which illustrate these lemmata are contained in Subsections 3.6.1 and 3.6.2.

3.5.1 Characterizing the Posterior

Here we collect some results that completely characterize the posterior measure μ^Y as a Gaussian measure with explicit formulae for its mean and covariance.



(a) $\alpha = 0.5$

(b) $\alpha = 1$

(c) $\alpha = 5$

(d) $\alpha = 0.5$

(e) $\alpha = 1$

(f) $\alpha = 5$

Figure 3.3: A numerical demonstration of Theorem 2 on the MNIST data set with digits 1, 4, and 7. The top panels showcase numerical estimates of $c_\epsilon = \frac{\partial \log(\mathcal{I})}{\partial \log(\epsilon)}$ for different α values and the bottom panels showcase the numerical estimates of $c_\gamma = \frac{\partial \log(\mathcal{I})}{\partial \log(\gamma)}$. In the dark blue regions, $c_\epsilon, c_\gamma \approx 0$, indicating that $\mathcal{I}$ stays approximately flat with respect to the respective variable ϵ or γ and so contraction has approximately ceased; the slope of the brighter regions is annotated in each panel and implies posterior contraction. The transition between the dark and bright regions occurs approximately at $\epsilon = \gamma^{2/\min\{1,\alpha\}}$. These results are similar to our synthetic experiment depicted in Figure 3.2.

Proposition 1. *Consider the posterior measure* μ^Y *given by* (3.8). *Then*

(i) $\mu^Y = \mathcal{N}(U^*, I_M \otimes C^*)$ *and has Lebesgue density*

$$\begin{aligned}\mu^Y(\mathrm{d}U) &= \frac{1}{\vartheta(Y)} \exp\left(-\frac{1}{2}\left\langle (U-U^*)^T, (C^*)^{-1}(U-U^*)^T\right\rangle_F\right)\mathrm{d}U \\ &\equiv \frac{1}{\vartheta(Y)} \prod_{m=1}^{M} \exp\left(-\frac{1}{2}\langle(\mathbf{u}_m - \mathbf{u}_m^*), (C^*)^{-1}(\mathbf{u}_m - \mathbf{u}_m^*)\rangle\right)\mathrm{d}\mathbf{u}_\ell.\end{aligned} \tag{3.23}$$

Here U^* *is the posterior mean with columns* $(\mathbf{u}_m^*)^T$ *and* C^* *is the covariance matrix of each row* $(\mathbf{u}_m^*)^T$.

(ii) The posterior means $\mathbf{u}_m^*$ *and covariances* C^* *are given by*

$$\mathbf{u}_m^* = \frac{1}{\gamma^2} C^* H^T \mathbf{y}_m, \qquad C^* = \left(C_\tau^{-1} + \frac{1}{\gamma^2}B\right)^{-1}, \tag{3.24}$$

where $B = H^T H$ *and* $\mathbf{y}_m^T$ *are the columns of* Y.

(iii) The columns $\mathbf{u}_m$ *of* $U \sim \mu^Y$ *are i.i.d. according to the Gaussian distribution* $\mathcal{N}(\mathbf{u}_\ell^*, C^*)$.

Proof. To show (i), we begin by expressing the likelihood in terms of the columns of U and Y; we get

$$\exp\left(-\Phi(U;Y)\right) = \exp\left(-\frac{1}{2\gamma^2}\left\|HU^T - Y^T\right\|_F^2\right) = \exp\left(-\frac{1}{2\gamma^2}\sum_{m=1}^{M}\left|H\mathbf{u}_m - \mathbf{y}_m\right|^2\right).$$

Combining the previous identity with (3.6), we can express the Lebesgue density of the posterior as

$$\begin{aligned}\mu^Y(\mathrm{d}U) &\propto \exp\left[-\frac{1}{2}\sum_{m=1}^{M}\left\langle \mathbf{u}_m, C_\tau^{-1}\mathbf{u}_m\right\rangle + \frac{1}{\gamma^2}\left|H\mathbf{u}_m - \mathbf{y}_m\right|^2\right] \\ &= \exp\left[-\frac{1}{2}\sum_{m=1}^{M}\left\langle \mathbf{u}_m, C_\tau^{-1}\mathbf{u}_m\right\rangle + \frac{1}{\gamma^2}\left(\langle \mathbf{u}_m, B\mathbf{u}_m\rangle - 2\langle \mathbf{u}_m, H^T\mathbf{y}_m\rangle + |\mathbf{y}_m|^2\right)\right] \\ &\propto \exp\left[-\frac{1}{2}\sum_{m=1}^{M}\left\langle \mathbf{u}_m, (C^*)^{-1}\mathbf{u}_m\right\rangle - 2\left\langle \mathbf{u}_m, \frac{1}{\gamma^2}H^T\mathbf{y}_m\right\rangle + \left\langle \mathbf{u}_m^*, (C^*)^{-1}\mathbf{u}_m^*\right\rangle\right]\end{aligned}$$

$$= \exp\left[-\frac{1}{2}\sum_{m=1}^{M}\left\langle \mathbf{u}_m, (C^*)^{-1}\mathbf{u}_m\right\rangle - 2\left\langle \mathbf{u}_m, (C^*)^{-1}\mathbf{u}_m^*\right\rangle + \left\langle \mathbf{u}_m^*, (C^*)^{-1}\mathbf{u}_m^*\right\rangle\right]$$

$$= \exp\left[-\frac{1}{2}\sum_{m=1}^{M}\left\langle \mathbf{u}_m - \mathbf{u}_m^*, (C^*)^{-1}(\mathbf{u}_m - \mathbf{u}_m^*)\right\rangle\right]$$

$$= \exp\left[-\frac{1}{2}\left\langle (U - U^*)^T, (C^*)^{-1}(U - U^*)^T\right\rangle_F\right],$$

with $\mathbf{u}_m^*$, and C^* as in (3.24). Assertion (ii) follows from (3.23) and the observation that the negative log posterior is a sum of identical positive-definite quadratic forms in each $\mathbf{u}_m$, from which the expressions for mean and variance of $\mathbf{u}_m$ may be inferred. Assertion (iii) is a consequence of the fact that uncorrelated Gaussian random variables are also independent.

□

3.5.2 Proofs of Lemmata 1–3

3.5.2.1 Proof of Lemma 1

Proof. Let $P_0 \in \mathbb{R}^{N\times N}$ denote the projection matrix onto $\operatorname{span}\{\bar{\boldsymbol{\chi}}_k\}_{k=1}^K$ (recall (3.11)) and define

$$\beta = \sqrt{\frac{K}{K+\zeta^2/4}}, \qquad \zeta := \min_{k\leq K}\min_{i\in Z_k}|\bar{\boldsymbol{\chi}}_k(i)|. \tag{3.25}$$

Our method of proof is to obtain lower bounds on the Dirichlet energy $\langle \mathbf{v}, (C_0^*)^{-1}\mathbf{v}\rangle$ for unit vectors $\mathbf{v} \in \mathbb{R}^N$ by considering two cases where $|P_0\mathbf{v}|\geq \beta$ of $|P_0\mathbf{v}|< \beta$. This translates to a lower bound on the smallest eigenvalue of $(C_0^*)^{-1}$. Since $\operatorname{Tr}(C_0^*) = \sum_{j=1}^N \lambda_{j,0}$, with $\lambda_{j,0}$ denoting the strictly positive eigenvalues of C_0^*, the lower bound on the Dirichlet energy of $(C_0^*)^{-1}$ translates to an upper bound on $\operatorname{Tr}(C_0^*)$.

Case 1 ($|P_0\mathbf{v}|\geq \beta$): Since $\mathbf{v}$ is a unit vector, we have that $\|(I-P_0)\mathbf{v}\|_\infty\leq |(I-P_0)\mathbf{v}|\leq \sqrt{1-\beta^2}$. The matrix $C_{\tau,0}$ and its inverse are positive definite, so

$$\left\langle \mathbf{v}, (C_0^*)^{-1}\mathbf{v}\right\rangle = \left\langle \mathbf{v}, \left(\frac{1}{\gamma^2}B\mathbf{v} + C_{\tau,0}^{-1}\right)\mathbf{v}\right\rangle \geq \left\langle \mathbf{v}, \frac{1}{\gamma^2}B\mathbf{v}\right\rangle = \frac{1}{\gamma^2}\sum_{i\in Z'} v_i^2, \tag{3.26}$$

where we used v_i to denote the entries of $\mathbf{v}$. Let us write $P_0\mathbf{v} = \sum_{k=1}^K c_k \bar{\boldsymbol{\chi}}_k$ with $c_k := \langle \mathbf{v}, \bar{\boldsymbol{\chi}}_k\rangle$ denoting the basis coefficients of $\mathbf{v}$ in span of $\{\bar{\boldsymbol{\chi}}_k\}_{k=1}^K$ and define

$$\mathfrak{k} := \arg\max_k |c_k|,$$

the index of the absolutely maximal coefficient amongst the c_k. The assumption $|P_0\mathbf{v}|\geq \beta$ implies $\sum_{k=1}^K c_k^2 \geq \beta^2$. It then follows that

$$K \max_{k\leq K} c_k^2 \geq \sum_{k=1}^K c_k^2 \geq \beta^2,$$

so $|c_{\mathfrak{k}}|= \max_{k\leq K}|c_k|\geq \beta/\sqrt{K}$. Since each $\bar{\boldsymbol{\chi}}_k$ is supported on $\widetilde{Z}_k$ on which it takes values that are at least ζ, we have

$$|(P_0\mathbf{v})_i|= |c_{\mathfrak{k}}|(\bar{\boldsymbol{\chi}}_{\mathfrak{k}})_i \geq \frac{\beta\zeta}{\sqrt{K}} \qquad \text{for} \qquad i \in \widetilde{Z}_{\mathfrak{k}},$$

where we used $(P_0\mathbf{v})_i$ to denote the i-th entry of the vector $P_0\mathbf{v}$. It then follows that for $i \in \widetilde{Z}_{\mathfrak{k}}$, we have

$$\begin{aligned}|v_i| &= |(P_0\mathbf{v})_i + ((I-P_0)\mathbf{v})_i|\geq \max\{0, |(P_0\mathbf{v})_i|-\|(I-P_0)\mathbf{v}\|_\infty\}\\ &\geq \max\left\{0, \frac{\beta\zeta}{\sqrt{K}} - \sqrt{1-\beta^2}\right\}.\end{aligned}$$

Substituting the value of β from (3.25), we obtain $|v_i|\geq (4K/\zeta^2+1)^{-1/2}$. Following Assumption 2, i.e. $\widetilde{Z}_k' \neq \emptyset$ for all k, we have

$$\frac{1}{\gamma^2}\sum_{j\in Z'} v_j^2 \geq \frac{1}{\gamma^2}|v_i|^2 \geq \gamma^{-2}\left(4K/\zeta^2+1\right)^{-1} \qquad \text{for some index } i \in \widetilde{Z}_{\mathfrak{k}}'.$$

Putting this lower bound together with (3.26) we conclude that for any $\mathbf{v}$ such that $|P_0\mathbf{v}|\geq \beta$, we have

$$\langle \mathbf{v}, (C_0^*)^{-1}\mathbf{v}\rangle \geq \gamma^{-2}\left(4K/\zeta^2+1\right)^{-1}.$$

Case 2 ($|P_0\mathbf{v}|< \beta$): We naturally have $|(I-P_0)\mathbf{v}|\geq \sqrt{1-\beta^2}$. Let $\{(\sigma_{k,0}, \boldsymbol{\phi}_{k,0})\}_{k=1}^N$ denote the eigenpairs of L_0, indexed by order of increasing eigenvalues. Recall from Subsection 3.3.1 that $\sigma_{k,0} = 0$ for $k = 1, 2, \dots, K$ and $\{\boldsymbol{\phi}_{k,0}\}_{k=1}^K \subset \operatorname{span}\{\bar{\boldsymbol{\chi}}_k\}_{k=1}^K$. Moreover, the orthonormal eigenvectors $\{\boldsymbol{\phi}_{k,0}\}_{k=1}^N$ are also eigenvectors of $C_{\tau,0}^{-1}$. With some abuse of notation, we define

$c_k := \langle \mathbf{v}, \boldsymbol{\phi}_{k,0} \rangle$ for $k = K+1, \dots, N$ and write $(I - P_0)\mathbf{v} = \sum_{k=K+1}^{N} c_k \boldsymbol{\phi}_{k,0}$. In light of this identity, we compute

$$\begin{aligned}\langle \mathbf{v}, (C_0^*)^{-1}\mathbf{v} \rangle = \left\langle \mathbf{x}, \left(\frac{1}{\gamma^2} B + C_{\tau,0}^{-1} \right) \mathbf{v} \right\rangle &\geq \langle \mathbf{v}, C_{\tau,0}^{-1} \mathbf{v} \rangle \\ &= \sum_{k=1}^{K} c_k^2 + \sum_{k=K+1}^{N} c_k^2 \tau^{-2\alpha} (\sigma_{k,0} + \tau^2)^{\alpha} \geq \sum_{k=K+1}^{N} c_k^2 \tau^{-2\alpha} (\sigma_{k,0} + \tau^2)^{\alpha}. \quad (3.27)\end{aligned}$$

For the first inequality, we have used the fact that B is positive semi-definite. From Assumption 1(b), it follows that $\sigma_{k,0} \geq \theta$ for $k \geq K$ and subsequently $\sigma_{k,0} + \tau^2 \geq \theta$ for $k \geq K$. With this observation and using the expression for β in (3.25), we continue the calculation in (3.27) to obtain the lower bound

$$\begin{aligned}\langle \mathbf{v}, (C_0^*)^{-1}\mathbf{v} \rangle &\geq \sum_{k=K+1}^{N} c_k^2 \tau^{-2\alpha} \theta^{\alpha} = \tau^{-2\alpha} \theta^{\alpha} |(I - P_0)\mathbf{v}|^2 \\ &\geq \frac{1}{4} \tau^{-2\alpha} \theta^{\alpha} \left(4K/\zeta^2 + 1 \right)^{-1}.\end{aligned}$$

Putting together the lower bounds from Cases 1 and 2 gives

$$\langle \mathbf{v}, (C_0^*)^{-1}\mathbf{v} \rangle \geq \min \left\{ \gamma^{-2} (4K/\zeta^2 + 1)^{-1}, \frac{1}{4} \tau^{-2\alpha} \theta^{\alpha} (4K/\zeta^2 + 1)^{-1} \right\}$$

for all unit vectors $\mathbf{v}$ and constants $\gamma, \tau, \alpha > 0$. Since the trace of a matrix coincides with the sum of its eigenvalues, we conclude that

$$\mathrm{Tr}(C_0^*) \leq N \max \left\{ \gamma^2 (4K/\zeta^2 + 1), 4\tau^{2\alpha} \theta^{-\alpha} \left(4K/\zeta^2 + 1 \right) \right\},$$

from which the desired result follows by taking $\Xi = N \left(4K/\zeta^2 + 1 \right) \max \left\{ 1, 4\theta^{-\alpha} \right\}$. □

3.5.2.2 Proof of Lemma 2

Proof. Recall (3.13). Then

$$C_0^* = C_0^* \left(\frac{1}{\gamma^2} B + C_{\tau,0}^{-1} \right) C_0^* = \frac{1}{\gamma^2} C_0^* B C_0^* + C_0^* C_{\tau,0}^{-1} C^*,$$

which gives the identity

$$\mathrm{Tr} \left(\frac{1}{\gamma^2} C_0^* B C_0^* \right) = \mathrm{Tr} \left(C_0^* \right) - \mathrm{Tr} \left(C_0^* C_{\tau,0}^{-1} C_0^* \right).$$

Both C_0^* and $C_{\tau,0}^{-1}$ are positive definite and so is their product $C_0^* C_{\tau,0}^{-1} C_0^*$. Therefore, $\mathrm{Tr} \left(C_0^* C_{\tau,0}^{-1} C_0^* \right) \geq 0$, so using Lemma 1 we have $\mathrm{Tr} \left(\frac{1}{\gamma^2} C_0^* B C_0^* \right) \leq \mathrm{Tr} \left(C_0^* \right) \leq \Xi \max \left\{ \gamma^2, \tau^{2\alpha} \right\}$. □

3.5.2.3 Proof of Lemma 3

Proof. Choose any vector $\mathbf{v} \in \text{span}\{\bar{\boldsymbol{\chi}}_1, \ldots, \bar{\boldsymbol{\chi}}_K\}$ and recall (3.13), the definition of C_0^*. Then

$$\begin{aligned}\left|\frac{1}{\gamma^2} C_0^* B\mathbf{v} - \mathbf{v}\right| = \left|C_0^* \left(\frac{1}{\gamma^2} B\mathbf{v} - (C_0^*)^{-1}\mathbf{v}\right)\right| &\le \|C_0^*\|_2 \left|\frac{1}{\gamma^2} B\mathbf{v} - (C_0^*)^{-1}\mathbf{v}\right| \\ &= \|C_0^*\|_2 \left|C_{\tau,0}^{-1}\mathbf{v}\right| \le \text{Tr}(C_0^*) \left|C_{\tau,0}^{-1}\mathbf{v}\right|.\end{aligned}$$

Recall from Subsection 3.3.1 that the vectors $\bar{\boldsymbol{\chi}}_k$ are eigenvectors of L_0 corresponding to an eigenvalue of 0, so they are also eigenvectors of $C_{\tau,0}^{-1}$ with eigenvalue 1. Therefore, since $\mathbf{v} \in \text{span}\left\{\bar{\boldsymbol{\chi}}_k\right\}_{k=1}^K$ it follows that $C_{\tau,0}^{-1}\mathbf{v} = \mathbf{v}$. Using this fact and Lemma 1, we conclude that

$$\left|\frac{1}{\gamma^2} C_0^* B\mathbf{v} - \mathbf{v}\right| \le \Xi \max\{\gamma^2, \tau^{2\alpha}\}|\mathbf{v}|.$$

The desired bound for the vectors $\mathbf{u}_m^\dagger$ now follows trivially from Assumption 3. □

3.5.3 Proofs of Lemmata 4–6

3.5.3.1 Proof of Lemma 4

Proof. We use a similar argument to the proof of Lemma 1 and obtain lower bounds on the Dirichlet energy $\langle \mathbf{v}, (C_\epsilon^*)^{-1}\mathbf{v}\rangle$ for unit vectors $\mathbf{v} \in \mathbb{R}^N$. Recall that $P_0 \in \mathbb{R}^{N\times N}$ denotes the projection matrix onto $\text{span}\{\bar{\boldsymbol{\chi}}_k\}_{k=1}^K$ and define ζ, β as in (3.25). Once again, we obtain the lower bounds in two cases: (1) $|P_0\mathbf{v}|\ge \beta$ and (2) $|P_0\mathbf{v}|< \beta$.

The case of $|P_0\mathbf{v}|\ge \beta$ follows from identical arguments to Case 1 in the proof of Lemma 1. In fact, the lower bound (3.26) holds for C_ϵ^* replacing C_0^*, so whenever $|P_0\mathbf{v}|\ge \beta$ we have

$$\langle \mathbf{v}, (C_\epsilon^*)^{-1}\mathbf{v}\rangle \ge \gamma^{-2}\left(4K/\zeta^2 + 1\right)^{-1}.$$

So we focus on the case where $|P_0\mathbf{v}|< \beta$, which implies that $|(I - P_0)\mathbf{v}|\ge \sqrt{1-\beta^2}$. Let $\{(\sigma_{j,\epsilon}, \boldsymbol{\phi}_{j,\epsilon})\}_{j=1}^N$ denote the eigenpairs of L_ϵ, indexed by order of increasing eigenvalue. Note that these orthonormal eigenvectors are also eigenvectors of $C_{\tau,\epsilon}^{-1}$. We let $P_\epsilon \in \mathbb{R}^{N\times N}$ denote the projection matrix onto $\text{span}\{\boldsymbol{\phi}_{1,\epsilon}, \boldsymbol{\phi}_{2,\epsilon}, \cdots, \boldsymbol{\phi}_{K,\epsilon}\}$. The key difference in this proof, compared to Case 2 in the proof of Lemma 1, is that we need to establish a lower bound on

$|(I-P_\epsilon)\mathbf{v}|$. We show that if $\epsilon \in (0, \epsilon_0)$ for a sufficiently small constant ϵ_0, then

$$|(I-P_\epsilon)\mathbf{v}| \geq \frac{1}{2}\sqrt{1-\beta^2} = \frac{1}{2}(4K/\zeta^2+1)^{-1/2}. \tag{3.28}$$

Using (3.21) and the fact that B is positive semi-definite, we can then write

$$\begin{aligned} \langle \mathbf{v}, (C_\epsilon^*)^{-1}\mathbf{v}\rangle &= \left\langle \mathbf{v}, \left(\frac{1}{\gamma^2}B + C_{\tau,\epsilon}^{-1}\right)\mathbf{v}\right\rangle \\ &\geq \langle \mathbf{v}, C_{\tau,\epsilon}^{-1}\mathbf{v}\rangle \geq \sum_{j=K+1}^{N} c_{j,\epsilon}^2 \tau^{-2\alpha}(\sigma_{j,\epsilon}+\tau^2)^\alpha, \end{aligned} \tag{3.29}$$

where $c_{j,\epsilon} := \langle \mathbf{v}, \boldsymbol{\phi}_{j,\epsilon}\rangle$. By [61, Lemm A.5], the graph Laplacian L_ϵ satisfies an expansion of the form

$$L_\epsilon = L_0 + \sum_{h=1}^{\infty} \epsilon^h L^{(h)},$$

where $\{\|L^{(h)}\|_2\}_{h=1}^\infty \in \ell^\infty$. Moreover, by [61, Prop. A7] and the binomial theorem, we have that

$$\begin{aligned} \tau^{-2\alpha}(\sigma_{K+1,\epsilon}+\tau^2)^\alpha &\geq \tau^{-2\alpha}\left(\theta + \tau^2 - \epsilon\sum_{h=1}^{\infty}\epsilon^{h-1}\|L^{(h)}\|_2\right)^\alpha \\ &> \theta^\alpha\tau^{-2\alpha}\left(1 - \frac{\epsilon}{\tau^2}\sum_{h=1}^{\infty}\epsilon^{h-1}\|L^{(h)}\|_2\right)^\alpha = \theta^\alpha\tau^{-2\alpha}\left(1-\frac{\epsilon}{\tau^2}\Xi_1\right)^\alpha, \end{aligned}$$

where $\Xi_1 := \sup_{\epsilon\in(0,\epsilon_0)}\sum_{h=1}^\infty \epsilon^{h-1}\|L^{(h)}\|_2$ which is bounded provided that $\epsilon_0 < 1$. Substituting this lower bound into (3.29) and recalling the increasing ordering of the $\sigma_{j,\epsilon}$ we obtain

$$\begin{aligned} \langle \mathbf{v}, (C_\epsilon^*)^{-1}\mathbf{v}\rangle &\geq \theta^\alpha\tau^{-2\alpha}\left(1-\frac{\epsilon}{\tau^2}\Xi_1\right)^\alpha \sum_{j=K+1}^{N} c_{j,\epsilon}^2 \\ &= \theta^\alpha\tau^{-2\alpha}\left(1-\frac{\epsilon}{\tau^2}\Xi_1\right)^\alpha |(I-P_\epsilon)\mathbf{v}|^2 \geq \frac{1}{4}\theta^\alpha\tau^{-2\alpha}\left(1-\frac{\epsilon}{\tau^2}\Xi_1\right)^\alpha (4K/\zeta^2+1)^{-1}. \end{aligned}$$

Putting this bound together with the lower bound from the first case where $|P_0\mathbf{v}| \geq \beta$, we conclude that

$$\langle \mathbf{v}, (C_\epsilon^*)^{-1}\mathbf{v}\rangle \geq \min\left\{\gamma^{-2}(4K/\zeta^2+1)^{-1}, \frac{1}{4}\tau^{-2\alpha}(1-\epsilon\tau^{-2}\Xi_1)^\alpha\theta^\alpha(4K/\zeta^2+1)^{-1}\right\},$$

from which it follows that

$$\mathrm{Tr}(C_\epsilon^*) \leq N\max\left\{\gamma^2(4K/\zeta^2+1), \frac{1}{4}\tau^{2\alpha}(1-\epsilon\tau^{-2}\Xi_1)^{-\alpha}\theta^{-\alpha}(4K/\zeta^2+1)\right\},$$

provided that $\epsilon_0 > 0$ is sufficiently small which concludes the proof of the lemma.

It remains for us to prove the bound (3.28). By [61, Prop. A.6 and proof of Prop. A.10], there exist uniform constants $\epsilon_1, \Xi_2 > 0$ so that $\forall \epsilon \in (0, \epsilon_1)$ and for any unit vector $\mathbf{v}$ we have

$$|(I - P_\epsilon)P_0\mathbf{v}|^2 \leq \Xi_2\epsilon^2 \quad \text{and} \quad |(I - P_0)P_\epsilon\mathbf{v}|^2 \leq \Xi_2\epsilon^2,$$

implying that the range of P_ϵ and P_0 are close when ϵ is small. Therefore, using the fact that P_0 and P_ϵ are symmetric and idempotent, as well as the Cauchy-Schwarz inequality, we can write

$$\begin{aligned}|(P_0 - P_\epsilon)\,\mathbf{v}|^2 &= \langle (P_0 - P_\epsilon)\mathbf{v}, P_0\mathbf{v}\rangle - \langle (P_0 - P_\epsilon)\mathbf{v}, P_\epsilon\mathbf{v}\rangle \\ &= \langle \mathbf{v}, (P_0 - P_\epsilon)\,P_0\mathbf{v}\rangle - \langle \mathbf{v}, (P_0 - P_\epsilon)\,P_\epsilon\mathbf{v}\rangle\,\langle \mathbf{v}, (I - P_\epsilon)P_0\mathbf{v}\rangle + \langle \mathbf{v}, (I - P_0)P_\epsilon\mathbf{v}\rangle \\ &\leq |\mathbf{v}|(|(I - P_\epsilon)P_0\mathbf{v}| + |(I - P_0)P_\epsilon\mathbf{v}|) \leq \Xi_3\epsilon.\end{aligned}$$

The lower bound (3.28) then follows from the following calculation

$$\begin{aligned}|(I - P_\epsilon)\mathbf{v}| = |(I - P_0)\mathbf{v} + (P_0 - P_\epsilon)\mathbf{v}| &\geq \max\{0, |(I - P_0)\mathbf{v}| - |(P_0 - P_\epsilon)\mathbf{v}|\} \\ &\geq \max\left\{0, \sqrt{1 - \beta^2} - (\Xi_3\epsilon)^{1/2}\right\} \geq \frac{\sqrt{1 - \beta^2}}{2} = \frac{1}{2}(4K/\zeta^2 + 1)^{-1/2}\end{aligned}$$

where the last inequality holds when $\epsilon_0 \leq \frac{1-\beta^2}{4\Xi_3}$. □

3.5.3.2 Proof of Lemma 6

Proof. The proof is nearly identical to that of Lemma 2 and is hence omitted. □

3.5.3.3 Proof of Lemma 6

Proof. We proceed similarly to the proof of Lemma 3 by choosing a vector $\mathbf{v} \in \text{span}\{\bar{\boldsymbol{\chi}}_k\}_{k=1}^K$. We then have

$$\begin{aligned}\left|\frac{1}{\gamma^2}C_\epsilon^* B\mathbf{v} - \mathbf{v}\right| &= \left|C_\epsilon^*\left(\frac{1}{\gamma^2}B\mathbf{v} - (C_\epsilon^*)^{-1}\mathbf{v}\right)\right| \\ &\leq \|C_\epsilon^*\|_2 \left|\frac{1}{\gamma^2}B\mathbf{v} - (C_\epsilon^*)^{-1}\mathbf{v}\right| = \|C_\epsilon^*\|_2 \left|C_{\tau,\epsilon}^{-1}\mathbf{v}\right|.\end{aligned}$$

Now decompose $\mathbf{v} = P_\epsilon \mathbf{v} + (I - P_\epsilon)\mathbf{v}$. Since we assumed that $\mathbf{v} \in \operatorname{span}\{\bar{\boldsymbol{\chi}}_\ell\}_{\ell=1}^K$, it follows from [61, Prop. A.6] that $|(I - P_\epsilon)\mathbf{v}| \leq \Xi_3 \epsilon |\mathbf{v}|$ for some $\Xi_3 > 0$ independent of ϵ, so

$$\begin{aligned}
\left|C_{\tau,\epsilon}^{-1}\mathbf{v}\right| &\leq \left|C_{\tau,\epsilon}^{-1} P_\epsilon \mathbf{v}\right| + \left|C_{\tau,\epsilon}^{-1}(I - P_\epsilon)\mathbf{v}\right| \\
&\leq \max_{k \leq K} \frac{(\sigma_{k,\epsilon} + \tau^2)^\alpha}{\tau^{2\alpha}} |P_\epsilon \mathbf{v}| + \max_{k > K} \frac{(\sigma_{k,\epsilon} + \tau^2)^\alpha}{\tau^{2\alpha}} |(I - P_\epsilon)\mathbf{v}| \\
&\leq \Xi_4 \left[\left(1 + \frac{\epsilon}{\tau^2}\right)^\alpha + \epsilon \left(1 + \frac{1}{\tau^{2\alpha}}\right) \right] |\mathbf{v}|.
\end{aligned}$$

The third inequality follows from the fact that the $\sigma_{k,\epsilon}$ are uniformly bounded for all $\epsilon \in (0, \epsilon_0)$ and $\epsilon_0 < 1$. In fact, by [61, Lemm. A.5], we have that

$$\begin{aligned}
\sigma_{k,\epsilon} = \langle \boldsymbol{\phi}_{k,\epsilon}, L_\epsilon \boldsymbol{\phi}_{k,\epsilon} \rangle &\leq |\langle \boldsymbol{\phi}_{k,\epsilon}, L_0 \boldsymbol{\phi}_{k,\epsilon} \rangle| + \sum_{h=1}^\infty \epsilon^h \left|\langle \boldsymbol{\phi}_{k,\epsilon}, L_h \boldsymbol{\phi}_{k,\epsilon} \rangle\right| \\
&\leq \|L_0\|_2 + \frac{\epsilon}{1-\epsilon} \left(\max_{h=1,2,\dots} \|L_h\|_2 \right) \leq \frac{1}{1-\epsilon} \left(\max_{h=0,1,\dots} \|L_h\|_2 \right).
\end{aligned}$$

Now bounding $\|C_\epsilon^*\|_2$ by $\operatorname{Tr}(C_\epsilon^*)$ and envoking Lemma 4 yields

$$\|C_\epsilon^*\|_2 \left|C_{\tau,\epsilon}^{-1}\mathbf{v}\right| \leq \Xi_0 \Xi_4 \max\left\{ \gamma^2, \left(\frac{\tau^2}{1 - \Xi_1 \epsilon/\tau^2}\right)^\alpha \right\} \left[\epsilon + \frac{\epsilon}{\tau^{2\alpha}} + \left(1 + \frac{\epsilon}{\tau^2}\right)^\alpha \right] |\mathbf{v}|.$$

The theorem follows by setting $\Xi_2 = \Xi_0 \Xi_4$. □

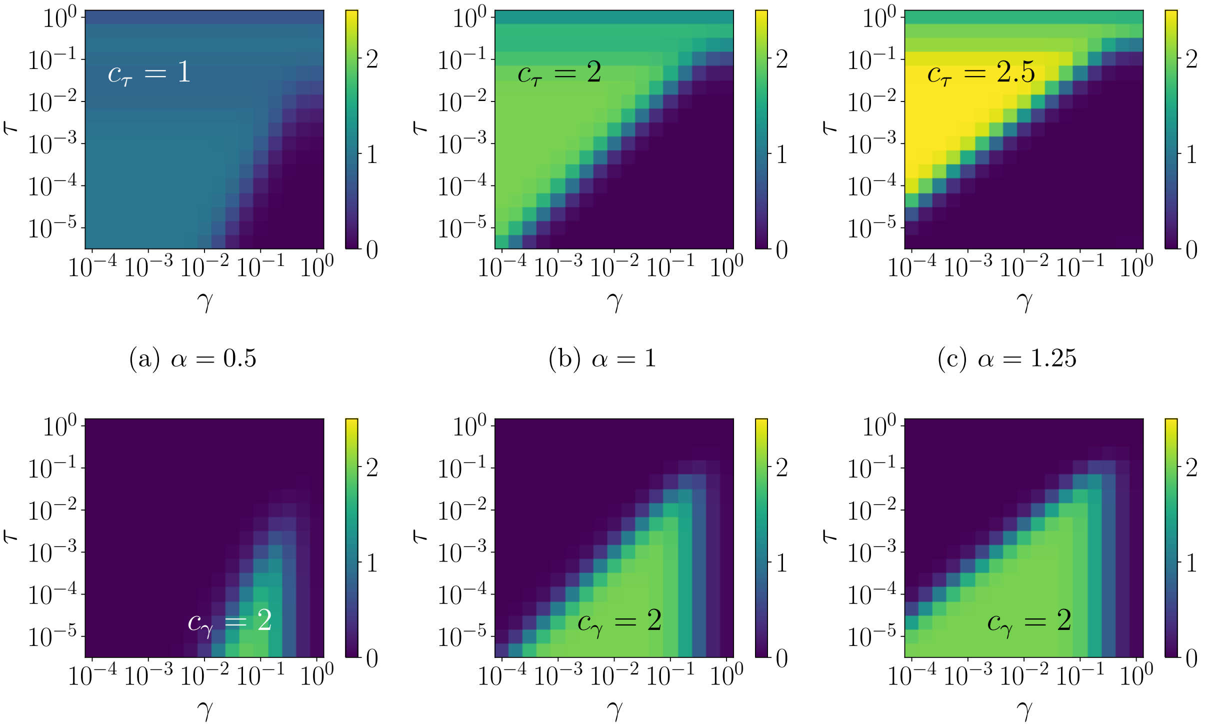
3.6 Numerical Demonstration of Lemmata

3.6.1 Numerics in Support of Lemmata 1–3

In Figures 3.4 and 3.5, we present numerics that illustrate the convergence results for Lemmata 1 and 3,respectively. These lemmata bound the first and third terms respectively of the decomposition of

$$\mathcal{I}(\gamma, \alpha, \tau) = M \operatorname{Tr}(C_0^*) + \frac{M}{\gamma^2} \operatorname{Tr}(C_0^* B C_0^*) + \sum_{m=1}^M \left| \frac{1}{\gamma^2} C_0^* B \mathbf{u}_m^\dagger - \mathbf{u}_m^\dagger \right|^2.$$

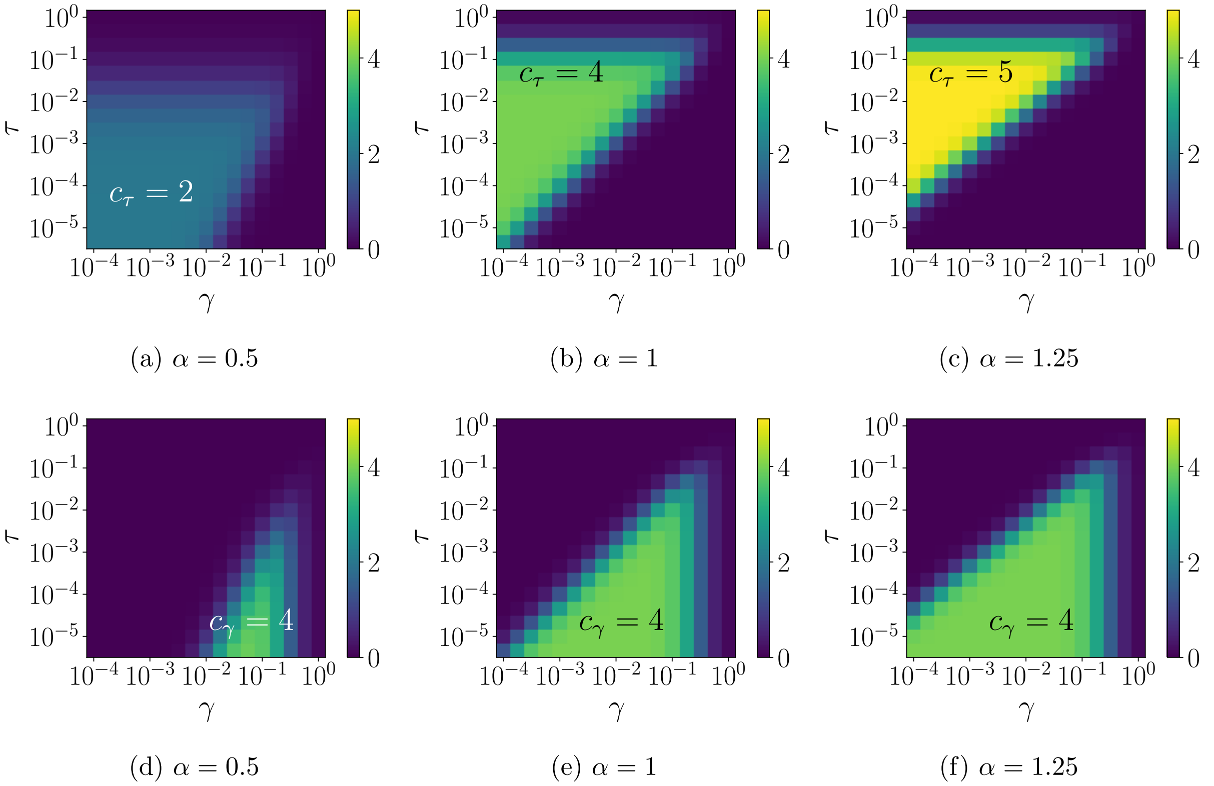
Numerics for the term $1/\gamma^2 \operatorname{Tr}(C_0^* B C_0^*)$ are omitted since the corresponding bound in Lemma 2 is derived from the bound found for $\operatorname{Tr}(C_0^*)$ in Lemma 1, and exhibit nearly identical behavior numerically. The top panels in Figure 3.4 show the estimated rate of



(a) $\alpha = 0.5$ (b) $\alpha = 1$ (c) $\alpha = 1.25$

(d) $\alpha = 0.5$ (e) $\alpha = 1$ (f) $\alpha = 1.25$

Figure 3.4: A numerical demonstration of Lemma 1 on the synthetic data set with $\epsilon = 0$. The top panels showcase the numerical estimates of the logarithmic slope $c_\tau := \frac{\partial \log(\mathrm{Tr}(C_0^*))}{\partial \log(\tau)}$ for different α values and the bottom panels showcase the numerical estimates of the logarithmic slope $c_\gamma := \frac{\partial \log(\mathrm{Tr}(C_0^*))}{\partial \log(\gamma)}$. In the dark blue region, $c_\tau, c_\gamma \approx 0$, indicating that $\mathrm{Tr}(C_0^*)$ stays approximately flat with respect to the respective variable τ or γ; the slope of the brighter regions is annotated in each panel. The transition between the dark and bright regions occurs approximately at $\tau = \gamma^{1/\alpha}$.



(a) $\alpha = 0.5$

(b) $\alpha = 1$

(c) $\alpha = 1.25$

(d) $\alpha = 0.5$

(e) $\alpha = 1$

(f) $\alpha = 1.25$

Figure 3.5: A numerical demonstration of Lemma 3 on the synthetic data set with $\epsilon = 0$. The top panels showcase the numerical estimates of the logarithmic slope $c_\tau := \frac{\partial \log(|C_0^* B\mathbf{u}_m^\dagger/\gamma^2 - \mathbf{u}_m^\dagger|^2)}{\partial \log(\tau)}$ for different α values and the bottom panels showcase the numerical estimates of the logarithmic slope $c_\gamma := \frac{\partial \log(|C_0^* B\mathbf{u}_m^\dagger/\gamma^2 - \mathbf{u}_m^\dagger|^2)}{\partial \log(\gamma)}$. In the dark blue region, $c_\tau, c_\gamma \approx 0$, indicating that $|C_0^* B\mathbf{u}_m^\dagger/\gamma^2 - \mathbf{u}_m^\dagger|^2$ stays approximately flat with respect to the respective variable τ or γ; the slope of the brighter regions is annotated in each panel. The transition between the dark and bright regions occurs approximately at $\tau = \gamma^{1/\alpha}$.

convergence of $\mathrm{Tr}(C_0^*)$ in terms of τ in the log-log scale, while the bottom panels show the estimated rate of convergence in terms of γ in the log-log scale. Figure 3.5 shows the estimated rate of convergence $\left|\frac{1}{\gamma^2}C_0^* B\mathbf{u}_m^\dagger - \mathbf{u}_m^\dagger\right|^2$ in the parameters τ and γ. From Figure 3.4, in the region where $\gamma^2 \ll \tau^{2\alpha}$, we see that $\frac{\partial \log(\mathrm{Tr}(C_0^*))}{\partial \log(\tau)}$ stays close to 2α, whereas $\frac{\partial \log(\mathrm{Tr}(C_0^*))}{\partial \log(\gamma)}$ is approximately 0. In the region where $\tau^{2\alpha} \ll \gamma^2$, we observe that $\frac{\partial \log(\mathrm{Tr}(C_0^*))}{\partial \log(\tau)}$ is close to 0, whereas $\frac{\partial \log(\mathrm{Tr}(C_0^*))}{\partial \log(\gamma)}$ is around 2. These results illustrate our bound in Lemma 1.

In Figure 3.5, in the region where $\gamma^2 \ll \tau^{2\alpha}$, we see that $\frac{\partial \log(|C_0^* B\mathbf{u}_m^\dagger/\gamma^2 - \mathbf{u}_m^\dagger|^2)}{\partial \log(\tau)}$ stays close to 4α, whereas $\frac{\partial \log(|C_0^* B\mathbf{u}_m^\dagger/\gamma^2 - \mathbf{u}_m^\dagger|^2)}{\partial \log(\gamma)}$ is approximately 0. In the region where $\tau^{2\alpha} \ll \gamma^2$, we observe that $\frac{\partial \log(|C_0^* B\mathbf{u}_m^\dagger/\gamma^2 - \mathbf{u}_m^\dagger|^2)}{\partial \log(\tau)}$ is close to 0, whereas $\frac{\partial \log(|C_0^* B\mathbf{u}_m^\dagger/\gamma^2 - \mathbf{u}_m^\dagger|^2)}{\partial \log(\gamma)}$ is around 4. These results illustrate our bounds presented in Lemma 3.

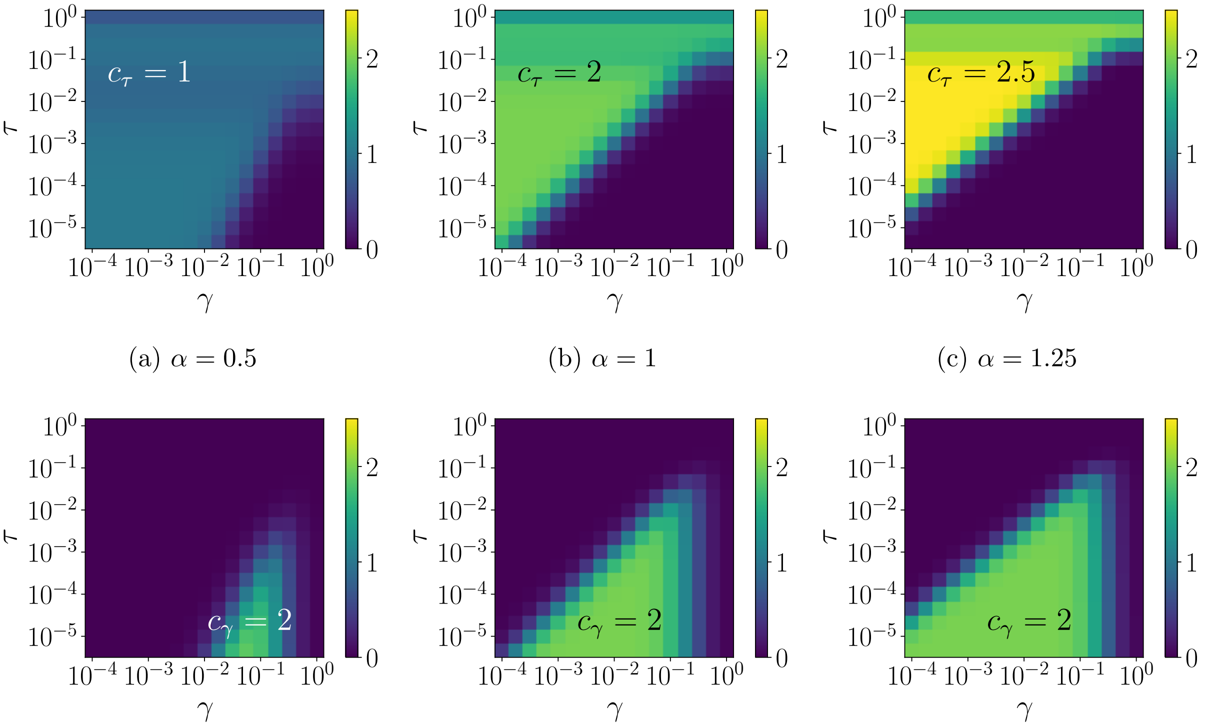
3.6.2 Numerics in Support of Lemmata 4–6

In Figures 3.6 and 3.7, we present numerics that illustrate the convergence results for Lemmata 4 and 6, respectively. These lemmata bound the first and third terms respectively of the decomposition of $\mathcal{I}$:

$$\mathcal{I}(\gamma, \alpha, \tau, \epsilon) = M\mathrm{Tr}(C_\epsilon^*) + \frac{M}{\gamma^2}\mathrm{Tr}(C_\epsilon^* B C_\epsilon^*) + \sum_{m=1}^{M}\left|\frac{1}{\gamma^2}C_\epsilon^* B\mathbf{u}_m^\dagger - \mathbf{u}_m^\dagger\right|^2.$$

Again, we omit the second term in this decomposition since the corresponding bound in Lemma 5 is derived from the bound found for $\mathrm{Tr}(C_\epsilon^*)$ in Lemma 4 and exhibit nearly identical behavior numerically. Just as in Figures 3.2 and 3.3, we have set the scaling $\epsilon = \tau^{\max\{2,2\alpha\}}$. The top panels in Figure 3.6 show the estimated rate of convergence of $\mathrm{Tr}(C_\epsilon^*)$ in terms of τ on a log-log scale, while the bottom panels show the estimated rate of convergence in terms of γ on a log-log scale. Figure 3.7 shows the estimated rate of convergence $|C_\epsilon^* B\mathbf{u}_m^\dagger/\gamma^2 - \mathbf{u}_m^\dagger|$ in the parameters ϵ and γ. From Figure 3.6, in the region where $\gamma^2 \ll \tau^{2\alpha}$, we see that $\frac{\partial \log(\mathrm{Tr}(C_\epsilon^*))}{\partial \log(\tau)}$ stays close to 2α, whereas $\frac{\partial \log(\mathrm{Tr}(C_\epsilon^*))}{\partial \log(\gamma)}$ is approximately 0. In the region where $\tau^{2\alpha} \ll \gamma^2$, we observe that $\frac{\partial \log(\mathrm{Tr}(C_\epsilon^*))}{\partial \log(\tau)}$ is close to 0, whereas $\frac{\partial \log(\mathrm{Tr}(C_\epsilon^*))}{\partial \log(\gamma)}$ is around 2. These results illustrate our bound presented in Lemma 4.

In Figure 3.7, in the region where $\gamma^2 \ll \tau^{2\alpha}$, we see that $\frac{\partial \log(|C_\epsilon^* B\mathbf{u}_m^\dagger/\gamma^2 - \mathbf{u}_m^\dagger|^2)}{\partial \log(\tau)}$ stays close



(a) $\alpha = 0.5$

(b) $\alpha = 1$

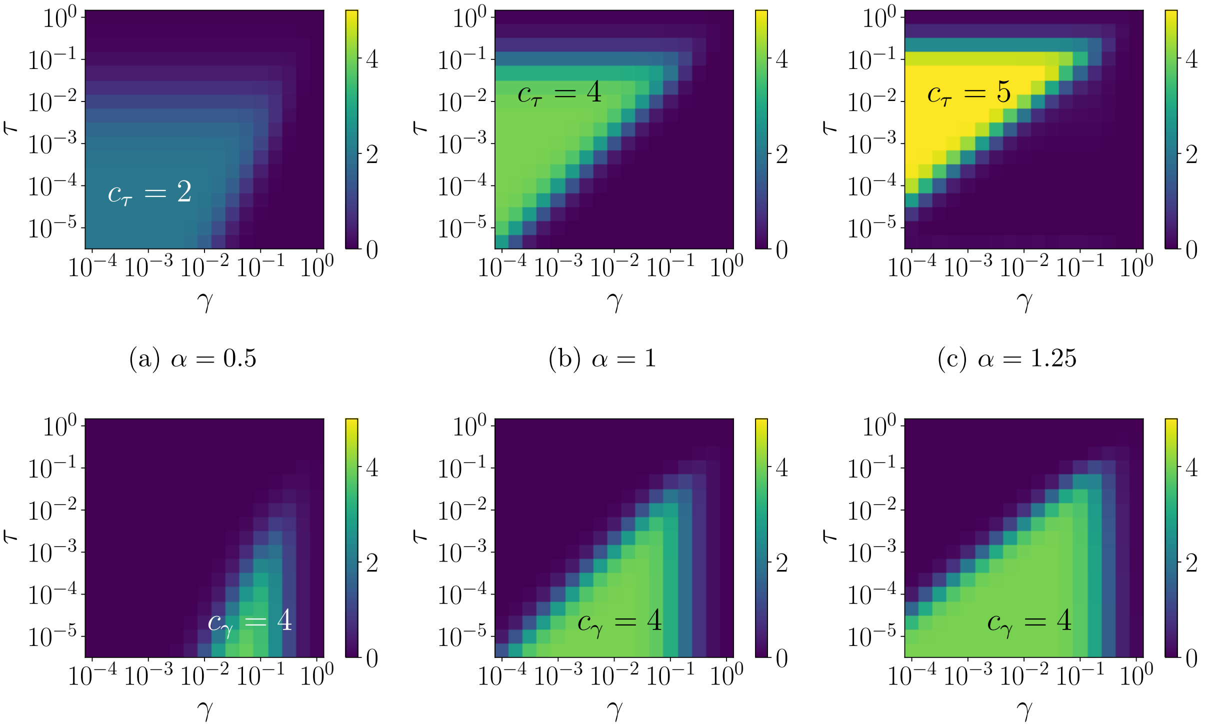
(c) $\alpha = 1.25$

(d) $\alpha = 0.5$

(e) $\alpha = 1$

(f) $\alpha = 1.25$

Figure 3.6: A numerical demonstration of Lemma 4 on the synthetic data set with $\epsilon = \tau^{2\alpha}$. The top panels showcase the numerical estimates of the logarithmic slope $c_\tau := \frac{\partial \log(\mathrm{Tr}(C_\epsilon^*))}{\partial \log(\tau)}$ for different α values and the bottom panels showcase the numerical estimates of the logarithmic slope $c_\gamma := \frac{\partial \log(\mathrm{Tr}(C_\epsilon^*))}{\partial \log(\gamma)}$. In the dark blue region, $c_\tau, c_\gamma \approx 0$, indicating that $\mathrm{Tr}(C_\epsilon^*)$ stays approximately flat with respect to the respective variable τ or γ; the slope of the brighter regions is annotated in each panel. The transition between the dark and bright regions occurs approximately at $\tau = \gamma^{1/\alpha}$.



(a) $\alpha = 0.5$

(b) $\alpha = 1$

(c) $\alpha = 1.25$

(d) $\alpha = 0.5$

(e) $\alpha = 1$

(f) $\alpha = 1.25$

Figure 3.7: A numerical demonstration of Lemma 6 on a synthetic data set with $\epsilon = \tau^{\max\{2,2\alpha\}}$. The top panels showcase the numerical estimates of the logarithmic slope $c_\tau := \frac{\partial \log(|C_\epsilon^* B\mathbf{u}_m^\dagger/\gamma^2 - \mathbf{u}_m^\dagger|^2)}{\partial \log(\tau)}$ for different α values and the bottom panels showcase the numerical estimates of the logarithmic slope $c_\gamma := \frac{\partial \log(|C_\epsilon^* B\mathbf{u}_m^\dagger/\gamma^2 - \mathbf{u}_m^\dagger|^2)}{\partial \log(\gamma)}$. In the dark blue region, $c_\tau, c_\gamma \approx 0$, indicating that $|C_\epsilon^* B\mathbf{u}_m^\dagger/\gamma^2 - \mathbf{u}_m^\dagger|^2$ stays approximately flat with respect to the respective variable τ or γ; the slope of the brighter regions is annotated in each panel. The transition between the dark and bright regions occurs approximately at $\tau = \gamma^{1/\alpha}$.

to 4α, whereas $\frac{\partial \log(|C_\epsilon^* B \mathbf{u}_m^\dagger/\gamma^2 - \mathbf{u}_m^\dagger|^2)}{\partial \log(\gamma)}$ is approximately 0. In the region where $\tau^{2\alpha} \ll \gamma^2$, we observe that $\frac{\partial \log(|C_\epsilon^* B \mathbf{u}_m^\dagger/\gamma^2 - \mathbf{u}_m^\dagger|^2)}{\partial \log(\tau)}$ is close to 0, whereas $\frac{\partial \log(|C_\epsilon^* B \mathbf{u}_m^\dagger/\gamma^2 - \mathbf{u}_m^\dagger|^2)}{\partial \log(\gamma)}$ is around 4. These results illustrate our bounds presented in Lemma 6.

3.7 Conclusion

The work in this chapter is, to the best of our knowledge, the first analysis of Bayesian posterior consistency in SSL. The regression formulation of SSL is convenient for both computations and analysis due to conjugacy of Gaussian likelihoods and priors, leading to a Gaussian posterior. The resulting closed form is useful in practice [172] and for theory, such as that developed in this chapter. We formulate SSR problem as a Bayesian inverse problem in which the unlabeled data defines the prior and the labeled data defines the likelihood. By postulating coherence between the labeled and unlabeled data we are able to quantify the convergence of the posterior distribution to the truth in terms of the noise in the labels and a measure of clustering in the data. As a by-product of the analysis we also learn about parameter choices within the prior construction.

CHAPTER 4

Uncertainty Quantification

4.1 Background

In this chapter, we continue the discussion of SSL problems in the BIP framework and present a method of uncertainty quantification (UQ). We generalize the UQ methodology — originally proposed in [19] to be paired with the binary-classification problem — to a multi-class setting. This chapter is a version of [126]. This work was done in collaboration with undergraduate researchers Yiling Qiao, Change Shi, and Chenjian Wang, whom I supervised together with Matt Haberland. The undergraduate researchers are responsible for conducting preliminary experiments based on the work by Xiyang Luo and Andrew Stuart. My contribution includes designing and implementing the algorithm and carrying out the final set of experiments of which the results are presented in this thesis. The whole project was supervised by Andrea Bertozzi.

UQ seeks to estimate a measure of uncertainty for a classification; it identifies data whose classification results are uncertain according to our classification model. We refer the reader to the books [136, 145] and the recent article [116] for a review of methodologies employed in the field of UQ. For application to machine-learning methods, the book [165] investigates UQ for a variety of machine-learning problems using a Gaussian-process prior. Except the above-mentioned book and the recent work [19], most machine learning methods, even those developed with a Bayesian way of thinking, focus on finding an optimal classification (and/or hyperparameters that produce the optimal classification) in an optimization context and do not consider or utilize UQ.

4.2 Methodology

We consider a Bayesian model similar to the one studied in Chapter 3. The posterior distribution of the label assignment function U takes the form

$$\mu^Y(\mathrm{d}U) \propto \exp(-\mathcal{J}(U)), \quad \mathcal{J}(U) = \frac{1}{2}\langle U, LU\rangle_F + \Phi(U;Y), \tag{4.1}$$

so a maximum a posteriori probability (MAP) estimator is a minimizer of $\mathcal{J}(U)$. We assume the prior on U is a Gaussian distribution,

$$\mu_0(\mathrm{d}U) \propto \exp\left(-\frac{1}{2}\langle U, LU\rangle_F\right).$$

To explicitly construct a sample U that follows the prior distribution, we employ the Karhunen-Loéve expansion. Recall that $L = \boldsymbol{\phi}\Lambda\boldsymbol{\phi}^T$ is the eigen-decomposition of the symmetrically normalized graph Laplacian (with $p = 1/2$ in (1.2)), where the columns of $\boldsymbol{\phi} \in \mathbb{R}^{N\times N}$ form an orthonormal basis of $\mathbb{R}^N$ and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_N)$ satisfies

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N.$$

We observe that L is positive semi-definite. Recall that $\mathbf{u}_m$ are the columns of U, so by the Karhunen-Loéve (KL) theorem, we construct a sample U by specifying its columns as the random sum

$$\mathbf{u}_m = \sum_{j=1}^{N} \frac{1}{\sqrt{\lambda_j}}\xi_{mj}\boldsymbol{\phi}_j, \tag{4.2}$$

with $\{(\lambda_j, \boldsymbol{\phi}_j)\}_{j=1}^N$ denoting the eigenpairs of L and $\xi_{mj} \overset{iid}{\sim} \mathcal{N}(0,1)$.

We note that the columns $\mathbf{u}_m$ live in $\mathrm{span}\{\boldsymbol{\phi}_1\}^\perp$ and U has the desired probability distribution

$$\mu_0(\mathrm{d}U) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^{N}\sum_{m=1}^{M}\lambda_i\langle \mathbf{u}_m, \boldsymbol{\phi}_i\rangle^2\right) = \exp\left(-\frac{1}{2}\langle U, LU\rangle_F\right). \tag{4.3}$$

In their recent work [19], the authors considered several likelihood functions $\mathbb{P}(Y|U)$ to connect the latent variable U to the ground-truth labeling Y for binary classification. In the previous two chapters, we considered least-squares likelihood. In this chapter, we primarily investigate the independent probit likelihood function. Suppose $\{\eta(i)\}_{i\in Z'}$ is a collection of independent M-variate normal random variables $\mathcal{N}(0, I_M)$. We connect U to Y via

$$V(i) = U(i) + \gamma\eta(i), \quad Y(i) = \text{threshold}\left(V(i)\right), \quad i \in Z',$$

where γ^2 is the noise variance. The threshold operator applied to a vector sets the largest element in the vector to be 1 and the rest to be 0. With the introduction of latent variables $\{V(i)\}_{i\in Z'}$, we have, from Bayes' formula, the following joint posterior probability distribution:

$$\mathbb{P}(U, V|Y) \propto \exp\left(-\frac{1}{2}\langle U, LU\rangle_F - \frac{1}{2\gamma^2}\sum_{i\in Z'}|U(i) - V(i)|^2\right)\prod_{i\in Z'} 1_{\text{threshold}(V(i))=Y(i)}.$$

Using a change of variable from u to ξ, for $\xi = (\xi_{mj})$, we can apply our chosen sampling method to $\mathbb{P}(\xi, V|U')$. We compute that the joint probability

$$\mathbb{P}(\xi, V|Y) \propto \exp\left(-\frac{1}{2}\langle \xi^T, \Lambda\xi^T\rangle_F - \frac{1}{2\gamma^2}\left|H\boldsymbol{\phi}\xi^T - Y\right|^2\right)\prod_{i\in Z'} 1_{\text{threshold}(V(i))=Y(i)}.$$

To sample from the joint posterior distribution, a Gibbs sampler will repeat the following three steps:

(1) Draw ξ from $\mathbb{P}(\xi|V, Y)$,

(2) Construct U from ξ via (4.2),

(3) Draw V from $\mathbb{P}(V|U, Y)$.

For Step (1), we note that for each $m \in \{1, 2, \cdots, M\}$, the conditional probability for each row of ξ, denoted as $\mathbb{P}(\xi_{m,:}|V, Y)$ has the same distribution as

$$\mathcal{N}\left(\mathbf{m}, C^*\right), \; C^* = \left(\Lambda + \frac{1}{\gamma^2}\boldsymbol{\phi}^T B\boldsymbol{\phi}\right)^{-1}, \; \mathbf{m} = \frac{1}{\gamma^2}C^*\boldsymbol{\phi}^T H^T\mathbf{y}_m.$$

In Step (3), for each $i \in Z'$, we need to sample an M-variate normal random variable subject to a linear inequality constraint; suppose $Y(i) = \mathbf{e}_m$, i.e., data point i belongs to class m according to the ground-truth label. Then we need to sample $V(i)$ according to the following conditions:

$$V(i) \sim \mathcal{N}\left(U(i), \gamma^2 I_M\right), \quad \mathbf{v}_m(i) \geq \mathbf{v}_{m'}(i) \quad \text{for all } m' \in \{1, 2, \cdots, M\}.$$

We use the algorithm from [22] to efficiently draw samples from the linearly constrained normal distribution.

4.3 Uncertainty Quantification

Given a set of samples $\{U^{(k)}\}_{j=1}^{N_s}$ from the Gibbs sampler, we investigate $\mathbb{E}_{U|Y}\left(\text{threshold}(U)\right)$; i.e. the posterior mean of threshold$\,(U)$; this can be approximated by the sample mean

$$s_m(i) = \mathbb{E}_{U|U'}\left(\text{threshold}(U(i))_m\right) \approx \frac{1}{N_s} \sum_{j=1}^{N_s} \text{threshold}(U(i))_m.$$

Since each element threshold$\,(U(i))_m$ is either zero or one, the expectation $s_m(i)$ gives the probability, under the posterior distribution, of the element being one; that is, $s_m(i)$ can be interpreted as the probability data point i belongs to class m. We note that for each data point, the probability of it belonging to each class should sum to one; i.e.,

$$\sum_{m=1}^{M} s_m(i) = 1. \tag{4.4}$$

This is obeyed by both the posterior mean and the sample mean approximation. We can use the posterior mean $s(i)$ as a classifier, which classifies data point i according to its largest entry.

Intuitively, a single large $s_m(i)$ for a data point i indicates a very confident classification of class m; in this case, the remaining entries in $s(i)$ are necessarily small due to the sum-to-one condition (4.4); this creates a large variance in the vector $s(i)$. If entries in the vector $s(i)$ are all roughly equal, meaning the data point is equally likely to be classified as either class, the classification has a lot of uncertainty, resulting in an $s(i)$ with a small variance. Based on this intuition, we measure the classification confidence of node i by the variance

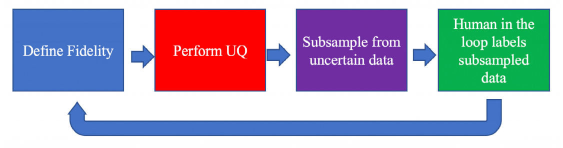

Define Fidelity
Perform UQ
Subsample from uncertain data
Human in the loop labels subsampled data

Figure 4.1: A flow chart summarizing the proposed human-in-the-loop system.

of $s(i)$; that is

$$S(i) = \mathrm{var}\left(s(i)\right) = \frac{1}{M}\sum_{m=1}^{M}\left(s_m(i) - \frac{1}{M}\sum_{m=1}^{M}s_m(i)\right)^2. \tag{4.5}$$

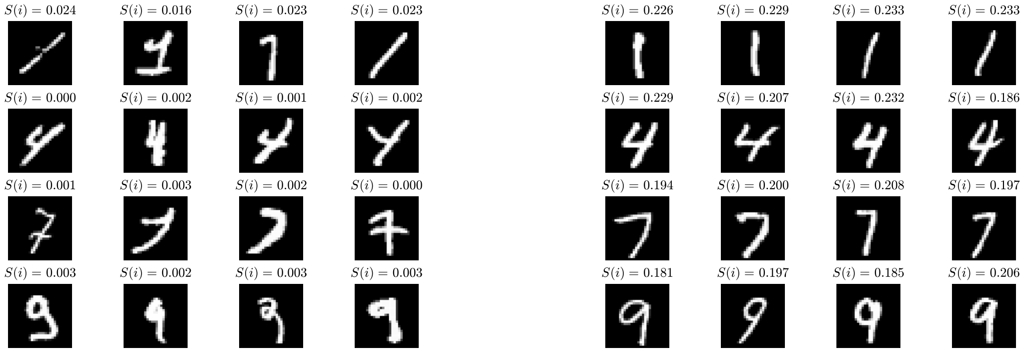
We emphasize that this variance is not the posterior variance. However, we can show the following connection between the quantity $S(i)$ and the posterior variance

$$S(i) = \frac{1}{M} - \frac{1}{M^2} - \frac{1}{M}\sum_{m=1}^{M}\mathrm{var}_{U|Y}\left(\mathrm{threshold}(U(i))_m\right),$$

where $\mathrm{var}_{U|Y}(\cdot)$ is the posterior variance. Therefore, the quantity $S(i)$ is a constant minus the mean posterior variance, which can be interpreted as a measure of uncertainty, averaged over all classes.

4.3.1 Human-in-the-Loop

In Section 6.4, we demonstrate a positive correlation between the proposed confidence score and the classification performance; the confidence score enables us to locate hard-to-classify data points, which we may label and use as additional fidelity. This naturally leads to the idea of using the confidence measure to intelligently select new fidelity points to achieve a better classification performance with limited human labeling effort. We design a human-in-the-loop system as follows (see Figure 4.1). We start with a small set of initial fidelity points and apply the Gibbs sampler to obtain a confidence score for the entire data set. We randomly sample, in a uniform fashion, additional candidate fidelity points with low confidence scores. The human in the loop then observes each of the candidate fidelity points to assign ground truth to them. We perform the UQ algorithm again to update the confidence scores and repeat the process until we reach the maximum number of fidelity points permitted (this



(a) Low confidence

(b) High confidence

Figure 4.2: Uncertainty quantification on the MNIST data set. Here $S(i)$ is the proposed confidence score. For each digit, we present four examples chosen from the top/bottom ten with the highest/lowest confidence scores within each class.

will be determined by the application). We observe in practice that adding data points with the lowest confidence scores does not benefit (sometimes even reduces) overall classification performance because these data points are often outliers. The significance of classifying these outliers correctly is scenario-dependent. In our experiments, we focus on the overall accuracy and do not sample fidelity from data points with confidence scores strictly below the tenth percentile.

4.4 Experiments

We perform UQ on (1) the MNIST data set [84], a handwritten digit data set, and (2) the HUJI EgoSeg data set [121], a body-worn video data set. Through these experiments, we illustrate some empirical properties of the confidence score (4.5); we demonstrate its correlation with the classification performance. We also put our human-in-the-loop system to test and showcase its ability to improve classification results upon uniformly randomly sampled training data.

4.4.1 MNIST

The MNIST data set [84] consists of 70,000 images of handwritten digits; each image is of the size 28×28 pixels. We choose uniformly at random 500 images each from the digits 1, 4, 7, and 9 to form a graph of 2000 nodes. We follow the graph construction procedure in [19]; each image is projected onto the lead 50 principal components yielding a 50-dimensional feature vector, and we construct a 15-nearest neighbor graph. The weighting constants r_{ij} are chosen according to [169]. For data point i, we compute the mean distance of its 15 nearest neighbors, denoted as r_i; then the weighting constant r_{ij} is given by $r_{ij} = r_i r_j$. We use the symmetrically normalized graph Laplacian (with $p = 1/2$ in (1.2)) and only its first 300 eigenvalues and eigenvectors. We perform the Gibbs sampler detailed in Section 4.1 with 3% uniformly randomly sampled fidelity points; we choose the noise variance to be $\gamma = 0.1$ and we draw 2×10^4 samples to estimate the uncertainty. We showcase examples of images with the highest or the lowest confidence scores in Figure 4.2. It is interesting to note that the lowest confidence score of the digit 1 is much higher than that of the other digits; we theorize that it is easier for the algorithm to differentiate 1 than the other three digits.

4.4.2 Body-Worn Videos

We also apply our method to the HUJI EgoSeg data set [121, 122]. This data set contains 65 hours of ego-centric videos, including 44 videos filmed using a head-mounted GoPro Hero3+, the Disney data set [42] and other YouTube videos[1]. In the recent paper [87], a graph-based semi-supervised learning method was applied to this data set to classify video segments according to camera-wearers' activities and showed promising results. This data set consists of footage of 7 activities: *Walking, Driving, Riding Bus, Biking, Standing, Sitting, and Static*. We follow the same feature extraction procedure described in [87] to obtain a 50-dimensional feature vector for every 4-second video segments; this yields 36,421 segments. To speed up our calculation, we sample every fifth segment. We construct the graph from

[1] This data set can be downloaded from http://www.vision.huji.ac.il/egoseg/.

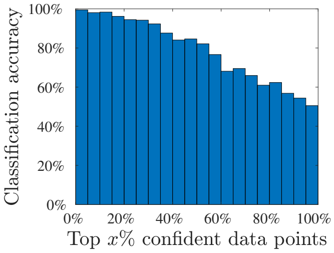
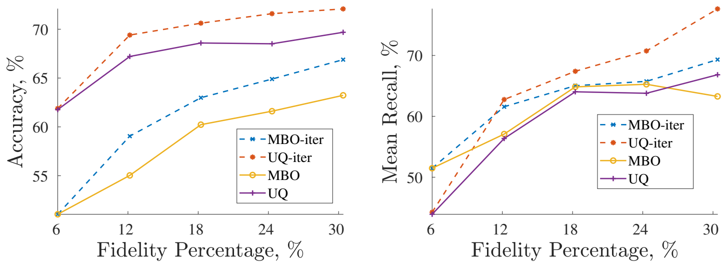


Figure 4.3: Classification accuracy on data points with top x% to $(x+5)$% confidence scores on the HUJI EgoSeg data set. We group data points based on their confidence score; each group contains 5% of data points and we evaluate the classification accuracy on each group.

the 50-dimensional feature vectors, and choose the constants $r_{ij} = r_i r_j$ according to [169], where r_i is the distance of the 40th nearest neighbor of node i. We employ the symmetrically normalized graph Laplacian and keep its first 400 eigenvalues and eigenvectors. We compute the eigenvectors using a low-rank approximation of the graph Laplacian via the Nystrom extension [47]. We apply the Gibbs sampler with $\gamma = 0.1$ and 2×10^4 iterations.

We separate the data set into a training and testing set, which are disjoint sets of videos; the training set contains around 65% of the data, measured in terms of the footage length. We refer readers to [122] for the details of the experimental protocol. However, we do not use the full training set, but instead take a portion of it as the fidelity; we train the model with the set of fidelity points. All classification performances are evaluated on the testing set only. We first investigate the correlation between the confidence score and the classification accuracy. We perform UQ with 12% of the training set. Recall that the classification is produced by taking the largest entry of the posterior mean $s(i)$ for each data point i. In Figure 4.3, we plot the classification accuracy of the top x% to $(x+5)$% confident data points for each $x \in \{0, 5, 10, \cdots, 95\}$. We observe that the classification is more accurate on data points with higher confidence scores. We also test our human-in-the-loop system on this data set. We start with 6% fidelity data and gradually increase the fidelity percentage to 30% over five iterations; at each iteration, we introduce additional 6% fidelity points sampled randomly in a uniform fashion from data points with confidence scores in the range



(a) Accuracy

(b) Recall

Figure 4.4: Classification performance of UQ and an MBO classifier using iteratively generated fidelity (UQ/MBO-iter) and uniformly randomly sampled fidelity (UQ/MBO) on the HUJI EgoSeg data set.

of the 10th and 50th percentile. We perform UQ as well as a graph-based semi-supervised learning method (an MBO scheme [17]) using the same set of fidelity points. We compare the classification performance, measured in terms of accuracy and mean recall averaged over seven activities, of both classifiers using iteratively generated fidelity against the same classifiers using uniformly randomly sampled fidelity. The results are presented in Figure 4.4. We observe that both classifiers benefit from the intelligently sampled fidelity — utilizing the confidence score produced by UQ — in terms of producing higher accuracy and mean recall than using uniformly randomly sampled fidelity.

4.5 Conclusion

In this chapter, we study UQ in a graph-based multi-class SSL problem . We generalize the probit model, originally proposed for the binary classification problem in [19], to the multi-class case. We propose a Gibbs sampler to sample from the posterior distribution and a confidence score that connects to the posterior variance. Through our experiments on the MNIST data set, we demonstrate that the proposed confidence score is easy to interpret; it is clear to see the contrast between the digit images with low confidence scores and ones with high confidence scores. The proposed confidence score also exhibits a correlation with the classification performance in our experiments on the HUJI EgoSeg data set. Based on these observations, we design a human-in-the-loop system to efficiently use human labeling effort to improve classification results. We test this system on the HUJI EgoSeg data set and observe that the classifiers that we study produce improved classification using the human-in-the-loop system than the same classifiers using uniformly randomly sampled fidelity.

CHAPTER 5

Active Learning with Probit Likelihood via Gaussian Approximations

5.1 Background

This chapter is a version of [103]. This work was done in collaboration with Kevin Miller under the supervision of Andrea Bertozzi. Kevin Miller proposed the novel active learning strategy and I contributed the theory and algorithm that enables us to execute the strategy efficiently. Together, we designed and conducted the experiments.

Active learning in SSL seeks to intelligently select training data to optimize the overall classification performance. We focus on *pool-based* active-learning paradigm, as opposed to online or streaming-based active learning [132]. That is, an active learner has access to a fixed "pool" of unlabeled data points from which it can decide the next training point. We consider querying only a single point at a time, as opposed to *batch-mode* active learning [62]. We assume the binary-classification case, in which the labels reside in $y_j \in \{\pm 1\}$ (or $\{0, 1\}$). In pool-based active learning, most methods alternate between: (1) training a model given the current labeled data $Z', \{y_j\}_{j \in Z'}$ and (2) choosing an active-learning query point in the unlabeled set $(Z')^c$ according to an acquisition function. We can classify most methods into a few categories: uncertainty [51, 65, 132], margin [10, 68, 149], clustering [34, 95], and look-ahead [27, 174] acquisition functions. Active-learning methods have been proposed for graph-based SSL models, which use a similarity graph to represent the geometric relationships between points in the data set, such as Gaussian random field (GRF) models [17, 19, 171]. Active learners implementing look-ahead expected risk [69, 174], model posterior covariance [67, 94], and other measures of uncertainty [80] have been produced for the GRF model of [171]. The conditional distribution of this foundational GRF model is a harmonic function on the graph and hence is referred to as the harmonic functions (HF) model.

Our contributions on this subject are (1) provide a unifying framework for active learning in many graph-based SSL models, (2) introduce an adaptation of non-Gaussian Bayesian models to allow efficient calculations previously done only on Gaussian models, and (3) introduce a novel "model-change" active-learning acquisition function built around our adaptation.

5.2 Graph-Based SSL Models

In this chapter, we only consider the binary-classification problem. Define a real-valued function on the nodes of the graph $u : Z \to \mathbb{R}$, $\mathbf{u} \in \mathbb{R}^N$ whose values reflect the classification of the data points. Given the current labeled set Z', one seeks the solution to the optimization problem

$$\mathbf{u}^* = \underset{\mathbf{u}\in\mathbb{R}^N}{\arg\min}\, \frac{1}{2}\langle \mathbf{u}, L_\tau \mathbf{u}\rangle + \sum_{i\in Z'} \ell(u_i, y_i) := \underset{\mathbf{u}\in\mathbb{R}^N}{\arg\min}\, \mathcal{J}_\ell(\mathbf{u};\mathbf{y}), \tag{5.1}$$

where $\ell : \mathbb{R} \times \mathbb{R} \to [0, \infty)$ is a chosen loss function and $\mathbf{y} \in \mathbb{R}^{|Z'|}$ is a vector of labels y_j. Common loss functions include $\ell(x, y) = (x - y)^2/2\gamma^2$ and $\ell(x, y) = -\log \Psi_\gamma(xy)$, where $\Psi_\gamma(t) = \int_{-\infty}^{t} \psi_\gamma(s)ds$ is the cumulative distribution function (CDF) of a log-concave probability density function (PDF) $\psi_\gamma(s)$.

This variational perspective has a probabilistic counterpart, from which Bayesian statistical methods can provide useful ways for devising well-principled acquisition functions. We can view the objective function in (5.1) as the negative log of an associated Bayesian posterior distribution; namely, the posterior distribution is proportional to

$$\exp\left(-\mathcal{J}_\ell(\mathbf{u};\mathbf{y})\right).$$

In the case of $\ell(x, y) = (x - y)^2/2\gamma^2$, we model the likelihood of observations $\mathbf{y}|\mathbf{u}$ by

$\mathcal{N}(H\mathbf{u}, \gamma^2 I_{|Z'|})$ where we recall that $H : \mathbb{R}^N \to \mathbb{R}^{|Z'|}$ is the projection of $\mathbf{u}$ onto the labeled indices Z'. This likelihood is Gaussian and therefore the posterior $\mathbb{P}(\mathbf{u}|\mathbf{y})$ is Gaussian $N(\mathbf{m}, C^*)$, with covariance $C^* = \left(L_\tau^{-1} + H^T H/\gamma^2\right)^{-1}$ and mean $\mathbf{m} = C^* H^T \mathbf{y}/\gamma^2$. We refer to this as the Gaussian regression (GR) model. The Gaussian structure of this posterior distribution allows us to efficiently calculate the posterior mean and covariance, including look-ahead calculations that is detailed in Section 5.2.2. Although the prior is Gaussian, the posterior distribution for general loss functions ℓ is not necessarily Gaussian. The key idea behind our method is to approximate a non-Gaussian distribution with a suitable Gaussian distribution to exploit the efficient calculations of the look-ahead posterior mean and covariance. This more general formulation allows us to use more realistic models for classification than just regression. An example of such a non-Gaussian posterior occurs when the loss function is $\ell(x, y) = -\log \Psi_\gamma(xy)$. In this case, the likelihood is derived from the model $y_j = \text{Sign}(u_j + \eta_j)$, where $\eta_j \sim \psi_\gamma$ [61]. We refer to this as the Probit model.

Some common acquisition functions originally derived for Gaussian models are

(1) MBR [174] $j_{\text{MBR}} = \arg\min_{j \in (Z')^c} \mathbb{E}_{y_j|\mathbf{m}} \left[\sum_{i=1}^N \text{Err}(i, \mathbf{m}^{j,y_j})\right]$

(2) VOpt [67] $j_V = \arg\max_{j \in (Z')^c} \frac{1}{\gamma^2 + C^*_{j,j}} \|C^*_{:,j}\|_2^2$

(3) ΣOpt [94] $j_\Sigma = \arg\max_{j \in (Z')^c} \frac{1}{\gamma^2 + C^*_{j,j}} \langle \mathbf{1}, C^*_{:,j} \rangle$

where $\text{Err}(i, \mathbf{m}^{j,y_j})$ is the estimated risk on the ith data point of the look-ahead mean $\mathbf{m}^{j,y_j}$. These acquisition functions were originally defined on the HF model [174], but have been generalized here to fit the GR model. To recover the HF model's acquisition functions, let $\gamma = 0$, $y_j \in \{0, 1\}$, and the posterior covariance C^* be defined only on the unlabeled nodes per the conditional nature of the HF model.

5.2.1 Laplace Approximation of the Probit Model

The Laplace approximation is a popular technique for approximating non-Gaussian distributions with a Gaussian distribution [127]. We approximate the Probit posterior with the

Gaussian distribution:

$$\hat{\mathbb{P}}(\mathbf{u}|\mathbf{y}) = \mathcal{N}(\hat{\mathbf{u}}, \hat{C}), \ \ \hat{\mathbf{u}} = \operatorname*{arg\,min}_{\mathbf{u}\in\mathbb{R}^N} \mathcal{J}_\ell(\mathbf{u};\mathbf{y}), \ \ \hat{C} = \left(\nabla\nabla\mathcal{J}_\ell(\mathbf{u};\mathbf{y})|_{\mathbf{u}=\hat{\mathbf{u}}}\right)^{-1}. \tag{5.2}$$

The mean of this Gaussian distribution $\hat{\mathbf{u}}$ is the MAP estimator of the true Probit posterior. This Gaussian distribution is in a form in which we can apply adaptations of acquisition functions of GR and HF models, such as VOpt [67], Σ-Opt [94], and MBR [174]. The Laplace approximations of the GR and HF models are themselves, because the mean and MAP estimator (i.e. mode) are the same for Gaussian distributions. Furthermore, this Laplace approximation of non-Gaussian posterior distributions incorporates labeling information that is not contained in the GR and HF models' covariance matrices.

5.2.2 Look-Ahead Updates

Acquisition functions such as MBR need a look-ahead model with index j and label y_j:

$$\operatorname*{arg\,min}_{\mathbf{u}\in\mathbb{R}^N} J^j(\mathbf{u};\mathbf{y}, y_j) := \operatorname*{arg\,min}_{\mathbf{u}\in\mathbb{R}^N} \frac{1}{2}\langle \mathbf{u}, L_\tau \mathbf{u}\rangle + \sum_{i\in Z'} \ell(u_i, y_i) + \ell(u_j, y_j).$$

This is simply the updated graph-based SSL problem, having added the index k and associated label y_k to the labeled data. As mentioned previously in Section 5.2, one convenience of Gaussian models is that we can solve for the look-ahead posterior distribution's parameters from the current posterior distribution without expensive model retraining. This is a crucial property for computing acquisition functions like MBR [174], which consider the effects of adding an index j with label y_j to the labeled data. There is no simple, closed-form solution for computing the look-ahead MAP estimator $\hat{\mathbf{u}}^{j,y_j}$ from the current $\hat{\mathbf{u}}$ in the Probit model (5.2) because of the loss function $-\ln\Psi_\gamma(xy)$. We approximate the look-ahead update $\tilde{\mathbf{u}}^{j,y_j}$ by computing a single step of Newton's Method on the look-ahead objective $\mathcal{J}^j(\mathbf{u};\mathbf{y}, y_j)$, starting with the current MAP estimator $\hat{\mathbf{u}}$:

$$\tilde{\mathbf{u}}^{j,y_j} = \hat{\mathbf{u}} - \left(\nabla\nabla\mathcal{J}^j(\hat{\mathbf{u}};\mathbf{y}, y_j)\right)^{-1}\left(\nabla\mathcal{J}^j(\hat{\mathbf{u}};\mathbf{y}, y_j)\right) = \hat{\mathbf{u}} - \frac{F(\hat{u}_j, y_j)}{1 + \hat{C}_{j,j}F'(\hat{u}_j, y_j)}\hat{C}_{:,j}, \tag{5.3}$$

where F, F' are the first and second derivatives of the loss function with respect to the first argument. We call this single step of Newton's method as a Newton approximation (NA) update. This is a rank-one update of the MAP estimator. The update requires storing the posterior covariance matrix $\hat{C}$; this is needed for all the aforementioned Gaussian-based acquisition functions, in this context. Due to the second-order nature of Newton's method, we find that this NA update $\tilde{\mathbf{u}}^{j,y_j}$ empirically is a good approximation of the true look-ahead MAP estimator $\hat{\mathbf{u}}^{j,y_j}$. We also derive a NA posterior covariance update:

$$\hat{C}^{j,y_j} = \left(\nabla\nabla\mathcal{J}^j(\hat{\mathbf{u}}^{j,y_j}; \mathbf{y}, y_j)\right)^{-1} \approx \hat{C} - \frac{F'(\tilde{u}_j^{j,y_j}, y_j)}{1 + \hat{C}_{j,j}F'(\tilde{u}_j^{j,y_j}, y_j)}\hat{C}_{:,j}\hat{C}_{:,j}^T =: \tilde{C}^{j,y_j}. \tag{5.4}$$

With these NA updates, we can straightforwardly apply the Gaussian-based acquisition functions to our approximation (5.2) of the Probit model. Furthermore, retraining models on new training data is approximated by using these NA updates of the MAP estimator and posterior covariance, as we demonstrate in Section 5.3.

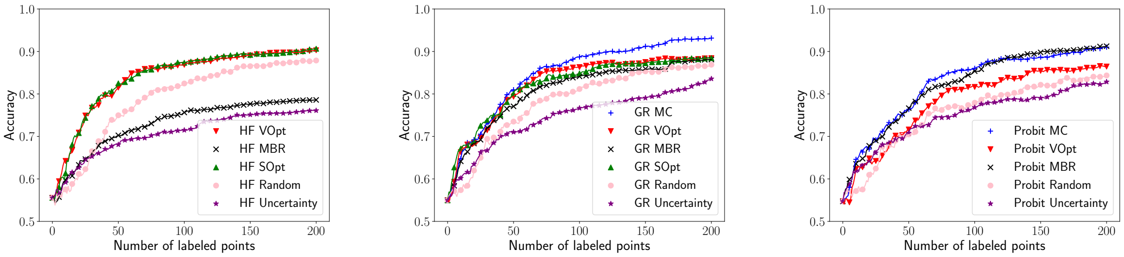
5.2.3 Model Change (MC) Acquisition Function

Approximating the change in a model (i.e. classifier) from the addition of an index j and associated label y_j has been investigated previously [27,72]. Employing our NA update (5.3), we propose an MC acquisition function for our approximated Probit model in a max-min framework:

$$j_{MC-P} = \operatorname*{arg\,max}_{j\in(Z')^c} \min_{y_j\in\{\pm1\}} \left|\hat{\mathbf{u}} - \hat{\mathbf{u}}^{j,y_j}\right| \approx \operatorname*{arg\,max}_{j\in(Z')^c} \min_{y_j\in\{\pm1\}} \left|\frac{F(\hat{u}_j, y_j)}{1 + \hat{C}_{j,j}F'(\hat{u}_j, y_j)}\hat{C}_{:,j}\right|.$$

5.3 Results

We present numerical results demonstrating our Gaussian approximations and subsequent NA updates in the Probit model on a synthetic data set (Checkerboard [174]) and a real-world data set (MNIST [84]). In each of the HF, GR, and Probit models, we show the performance of the MC method of Section 5.2.3, VOpt [67], MBR [174], uncertainty [132], and sampling new training data uniformly at random. We use "model–method" to indicate a model and acquisition function combination (e.g. GR–Vopt is referring to using Vopt acquisition function on the GR model). We calculate the average accuracies over five trials



(a) HF

(b) GR

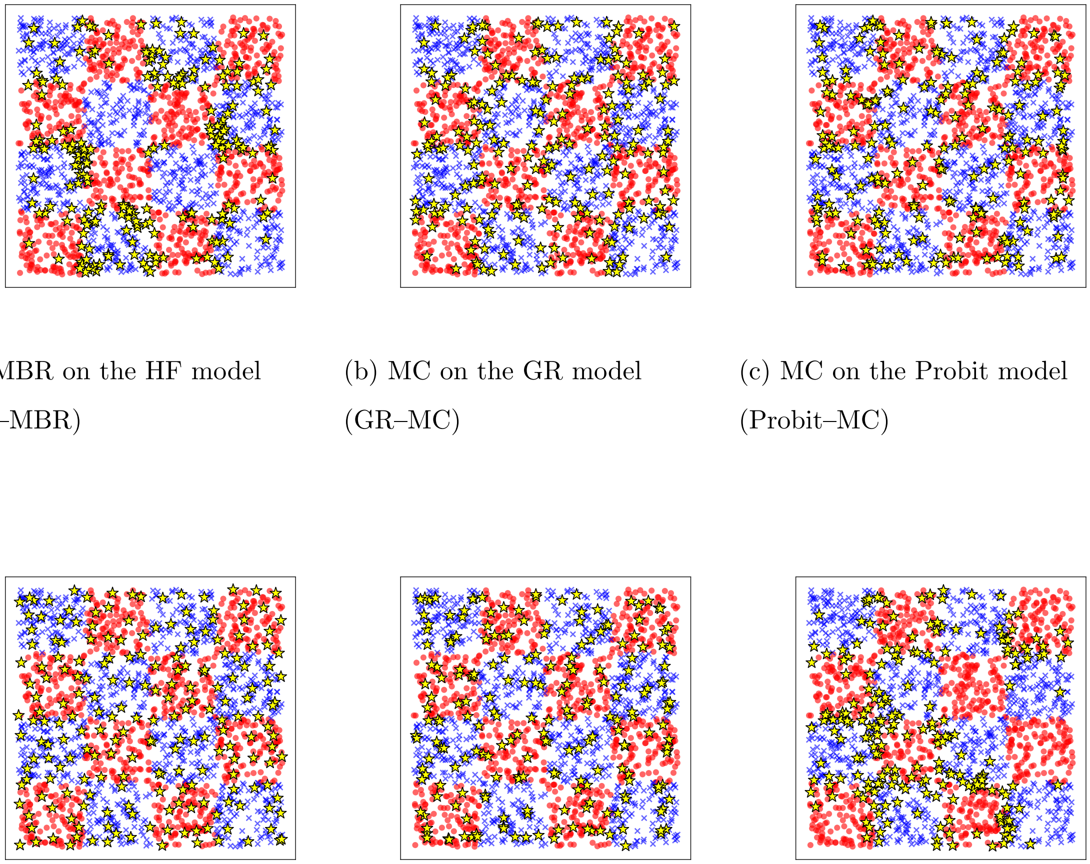
(c) Probit

Figure 5.1: Classification performances of different combinations of active-learning acquisition functions and classification models on a checkerboard data set.

according to the underlying SSL classifier of the acquisition function. After comparing accuracies across all methods with a common classifier (of the Probit model), we find that each method's query choices improve the accuracy of its underlying classifier. In Figure 5.3, we demonstrate how closely the NA updates $\tilde{\mathbf{u}}^{j,y_j}, \tilde{C}^{j,y_j}$ ((5.3), (5.4)) approximate the active learning choices from retraining the model (i.e. $\hat{\mathbf{u}}^{j,y_j}, \hat{C}^{j,y_j}$).

5.3.1 Checkerboard Data Set

The checkerboard data set [174] consists of $2,000$ points uniformly sampled on the unit square $[0,1]^2 \subset \mathbb{R}^2$, and we divide into two classes based on a 4×4 checkerboard pattern. For each of the five trials, we choose ten points uniformly at random to label initially (five from each class), and then sequentially choose 200 query points via our list of acquisition functions. Similar to [80], we showcase this data set because successful active learning in this data set requires properly "exploring" the many different clusters as well as "exploiting" the learned decision boundaries efficiently. The best performing methods are the MC methods in the GR and Probit models, as well as Probit–MBR. These methods not only identify each of the clusters in the grid (Figure 5.2b, 5.2c) but also explore the decision boundaries between clusters. In the Probit–Uncertainty(Figure 5.2f) and HF–MBR(Figure 5.2a), the methods have not explored the extent of the clustering structure and are not as accurate (Figure 5.1). Conversely, the VOpt acquisition function in each model only identifies points that are inside each of the clusters. As seen in Figure 5.2d and 5.2e, these acquisition functions have not explored the boundaries between the clusters and thus do not achieve as high of accuracy.



(a) MBR on the HF model (HF–MBR)

(b) MC on the GR model (GR–MC)

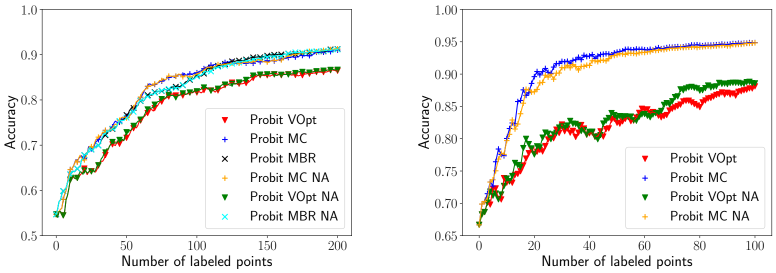
(c) MC on the Probit model (Probit–MC)

(d) Vopt on the HF model (HF–Vopt)

(e) Vopt on the GR model (GR-Vopt)

(f) Uncertainty on the Probit model (Probit–uncertainty)

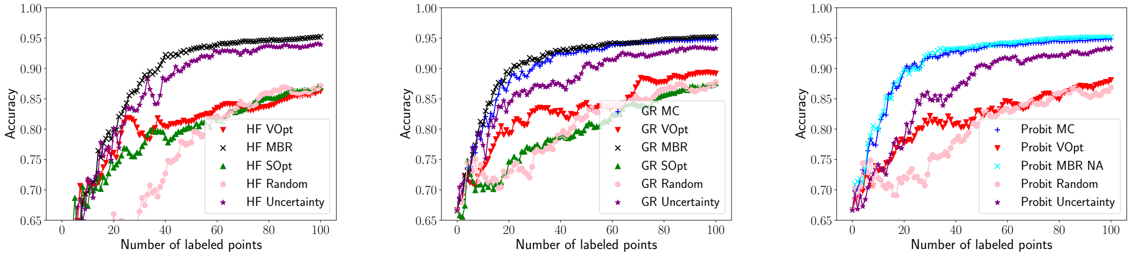
Figure 5.2: Acquisition function choices on a checkerboard data set. Yellow stars show the 200 points chosen by each of the given acquisition functions.



(a) Checkerboard

(b) MNIST

Figure 5.3: Accuracy comparison for query choices using the true posterior updates $\hat{\mathbf{u}}^{j,y_j}, \hat{C}^{j,y_j}$ compared to the NA updates $\tilde{\mathbf{u}}^{j,y_j}, \tilde{C}^{j,y_j}$. NA update denoted with "NA" in legend.



(a) HF

(b) GR

(c) Probit

Figure 5.4: Classification performances of different combinations of active-learning acquisition functions and classification models on the MNIST data Set.

5.3.2 MNIST

MNIST [84] is a data set of 70,000 grayscale 28×28 pixel images of handwritten digits (0–9). Each image is represented by a 784-dimensional vector $\mathbf{x}_i$ and we normalize the pixel values to range from 0 to 1. We form a set of 4,000 data points by choosing uniformly at random 400 images from each digit. We construct a 15-nearest-neighbor graph among the data points with weights $w_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/380^2)$. We consider the binary-classification problem of classifying even digits versus odd digits. For each of the five trials, we start with ten initial training points evenly distributed between the two classes (not necessarily among the digits) and use the active learners to query 100 points. The average classification accuracies are presented in Figure 5.4. Though the MBR methods perform the best, they are more costly to compute than our competitive MC acquisition functions.

5.4 Conclusion

Under this unifying Bayesian perspective of active learning in graph-based SSL, we use Laplace and Newton approximations to allow non-Gaussian models to employ acquisition functions previously only used in Gaussian models. We introduce a novel MC acquisition function that is both efficient to compute and provides competitive results. Future work may extend these results to batch-mode active learning, multi-class classification, and kernel methods other than graph-based SSL.

CHAPTER 6

Spatiotemporal Hawkes Processes and Network Reconstruction

6.1 Background

This chapter is a version of [168]. This work is done in collaboration with Baichuan Yuan under the supervision of Andrea Bertozzi, Jeffrey Brantingham, and Mason Porter. Baichuan Yuan contributed the algorithm and I am responsible for designing and conducting experiments on a synthetic data set and a Gowalla data set.

Digital devices such as smart phones and tablets generate a massive amount of spatiotemporal data about human activities, providing a wonderful opportunity for researchers to gain insight into human dynamics through our "digital footprints". A broad variety of human activities are analyzed using such data, creating new disciplines [83] such as computational social science and digital humanities. Examples of such activities include online check-ins in large cities [29], effects of human mobility [9] and currency flow [23] on the spread of contagious diseases, online communications during Occupy Wall Street [30], crime reports in Los Angeles county [78], and many others.

Network analysis is a powerful approach for representing and analyzing complex systems of interacting components [111], and network-based methods can provide considerable insights into the structure and dynamics of complex spatiotemporal data [13]. It has been valuable for studies of both digital human footprints and human mobility [11]. To give one recent example, Noulas et al. [113] studied geographic online social networks to illustrate similarities and heterogeneities in human mobility patterns.

Suppose that each node in a network represents an entity, and that the edges (which can be either undirected or directed, and can be either unweighted or weighted) represent spatiotemporal connections between pairs of entities. For instance, in a check-in data set from a social medium, one can model each user as a node, which has associated check-in time and locations. In this case, one can suppose that an edge exists between a pair of users if they follow each other on the social medium. One can use edge weights to quantify the amount of "influence" between users, where a larger weight signifies a larger impact. In our investigation, we assume that the relationships between nodes are time-independent.[1] In some cases, the entities and relationships are both known, and one can investigate the structure and dynamics of the associated networks. However, in many situations, network data is incomplete — with potentially a large amount of missing data, in the form of missing entities, interactions, and/or metadata [143] — and the relationships between nodes may not be directly observable [130]. For example, social-media companies attempt to infer friendship relationships among their users to provide accurate friendship recommendations for online social networks.

In the last few years, there has been a considerable amount of work on inferring missing data (both structure and weights) in networks. A basic approach for inferring relationships among entities is to calculate cross-correlations of their associated time series [82]. Another approach is to use coefficients from a generalized linear model (GLM) [109], a generalization of linear regression that allows response variables to have a non-Gaussian error-distribution. Recently, people have begun to use point-process methods [134] in network reconstruction. For example, Perry and Wolfe [119] modeled networks as a multivariate point process and then inferred covariate-based edges (both their existence and their weights) by estimating a point process. Among point-process models, it is very popular to use Hawkes processes (also known as self-exciting point processes) for studying human dynamics [49, 89]. Hawkes-process models are characterized by mutual "triggering" among events [114], as one event may increase the probability for subsequent events to occur. Such models can capture in-

[1] For other regimes of relative time scales between spatiotemporal processes and network dynamics, it is necessary to consider time-dependent edges [63, 123].

homogeneous inter-event times and causal (temporal) correlations, which have both been observed in human dynamics [76]. These properties make it a useful approach in social-network applications [71]. It thus seems promising to use such processes for network inference on dynamic human data, such as crime events or online social activity. For example, Linderman and Adams [89] proposed a fully-bayesian Hawkes model that they reported to be more accurate for their data at inferring missing edges than GLMs, cross-correlations, and a simple self-exciting point process with an exponential kernel. Very recently, self-exciting point processes were applied in [146] to reconstruct multilayer networks [75], a generalization of ordinary graphs. However, the aforementioned temporal point-process models are not without limitations. For example, most of these models do not use spatial information, even when it plays a significant role in a system's dynamics. Furthermore, many assume an a priori model [89] or a specific parametrization [148] for their point processes.

In this chapter, we consider a nonparametric and multivariate version of the spatiotemporal Hawkes process. Spatiotemporal Hawkes processes have been used previously to study numerous topics, including crime [105], social media [81], and earthquake prediction [48]. In our model, each node in a network is associated with a spatiotemporal Hawkes process. The nodes can "trigger" each other, so events that are associated with one node increase the probability that there will be events associated with the other nodes. We measure the extent of such mutual-triggering effects using a $U \times U$ "triggering matrix" $\mathbf{K}$, where U is the number of nodes. If one considers an exclusively temporal scenario, a point process u does not "cause" (in the Granger sense [56]) a point process v if and only if $\mathbf{K}(u, v) = 0$ [39]. Because triggering between point processes reflects an underlying connection, one can try to recover latent relationships in a network from $\mathbf{K}$. Such triggering should decrease with both distance and time according to some spatial and temporal kernels. In our work, instead of assuming exponential decay [49] or some other distribution [89, 148], we adopt a nonparametric approach [97] to learn both spatial and temporal kernels from data using an expectation-maximization-type (EM-type) algorithm [160].

We compare our approach with other recent point-process network-reconstruction methods [49, 89] on both synthetic and real-world data sets with spatial information. Our two examples of the latter data sets come from a location-based social-networking website and crime topics. We illustrate the importance both of incorporating spatial information and of using nonparametric kernels. Although we assume that the relationships among nodes are time-independent, our model still recovers a causal structure among events in synthetic data sets. We also build event-causality networks on data sets about violent crimes of gangs and examine gang retaliation patterns using motif analysis.

6.2 Self-Exciting Point Processes

A *point process* S is a random measure on a complete separable metric space that takes values on $\{0, 1, 2, \ldots\} \cup \{\infty\}$ [131]. We first consider a *temporal point process*, which consists of a list $\{t_1, t_2, \ldots, t_N\}$ of N time points, with corresponding events $1, 2, \ldots, N$. Let $S[a, b)$ denote the number of points (i.e., events) that occur in a finite time interval $[a, b)$, with $a < b$. One typically models the behavior of a simple temporal point process (multiple events cannot occur at the same time) by specifying its conditional intensity function $\lambda(t)$, which represents the rate at which events are expected to occur around a particular time t, conditional on the prior history of the point process before time t. Specifically, when $H_t = \{t_i | t_i < t\}$ is the history of the process up to time t, one defines the *conditional intensity function*

$$\lambda(t) = \lim_{\Delta t \downarrow 0} \frac{\mathbb{E}[S[t, t + \Delta t) | H_t]}{\Delta t} \,.$$

One important point-process model is a *Poisson process*, in which the number of points in any time interval follows a Poisson distribution and the number of points in disjoint sets are independent. A Poisson process is called *homogeneous* if $\lambda(t) \equiv$ constant and is thus characterized by a constant rate at which events are expected to occur per unit time. It is called *inhomogeneous* if the conditional intensity function $\lambda(t)$ depends on the time t (e.g., $\lambda(t) = \sin(t)$). In both situations, the numbers of points (i.e., events) in disjoint intervals are independent random variables.

We now discuss self-exciting point processes, which allow one to examine a notion of causality in a point process. If we consider a list $\{t_1, t_2, \ldots, t_N\}$ of time stamps, we say that a point process is *self-exciting* if

$$\mathrm{Cov}\left[S(t_{k-1}, t_k), S(t_k, t_{k+1})\right] > 0 \quad \text{for } k \text{ such that } \quad t_{k-1} < t_k < t_{k+1}\,.$$

That is, if an event occurs, another event becomes more likely to occur locally in time.

A *univariate temporal Hawkes process* has the following conditional intensity function:

$$\lambda(t) = \mu(t) + K \sum_{t_k < t} g(t - t_k)\,, \tag{6.1}$$

where the background rate $\mu(t)$ can either be a constant or a time-dependent function that describes how the likelihood of some process (crimes, e-mails, tweets, and so on) evolves in time. For example, violent crimes are more likely to happen at night than during the day, and business e-mails are less likely to be sent during the weekend than on a weekday. One can construe the rate $\mu(t)$ as a process that designates the likelihood of an event to occur, independent of the other events. The summation term in (6.1) describes the self-excitation: past events increase the current conditional intensity. The function $g(t)$ is called the *triggering kernel*, and the parameter K denotes the mean number of events that are triggered by an event. One standard example is a Hawkes process with an exponential kernel $g(t) = \omega e^{-\omega t}$, where ω is a constant decay rate for the triggering kernel that controls how fast the rate $\lambda(t)$ returns to its baseline level $\mu(t)$ after an event occurs.

6.2.1 Temporal Multivariate Models

In network reconstruction, one seeks to infer the relationships (i.e., edges) and the strengths of such relationships (i.e., edge weights) among a set of entities (i.e., nodes). When modeling the relationships in a network, it is more appropriate to use a multivariate point process than a univariate one. In a temporal multivariate point process, there are U different point processes $(S_u)_{u=1,\ldots,U}$, and the corresponding conditional intensity functions are $(\lambda_u(t))_{u=1,\ldots,U}$. We seek to infer the intensity functions from observed data $(t_j, u_j)_{j=1,\ldots,N}$ in a time window $[0, T]$, where t_j and u_j, respectively, are the time and point-process index of event j. There are numerous applications of temporal multivariate point processes, such as financial markets [8], real-time crime forecasting [163] and neural spike trains [24]. Here we focus on the specific application of network reconstruction.

A trivial example of a multivariate point process is the multivariate Poisson process, in which each point process is a univariate Poisson process. Another example is the multivariate Cox process, which consists of doubly stochastic Poisson processes in which the conditional intensity itself is a stochastic process. Perry and Wolfe [119] used a Cox process to model e-mail interactions (edges) among a set of users (nodes). Neither the multivariate Poisson nor the multivariate Cox process are self-exciting.

Instead of modeling edges as Cox processes, Fox et al. [49] used multivariate Hawkes processes to model people (nodes) communicating with each other via e-mail. Their conditional intensity function has an exponential kernel and a nonparametric background function $\mu_u(t)$ for each person (process) u:

$$\lambda_u(t) = \mu_u(t) + \sum_{t_i<t} K_{u_i u} \omega e^{-\omega(t-t_i)} \,, \tag{6.2}$$

where $K_{uv} = \mathbf{K}(u,v)$ is the expected number of events of person v that are triggered by one event of person u. One can estimate the set of parameters Θ by minimizing the negative log-likelihood function

$$-\log(L(\Theta)) = -\sum_{k=1}^{N} \log(\lambda_{u_k}(t_k)) + \sum_{u=1}^{U} \int_0^T \lambda_u(t) \mathrm{d}t \,. \tag{6.3}$$

Recall that u_k is the point process associated with event k.

There are several variants of the multivariate Hawkes process. One is to add regularization terms to (6.3) to improve the accuracy of parameter estimation. Lewis and Mohler [86] used maximum-penalized likelihood estimation, which enforces some regularity on the model parameters, to infer Hawkes processes. Zhou et al. [170] extended this idea and promoted the low-rank and sparsity properties of $\mathbf{K}$ by adding nuclear and L_1 norms of $\mathbf{K}$ to (6.3) with the conditional intensity function $\lambda_u(t)$ from (6.2). Linderman et al. [89] added random-graph priors on $\mathbf{K}$ and developed a fully Bayesian multivariate Hawkes model. See [96] for theoretical guarantees on inferring Hawkes processes with a regularizer. Another research direction is to speed up the parameter estimation of point-process models. For example, Hall et al. [57] tried to learn the triggering matrix $\mathbf{K}$ via an online learning framework for streaming data. Instead of using a likelihood-based method, Achab et al. [2] developed a fast moment-matching method to estimate the matrix $\mathbf{K}$.

6.2.2 Spatiotemporal Point Processes

Many real-world data sets include not only time stamps but also accompanying spatial information, which can be particularly important for correctly inferring and understanding the associated dynamics [13]. In earthquakes, for example, most aftershocks usually occur geographically near the main shock [115]. In online social media, if two people often check in at the same location at closely proximate times, there is more likely to be a connection between them than if such "joint check-ins" occur rarely [29]. These situations suggest that it is important to examine spatiotemporal point processes, rather than just temporal ones. Indeed, there are myriad applications of spatiotemporal Hawkes processes, including crime prediction [105], seismology [115], and Twitter topics [81]. The successful employment of such processes in earthquake prediction and predictive policing [106] have helped inspire our work, in which we extend these ideas to network reconstruction.

We characterize a spatiotemporal point process $S(t, x, y)$ via its conditional intensity $\lambda(t, x, y)$, which is the expected rate of the accumulation of points around a particular spatiotemporal location. Given the history $\mathcal{H}_t$ of all points up to time t, we write

$$\lambda(t, x, y) = \lim_{\Delta t, \Delta x, \Delta y \downarrow 0} \left(\frac{\mathbb{E}\left[S\{(t, t+\Delta t) \times (x, x+\Delta x) \times (y, y+\Delta y)\} | \mathcal{H}_t\right]}{\Delta t\, \Delta x\, \Delta y} \right) .$$

For the purpose of modeling earthquakes, [115] used a self-exciting point process with a conditional intensity of the form

$$\lambda(t, x, y) = \mu(x, y) + \sum_{t > t_i} g(x - x_i, y - y_i, t - t_i) .$$

In this setting, if an earthquake occurs, aftershocks are more likely to occur locally in time and space. The choice of the triggering kernel $g(t, x, y)$ is inspired by physical properties of earthquakes. For example, [115] used a modified Omori formula (a power law) [114] to describe the frequency of aftershocks per unit time. In sociological applications, there is no direct theory to indicate appropriate choices for the kernel function. Some researchers have chosen specific kernels (e.g., exponential kernels) that are easy to compute. For example, Tita et al. [148] used a spatiotemporal point process to infer missing information about event participants. They modeled interactions between event participants as a combination of a spatial Gaussian mixture model and a temporal Hawkes process with an exponential kernel. A key problem is how to justify kernel choices in specific applications.

6.3 Spatiotemporal Models for Network Reconstruction

Many network-reconstruction methods using self-exciting point processes, such as [49, 89], have inferred time-independent relationships (i.e., edges) among entities (i.e., nodes) with corresponding (exclusively) temporal point processes. Entity (process) u is adjacent to v if $\mathbf{K}(u, v) > 0$, where one estimates the triggering matrix $\mathbf{K}$ from the data. Entity u is not adjacent to v if entity u's point process does not cause entity v's point process in time (in the Granger sense [39]). For many problems, it is desirable — or even crucial — to incorporate spatial information [13, 32]. For example, spatial information is an important part of online fingerprints in human activity, and it has a significant impact on most other social networks. In crime modeling, for example, there is a "near repeat" phenomenon in crime locations, indicating the necessity of including spatial information. Specifically, the spatial neighborhood of an initial burglary has a higher risk of repeat victimization than more-distant locations [133]. In our work, we propose multivariate spatiotemporal Hawkes processes to infer relationships in networks and provide a novel approach for analyzing spatiotemporal dynamics.

Another important issue is the assumptions on triggering kernels for a Hawkes process. In seismology, for example, researchers attempt to use an underlying physical model to help determine a good kernel. However, it is much more difficult to validate such models in social networks than for physical or even biological phenomena [124]. The content of social data is often unclear, and there is often little understanding of the underlying mechanisms that produce them. With less direct knowledge of possible triggering kernels, it is helpful to employ a data-driven method for kernel selection. Using a kernel with an inappropriate decay rate may lead to either underestimation or overestimation of the elements in the triggering matrix $\mathbf{K}$, which may also include false negatives or positives in the inferred relationships between entities. Therefore, we ultimately use a nonparametric approach to learn triggering kernels in various applications to avoid a priori assumptions about a specific parametrization.

A multivariate spatiotemporal Hawkes process is a sequence $\{(t_i, x_i, y_i, u_i)\}_{i=1}^{N}$ with N events, where t_i and (x_i, y_i) are spatiotemporal stamps and u_i is the point-process index of event i. Each of the U nodes is a *marginal process*. The conditional intensity function for node u is

$$\lambda_u(t, x, y) = \mu_u(x, y) + \sum_{t > t_i} K_{u_i u} g(x - x_i, y - y_i, t - t_i)\,. \tag{6.4}$$

The above Hawkes process assumes that each node u has a background Poisson process that is constant in time but inhomogeneous in space with conditional intensity $\mu_u(x, y)$. There is also self-excitation, as past events increase the likelihood of subsequent events. We quantify the amount of impact that events associated with node u_i have on subsequent events of node u_j with a spatiotemporal kernel and the element $\mathbf{K}(u_i, u_j) = K_{u_i u_j}$ of the triggering matrix.

6.3.1 A Parametric Model

We first propose a multivariate Hawkes process with a specific parametric form. We use this model to generate spatiotemporal events on synthetic networks and provide a form of "ground truth" that we can use later.

The background rate μ_u and the triggering kernel g for (6.4) are given by

$$g(x, y, t) = g_1(t) \times g_2(x, y) = \omega \exp\left(-\omega t\right) \times \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)\,,$$

$$\mu_u(x, y) = \sum_{i=1}^{N} \frac{\beta_{u_i u}}{2\pi\eta^2 T} \times \exp\left(-\frac{(x - x_i)^2 + (y - y_i)^2}{2\eta^2}\right)\,.$$

For simplicity, we use exponential decay in time [114] and a Gaussian kernel in space [104]. We let T denote the time window of a data set; $K_{u_i u}$ denote the mean number of the events in process u that are triggered by each event in the process u_i; the quantity $\beta_{u_i u}$ denote the extent to which events in process u_i contribute to the background rate for events in the process u; and σ and η, respectively, denote the standard deviations in the triggering kernel and background rate. The value of σ determines the spreading scale of the triggering effect in space.

6.3.2 A Nonparametric Model

With the conditional intensity given in (6.4), we estimate the triggering kernel $g(x, y, t) = g_1(t) \times g_2(x, y)$ nonparametrically using histogram estimators [97]. We assume that g_2 is isotropic, which entails that $g_2(x, y) = g_2(r)$, where $r = \sqrt{x^2 + y^2}$. We let $h(r)$ be the spatial triggering kernel in the polar coordinate: $h(r) = 2\pi r g_2(r)$. We extend the background rate that was proposed in [48] to the multivariate case and write

$$\mu_u(x, y) = \gamma_u \tau(x, y) = \frac{\gamma_u}{T} \sum_{i=1}^{N} \frac{p_{ii}}{2\pi d_i^2} \exp\left(-\frac{(x - x_i)^2 + (y - y_i)^2}{2d_i^2} \right), \tag{6.5}$$

where γ_u is the background intensity of process u and p_{ii} is the probability that event i is a background event (i.e., it is not triggered by any event). We compute d_i by determining the radius of the smallest disk centered at (x_i, y_i) that includes at least n_p other events and is at least as large as some small value ϵ that represents the error in location.

Once we fit the model to spatiotemporal data, the triggering matrix $\mathbf{K}$ gives our inferences for the underlying relationships between entities. For two entities u and v, the matrix element $\mathbf{K}(u, v)$ indicates a mixture of temporal causality and spatial dependence between them. In inferring latent relationships in a network, we assume that entity u is not related to v if $\mathbf{K}(u, v) = 0$. We threshold the matrix $\mathbf{K}$ at a certain level: we set elements that are smaller than the threshold value to 0 and either maintain the values of larger or equal elements to obtain a weighted network or set them to 1 to produce an unweighted network. We use $\tilde{\mathbf{K}}$ to denote the thresholded matrix $\mathbf{K}$. We interpret that there is no relation between two nodes u and v if $\tilde{\mathbf{K}}(u, v) = \tilde{\mathbf{K}}(v, u) = 0$.

6.3.3 Model Estimation

We use an EM-type algorithm [160] to estimate the parameters and kernel functions of our model. This EM-type algorithm gives us an iterative method to find maximum-likelihood estimates of the parameters. We assume that the original model depends on unobservable latent variables. Suppose that we have data X and want to estimate parameters Θ. One can view the likelihood function $L(\Theta; X)$ as the marginal likelihood function of $L(\Theta; Y, X)$, where Y is a latent variable. We call $L(\Theta; Y, X)$ the "complete-data likelihood function" and $L(\Theta; X)$ the "incomplete-data likelihood function". Because both Y and $L(\Theta; Y, X)$ are random variables, we cannot estimate them directly. Therefore, we consider the following expectation function:

$$\begin{aligned} Q(\Theta, \Theta^{i-1}) &= \mathbb{E}\left[\log(L(\Theta; Y, X))|X, \Theta^{i-1}\right] \\ &= \int \log(L(\Theta; Y, X)) f(Y|X, \Theta^{i-1}) \mathrm{d}Y\,, \end{aligned} \tag{6.6}$$

where $f(Y|X, \Theta^{i-1})$ is the probability density function of Y, given the data X and Θ^{i-1}. We update parameters by solving the following equation:

$$\hat{\Theta}^i = \arg\max_{\Theta} Q(\Theta, \Theta^{i-1})\,.$$

6.3.3.1 Parametric Model

The log-likelihood for the parametric model defined in (6.4) in a spatial region R and time window $[0, T]$ is

$$\log(L(\Theta; X)) = \sum_{k=1}^{N} \log(\lambda_{u_k}(t_k)) - \sum_{u=1}^{U} \iint_R \int_0^T \lambda_u(t)\,\mathrm{d}t\,\mathrm{d}x\,\mathrm{d}y\,. \tag{6.7}$$

We define random variables Y_{ij} and Y_{ij}^b using the approach from [104]. If event j triggers event i via the kernel g, then $Y_{ij} = 1$; otherwise, $Y_{ij} = 0$. The equality $Y_{ij}^b = 1$ indicates that event i is triggered by event j at a background rate of μ. We define two expectation matrices $\mathbf{P}(i,j) = p_{ij} = \mathbb{E}[Y_{ij}]$ and $\mathbf{P}^b(i,j) = p_{ij}^b = \mathbb{E}[Y_{ij}^b]$. We convert the incomplete-data log-likelihood function in (6.7) into the following complete-data log-likelihood function:

$$\log(L(\Theta; X, Y)) = \sum_{j<i} Y_{ij} \log\left(K_{u_i u_j} g(t_i - t_j, x_i - x_j, y_i - y_j)\right) - \sum_{u=1}^{U} \sum_{i=1}^{N} \beta_{uu_i} \\ - \sum_{u=1}^{U} \sum_{i=1}^{N} K_{u_i u}\left(1 - e^{-w(T-t_i)}\right) + \sum_{i=1}^{N} \sum_{j=1}^{N} Y_{ij}^{b} \log(\mu_{u_i})\,.$$

We then calculate the expectation function using (6.6) to obtain

$$Q(\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{N} p_{ij}^{b} \log\left(\frac{\beta_{u_j u_i}}{2\pi\eta^2 T} \exp\left(-\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{2\eta^2}\right)\right) - \sum_{u=1}^{U} \sum_{i=1}^{N} \beta_{u_i u} \\ + \sum_{j<i} p_{ij} \log\left(\omega K_{u_j u_i} e^{-\omega(t_i - t_j)} \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{2\sigma^2}\right)\right) \\ - \sum_{u=1}^{U} \sum_{i=1}^{N} K_{u_i u}\left(1 - e^{-w(T-t_i)}\right)\,.$$

We perform the maximization step of the EM-type algorithm (a projected gradient ascent) [86] directly by taking derivatives with respect to the parameters and setting them to 0. For the expectation step, we use the "optimal" parameter values from the prior maximization step to update the probabilities p_{ij} and p_{ij}^{b}. By (alternately) iterating these the expectation and maximization steps, we obtain (4) for the parametric model. For initialization, we sample Θ^0, p_{ij}, and p_{ij}^{b} uniformly at random. Note additionally that $p_{ij} = 0$ for $i > j$.

6.3.3.2 Nonparametric Model

The log-likelihood function of the nonparametric model is the same as for the parametric model in (6.7). We use a similar approach as before to derive an EM-type algorithm for the nonparametric model. The main differences are that (1) only Y_{ij} are latent variables and $Y_{ii} = 1$ signifies that event i is a background event, whereas $Y_{ji} = 1$ signifies that event i is triggered by event j; and (2) we assume that the triggering kernels $g_1(t)$ and $g_2(r)$ are piecewise constant functions. We discretize space and time into n_t^{bins} temporal bins and n_r^{bins} spatial bins, and the kernel takes a constant value in each spatiotemporal bin.

To formally present the EM-type algorithm (see algorithm 5), we borrow notation from [48]. Let C_k denote the set of event pairs (i, j) for which $t_j - t_i$ belongs to the k^{th} temporal

Algorithm 4 EM-type Algorithm for the Parametric Model

```
**Inputs**: point process: $\{(u_i, t_i, x_i, y_i)\}_{i=1}^{N}$; initial guesses for parameters: $\Theta^{(0)} = \left(\{K_{uv}^{(0)}\}_{u,v=1}^{U}, \{\beta_{uv}^{(0)}\}_{u,v=1}^{U}, \sigma^{(0)}, \omega^{(0)}\right)$ and $\{p_{ij}^{(0)}\}_{i,j=1}^{N}, \{p_{ij}^{b,(0)}\}_{i,j=1}^{N}$; termination threshold: $\epsilon$.
**Outputs**: model parameters $\Theta = \big(\{K_{uv}\}_{u,v=1}^{U}, \{\beta_{uv}\}_{u,v=1}^{U}, \sigma, \omega\big)$.
Initialize $\delta = 1$ and $k = 0$.
**while** $\delta > \epsilon$ **do**
    Let $\eta^{2,(k)}$ and $\sigma^{2,(k)}$ be the value of $\eta^2$ and $\sigma^2$ at the $k$th iteration.
    **Expectation step**: for $i, j \in \{1, 2, \cdots, N\}$,
    $p_{ij}^{(k)} = \big(K_{u_j u_i} g\left(t_i - t_j, x_i - x_j, y_i - y_j\right)\big) / \lambda\left(x_i, y_i, t_i\right).$
    $p_{ij}^{b,(k)} = \beta_{u_j u_i}^{(k)} \exp\left(-\frac{(x_j - x_i)^2 + (y_j - y_i)^2}{2\eta^{2,(k)}}\right) / 2\pi\eta^{2,(k)} T \lambda(x_i, y_i, t_i).$
    **Maximization step**: for $u, \hat{u} \in \{1, 2, \cdots, U\}$,
    $\omega^{(k+1)} = \dfrac{\sum_{j<i} p_{ij}^{(k)}}{\sum_{j<i} p_{ij}^{(k)}(t_i - t_j) + \sum_{u=1}^{U} \sum_{i=1}^{N} K_{u_i u}(T - t_i) e^{-\omega(T - t_i)}},$
    Let $n_u$ denote the number of events in point process $u$; and let $i_l^u$, with $l \in \{1, \ldots, n_u\}$, index the events for process $u$.
    $K_{\hat{u}u}^{(k+1)} = \sum_{l=1}^{n_u} \sum_{t_{i_{\hat{l}}^{\hat{u}}} < t_{i_l^u}} p_{i_l^u i_{\hat{l}}^{\hat{u}}}^{(k)} / \sum_{l=1}^{n_{\hat{u}}} \left(1 - \exp\left(-w\left(T - t_{i_{\hat{l}}^{\hat{u}}}\right)\right)\right),$
    $\beta_{\hat{u}u}^{(k+1)} = \sum_{i=1}^{n_u} \sum_{j=1}^{n_{\hat{u}}} p_{i_l^u i_{\hat{l}}^{\hat{u}}}^{b,(k)} / n_{\hat{u}}.$
    $\sigma^{2,(k+1)} = \sum_{i,j=1}^{N} \left(p_{ij}^{b,(k)} + p_{ij}^{(k)}\right) \left((x_i - x_j)^2 + (y_i - y_j)^2\right) / \sum_{i,j=1}^{N} 2\left(p_{ij}^{b,(k)} + p_{ij}^{(k)}\right).$
    $\eta^{2,(k+1)} = \sigma^{2,(k+1)}.$
    $\delta = \|\Theta^{(k)} - \Theta^{(k+1)}\|.$
    $k = k + 1.$
**end while**
```

bin, D_k denote the set of event pairs (i, j) for which r_{ij} (the distance between nodes i and j) belongs to the k^{th} spatial bin, N_u denote the number of events that include node u, the parameter Δt_k denote the size of the k^{th} temporal bin, and Δr_k denote the size of the k^{th} spatial bin.

Algorithm 5 EM-type Algorithm for our Nonparametric Model

```
**Inputs**: point process: $\{(u_i, t_i, x_i, y_i)\}_{i=1}^{N}$; initial guesses of parameters: $\{K_{uv}^{(0)}\}_{u,v=1}^{U}$ and $\{p_{ij}^{(0)}\}_{i,j=1}^{N}$; termination threshold: $\epsilon$.
**Outputs**: model parameters: $\{K_{uv}\}_{u,v=1}^{U}$; triggering probability between events: $\{p_{ij}\}_{i,j=1}^{N}$; temporal triggering kernel: $g_1$; spatial triggering kernel: $g_2$.
Initialize $\delta = 1$ and $\eta = 0$ .
**while** $\delta > \epsilon$ **do**
    Update background kernel $\tau^{\eta}(x, y)$ (see (6.5))
    $\gamma_u^{(\eta)} = \sum_{u_i=u} p_{ii}^{(\eta)}/Z^{(\eta)}$ , where $Z^{(\eta)}$ satisfies $\int_0^T \iint_S \tau^{\eta}(x, y)\mathrm{d}s\,\mathrm{d}t = Z^{(\eta)}$ for a bounded spatial domain $S$ and for $u \in \{1, \dots, U\}$.
    $K_{uv}^{(\eta)} = \sum_{u_i=u} \sum_{u_j=v} p_{ij}^{(\eta)}/N_u$ for $u, v \in \{1, \dots, U\}$.
    $g_1^{(\eta)}(t) = \sum_{i,j \in C_k} p_{ij}^{(\eta)}/\Delta t_k \sum_{i<j} p_{ij}^{(\eta)}$ for $t$ in the $k$th temporal bin.
    $h^{(\eta)}(r) = \sum_{i,j \in D_k} p_{ij}^{(\eta)}/\Delta r_k \sum_{i<j} p_{ij}^{(\eta)}$ for $r$ in the $k$th spatial bin. Set $g_2^{(\eta)}(r) = h^{(\eta)}(r)/(2\pi r)$ .
    $p_{ij}^{(\eta+1)} = K_{u_i u_j}^{(\eta)} g_1^{(\eta)}(t_j - t_i) g_2^{(\eta)}(r_{ij})$ for $i < j$ and $p_{jj}^{(\eta+1)} = \mu_{u_j}^{(\eta)}(x_j, y_j)$.
    Normalize $p_{ij}^{(\eta+1)}$ so that $\sum_{i=1}^{N} p_{ij}^{(\eta+1)} = 1$ for any $j$.
    $\delta = \max_{i,j} \|p_{ij}^{(\eta+1)} - p_{ij}^{(\eta)}\|$ and $\eta = \eta + 1$.
**end while**
```

6.3.4 Simulations

To generate synthetic data for model comparisons, we need to simulate self-exciting point processes with the conditional intensity in (6.4) for each process u. We use the branching structures [175] of self-exciting point processes to develop algorithm 6 for our simulations.

Algorithm 6 Simulation of a Multivariate Hawkes Process

```
**Inputs**: time-window size: $T$; spatial region: $S \subset \mathbb{R}^2$; background rate: $\{\gamma_u\}_{u=1}^{U}$; triggering matrix: $\{K_{uv}\}_{u,v=1}^{U}$; temporal and spatial triggering kernels: $g_1(t)$, $g_2(x, y)$.
**Output**: point process: $\mathbf{C} = \{(u_i, t_i, x_i, y_i)\}_{i=1}^{N}$.
Initialize an empty set $\mathbf{C}$ and an empty stack $\mathbf{Q}$.
**Generate background events:**
  Draw $N_u$, the number of background events of type $u$, from a Poisson distribution with parameter $\lambda = \gamma_u T$ for each $u \leq U$.
  Add each background event $i \leq \sum_{u=1}^{U} N_u$ — i.e., $(x_i, y_i, t_i, u_i)$ — to the set $\mathbf{C}$ and the stack $\mathbf{Q}$, where $(x_i, y_i, t_i)$ is drawn from the uniform spatiotemporal distribution over the time interval $[0, T]$ and a bounded spatial region $S$.
**Generate triggered events:**
  **while** $\mathbf{Q}$ is not empty **do**
    Remove the most recently added element $(x_i, y_i, t_i, u_i)$ from the stack $\mathbf{Q}$.
    Draw $N_i$, the number of events triggered by event $i$, from a Poisson distribution with parameter $\lambda_i = \sum_{u'=1}^{U} K_{u_i u'}$.
    Generate events $(x_k, y_k, t_k, u_k)$ for each $k \leq N_i$ as follows:
      Sample $t_k$, $(x_k, y_k)$ and $u_k$ according to $g_1(t - t_i)$, $g_2(x - x_i, y - y_i)$, and $P(u_k = \tilde{u}) = \frac{K_{u_i \tilde{u}}}{\sum_{v=1}^{U} K_{u_i v}}$, respectively.
      Add $(x_k, y_k, t_k, u_k)$ to the set $\mathbf{C}$.
      **if** $t_k \leq T$ **then**
        Add the element $(x_k, y_k, t_k, u_k)$ to the stack $\mathbf{Q}$.
      **end if**
  **end while**
```

6.4 Numerical Experiments and Results

We apply our algorithm to both synthetic and real-world data sets to demonstrate the usefulness of incorporating spatial information and of our nonparametric approach. We consider a synthetic data set in Section 6.4.1 and a Gowalla data set in Section 6.4.2. We compare our nonparametric model ("Nonparametric Hawkes") with the Bayesian Hawkes model[2] in [89] ("Bayesian Hawkes"), the exclusively temporal Hawkes model with kernel $g(t) = \omega \exp(-\omega t)$ from [49] ("Temporal Hawkes"), and the parametric spatiotemporal model detailed in Section 6.3.1 ("Parametric Hawkes"). We make comparisons by examining how well the following properties are recovered in the inferred triggering matrix: (1) symmetry and reciprocity; (2) existence of edges; and (3) community structure. We also demonstrate the ability of our algorithm to infer the triggering kernel g.

6.4.1 Synthetic Data

We first generate synthetic triggering matrices $\mathbf{K}$ using a weighted stochastic block model (WSBM) [4, 118]. We assign a network's nodes to four sets (called "communities") and assign edges to adjacency-matrix blocks based on the set memberships of the nodes. Two of the communities consist of ten nodes each, and the other two communities consist of five nodes each. For each edge, we first draw a Bernoulli random variable to determine whether it exists, and we then draw an exponential random variable to determine the weight of the edge (if it exists). The parameter of the Bernoulli random variable is 0.68 for there to be an edge between nodes from the same community and 0.2 for an edge between nodes from different communities. The decay-rate parameter for the exponential random variable in these two situations is 0.1 and 0.01, respectively. By construction, our triggering matrices are symmetric.

The triggering matrices that we generate in this way are not guaranteed to satisfy the

[2] We use code from the authors of [89]; it is available at `https://github.com/slinderman/pyhawkes`. In all of our experiments, we use the default hyperparameters that come with the published code.

stability condition for Hawkes processes; this condition is that the largest-magnitude eigenvalue of $\mathbf{K}$ is smaller than one [33]. When this condition is satisfied, each event has, almost surely, finitely many subsequent events as "offspring". In our work, we discard any simulated adjacency matrix that does not satisfy the stability condition, and we generate a new one to replace it. (With our choices of the parameters, we discard about 65% of the generated adjacency matrices.)

With each triggering matrix $\mathbf{K}$, we use algorithm 6 to simulate a multivariate spatiotemporal Hawkes process with our parametric model in Section 6.3.1 with $\omega = 0.6$, $\sigma^2 = 0.3$, $T = 250$, $S = [0, 1] \times [0, 1]$, and a homogeneous value $\gamma_u = 0.2$ for all nodes u. We then reconstruct the underlying networks and the triggering kernels from the simulated data.

6.4.1.1 Symmetry and Reciprocity

As we noted in Section 6.4.1, our simulated triggering matrices are symmetric, but our reconstructed adjacency matrices generally are not symmetric. Measuring deviation from symmetry gives one way to evaluate the performance of our inference methods. We use various reciprocity measures to quantify such deviation.

We conduct two sets of experiments. In the first one, we fix a single synthetic triggering matrix and simulate ten multivariate spatiotemporal Hawkes point processes. We then estimate the triggering matrix $\mathbf{K}$ from each point process using various methods, which we thereby compare with each other. In a second set of experiments, instead of fixing a single triggering matrix, we generate ten different triggering matrices using the same WSBM model and parameters, and we simulate one point process for each triggering matrix.

There is no standard way of measuring reciprocity in a weighted network. In our calculations, we use diagnostics that were proposed in [140] and [6]. First, as in [140], we compute the reciprocated edge weight $K_{uv}^{\leftrightarrow} = \min\{K_{uv}, K_{vu}\}$, and we then calculate a network-level reciprocity score R_1 as the ratio between the total reciprocated weight $W^{\leftrightarrow} = \sum_{u \neq v} K_{uv}^{\leftrightarrow}$ and the total weight $W = \sum_{u \neq v} K_{uv}$. That is, the "reciprocity" is $R_1 := W^{\leftrightarrow}/W$. Second, Akoglu et al. [6] proposed three node-level measures of reciprocity: (1) the "ratio"

Table 6.1: Reciprocity of the triggering matrices that we infer using different methods. We report the mean and standard deviation (in parentheses) over ten simulations with the same (ground-truth) triggering matrix.

	Nonparametric	Temporal	Parametric	Bayesian
R_1	0.59 (0.05)	0.29 (0.06)	0.54 (0.03)	0.36 (0.03)
Correlation	0.84 (0.05)	0.36 (0.16)	0.79 (0.05)	0.30 (0.14)
Ratio	0.55 (0.02)	0.37 (0.11)	0.58 (0.02)	0.32 (0.02)
Coherence	0.75 (0.01)	0.63 (0.03)	0.71 (0.02)	0.68 (0.02)
Entropy	0.71 (0.01)	0.59 (0.03)	0.68 (0.02)	0.60 (0.02)

$R_{\text{ratio}} := \min\{K_{uv}, K_{vu}\}/\max\{K_{uv}, K_{vu}\}$; (2) "coherence" $R_{\text{coher}} = 2\sqrt{K_{uv}K_{vu}}/(K_{uv} + K_{vu})$; and (3) "entropy" $R_{\text{entropy}} := -r_{uv}\log_2(r_{uv}) - r_{vu}\log_2(r_{vu})$, where $r_{uv} = K_{uv}/(K_{uv} + K_{vu})$. These last three measures of reciprocity are measured at a node level, whereas R_1 is a network-level measure. For the other measures, we obtain a network-level measure by calculating those scores for each pair of nodes and then taking a mean over all pairs of nodes. Each of the above quantities gives a score between 0 and 1, where a larger value indicates a stronger tendency for the nodes in a network to reciprocate. In a perfectly symmetric and reciprocal network, each of the four methods gives a value of 1.

In Table 6.1, we report the mean reciprocity and the standard deviation over ten simulations with the same triggering matrix. In Table 6.2, we report the mean results from ten different triggering matrices. Both spatiotemporal models give higher scores than the exclusively temporal models, which is what we expected, as the temporal models discard spatial information. According to these measures of success, the nonparametric model has the best performance.

6.4.1.2 Edge Reconstruction

We also evaluate the reconstruction methods based on their ability to recover the existence of edges. This is particularly relevant if we want to know whether there is a connection between two entities. We will discuss this application in detail using the Gowalla data set

Table 6.2: Reciprocity of the triggering matrices that we infer using different methods. We report the mean and standard deviation (in parentheses) over ten simulations, each with a different (ground-truth) triggering matrix.

	Nonparametric	Temporal	Parametric	Bayesian
R_1	0.61 (0.12)	0.36 (0.12)	0.55 (0.10)	0.40 (0.05)
Correlation	0.81 (0.16)	0.48 (0.27)	0.76 (0.15)	0.23 (0.14)
Ratio	0.63 (0.04)	0.43 (0.06)	0.62 (0.03)	0.33 (0.03)
Coherence	0.78 (0.04)	0.62 (0.03)	0.72 (0.03)	0.70 (0.03)
Entropy	0.75 (0.05)	0.58 (0.03)	0.69 (0.03)	0.62 (0.04)

(see Section 6.4.2).

In our model, we consider an edge to exist if the corresponding weighted entry in the inferred triggering matrix exceeds a certain threshold. For different threshold levels, we compute the numbers of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) for a given ground-truth triggering matrix. We summarize our results in a receiver operating characteristic (ROC) plot (see Figure 6.1), in which we plot the true-positive rate (TPR) (where TPR = TP/(TP + FN)) versus the false-positive rate (FPR) (where FPR = FP/(FP + TN)). A better inference of a triggering matrix gives a larger value of TPR at a fixed FPR.

Based on the ROC plot in Figure 6.1, we conclude that the spatiotemporal models — both the parametric and nonparametric Hawkes models that we proposed in Section 6.3 — outperform the exclusively temporal ones. Therefore, incorporating spatial information improves the quality of our reconstructed binary networks, at least according to this measure of success. The best results are from our parametric model, which is not surprising, given that we use the same model to simulate the data. The performance of our nonparametric model is very close to that of the parametric model, confirming its effectiveness at inferring the existence of edges.

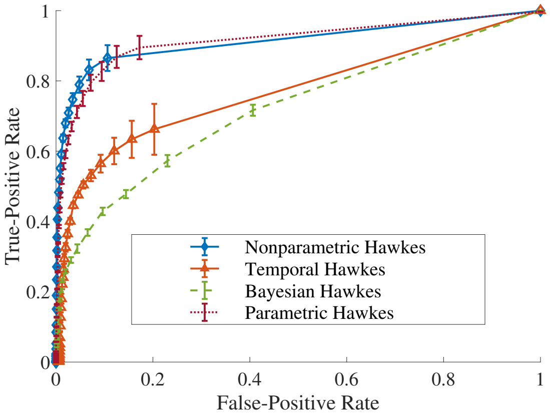


Figure 6.1: Model comparison using synthetic networks. We show the mean ROC curves with error bars (averaged over ten simulations, each with a different triggering matrix) on edge reconstruction. The ROC curve of a better reconstruction should be closer to 1 for a larger range of horizontal-axis values, such that it has a larger area under the curve (AUC), which is equal to the probability that a uniformly-randomly chosen existing edge in a ground-truth network has a larger weight than a uniformly-randomly chosen missing edge in the inferred network.

Table 6.3: The L_1 errors of the inferred spatial and temporal kernels. We simulate ten point processes with the same triggering matrix and triggering kernel. We report the mean and standard deviation (in parentheses) of the L_1 errors averaged over the ten simulations with the same triggering kernel and matrix. Note that the exclusively temporal model does not estimate a spatial kernel.

	Nonparametric	Temporal	Parametric
Temporal kernel	0.07 (0.02)	0.20 (0.06)	0.02 (0.02)
Spatial kernel	0.06 (0.02)	-	0.12 (0.02)

6.4.1.3 Inferred Kernels

We report the inferred kernels of the different models in Figure 6.2. Recall that the ground-truth kernels that we use to simulate point processes are $g_1(t) = \omega \exp(-\omega t)$ and $h(r) = 2\pi r g_2(r) = \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right)$, where $r^2 = x^2 + y^2$, $\omega = 0.6$, and $\sigma^2 = 0.3$. Let $\hat{g_1}$ and $\hat{h}$ denote the inferred temporal and spatial kernels, respectively.

We calculate the L_1 errors $\int |g_1(t) - \hat{g_1}(t)| \, \mathrm{d}t$ and $\int |h(r) - \hat{h}(r)| \, \mathrm{d}r$. We report these errors in Table 6.3 and present visualizations of the inferred kernels in Figure 6.2. As expected, both spatiotemporal Hawkes models give more accurate kernel inference than the exclusively temporal model. The nonparametric Hawkes model does not use any information about the ground-truth kernels. Surprisingly, it is more accurate, in terms of the L_1 error, at inferring the spatial trigger kernel than the parametric model, whose kernel shares the same parametric form as the ground-truth kernel.

6.4.1.4 Community-Structure Recovery

We also evaluate the quality of the inferred networks based on their community structure, in which dense sets of nodes in a network are connected sparsely to other dense sets of nodes [46,125]. Recall that we have planted a four-community structure in the synthetic triggering matrices (see Section 6.4.1). We apply the community-detection methods from [4] (an inference method for a WSBM), [79] (symmetric non-negative matrix factorization; NMF),

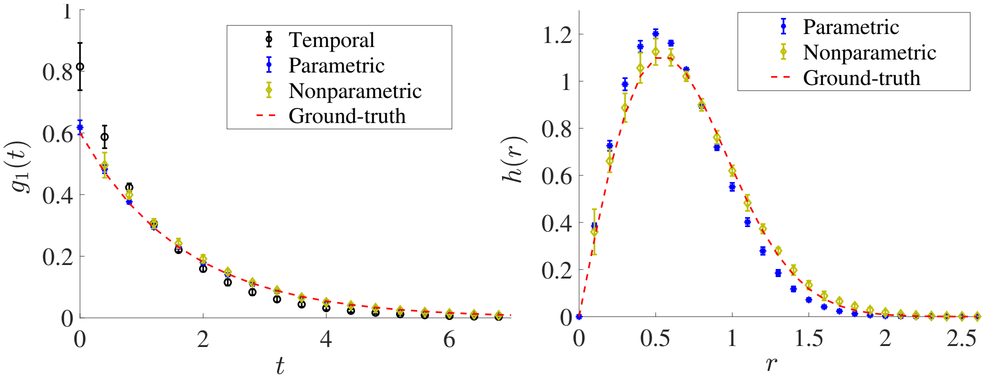


Figure 6.2: Model comparison using synthetic networks: Inferred (left) temporal and (right) spatial kernels using different methods: Temporal Hawkes, Parametric Hawkes and Nonparametric Hawkes. The dashed lines are ground-truth kernels used for the synthetic data.

and [66,108,110,110] (modularity maximization[3]). The WSBM that we infer for community detection is the same that one we use to construct the synthetic adjacency matrices (see Section 6.4.1). To evaluate our inferred community structure, we use the square-root variant of *normalized mutual information* (NMI) [144] between the inferred community assignment and "ground truth" community labels. Specifically, Let S_1 and S_2 be community assignments of the U nodes to C_1 and C_2 communities, respectively; and let $S_{\ell k}$, with $\ell \in \{1, 2\}$ and $k \in \{1, 2, \cdots, C_\ell\}$, denote the set of nodes in the kth community in assignment S_ℓ. The NMI between S_1 and S_2 is

$$\text{NMI}(S_1, S_2) = \frac{I(S_1, S_2)}{\sqrt{H(S_1)H(S_2)}} \in [0, 1]\,,$$

where $I(S_1, S_2) = \sum_{i=1}^{C_1} \sum_{j=1}^{C_2} \frac{|S_{1i} \cap S_{2j}|}{U} \log \frac{|S_{1i} \cap S_{2j}|/U}{|S_{1i}||S_{2j}|/U^2}$ (where $|J|$ denotes the cardinality of the set J) and the entropy is $H(S_\ell) = -\sum_{i=1}^{N_\ell} \frac{|S_{\ell i}|}{N} \log \frac{|S_{\ell i}|}{N}$ (with $\ell \in \{1, 2\}$). Intuitively, NMI measures the amount of information that is shared by two community assignments. If they are the same after permuting community labels, the NMI is equal to 1. A larger NMI score implies that the inferred community assignment shares more information with the ground-truth labels. See [151] for a discussion of other approaches for comparing different community

[3] For modularity maximization, we use the implementation of a (locally greedy) Louvain-like [21] method (called GenLouvain) from [66] with the default resolution-parameter value of 1 and the Newman–Girvan null model.

Table 6.4: Normalized mutual information (NMI) between the outputs of different community-detection methods applied to the inferred networks and the ground-truth community structure (averaged over ten simulations, each with a different triggering matrix).

	Nonparametric	Temporal	Parametric	Bayesian
Weighted SBM	0.80	0.38	0.83	0.36
Symmetric NMF	0.62	0.31	0.66	0.19
Modularity Maximization	0.64	0.47	0.71	0.28

assignments in networks.

There are numerous approaches for detecting communities in networks [46,118,125], and we use methods with readily-available code. As we show in Table 6.4, all of these community-detection methods perform better when we infer triggering matrices using both spatial and temporal information than with with exclusively temporal information. One can, of course, repeat our experiments using other methods.

6.4.2 Gowalla Friendship Network

Gowalla is a location-based social-media website in which users share their locations by checking in. We use a Gowalla data set — collected in [29] using Gowalla's public API — of a "friendship" network with 196,591 users, 950,327 edges, and a total of 6,442,890 check-ins of these users between February 2009 and October 2010. The data set also includes the latitude and longitude coordinates and the time (with a precision of one second) of each check-in. Similar to a Facebook "friendship" network, the Gowalla friendship network is undirected. The mean number of friends for each user is 9.7, the median is 3, and the maximum is $14,730$. We study several subnetworks in the Gowalla data set. We view the spatiotemporal check-ins of Gowalla users within each subnetwork as events in a multivariate point process and infer relationships between these users.

We compare our Nonparametric Hawkes method with the Bayesian Hawkes and the exclusively Temporal Hawkes in terms of how well our inferred edges match the Gowalla friendships. Because a Gowalla friendship network is undirected, we first symmetrize the inferred triggering matrix (via $\tilde{\mathbf{K}} = \left(\mathbf{K} + \mathbf{K}^T\right)/2$) to obtain an undirected network. We then calculate FPRs and TPRs in the same fashion as Section 6.4.1.2 using $\tilde{\mathbf{K}}$'s associated "ground-truth" friendship network and generate the corresponding ROC curves. In the ROC curves of three different cities in Figure 6.4, we observe that the best results are from our nonparametric model that incorporates spatial information. The mean AUCs are 0.4277 (with a standard deviation of 0.1042) for the Temporal Hawkes method; 0.5301 (with a standard deviation of 0.0585) for the Bayesian Hawkes method; and 0.6692 (with a standard deviation of 0.0421) for our Nonparametric Hawkes method in all of the examined subnetworks.

6.4.2.1 New York City (NYC)

We study check-ins in New York City (NYC) during the period April–October 2010. We use a bounding box (with a north latitude of 40.92, a south latitude of 40.48, an east longitude of -73.70, and a west longitude of -74.26)[4] to locate check-ins in NYC. We consider "active" users, who have at least 100 check-ins during the period. To alleviate the computational burden, we also only consider users who have at most 500 check-ins during the period to reduce the number of users and the total number of check-ins. Our inference process requires computing a triggering probability for each pair of events (i.e., check-ins), which results in a full upper-triangular matrix. The number of nonzero entries in this matrix scales with the square of the total number of events, so the memory requirement also scales quadratically with the number of events. We perform experiments only for cases in which the total number of events is at most $10,000$ to be able to store triggering probabilities for all pairs of events in 4-gigabyte memory. There are $5,801$ unique users with at least one check-in in NYC during the period, and there are $101,329$ check-ins in total. After removing "inactive" users (i.e., those with strictly fewer than 100 check-ins) and overly active users (i.e., those with strictly

[4] We obtain latitude and longitude coordinates from http://www.mapdevelopers.com/geocode_bounding_box.php.

more than 500 check-ins), we are left with 160 users and a total of 29, 118 check-ins. We also restrict ourselves to users in the largest connected component (LCC) of the network. This yields 46 users and 8, 495 check-ins, on which we apply our inference methodology.

6.4.2.2 Los Angeles (LA)

We apply the same procedure as in Section 6.4.2.1 on the check-in data for Los Angeles (LA). The bounding box that we use for LA has a north latitude of 34.34, a south latitude of 33.70, an east longitude of −188.16, and a west longitude of −188.67. We restrict the area of LA to be the same as that of NYC, although LA's geographic area is much larger than that of NYC. After selecting only users in the LCC of the Gowalla network among users who are active (with at least 150 check-ins) but not overly active (with at most 1000 check-ins) users, we are left with 23 users and 6, 203 check-ins.

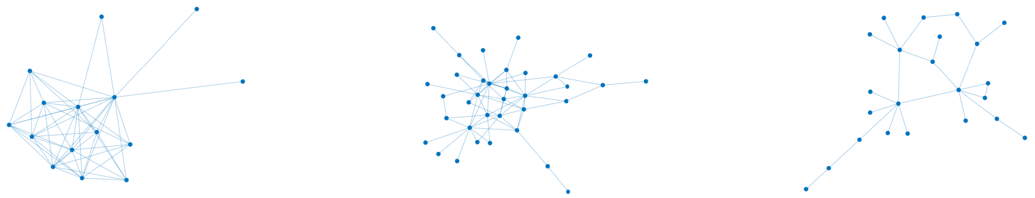
6.4.2.3 San Francisco (SF)

To look at a different type of example, we also examine the 1-ego network of the most popular user (with 14 friends) in San Francisco (SF). (A 1-ego network [154] of a node is an induced subgraph that includes a focal node — the ego — and its direct neighbors.) The bounding box that we use for SF has a north latitude of 37.93, a south latitude of 37.64, an east longitude of −122.28, and a west longitude of −123.17. In this 1-ego network, there are 9, 887 check-ins.

6.5 Conclusion

In this chapter, we use point-process models to infer latent networks from synthetic and real-world spatiotemporal data sets. We then apply tools from network analysis to examine the inferred networks. We study the role of spatial information and nonparametric techniques in network reconstruction.

As we have illustrated, it is very important to incorporate spatial information. However,

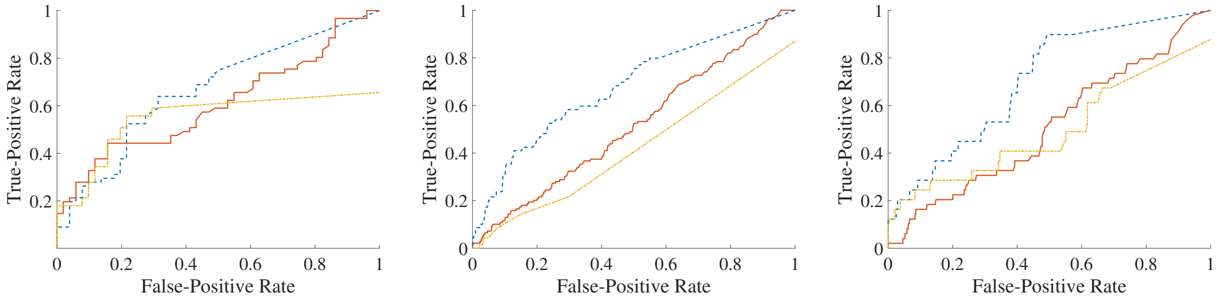


(a) 1-Ego network of a user of Gowalla in SF.

(b) Largest connected component of the Gowalla network in NYC.

(c) Largest connected component of the Gowalla network in LA.

Figure 6.3: Three different friendships networks in the Gowalla data set. We compare different network reconstruction methods for these networks.



(a) San Fransisco

(b) New York City

(c) Los Angeles

Figure 6.4: ROC curves of different methods for reconstructing three Gowalla friendship networks. Here dashed lines are for our Nonparametric Hawkes; dotted lines for Temporal Hawkes; and solid lines for Bayesian Hawkes.)

using such information effectively requires making a good choice of spatiotemporal triggering kernels. We achieve this using a nonparametric approach. Through experiments on synthetic data sets, we show that our nonparametric Hawkes method is capable of doing a good job of successfully recovering spatial and temporal triggering kernels. Moreover, our approach is able to infer a network structure that better recovers — compared to other network reconstruction methods that we studied — symmetry and reciprocity, edge reconstruction, and community structures. Through experiments on real-world data sets, we illustrat that the inferred networks of our approach are meaningful, in the sense that they have large positive correlations with some metadata.

Bibliography

[1] Girmaw Abebe and Andrea Cavallaro. A long short-term memory convolutional neural network for first-person vision activity recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1339–1346, 2017.

[2] Massil Achab, Emmanuel Bacry, Stéphane Gaïffas, Iacopo Mastromatteo, and Jean-François Muzy. Uncovering causality from multivariate Hawkes integrated cumulants. *The Journal of Machine Learning Research*, 18(1):6998–7025, 2017.

[3] Sergios Agapiou, Stig Larsson, and Andrew M. Stuart. Posterior contraction rates for the Bayesian approach to linear ill-posed inverse problems. *Stochastic Processes and their Applications*, 123(10):3828–3860, 2013.

[4] Christopher Aicher, Abigail Z. Jacobs, and Aaron Clauset. Learning latent block structure in weighted networks. *Journal of Complex Networks*, 3(2):221–248, 2014.

[5] Kiyoharu Aizawa, Kenichiro Ishijima, and Makoto Shiina. Summarizing wearable video. In *Proceedings to 2001 International Conference on Image Processing*, volume 3, pages 398–401. IEEE, 2001.

[6] Leman Akoglu, Pedro O.S. Vaz de Melo, and Christos Faloutsos. Quantifying reciprocity in large weighted communication networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 85–96. Springer, 2012.

[7] Samuel Miller Allen and John W Cahn. Ground state structures in ordered binary alloys with second neighbor interactions. *Acta Metallurgica*, 20(3):423–433, 1972.

[8] Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. Hawkes processes in finance. *Market Microstructure and Liquidity*, 1(01):1550005, 2015.

[9] Duygu Balcan, Vittoria Colizza, Bruno Gonçalves, Hao Hu, José J Ramasco, and Alessandro Vespignani. Multiscale mobility networks and the spatial spreading of infectious diseases. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 106, pages 21484–21489. National Academy Sciences, 2009.

[10] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, 2009.

[11] Hugo Barbosa, Marc Barthelemy, Gourab Ghoshal, Charlotte R. James, Maxime Lenormand, Thomas Louail, Ronaldo Menezes, José J. Ramasco, Filippo Simini, and Marcello Tomasini. Human mobility: Models and applications. *Physics Reports*, 734:1–74, 2018.

[12] John L. Barron, David J. Fleet, and Steven S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[13] Marc Barthelemy. *Morphogenesis of Spatial Networks*. Springer, Cham, 2018.

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

[15] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems*, pages 585–591, 2002.

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

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

[18] Andrea L. Bertozzi, Bamdad Hosseini, Hao Li, Kevin Miller, and Andrew M. Stuart. Posterior consistency of semi-supervised regression on graphs. *arXiv preprint arXiv:2007.12809*, 2020.

[19] 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.

[20] Bharat Lal Bhatnagar, Suriya Singh, Chetan Arora, and C.V. Jawahar. Unsupervised learning of deep feature representation for clustering egocentric actions. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1447–1453, 2017.

[21] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):10008, 2008.

[22] Zdravko I. Botev and Pierre L'Ecuyer. Efficient probability estimation and simulation of the truncated multivariate student-t distribution. In *Proceedings of the 2015 Winter Simulation Conference*, pages 380–391. IEEE Press, 2015.

[23] D. Brockmann, L. Hufnagel, and T. Geisel. The scaling laws of human travel. *Nature*, 439:462–465, 2006.

[24] Emery N. Brown, Robert E. Kass, and Partha P. Mitra. Multiple neural spike train data analysis: State-of-the-art and future challenges. *Nature Neuroscience*, 7(5):456–461, 2004.

[25] Lawrence D. Brown and Mark G. Low. Asymptotic equivalence of nonparametric regression and white noise. *The Annals of Statistics*, 24(6):2384–2398, 1996.

[26] Jeremy Budd and Yves van Gennip. Graph Merriman–Bence–Osher as a semi-discrete implicit Euler scheme for graph Allen–Cahn. *SIAM Journal on Mathematical Analysis*, 52(5):4104–4139, 2020.

[27] W. Cai, Y. Zhang, and J. Zhou. Maximizing expected model change for active learning in regression. In *2013 IEEE 13th International Conference on Data Mining*, pages 51–60, 2013.

[28] Daniela Calvetti and Erkki Somersalo. *An Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing*, volume 2. Springer Science & Business Media, 2007.

[29] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: User movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1082–1090. ACM, 2011.

[30] Michael D. Conover, Clayton Davis, Emilio Ferrara, Karissa McKelvey, Filippo Menczer, and Alessandro Flammini. The geospatial characteristics of a social movement communication network. *PloS ONE*, 8(3):e55957, 2013.

[31] Simon L. Cotter, Gareth O. Roberts, Andrew M. Stuart, and David White. MCMC methods for functions: Modifying old algorithms to make them faster. *Statistical Science*, pages 424–446, 2013.

[32] Noel Cressie. *Statistics for Spatial Data*. John Wiley & Sons, 2015.

[33] Daryl J. Daley and David Vere-Jones. *An Introduction to the Theory of Point Processes*. Springer Science & Business Media, 2007.

[34] Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*, pages 208–215, Helsinki, Finland, July 2008. Association for Computing Machinery.

[35] Masoumeh Dashti, Kody J. H. Law, Andrew M. Stuart, and Jochen Voss. MAP estimators and their consistency in Bayesian nonparametric inverse problems. *Inverse Problems*, 29(9):095017, 2013.

[36] Masoumeh Dashti and Andrew M. Stuart. *The Bayesian Approach to Inverse Problems*. Springer International Publishing, Cham, 2017.

[37] Ana Garcia del Molino, Cheston Tan, Joo-Hwee Lim, and Ah-Hwee Tan. Summarization of egocentric videos: a comprehensive survey. *IEEE Transactions on Human-Machine Systems*, 47(1):65–76, 2017.

[38] Persi Diaconis and David Freedman. On the consistency of Bayes estimates. *The Annals of Statistics*, 14(1):1–26, 1986.

[39] Michael Eichler, Rainer Dahlhaus, and Johannes Dueck. Graphical modeling for multivariate Hawkes processes with nonparametric link functions. *Journal of Time Series Analysis*, 38(2):225–242, 2017.

[40] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of Inverse Problems*, volume 375. Springer Science & Business Media, 1996.

[41] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian Conference on Image Analysis*, pages 363–370. Springer, 2003.

[42] Alircza Fathi, Jessica K. Hodgins, and James M. Rehg. Social interactions: A first-person perspective. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1226–1233. IEEE, 2012.

[43] Alireza Fathi, Ali Farhadi, and James M. Rehg. Understanding egocentric activities. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 407–414. IEEE, 2011.

[44] Alireza Fathi, Yin Li, and James M Rehg. Learning to recognize daily actions using gaze. In *European Conference on Computer Vision*, pages 314–327. Springer, 2012.

[45] Denis Fortun, Patrick Bouthemy, and Charles Kervrann. Optical flow modeling and computation: A survey. *Computer Vision and Image Understanding*, 134:1–21, 2015.

[46] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1–44, November 2016.

[47] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the Nyström method. *IEEE transactions on pattern analysis and machine intelligence*, 26(2):214–225, 2004.

[48] Eric W. Fox, Frederic P. Schoenberg, and Joshua Seth Gordon. Spatially inhomogeneous background rate estimators and uncertainty quantification for nonparametric Hawkes point process models of earthquake occurrences. *The Annals of Applied Statistics*, 10(3):1725–1756, 2016.

[49] Eric W. Fox, Martin B. Short, Frederic P. Schoenberg, Kathryn D. Coronges, and Andrea L. Bertozzi. Modeling e-mail networks and inferring leadership using self-exciting point processes. *Journal of the American Statistical Association*, 111(514):564–584, 2016.

[50] David Freedman. Wald lecture: On the Bernstein-von Mises theorem with infinite-dimensional parameters. *The Annals of Statistics*, 27(4):1119–1141, 1999.

[51] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1183–1192, Sydney, NSW, Australia, 08 2017. JMLR.org.

[52] Cristina Garcia-Cardona, Ekaterina Merkurjev, Andrea L. Bertozzi, Arjuna Flenner, and Allon G. Percus. Multiclass data segmentation using diffuse interface methods on graphs. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1600–1613, 2014.

[53] Subhashis Ghosal, Jayanta K. Ghosh, and Aad W. Van Der Vaart. Convergence rates of posterior distributions. *Annals of Statistics*, 28(2):500–531, 2000.

[54] Guy Gilboa and Stanley Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005–1028, 2008.

[55] Evarist Giné and Richard Nickl. *Mathematical Foundations of Infinite-Dimensional Statistical Models*. Cambridge Series In Statistical and Probabilistic Mathematics. Cambridge University Press, New York, 2016.

[56] Clive W.J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438, 1969.

[57] Eric C. Hall and Rebecca M. Willett. Tracking dynamic point processes on networks. *IEEE Transactions on Information Theory*, 62(7):4327–4346, 2016.

[58] Jarno Hartog and Harry van Zanten. Nonparametric Bayesian label prediction on a graph. *Computational Statistics and Data Analysis*, 120:111–131, 2018.

[59] Jarno Hartog and J. Harry van Zanten. Nonparametric Bayesian label prediction on a large graph using truncated Laplacian regularization. *Communications in Statistics - Simulation and Computation*, pages 1–18, 2019.

[60] Franca Hoffmann, Bamdad Hosseini, Assad A. Oberai, and Andrew M. Stuart. Spectral analysis of weighted Laplacians arising in data clustering. *arXiv preprint arXiv:1909.06389*, 2019.

[61] Franca Hoffmann, Bamdad Hosseini, Zhi Ren, and Andrew M. Stuart. Consistency of semi-supervised learning algorithms on graphs: Probit and one-hot methods. *Journal of Machine Learning Research*, 21(186):1–55, 2020.

[62] Steven C.H. Hoi, Rong Jin, Jianke Zhu, and Michael R. Lyu. Semi-supervised SVM batch mode active learning for image retrieval. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, June 2008. ISSN: 1063-6919.

[63] Petter Holme. Modern temporal network theory: A colloquium. *The European Physical Journal B*, 88(9):234, 2015.

[64] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[65] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[66] Lucas G. S. Jeub, Marya Bazzi, Inderjit S. Jutla, and Peter J. Mucha. *A generalized Louvain method for community detection implemented in Matlab*, 2011–2019. https://github.com/GenLouvain/GenLouvain.

[67] Ming Ji and Jiawei Han. A variance minimization criterion to active learning on graphs. In *Artificial Intelligence and Statistics*, pages 556–564, 2012.

[68] Heinrich Jiang and Maya Gupta. Minimum-margin active learning. *arXiv preprint arXiv:1906.00025*, 2019.

[69] Kwang-Sung Jun and Robert Nowak. Graph-based active learning: A new look at expected error minimization. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1325–1329. IEEE, 2016.

[70] Jari Kaipio and Erkki Somersalo. *Statistical and Computational Inverse Problems*, volume 160. Springer Science & Business Media, 2006.

[71] Márton Karsai, Hang-Hyun Jo, and Kimmo Kaski. *Bursty Human Dynamics*. Springer, 2018.

[72] Mina Karzand and Robert D. Nowak. Maximin active learning in overparameterized model classes. *IEEE Journal on Selected Areas in Information Theory*, 2020.

[73] Alisa Kirichenko and Harry van Zanten. Estimating a smooth function on a large graph by Bayesian Laplacian regularisation. *Electron. J. Statist.*, 11(1):891–915, 2017.

[74] Kris M. Kitani, Takahiro Okabe, Yoichi Sato, and Akihiro Sugimoto. Fast unsupervised ego-action learning for first-person sports videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3241–3248, 2011.

[75] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.

[76] Mikko Kivelä and Mason A. Porter. Estimating interevent time distributions from finite observation periods in communication networks. *Physical Review E*, 92(5):052813, 2015.

[77] Georgios Kostopoulos, Stamatis Karlos, Sotiris Kotsiantis, and Omiros Ragos. Semi-supervised regression: A recent review. *Journal of Intelligent & Fuzzy Systems*, pages 1–18, 2018.

[78] Da Kuang, P. Jeffrey Brantingham, and Andrea L. Bertozzi. Crime topic modeling. *Crime Science*, 6(1):12, 2017.

[79] Da Kuang, Chris Ding, and Haesun Park. Symmetric nonnegative matrix factorization for graph clustering. In *Proceedings of the 2012 SIAM international conference on data mining*, pages 106–117. SIAM, 2012.

[80] Dan Kushnir and Luca Venturi. Diffusion-based deep active learning. *arXiv preprint arXiv:2003.10339*, 2020.

[81] Eric L. Lai, Daniel Moyer, Baichuan Yuan, Eric W. Fox, Blake Hunter, Andrea L. Bertozzi, and P Jeffrey Brantingham. Topic time series analysis of microblogs. *IMA Journal of Applied Mathematics*, 81(3):409–431, 2016.

[82] Steffen L. Lauritzen. *Graphical Models*, volume 17. Clarendon Press, 1996.

[83] David Lazer, Alex Pentland, Lada Adamic, Sinan Aral, Albert-László Barabási, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, Tony Jebara, Gary King, Michael Macy, Deb Roy, and Marshall Van Alstyne. Computational social science. *Science*, 323(5915):721–723, 2009.

[84] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. http://yann.lecun.com/exdb/mnist/.

[85] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562, 2001.

[86] Erik Lewis and George O. Mohler. A nonparametric EM algorithm for multiscale Hawkes processes. *preprint*, 2011. http://paleo.sscnet.ucla.edu/Lewis-Molher-EM_Preprint.pdf.

[87] Hao Li, Honglin Chen, Alexander Song, Matt Haberland, Osman Akar, Adam Dhillon, Tiankuang Zhou, Andrea L. Bertozzi, and P. Jeffrey Brantingham. PDEs on graphs for semi-supervised learning applied to first-person activity recognition in body-worn video. *arXiv preprint arXiv:1904.09062*, 2020.

[88] Yin Li, Zhefan Ye, and James M Rehg. Delving into egocentric actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 287–295, 2015.

[89] Scott Linderman and Ryan Adams. Discovering latent network structure in point process data. In *International Conference on Machine Learning*, pages 1413–1421, 2014.

[90] Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011.

[91] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 1981 DARPA Image Understanding Workshop*, pages 121–130, April 1981.

[92] Xiyang Luo and Andrea L. Bertozzi. Convergence of the graph Allen–Cahn scheme. *Journal of Statistical Physics*, 167(3-4):934–958, 2017.

[93] Minghuang Ma, Haoqi Fan, and Kris M. Kitani. Going deeper into first-person activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1894–1903, 2016.

[94] Yifei Ma, Roman Garnett, and Jeff Schneider. Σ-Optimality for active learning on Gaussian random fields. In *Advances in Neural Information Processing Systems 26*, pages 2751–2759. Curran Associates, Inc., 2013.

[95] Mauro Maggioni and James M. Murphy. Learning by active nonlinear diffusion. *Foundations of Data Science*, 1(3):271–291, 2019.

[96] Benjamin Mark, Garvesh Raskutti, and Rebecca M. Willett. Network estimation from point process data. *IEEE Transactions on Information Theory*, 65(5):2953–2975, 2018.

[97] David Marsan and Olivier Lengline. Extending earthquakes' reach through cascading. *Science*, 319(5866):1076–1079, 2008.

[98] 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*, pages 17–31. Springer, 2016.

[99] Zhaoyi Meng, Javier Sánchez, Jean-Michel Morel, Andrea L. Bertozzi, and P. Jeffrey Brantingham. Ego-motion classification for body-worn videos. In *International Conference on Imaging, Vision and Learning based on Optimization and PDEs*, pages 221–239. Springer, 2016.

[100] Ekaterina Merkurjev, Cristina Garcia-Cardona, Andrea L. Bertozzi, Arjuna Flenner, and Allon G. Percus. Diffuse interface methods for multiclass segmentation of high-dimensional data. *Applied Mathematics Letters*, 33:29–34, 2014.

[101] 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*, pages 689–693. IEEE, 2014.

[102] Barry Merriman, James K. Bence, and Stanley Osher. Motion of multiple junctions: A level-set approach. *Journal of Computational Physics*, 112(2):334–363, June 1994.

[103] Kevin Miller, Hao Li, and Andrea L. Bertozzi. Efficient graph-based active learning with probit likelihood via Gaussian approximations. In *Workshop on Real World Experiment Design and Active Learning, at the International Conference on Machine Learning (ICML)*, 2020. arXiv preprint arXiv:2007.11126.

[104] George Mohler. Marked point process hotspot maps for homicide and gun crime prediction in Chicago. *International Journal of Forecasting*, 30(3):491–497, 2014.

[105] George O. Mohler, Martin B. Short, P. Jeffrey Brantingham, Frederic P. Schoenberg, and George E. Tita. Self-exciting point process modeling of crime. *Journal of the American Statistical Association*, 106(493):100–108, 2011.

[106] George O. Mohler, Martin B. Short, Sean Malinowski, Mark Johnson, George E. Tita, Andrea L. Bertozzi, and P. Jeffrey Brantingham. Randomized controlled field trials of predictive policing. *Journal of the American Statistical Association*, 110(512):1399–1411, 2015.

[107] François Monard, Richard Nickl, and Gabriel P. Paternain. Consistent inversion of noisy non-abelian X-ray transforms. *Communications on Pure and Applied Mathematics*, 2020. DOI: https://doi.org/10.1002/cpa.21942.

[108] Peter J. Mucha, Thomas Richardson, Kevin Macon, Mason A. Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.

[109] John Ashworth Nelder and Robert W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.

[110] Mark E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104, 2006.

[111] Mark E. J. Newman. *Networks.* OUP Oxford; 2nd Edition, 2018.

[112] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856, 2002.

[113] Anastasios Noulas, Salvatore Scellato, Renaud Lambiotte, Massimiliano Pontil, and Cecilia Mascolo. A tale of many cities: Universal patterns in human urban mobility. *PloS one*, 7(5):e37027, 2012.

[114] Yosihiko Ogata. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical Association*, 83(401):9–27, 1988.

[115] Yosihiko Ogata. Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, 50(2):379–402, 1998.

[116] Houman Owhadi, Clint Scovel, Timothy John Sullivan, Mike McKerns, and Michael Ortiz. Optimal uncertainty quantification. *SIAM Review*, 55(2):271–345, 2013.

[117] Fatih Özkan, Mehmet Ali Arabaci, Elif Surer, and Alptekin Temizel. Boosted multiple kernel learning for first-person activity recognition. In *Signal Processing Conference (EUSIPCO), 2017 25th European*, pages 1050–1054. IEEE, 2017.

[118] Tiago P. Peixoto. Bayesian stochastic blockmodeling. *Advances in Network Clustering and Blockmodeling*, pages 289–332, 2019.

[119] Patrick O. Perry and Patrick J. Wolfe. Point process modelling for directed interaction networks. *Journal of the Royal Statistical Society: SERIES B: Statistical Methodology*, pages 821–849, 2013.

[120] Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2847–2854. IEEE, 2012.

[121] Yair Poleg, Chetan Arora, and Shmuel Peleg. Temporal segmentation of egocentric videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2537–2544, 2014.

[122] Yair Poleg, Ariel Ephrat, Shmuel Peleg, and Chetan Arora. Compact CNN for indexing egocentric videos. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016.

[123] Mason A. Porter and James P. Gleeson. *Dynamical Systems on Networks: A Tutorial*, volume 4. Springer International Publishing, 2016.

[124] Mason A. Porter and Sam D. Howison. The role of network analysis in industrial and applied mathematics. *arXiv preprint arXiv:1703.06843*, 2017.

[125] Mason A. Porter, Jukka-Pekka Onnela, and Peter J. Mucha. Communities in networks. *Notices of the AMS*, 56(9):1082–1097, 1164–1166, 2009.

[126] Yiling Qiao, Chang Shi, Chenjian Wang, Hao Li, Matt Haberland, Xiyang Luo, Andrew M. Stuart, and Andrea L. Bertozzi. Uncertainty quantification for semi-supervised multi-class classification in image processing and ego-motion analysis of body-worn videos. *Electronic Imaging*, 2019(11):264–1, 2019.

[127] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2006.

[128] Michael S. Ryoo and Larry Matthies. First-person activity recognition: What are they doing to me? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2730–2737, 2013.

[129] Michael S. Ryoo, Brandon Rothrock, and Larry Matthies. Pooled motion features for first-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 896–904, 2015.

[130] Salvatore Scellato, Anastasios Noulas, and Cecilia Mascolo. Exploiting place features in link prediction on location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1046–1054, 2011.

[131] Frederic Paik Schoenberg, David R. Brillinger, and Peter Guttorp. Point processes, spatial-temporal. *Encyclopedia of Environmetrics*, 3:1573–1577, 2002.

[132] Burr Settles. Active Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, June 2012.

[133] Martin B. Short, P. Jeffrey Brantingham, Andrea L. Bertozzi, and George E. Tita. Dissipation and displacement of hotspots in reaction-diffusion models of crime. In *Proceedings of the National Academy of Sciences*, volume 107, pages 3961–3965. National Acad Sciences, 2010.

[134] Aleksandr Simma and Michael I. Jordan. Modeling events with cascades of Poisson processes. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, UAI'10, page 546–555, Arlington, Virginia, USA, 2010. AUAI Press.

[135] Suriya Singh, Chetan Arora, and C.V. Jawahar. Trajectory aligned features for first person action recognition. *Pattern Recognition*, 62:45–55, 2017.

[136] Ralph C. Smith. *Uncertainty Quantification: Theory, Implementation, and Applications*, volume 12. SIAM, 2013.

[137] Daniel A. Spielman and Shang-Hua Teng. Spectral partitioning works: Planar graphs and finite element meshes. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 96–105. IEEE, 1996.

[138] Daniel A. Spielman and Shang-Hua Teng. Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications*, 421(2-3):284–305, 2007.

[139] Ekaterina H. Spriggs, Fernando De La Torre, and Martial Hebert. Temporal segmentation and activity classification from first-person sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17–24, 2009.

[140] Tiziano Squartini, Francesco Picciolo, Franco Ruzzenenti, and Diego Garlaschelli. Reciprocity of weighted networks. *Scientific Reports*, 3(1):1–9, 2013.

[141] Ingo Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2(Nov):67–93, 2001.

[142] Ingo Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE Transactions on Information Theory*, 51(1):128–142, 2005.

[143] Alexey Stomakhin, Martin B. Short, and Andrea L. Bertozzi. Reconstruction of missing data in social networks based on temporal patterns of interactions. *Inverse Problems*, 27(11):115013, 2011.

[144] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

[145] Timothy John Sullivan. *Introduction to Uncertainty Quantification*, volume 63. Springer, 2015.

[146] Peiyuan Suny, Jianxin Li, Yongyi Mao, Richong Zhang, and Lihong Wang. Inferring multiplex diffusion network via multivariate marked Hawkes process. *arXiv preprint arXiv:1809.07688*, 2018.

[147] Ambuj Tewari and Peter L. Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8:1007–1025, 2007.

[148] George Tita, P. Jeffrey Brantingham, Aram Galstyan, and Yoon-Sik Cho. Latent self-exciting point process model for spatial-temporal networks. *Discrete and Continuous Dynamical Systems - Series B*, 19(5):1335–1354, April 2014.

[149] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2(Nov):45–66, 2001.

[150] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 4489–4497. IEEE, 2015.

[151] Amanda L. Traud, Eric D. Kelsic, Peter J. Mucha, and Mason A. Porter. Comparing community structure to characteristics in online collegiate social networks. *SIAM Review*, 53(3):526–543, 2011.

[152] Nicolás García Trillos, Moritz Gerlach, Matthias Hein, and Dejan Slepčev. Error estimates for spectral convergence of the graph Laplacian on random geometric graphs toward the Laplace–Beltrami operator. *Foundations of Computational Mathematics*, 20(4):827–887, 2020.

[153] Nicolas Garcia Trillos and Dejan Slepčev. A variational approach to the consistency of spectral clustering. *Applied and Computational Harmonic Analysis*, 45(2):239–281, 2018.

[154] Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. The anatomy of the facebook social graph. *arXiv preprint arXiv:1111.4503*, 2011.

[155] Aad W. van der Vaart. *Asymptotic Statistics*, volume 3. Cambridge University Press, 2000.

[156] Aad W. van der Vaart and J. Harry van Zanten. Rates of contraction of posterior distributions based on Gaussian process priors. *The Annals of Statistics*, 36(3):1435–1463, 2008.

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

[158] 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.

[159] Vladimir Vapnik. *Statistical Learning Theory.* Wiley, 1998.

[160] Alejandro Veen and Frederic P. Schoenberg. Estimation of space–time branching process models in seismology using an EM–type algorithm. *Journal of the American Statistical Association*, 103(482):614–624, 2008.

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

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

[163] 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. In *4th Workshop on Mining and Learning from Time Series (MileTS), KDD, London*, 2018.

[164] Xuanhan Wang, Lianli Gao, Jingkuan Song, Xiantong Zhen, Nicu Sebe, and Heng Tao Shen. Deep appearance and motion learning for egocentric activity recognition. *Neurocomputing*, 275:438–447, 2018.

[165] Christopher K.I. Williams and Carl Edward Rasmussen. Gaussian processes for regression. In *Advances in Neural Information Processing Systems*, pages 514–520, 1996.

[166] Mingrui Wu and Bernhard Schölkopf. Transductive classification via local learning regularization. In *Artificial Intelligence and Statistics*, pages 628–635, 2007.

[167] Qiang Wu and Ding-Xuan Zhou. Analysis of support vector machine classification. *Journal of Computational Analysis & Applications*, 8(2), 2006.

[168] Baichuan Yuan, Hao Li, Andrea L Bertozzi, P Jeffrey Brantingham, and Mason A Porter. Multivariate spatiotemporal Hawkes processes and network reconstruction. *SIAM Journal on Mathematics of Data Science*, 1(2):356–382, 2019.

[169] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2005.

[170] Ke Zhou, Hongyuan Zha, and Le Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Artificial Intelligence and Statistics*, pages 641–649, 2013.

[171] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report TR1530, University of Wisconsin-Madison, Computer Sciences Department, 2005. https://minds.wisconsin.edu/bitstream/handle/1793/60444/TR1530.pdf.

[172] Xiaojin Zhu. *Semi-supervised Learning with Graphs*. PhD thesis, Pittsburgh, PA, USA, 2005. AAI3179046.

[173] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine learning (ICML-03)*, pages 912–919, 2003.

[174] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions. In *ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, pages 58–65, 2003.

[175] Jiancang Zhuang, Yosihiko Ogata, and David Vere-Jones. Analyzing earthquake clustering features by using stochastic reconstruction. *Journal of Geophysical Research: Solid Earth*, 109(B5), 2004.