

## Uncertainty Quantification in Graph-Based Classification of High Dimensional Data\*

Andrea L. Bertozzi<sup>†</sup>, Xiyang Luo<sup>†</sup>, Andrew M. Stuart<sup>‡</sup>, and Konstantinos C. Zygalakis<sup>§</sup>

**Abstract.** Classification of high dimensional data finds wide-ranging applications. In many of these applications equipping the resulting classification with a measure of uncertainty may be as important as the classification itself. In this paper we introduce, develop algorithms for, and investigate the properties of a variety of Bayesian models for the task of binary classification; via the posterior distribution on the classification labels, these methods automatically give measures of uncertainty. The methods are all based on the graph formulation of semisupervised learning. We provide a unified framework which brings together a variety of methods that have been introduced in different communities within the mathematical sciences. We study probit classification [C. K. Williams and C. E. Rasmussen, “Gaussian Processes for Regression,” in *Advances in Neural Information Processing Systems 8*, MIT Press, 1996, pp. 514–520] in the graph-based setting, generalize the level-set method for Bayesian inverse problems [M. A. Iglesias, Y. Lu, and A. M. Stuart, *Interfaces Free Bound.*, 18 (2016), pp. 181–217] to the classification setting, and generalize the Ginzburg–Landau optimization-based classifier [A. L. Bertozzi and A. Flenner, *Multiscale Model. Simul.*, 10 (2012), pp. 1090–1118], [Y. Van Gennip and A. L. Bertozzi, *Adv. Differential Equations*, 17 (2012), pp. 1115–1180] to a Bayesian setting. We also show that the probit and level-set approaches are natural relaxations of the harmonic function approach introduced in [X. Zhu et al., “Semi-supervised Learning Using Gaussian Fields and Harmonic Functions,” in ICML, Vol. 3, 2003, pp. 912–919]. We introduce efficient numerical methods, suited to large datasets, for both MCMC-based sampling and gradient-based MAP estimation. Through numerical experiments we study classification accuracy and uncertainty quantification for our models; these experiments showcase a suite of datasets commonly used to evaluate graph-based semisupervised learning algorithms.

**Key words.** graph classification, uncertainty quantification, Gaussian prior

**AMS subject classifications.** 62-09, 65S05, 94-04

**DOI.** 10.1137/17M1134214

### 1. Introduction.

**1.1. The central idea.** Semisupervised learning has attracted the attention of many researchers because of the importance of combining unlabeled data with labeled data. In many

\*Received by the editors June 16, 2017; accepted for publication (in revised form) February 5, 2018; published electronically April 26, 2018.

<http://www.siam.org/journals/juq/6-2/M113421.html>

**Funding:** The work of the first author was funded by NSF grant DMS-1118971, NSF grant DMS-1417674, and ONR grant N00014-16-1-2119. The work of the third author was funded by DARPA (EQUIPS) and EPSRC (Programme Grant). KCZ was partially supported by a grant from the Simons Foundation and by the Alan Turing Institute under the EPSRC grant EP/N510129/1. Part of this work was done during the authors’ stay at the Newton Institute for the program Stochastic Dynamical Systems in Biology: Numerical Methods and Applications.

<sup>†</sup>Department of Mathematics, University of California, Los Angeles, CA ([bertozzi@math.ucla.edu](mailto:bertozzi@math.ucla.edu), [mathluo@math.ucla.edu](mailto:mathluo@math.ucla.edu)).

<sup>‡</sup>Department of Computing and Mathematical Sciences, Caltech, Pasadena, CA ([astuart@caltech.edu](mailto:astuart@caltech.edu)).

<sup>§</sup>School of Mathematics, University of Edinburgh, Edinburgh, Scotland ([k.zygalakis@ed.ac.uk](mailto:k.zygalakis@ed.ac.uk)).

applications the number of unlabeled data points is so large that it is only feasible to label a small subset of these points by hand. Therefore, the problem of effectively utilizing correlation information in unlabeled data to automatically extend the labels to all the data points is an important research question in machine learning. This paper concerns the issue of how to address uncertainty quantification in such classification methods. In doing so we bring together a variety of themes from the mathematical sciences, including optimization, PDEs, probability, and statistics. We will show that a variety of different methods, arising in very distinct communities, can all be formulated around a common objective function,

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

for a real-valued function  $w$  on the nodes of a graph representing the data points. The matrix  $P$  is proportional to a graph Laplacian derived from the unlabeled data, and the function  $\Phi$  involves the labeled data. The variable  $w$  is used for classification. Minimizing this objective function is one approach to such a classification. However, if uncertainty is modeled, then it is natural to consider the probability distribution with density  $\mathbb{P}(w)$  proportional to  $\exp(-J(w))$ ; this will be derived using Bayesian formulations of the problem.

We emphasize that the variety of probabilistic models considered in this paper arises from different assumptions concerning the structure of the data. Our objective is to introduce a unified framework within which to propose and evaluate algorithms to sample  $\mathbb{P}(w)$  or minimize  $J(w)$ . We will focus on the Monte Carlo Markov chain (MCMC) and gradient descent. The objective of the paper is not to assess the validity of the assumptions leading to the different models; this is an important modeling question best addressed through understanding of specific classes of data arising in applications.

**1.2. Literature review.** There are a large number of possible approaches to the semisupervised learning problem, developed in both the statistics and machine learning communities. The review [56] provides an excellent overview. In this paper we will concentrate exclusively on graph-based methods. These have the attractive feature of reducing potentially high dimensional unlabeled data to a real-valued function on the edges of a graph, quantifying affinity between the nodes; each node of the graph is identified by an unlabeled data point.

Early graph-based approaches were combinatorial. Blum et al. posed the binary semisupervised classification problem as a Markov random field (MRF) over the discrete state space of binary labels, the MAP estimation of which can be solved using a graph-cut algorithm in polynomial time [56]. In general, inference for multilabel discrete MRFs is intractable [20]. However, several approximate algorithms exist for the multilabel case [14, 13, 34] and have been applied to many imaging tasks [15, 5, 31].

A different line of work is based on using the affinity function on the edges to define a real-valued function  $w$  on the nodes of the graph, a substantial compression of high dimensional unlabeled data at each node. The Dirichlet energy  $J_0(w) := \frac{1}{2} \langle w, Pw \rangle$ , with  $P$  proportional to the graph Laplacian formed from the affinities on the edges, plays a central role. A key conceptual issue in the graph-based approach is then to connect the labels, which are discrete, to this real-valued function. Strategies to link the discrete and continuous data then constitute different modeling assumptions. The line of work initiated in [57] makes the assumption that

the labels are also real-valued and take the real values  $\pm 1$ , linking them directly to the real-valued function on the nodes of the graph; this may be thought of as a continuum relaxation of the discrete state space MRF in [12]. The basic method is to minimize  $J_0(w)$  subject to the hard constraint that  $w$  agrees with the label values; alternatively this constraint may be relaxed to a soft additional penalty term added to  $J_0(w)$ . These methods are a form of kriging, or Gaussian process regression [50, 51], on a graph. A Bayesian interpretation of the approach in [57] is given in [58], with further developments in [29]. The Laplacian-based approach has since been generalized in [55, 4, 47, 45, 32]; in particular this line of work was developed to study the transductive problem of assigning predictions to data points off the graph. A formal framework for graph-based regularization, using  $J_0(w)$ , can be found in [3, 43]. We also mention related methodologies such as the support vector machine (SVM) [11] and robust convex minimization methods [1, 2], which may be based on minimization of  $J_0(w)$  with an additional soft penalty term; however, since these do not have a Bayesian interpretation we do not consider them here. Other forms of regularization have been considered such as the graph wavelet regularization [42, 24].

The underlying assumption in much of the work described in the previous paragraph is that the labels are real valued. An arguably more natural modeling assumption is that there is a link function, such as the `sign` function, connecting a real-valued function on the graph nodes to the labels via thresholding. This way of thinking underlies the probit approach [51] and the Bayesian level-set method [28, 21], both of which we will study in this paper. Lying between the approaches initiated by [57] and those based on thresholding are the methods based on optimization over real-valued variables which are penalized from taking values far from  $\pm 1$ ; this idea was introduced in the work of Bertozzi et al. [6, 7, 48]. It is based on a Ginzburg–Landau relaxation of the discrete total variation functional, which coincides with the graph-cut energy. This was generalized to multiclass classification in [23]. Following this line of work, several new algorithms were developed for semisupervised and unsupervised classification problems on weighted graphs [27, 35, 38].

The Bayesian way of thinking is foundational in artificial intelligence and machine learning research [11]. However, while the book [51] conducts a number of thorough uncertainty quantification studies for a variety of learning problems using Gaussian process priors, most of the above-mentioned papers studying graph-based learning primarily use the Bayesian approach to learn hyperparameters in an optimization context and do not consider uncertainty quantification. Thus the careful study of, and development of algorithms for, uncertainty quantification in classification remains largely open. There is a wide range of methodologies employed in the field of uncertainty quantification, and the reader may consult the books [44, 46, 53] and the recent article [40] for details and further references. Underlying all of these methods is a Bayesian methodology which we adopt and which is attractive both for its clarity with respect to modeling assumptions and its basis for application of a range of computational tools. Nonetheless it is important to be aware of limitations in this approach, in particular with regard to its robustness with respect to the specification of the model, and in particular the prior distribution on the unknown of interest [39].

**1.3. Our contribution.** In this paper, we focus exclusively on the problem of binary semi-supervised classification; however, the methodology and conclusions will extend beyond this

setting, and to multiclass classification in particular. Our focus is on a presentation which puts uncertainty quantification at the heart of the problem formulation, and we make four primary contributions:

- we define a number of different Bayesian formulations of the graph-based semisupervised learning problem and connect them to one another, to binary classification methods and to a variety of PDE-inspired approaches to classification; in so doing we provide a single framework for a variety of methods which have arisen in distinct communities and open up a number of new avenues of study for the problem area;
- we highlight the pCN-MCMC method for posterior sampling which, based on analogies with its use for PDE-based inverse problems [19], has the potential to sample the posterior distribution in a number of steps which is independent of the number of graph nodes;
- we introduce approximations exploiting the empirical properties of the spectrum of the graph Laplacian, generalizing methods used in the optimization context in [7], allowing for computations at each MCMC step which scale well with respect to the number of graph nodes;
- we demonstrate, by means of numerical experiments on a range of problems, both the feasibility and value of Bayesian uncertainty quantification in semisupervised, graph-based learning using the algorithmic ideas introduced in this paper.

**1.4. Overview and notation.** The paper is organized as follows. In section 2, we give some background material needed for problem specification. In section 3 we formulate the three Bayesian models used for the classification tasks. Section 4 introduces the MCMC and optimization algorithms that we use. In section 5, we present and discuss results of numerical experiments to illustrate our findings; these are based on four examples of increasing size: the house voting records from 1984 (as used in [7]), the tunable two moons dataset [17], the MNIST digit database [30], and the hyperspectral gas plume imaging problem [16]. We conclude in section 6. To aid the reader, we give here an overview of the notation used throughout the paper:

- $Z$ , the set of nodes of the graph, with cardinality  $N$ ;
- $Z'$ , the set of nodes where labels are observed, with cardinality  $J \leq N$ ;
- $x : Z \mapsto \mathbb{R}^d$ , feature vectors;
- $u : Z \mapsto \mathbb{R}$ , latent variable characterizing nodes, with  $u(j)$  denoting evaluation of  $u$  at node  $j$ ;
- $S : \mathbb{R} \mapsto \{-1, 1\}$ , thresholding function;
- $S_\epsilon$ , relaxation of  $S$  using gradient flow in double-well potential  $W_\epsilon$ ;
- $l : Z \mapsto \{-1, 1\}$ , label value at each node with  $l(j) = S(u(j))$ ;
- $y : Z' \mapsto \{-1, 1\}$  or  $y : Z' \mapsto \mathbb{R}$ , label data;
- $v : Z \mapsto \mathbb{R}$ , relaxation of the label variable  $l$ ;
- $A$ , weight matrix of the graph, and  $L$ , resulting normalized graph Laplacian;
- $P$ , precision matrix, and  $C$ , covariance matrix, both found from  $L$ ;
- $\{q_k, \lambda_k\}_{k=0}^{N-1}$ , eigenpairs of  $L$ ;
- $U$ , orthogonal complement of the null space of the graph Laplacian  $L$ , given by  $q_0^\perp$ ;
- GL, Ginzburg–Landau functional;

- $\mu_0, \nu_0$ , prior probability measures;
- measures denoted  $\mu$  typically take argument  $u$  and are real valued; measures denoted  $\nu$  take argument  $l$  on label space, or argument  $v$  on a real-valued relaxation of label space;
- $\mathcal{N}(m, \Sigma)$ , Gaussian random variable with mean  $m$  and covariance  $\Sigma$ ;
- $\mathbb{P}$  and  $\mathbb{E}$ , probability of an event, and the expectation of a random variable, respectively; the underlying probability measure will be made explicit with a subscript when it is necessary to do so.

**2. Problem specification.** In subsection 2.1 we formulate semisupervised learning as a problem on a graph. Subsection 2.2 defines the relevant properties of the graph Laplacian, and in subsection 2.3 these properties are used to construct a Gaussian probability distribution; in section 3 this Gaussian will be used to define our prior information about the classification problem. In subsection 2.4 we discuss thresholding, which provides a link between the real-valued prior information and the label data provided for the semisupervised learning task; in section 3 this will be used to define our likelihood.

**2.1. Semisupervised learning on a graph.** We are given a set of feature vectors  $X = \{x(j), \dots, x(j), \dots, x(N)\}$  for each  $j \in Z := \{1, \dots, N\}$ . For each  $j$  the feature vector  $x(j)$  is an element of  $\mathbb{R}^d$ , so that  $X \in \mathbb{R}^{d \times N}$ . Graph learning starts from the construction of an undirected graph  $G$ , with vertices  $Z$  and edge weights  $\{A\}_{ij} = a_{ij}$  computed from the feature set  $X$ . The weights  $a_{ij}$  will depend only on  $x(i)$  and  $x(j)$  and will decrease monotonically with some measure of distance between  $x(i)$  and  $x(j)$ ; the weights thus encode affinities between nodes of the graph. Although choosing these weights is a very important modeling problem, we do not discuss it here. For graph semisupervised learning, we are also given a partial set of (possibly noisy) labels  $y = \{y(j) | j \in Z'\}$ , where  $Z' \subseteq Z$  has size  $J \leq N$ . The task is to infer the labels for all nodes in  $Z$ , using the weighted graph  $G = (Z, A)$  and also the set of noisily observed labels  $y$ . In the Bayesian formulation which we adopt, the feature set  $X$ , and hence the graph  $G$ , is viewed as prior information, describing correlations among the nodes of the graph, and we combine this with a likelihood based on the noisily observed labels  $y$  to obtain a posterior distribution on the labeling of all nodes. Various Bayesian formulations, which differ in the specification of the observation model and/or the prior, are described in section 3. In the remainder of this section we give the background needed to understand all of these formulations, thereby touching on the graph Laplacian itself, and its link to Gaussian probability distributions and, via thresholding, to non-Gaussian probability distributions and to the Ginzburg–Landau functional. An important point to appreciate is that building our priors from Gaussians confers considerable computational advantages for large graphs; for this reason the non-Gaussian priors will be built from Gaussians via change of measure or push-forward under a nonlinear map.

**2.2. The graph Laplacian.** The graph Laplacian is central to many graph-learning algorithms. There are a number of variants used in the literature; see [7, 49] for a discussion. We will work with the normalized Laplacian, defined from the weight matrix  $A = \{a_{ij}\}$  as follows. We define the diagonal matrix  $D = \text{diag}\{d_{ii}\}$  with entries  $d_{ii} = \sum_{j \in Z} a_{ij}$ . If we assume that the graph  $G$  is connected, then  $d_{ii} > 0$  for all nodes  $i \in Z$ . We can then define the normalized

graph Laplacian<sup>1</sup> as

$$(1) \quad L = I - D^{-1/2}AD^{-1/2}$$

and the graph Dirichlet energy as  $J_0(u) := \frac{1}{2}\langle u, Lu \rangle$ . Then

$$(2) \quad J_0(D^{\frac{1}{2}}u) = \frac{1}{4} \sum_{\{i,j\} \in Z \times Z} a_{ij}(u(i) - u(j))^2.$$

Thus, similarly to the classical Dirichlet energy, this quadratic form penalizes nodes to keep them from having different function values, with the penalty being weighted with respect to the similarity weights from  $A$ . Furthermore the identity shows that  $L$  is positive semidefinite. Indeed the vector of ones  $\mathbb{1}$  is in the null space of  $D - A$  by construction, and hence  $L$  has a zero eigenvalue with corresponding eigenvector  $D^{\frac{1}{2}}\mathbb{1}$ .

We let  $(q_k, \lambda_k)$  denote the eigenpairs of the matrix  $L$ , so that<sup>2</sup>

$$(3) \quad \lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_{N-1} \leq \lambda_{\max} < \infty, \quad \langle q_j, q_k \rangle = \delta_{jk}.$$

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

$$(4) \quad \frac{1}{2}\langle u, Lu \rangle = \frac{1}{2} \sum_{j=1}^{N-1} \lambda_j (\langle u, q_j \rangle)^2;$$

this is analogous to decomposing the classical Dirichlet energy using Fourier analysis.

**2.3. Gaussian measure.** We now show how to build a Gaussian distribution with negative log density proportional to  $J_0(u)$ . Such a Gaussian prefers functions that have larger components on the first few eigenvectors of the graph Laplacian, where the eigenvalues of  $L$  are smaller. The corresponding eigenvectors carry rich geometric information about the weighted graph. For example, the second eigenvector of  $L$  is the *Fiedler vector* and solves a relaxed normalized min-cut problem [49, 26]. The Gaussian distribution thereby connects geometric intuition embedded within the graph Laplacian to a natural probabilistic picture.

To make this connection concrete we define diagonal matrix  $\Sigma$  with entries defined by the vector

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

---

<sup>1</sup>In the majority of the paper the only property of  $L$  that we use is that it is symmetric positive semidefinite. We could therefore use other graph Laplacians, such as the unnormalized choice  $L = D - A$ , in most of the paper. The only exception is the spectral approximation sampling algorithm introduced later; that particular algorithm exploits empirical properties of the symmetrized graph Laplacian. Note, though, that the choice of which graph Laplacian to use can make a significant difference; see [7] and Figure 2.1 therein. To make our exposition more concise we confine our presentation to the graph Laplacian (1).

<sup>2</sup>For the normalized graph Laplacian, the upper bound  $\lambda_{\max} = 2$  may be found in [18, Lemma 1.7, Chapter 1], but with further structure on the weights, many spectra saturate at  $\lambda_{\max} \approx 1$  (see supplementary materials (M113421\_01.pdf [local/web 469KB])), a fact we will exploit later.



and define the positive semidefinite covariance matrix  $C = cQ\Sigma Q^*$ ; the choice of the scaling  $c$  will be discussed below. We let  $\mu_0 := \mathcal{N}(0, C)$ . Note that the covariance matrix is that of a Gaussian with variance proportional to  $\lambda_j^{-1}$  in direction  $q_j$ , thereby leading to structures which are more likely to favor the Fiedler vector ( $j = 1$ ), and lower values of  $j$  in general, than it does for higher values. The fact that the first eigenvalue of  $C$  is zero ensures that any drawn from  $\mu_0$  changes sign, because it will be orthogonal to  $q_0$ .<sup>3</sup> To make this intuition explicit we recall the Karhunen–Loève expansion, which constructs a sample  $u$  from the Gaussian  $\mu_0$  according to the random sum

$$(5) \quad u = c^{\frac{1}{2}} \sum_{j=1}^{N-1} \lambda_j^{-\frac{1}{2}} q_j z_j,$$

where the  $\{z_j\}$  are i.i.d.  $\mathcal{N}(0, 1)$ . Equation (3) thus implies that  $\langle u, q_0 \rangle = 0$ .

We choose the constant of proportionality  $c$  as a rescaling, which enforces the property  $\mathbb{E}|u|^2 = N$  for  $u \sim \mu_0 := \mathcal{N}(0, C)$ ; in other words, the per-node variance is 1. Note that, using the orthogonality of the  $\{q_j\}$ ,

$$(6) \quad \mathbb{E}|u|^2 = c \sum_{j=1}^{N-1} \lambda_j^{-1} \mathbb{E}z_j^2 = c \sum_{j=1}^{N-1} \lambda_j^{-1} \implies c = N \left( \sum_{j=1}^{N-1} \lambda_j^{-1} \right)^{-1}.$$

We reiterate that the support of the measure  $\mu_0$  is the space  $U := q_0^\perp = \text{span}\{q_1, \dots, q_{N-1}\}$  and that, on this space, the probability density function is proportional to

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

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

**2.4. Thresholding and non-Gaussian probability measure.** For the models considered in this paper, the label space of the problem is discrete, while the latent variable  $u$  through which we will capture the correlations amongst nodes of the graph, encoded in the feature vectors, is real-valued. We describe thresholding, and a relaxation of thresholding, to address the need to connect these two differing sources of information about the problem. In what follows, the latent variable  $u : Z \rightarrow \mathbb{R}$  (appearing in the probit and Bayesian level-set methods) is thresholded to obtain the label variable  $l : Z \rightarrow \{-1, 1\}$ . The variable  $v : Z \rightarrow \mathbb{R}$  (appearing in the Ginzburg–Landau method) is a real-valued relaxation of the label variable  $l$ . The variable  $u$  will be endowed with a Gaussian probability distribution. From this the variables  $l$  (which lives on a discrete space) and  $v$  (which is real valued, but concentrated near the discrete space supporting  $l$ ) will be endowed with non-Gaussian probability distributions.

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

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

---

<sup>3</sup>Other treatments of the first eigenvalue are possible and may be useful, but for simplicity of exposition we do not consider them in this paper.

This will be used to connect the latent variable  $u$  with the label variable  $l$ . The function  $S$  may be relaxed by defining  $S_\epsilon(u) = v|_{t=1}$ , where  $v$  solves the gradient flow

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

This will be used, indirectly, to connect the latent variable  $u$  with the real-valued relaxation of the label variable,  $v$ . Note that  $S_\epsilon(\cdot) \rightarrow S(\cdot)$ , pointwise, as  $\epsilon \rightarrow 0$ , on  $\mathbb{R} \setminus \{0\}$ . This reflects the fact that the gradient flow minimizes  $W_\epsilon$ , asymptotically as  $t \rightarrow \infty$ , whenever started on  $\mathbb{R} \setminus \{0\}$ .

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

$$(S^\# \mu_0)(l) = \mu_0\left(\{u | S(u(j)) = l(j) \forall 1 \leq j \leq N\}\right),$$

recalling that  $N$  is the cardinality of  $Z$ . The definition is readily extended to components of  $l$  defined only on subsets of  $Z$ . Thus  $S^\# \mu_0$  is a measure on the label space  $\{-1, 1\}^N$ . The second approach is to work with a change of measure from the Gaussian  $\mu_0$  in such a way that the probability mass on  $U \subset \mathbb{R}^N$  concentrates close to the label space  $\{-1, 1\}^N$ . We may achieve this by defining the measure  $\nu_0$  via its Radon–Nikodym derivative

$$(7) \quad \frac{d\nu_0}{d\mu_0}(v) \propto e^{-\sum_{j \in Z} W_\epsilon(v(j))}.$$

We name  $\nu_0$  the Ginzburg–Landau measure, since the negative log density function of  $\nu_0$  is the graph Ginzburg–Landau functional

$$(8) \quad \text{GL}(v) := \frac{1}{2c} \langle v, Lv \rangle + \sum_{j \in Z} W_\epsilon(v(j)).$$

The Ginzburg–Landau distribution defined by  $\nu_0$  can be interpreted as a nonconvex ground relaxation of the discrete MRF model [56], in contrast to the convex relaxation which is the Gaussian field [57]. Since the double well has minima at the label values  $\{-1, 1\}$ , the probability mass of  $\nu_0$  is concentrated near the modes  $\pm 1$ , and  $\epsilon$  controls this concentration effect.

**3. Bayesian formulation.** In this section we formulate three different Bayesian models for the semisupervised learning problem. The three models all combine the ideas described in the previous section to define three distinct posterior distributions. It is important to realize that these different models will give different answers to the same questions about uncertainty quantification. The choice of which Bayesian model to use is related to the data itself, and making this choice is beyond the scope of this paper. Currently the choice must be addressed on a case-by-case basis, as is done when choosing an optimization method for classification. Nonetheless we will demonstrate that the shared structure of the three models means that a



common algorithmic framework can be adopted, and we will make some conclusions about the relative costs of applying this framework to the three models.

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

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

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

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

We will use as prior the non-Gaussian  $\mathbb{P}(v)dv = \nu_0(dv)$ .

For the probit and level-set models we now explicitly state the prior density  $\mathbb{P}(u)$ , the likelihood function  $\mathbb{P}(y|u)$ , and the posterior density  $\mathbb{P}(u|y)$ ; in the Ginzburg–Landau case  $v$  will replace  $u$ , and we will define the densities  $\mathbb{P}(v)$ ,  $\mathbb{P}(y|v)$ , and  $\mathbb{P}(v|y)$ . Prior and posterior probability measures associated with the letter  $\mu$  are on the latent variable  $u$ ; measures associated with the letter  $\nu$  are on the label space, or real-valued relaxation of the label space.

**3.1. Probit.** The probit method is designed for classification and is described in [51]. In that context Gaussian process priors are used and, unlike the graph Laplacian construction used here, do not depend on the unlabeled data. Combining Gaussian process priors and graph Laplacian priors was suggested and studied in [4, 43, 32]. A recent fully Bayesian treatment of the methodology using unweighted graph Laplacians may be found in the paper [25]. In detail, our model is as follows.

**Prior:** We take as prior on  $u$  the Gaussian  $\mu_0$ . Thus

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

**Likelihood:** For any  $j \in Z'$

$$y(j) = S\left(u(j) + \eta(j)\right),$$

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

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

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

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

similarly

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

**Posterior:** Bayes' theorem gives posterior  $\mu_p$  with probability density function (pdf)

$$\mathbb{P}_p(u|y) \propto \exp\left(-\frac{1}{2}\langle u, Pu \rangle - \Phi_p(u; y)\right),$$

where

$$\Phi_p(u; y) := -\sum_{j \in Z'} \log\left(\Psi(y(j)u(j); \gamma)\right).$$

We let  $\nu_p$  denote the push-forward under  $S$  of  $\mu_p$ :  $\nu_p = S^\# \mu_p$ .

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

$$J_p(u) = \frac{1}{2}\langle u, Pu \rangle - \sum_{j \in Z'} \log\left(\Psi(y(j)u(j); \gamma)\right).$$

This is a convex function, a fact which is well known in related contexts, but which we state and prove in Proposition 1, section 2, of the supplementary materials (M113421\_01.pdf [local/web 469KB]) for the sake of completeness. In view of the close relationship between this problem and the level-set formulation described next, for which there are no minimizers, we expect that minimization may not be entirely straightforward in the  $\gamma \ll 1$  limit. This is manifested in the presence of near-flat regions in the probit log likelihood function when  $\gamma \ll 1$ .

Our variant on the probit methodology differs from that in [25] in several ways: (i) our prior Gaussian is scaled to have per-node variance one, while in [25] the per-node variance is a hyperparameter to be determined; (ii) our prior is supported on  $U = q_0^\perp$ , while in [25] the prior precision is found by shifting  $L$  and taking a possibly fractional power of the resulting matrix, resulting in support on the whole of  $\mathbb{R}^N$ ; (iii) we allow for a scale parameter  $\gamma$  in the observational noise, while in [25] the parameter  $\gamma = 1$ .

**3.2. Level set.** This method is designed for problems considerably more general than classification on a graph [28]. For the current application, this model is exactly the same as probit except for the order in which the noise  $\eta(j)$  and the thresholding function  $S(u)$  are applied in the definition of the data. Thus we again take as prior for  $u$  the Gaussian  $\mu_0$ . Then we have the following.

**Likelihood:** For any  $j \in Z'$

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

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

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

**Posterior:** Bayes' theorem gives posterior  $\mu_{\text{ls}}$  with pdf

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

where

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

We let  $\nu_{\text{ls}}$  denote the push-forward under  $S$  of  $\mu_{\text{ls}} : \nu_{\text{ls}} = S^{\#}\mu_{\text{ls}}$ .

**MAP estimator functional:** The negative of the log posterior is, in this case, given by

$$J_{\text{ls}}(u) = \frac{1}{2} \langle u, Pu \rangle + \Phi_{\text{ls}}(u; y).$$

However, unlike the probit model, the Bayesian level-set method has no MAP estimator—the infimum of  $J_{\text{ls}}$  is not attained, and this may be seen by noting that if the infimum were attained at any nonzero point  $u^*$ , then  $\epsilon u^*$  would reduce the objective function for any  $\epsilon \in (0, 1)$ ; however, the point  $u^* = 0$  does not attain the infimum. This proof is detailed in [28] for a closely related PDE-based model, and the proof is easily adapted.

**3.3. Ginzburg–Landau.** For this model, we take as prior the Ginzburg–Landau measure  $\nu_0$  defined by (7) and employ a Gaussian likelihood for the observed labels. This construction gives the Bayesian posterior whose MAP estimator is the objective function introduced and studied in [7, 33].

**Prior:** We define the prior on  $v$  to be the Ginzburg–Landau measure  $\nu_0$  given by (7) with density

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

**Likelihood:** For any  $j \in Z'$

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

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

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

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

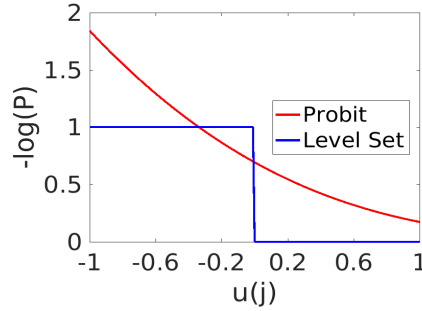
$$\begin{aligned} \mathbb{P}_{\text{gl}}(v|y) &\propto \exp\left(-\frac{1}{2} \langle v, Pv \rangle - \Phi_{\text{gl}}(v; y)\right), \\ \Phi_{\text{gl}}(v; y) &:= \sum_{j \in Z} W_{\epsilon}(v(j)) + \sum_{j \in Z'} \left( \frac{1}{2\gamma^2} |y(j) - v(j)|^2 \right). \end{aligned}$$

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

$$J_{\text{gl}}(v) = \frac{1}{2} \langle v, Pv \rangle + \Phi_{\text{gl}}(v; y).$$

This objective function was introduced in [7] as a relaxation of the min-cut problem, penalized by data; the relationship to min-cut was studied rigorously in [48]. The minimization problem for  $J_{\text{gl}}$  is nonconvex and has multiple minimizers, reflecting the combinatorial character of the min-cut problem of which it is a relaxation.

**3.4. Uncertainty quantification for graph-based learning.** In Figure 1 we plot the component of the negative log likelihood at a labeled node  $j$ , as a function of the latent variable  $u = u(j)$  with data  $y = y(j)$  fixed, for the probit and Bayesian level-set models. The log likelihood for the Ginzburg–Landau formulation is not directly comparable, as it is a function of the relaxed label variable  $v(j)$ , with respect to which it is quadratic with minimum at the data point  $y(j)$ .



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

The probit and Bayesian level-set models lead to posterior distributions  $\mu$  (with different subscripts) in latent variable space, and push-forwards under  $S$ , denoted  $\nu$  (also with different subscripts), in label space. The Ginzburg–Landau formulation leads to a measure  $\nu_{gl}$  in (relaxed) label space. Uncertainty quantification for semisupervised learning is concerned with completely characterizing these posterior distributions. In practice this may be achieved by sampling using MCMC methods. In this paper we will study four measures of uncertainty:

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

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

$$(9) \quad s_j^l := \mathbb{E}_\nu(l(j)),$$

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

$$\text{Var}_\nu(l(j)) = 4q(1 - q) = 1 - (s_j^l)^2.$$

Later we will find it useful to consider the variance averaged over all nodes and hence define<sup>4</sup>

$$(10) \quad \text{Var}(l) = \frac{1}{N} \sum_{j=1}^N \text{Var}_\nu(l(j)).$$

Note that the maximum value obtained by  $\text{Var}(l)$  is 1. This maximum value is attained under the Gaussian prior  $\mu_0$  that we use in this paper. The deviation from this maximum under the posterior is a measure of the information content of the labeled data. Note, however, that the prior does contain information about classifications in the form of correlations between vertices; this is not captured in (10).

**4. Algorithms.** From section 3, we see that for all of the models considered, the posterior  $\mathbb{P}(w|y)$  has the form

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

for some function  $\Phi$ , different for each of the three models (acknowledging that in the Ginzburg–Landau case the independent variable is  $w = v$ , a real-valued relaxation of label space, whereas for the other models  $w = u$ , an underlying latent variable which may be thresholded by  $S(\cdot)$  into label space). Furthermore, the MAP estimator is the minimizer of  $J$ . Note that  $\Phi$  is differentiable for the Ginzburg–Landau and probit models, but not for the level-set model. We introduce algorithms for both sampling (MCMC) and MAP estimation (optimization) that apply in this general framework. The sampler we employ does not use information about the gradient of  $\Phi$ ; the MAP estimation algorithm does, but is only employed on the Ginzburg–Landau and probit models. Both sampling and optimization algorithms use spectral properties of the precision matrix  $P$ , which is proportional to the graph Laplacian  $L$ .

**4.1. MCMC.** Broadly speaking there are two strong competitors as samplers for this problem: Metropolis–Hastings-based methods, and Gibbs-based samplers. In this paper we focus entirely on Metropolis–Hastings methods, as they may be used on all three of the models considered here. In order to induce scalability with respect to the size of  $Z$  we use the preconditioned Crank–Nicolson (pCN) method described in [19] and introduced in the context of diffusions by Beskos et al. in [10] and by Neal in the context of machine learning [37]. The method is also robust with respect to the small noise limit  $\gamma \rightarrow 0$  in which the label data is perfect. The pCN-based approach is compared with Gibbs-like methods for probit, to which they both apply, in [8]; both large data sets  $N \rightarrow \infty$  and small noise  $\gamma \rightarrow 0$  limits are considered.

The standard random walk Metropolis (RWM) algorithm suffers from the fact that the optimal proposal variance or stepsize scales inverse proportionally to the dimension of the state space [41], which is the graph size  $N$  in this case. The pCN method is designed so that the proposal variance required to obtain a given acceptance probability scales independently of the dimension of the state space (here the number of graph nodes  $N$ ), hence in practice giving faster convergence of the MCMC when compared with RWM [9]. We restate the pCN

---

<sup>4</sup>Strictly speaking,  $\text{Var}(l) = N^{-1} \text{Tr}(\text{Cov}(l))$ .

method as Algorithm 1 and then follow with various variants of it in Algorithms 2 and 3. In all three algorithms  $\beta \in [0, 1]$  is the key parameter which determines the efficiency of the MCMC method: small  $\beta$  leads to high acceptance probability but small moves; large  $\beta$  leads to low acceptance probability and large moves. Somewhere between these extremes is an optimal choice of  $\beta$  which minimizes the asymptotic variance of the algorithm when applied to compute a given expectation.

---

**Algorithm 1.** pCN algorithm.

---

- 1: Input:  $L$ .  $\Phi(u)$ .  $u^{(0)} \in U$ .
  - 2: Output:  $M$  approximate samples from the posterior distribution.
  - 3: Define:  $\alpha(u, w) = \min\{1, \exp(\Phi(u) - \Phi(w))\}$ .
  - 4: **while**  $k < M$  **do**
  - 5:    $w^{(k)} = \sqrt{1 - \beta^2}u^{(k)} + \beta\xi^{(k)}$ , where  $\xi^{(k)} \sim \mathcal{N}(0, C)$  via (11).
  - 6:   Calculate acceptance probability  $\alpha(u^{(k)}, w^{(k)})$ .
  - 7:   Accept  $w^{(k)}$  as  $u^{(k+1)}$  with probability  $\alpha(u^{(k)}, w^{(k)})$ , otherwise  $u^{(k+1)} = u^{(k)}$ .
  - 8: **end while**
- 

The value  $\xi^{(k)}$  is a sample from the prior  $\mu_0$ . If the eigenvalues and eigenvectors of  $L$  are all known, then the Karhunen–Loève expansion (11) gives

$$(11) \quad \xi^{(k)} = c^{\frac{1}{2}} \sum_{j=1}^{N-1} \lambda_j^{-\frac{1}{2}} q_j z_j,$$

where  $c$  is given by (6), the  $z_j, j = 1, \dots, N - 1$ , are i.i.d. centered unit Gaussians, and the equality is in law.

**4.2. Spectral projection.** For graphs with a large number of nodes  $N$ , it is prohibitively costly to directly sample from the distribution  $\mu_0$ , since doing so involves knowledge of a complete eigendecomposition of  $L$  in order to employ (11). A method that is frequently used in classification tasks is to restrict the support of  $u$  to the eigenspace spanned by the first  $\ell$  eigenvectors with the smallest nonzero eigenvalues of  $L$  (hence largest precision), and this idea may be used to approximate the pCN method; this leads to a low rank approximation. In particular we approximate samples from  $\mu_0$  by

$$(12) \quad \xi_\ell^{(k)} = c_\ell^{\frac{1}{2}} \sum_{j=1}^{\ell-1} \lambda_j^{-\frac{1}{2}} q_j z_j,$$

where  $c_\ell$  is given by (6) truncated after  $j = \ell - 1$ , the  $z_j$  are i.i.d. centered unit Gaussians, and the equality is in law. This is a sample from  $\mathcal{N}(0, C_\ell)$ , where  $C_\ell = c_\ell Q \Sigma_\ell Q^*$  and the diagonal entries of  $\Sigma_\ell$  are set to zero for the entries after  $\ell$ . In practice, to implement this algorithm, it is only necessary to compute the first  $\ell$  eigenvectors of the graph Laplacian  $L$ . This gives Algorithm 2.

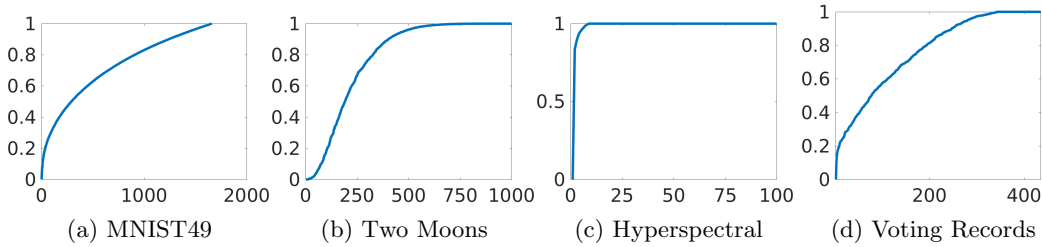
**4.3. Spectral approximation.** Spectral projection often leads to good classification results, but may lead to reduced posterior variance and a posterior distribution that is overly

---

**Algorithm 2.** pCN algorithm with spectral projection.

---

- 1: Input:  $L$ .  $\Phi(u)$ .  $u^{(0)} \in U$ .
  - 2: Output:  $M$  approximate samples from the posterior distribution.
  - 3: Define:  $\alpha(u, w) = \min\{1, \exp(\Phi(u) - \Phi(w))\}$ .
  - 4: **while**  $k < M$  **do**
  - 5:    $w^{(k)} = \sqrt{1 - \beta^2}u^{(k)} + \beta\xi_\ell^{(k)}$ , where  $\xi_\ell^{(k)} \sim \mathcal{N}(0, C_\ell)$  via (12).
  - 6:   Calculate acceptance probability  $\alpha(u^{(k)}, w^{(k)})$ .
  - 7:   Accept  $w^{(k)}$  as  $u^{(k+1)}$  with probability  $\alpha(u^{(k)}, w^{(k)})$ , otherwise  $u^{(k+1)} = u^{(k)}$ .
  - 8: **end while**
- 



**Figure 2.** Spectra of graph Laplacian of various datasets. See section 5 for the description of the datasets and graph construction parameters. The y-axis indicates the eigenvalues, and the x-axis the index of ordering.

smooth on the graph domain. We propose an improvement to the method that preserves the variability of the posterior distribution but still only involves calculating the first  $\ell$  eigenvectors of  $L$ . This is based on the empirical observation that in many applications the spectrum of  $L$  saturates and satisfies, for  $j \geq \ell$ ,  $\lambda_j \approx \bar{\lambda}$  for some  $\bar{\lambda}$ . Such behavior may be observed in (b), (c), and (d) of Figure 2; in particular note that in the hyperspectral case  $\ell \ll N$ . We assume such behavior in deriving the low rank approximation used in this subsection. (See supplementary materials (M113421\_01.pdf [local/web 469KB]) for a detailed discussion of the graph Laplacian spectrum.) We define  $\Sigma_{\ell,o}$  by overwriting the diagonal entries of  $\Sigma$  from  $\ell$  to  $N - 1$  with  $\bar{\lambda}^{-1}$ . We then set  $C_{\ell,o} = c_{\ell,o}Q\Sigma_{\ell,o}Q^*$  and generate samples from  $\mathcal{N}(0, C_{\ell,o})$  (which are approximate samples from  $\mu_0$ ) by setting

$$(13) \quad \xi_{\ell,o}^{(k)} = c_{\ell,o}^{\frac{1}{2}} \sum_{j=1}^{\ell-1} \lambda_j^{-\frac{1}{2}} q_j z_j + c_{\ell,o}^{\frac{1}{2}} \bar{\lambda}^{-\frac{1}{2}} \sum_{j=\ell}^{N-1} q_j z_j,$$

where  $c_{\ell,o}$  is given by (6) with  $\lambda_j$  replaced by  $\bar{\lambda}$  for  $j \geq \ell$ , the  $\{z_j\}$  are centered unit Gaussians, and the equality is in law. Importantly samples according to (13) can be computed very efficiently. In particular there is no need to compute  $q_j$  for  $j \geq \ell$ , and the quantity  $\sum_{j=\ell}^{N-1} q_j z_j$  can be computed by first taking a sample  $\bar{z} \sim \mathcal{N}(0, I_N)$  and then projecting  $\bar{z}$  onto  $U_\ell := \text{span}(q_\ell, \dots, q_{N-1})$ . Moreover, projection onto  $U_\ell$  can be computed only using  $\{q_1, \dots, q_{\ell-1}\}$ , since the vectors span the orthogonal complement of  $U_\ell$ . Concretely, we have

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



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

$$(14) \quad \xi_{\ell,o}^{(k)} = c_{\ell,o}^{\frac{1}{2}} \sum_{j=1}^{\ell-1} \lambda_j^{-\frac{1}{2}} q_j z_j + c_{\ell,o}^{\frac{1}{2}} \bar{\lambda}^{-\frac{1}{2}} \left( \bar{z} - \sum_{j=1}^{\ell-1} q_j \langle q_j, \bar{z} \rangle \right).$$

The vector  $\xi_{\ell,o}^{(k)}$  is a sample from  $\mathcal{N}(0, C_{\ell,o})$  and results in Algorithm 3. Under the stated empirical properties of the graph Laplacian, we expect this to be a better approximation of the prior covariance structure than the approximation of the previous subsection.

---

**Algorithm 3.** pCN algorithm with spectral approximation.

---

- 1: Input:  $L, \Phi(u), u^{(0)} \in U$ .
  - 2: Output:  $M$  approximate samples from the posterior distribution.
  - 3: Define:  $\alpha(u, w) = \min\{1, \exp(\Phi(u) - \Phi(w))\}$ .
  - 4: **while**  $k < M$  **do**
  - 5:    $w^{(k)} = \sqrt{1 - \beta^2} u^{(k)} + \beta \xi_{\ell,o}^{(k)}$ , where  $\xi_{\ell,o}^{(k)} \sim \mathcal{N}(0, C_{\ell,o})$  via (14).
  - 6:   Calculate acceptance probability  $\alpha(u^{(k)}, w^{(k)})$ .
  - 7:   Accept  $w^{(k)}$  as  $u^{(k+1)}$  with probability  $\alpha(u^{(k)}, w^{(k)})$ , otherwise  $u^{(k+1)} = u^{(k)}$ .
  - 8: **end while**
- 

**4.4. MAP estimation: Optimization.** Recall that the objective function for the MAP estimation has the form  $\frac{1}{2} \langle u, Pu \rangle + \Phi(u)$ , where  $u$  is supported on the space  $U$ . For Ginzburg–Landau and probit, the function  $\Phi$  is smooth, and we can use a standard projected gradient method for the optimization. Since  $L$  is typically ill-conditioned, it is preferable to use a semi-implicit discretization as suggested in [7], since convergence to a stationary point can be shown under a graph-independent learning rate. Furthermore, the discretization can be performed in terms of the eigenbasis  $\{q_1, \dots, q_{N-1}\}$ , which allows us to easily apply spectral projection when only a truncated set of eigenvectors is available. We state the algorithm in terms of the (possibly truncated) eigenbasis below. Here  $P_\ell$  is an approximation to  $P$  found by setting  $P_\ell = Q_\ell D_\ell Q_\ell^*$ , where  $Q_\ell$  is the matrix with columns  $\{q_1, \dots, q_{\ell-1}\}$  and  $D_\ell = \text{diag}(d)$  for  $d(j) = c_\ell \lambda_j, j = 1, \dots, \ell - 1$ . Thus  $P_{N-1} = P$ .

---

**Algorithm 4.** Linearly implicit gradient flow with spectral projection.

---

- 1: **Input:**  $Q_m = (q_1, \dots, q_m), \Lambda_m = (\lambda_1, \dots, \lambda_m), \Phi(u), u^{(0)} \in U$ .
  - 2: **while**  $k < M$  **do**
  - 3:    $u^{(*)} = u^{(k)} - \beta \nabla \Phi(u^{(k)})$ .
  - 4:    $u^{(k+1)} = (I + \beta P_m)^{-1} u^{(*)}$ .
  - 5: **end while**
- 

**5. Numerical experiments.** In this section we conduct a series of numerical experiments on four different datasets that are representative of the field of graph semisupervised learning. There are three main purposes for the experiments. First, we perform uncertainty quantification, as explained in subsection 3.4. Second, we study the spectral approximation and

projection variants on pCN sampling, as these scale well to massive graphs. Finally, we make some observations about the cost and practical implementation details of these methods for the different Bayesian models we adopt; these will help guide the reader in making choices about which algorithm to use. We present the results for MAP estimation in section 2 of the supplementary materials (M113421\_01.pdf [local/web 469KB]), alongside the proof of convexity of the probit MAP estimator.

The quality of the graph constructed from the feature vectors is central to the performance of any graph learning algorithms. In the experiments below, we follow the graph construction procedures used in the previous papers [7, 27, 35], which applied graph semisupervised learning to all of the datasets that we consider in this paper. Moreover, we have verified that for all the reported experiments below, the graph parameters are in a range such that spectral clustering [49] (an unsupervised learning method) gives a reasonable performance. The methods we employ lead to refinements over spectral clustering (improved classification) and, of course, to uncertainty quantification (which spectral clustering does not address).

**5.1. Datasets.** We introduce the datasets and describe the graph construction for each dataset. In all cases we numerically construct the weight matrix  $A$ , and then the graph Laplacian  $L$ .<sup>5</sup>

**5.1.1. Two moons.** The two moons artificial dataset is constructed to give noisy data which lies near a nonlinear low dimensional manifold embedded in a high dimensional space [17]. The dataset is constructed by sampling  $N$  data points uniformly from two semicircles centered at  $(0, 0)$  and  $(1, 0.5)$  with radius 1, embedding the data in  $\mathbb{R}^d$ , and adding Gaussian noise with standard deviation  $\sigma$ . We set  $N = 2,000$  and  $d = 100$  in this paper; recall that the graph size is  $N$  and each feature vector has length  $d$ . We will conduct a variety of experiments with differently labeled data size  $J$ , and in particular study variation with  $J$ . The default value, when not varied, is  $J$  at 3% of  $N$ , with the labeled points chosen at random.

We take each data point as a node on the graph and construct a fully connected graph using the self-tuning weights of Zelnik-Manor and Perona [54], with  $K = 10$ . Specifically we let  $x_i, x_j$  be the coordinates of the data points  $i$  and  $j$ . Then weight  $a_{ij}$  between nodes  $i$  and  $j$  is defined by

$$(15) \quad a_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\tau_i\tau_j}\right),$$

where  $\tau_j$  is the distance of the  $K$ th closest point to the node  $j$ .

**5.1.2. House voting records from 1984.** This dataset contains the voting records of 435 U.S. House of Representatives; for details see [7] and the references therein. The votes were recorded in 1984 from the second session of the 98th United States Congress. The votes for each individual are vectorized by mapping a yes vote to 1, a no vote to  $-1$ , and an abstention/no-show to 0. The dataset contains 16 votes that are believed to be well correlated with partisanship, and we use only these votes as feature vectors for constructing the graph. Thus the graph size is  $N = 435$ , and feature vectors have length  $d = 16$ . The goal is to predict

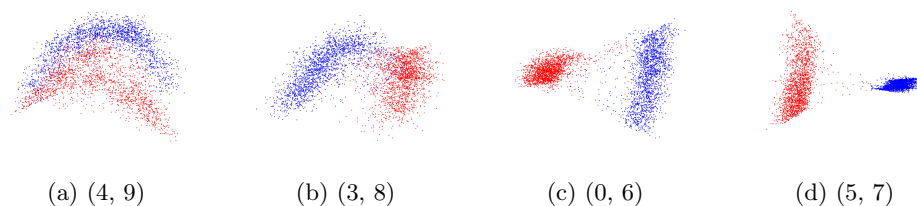
---

<sup>5</sup>The weight matrix  $A$  is symmetric in theory; in practice we find that symmetrizing via the map  $A \mapsto \frac{1}{2}A + \frac{1}{2}A^*$  is helpful.

the party affiliation of each individual, given a small number of known affiliations (labels). We pick 3 Democrats and 2 Republicans at random to use as the observed class labels; thus  $J = 5$ , corresponding to less than 1.2% of fidelity (i.e., labeled) points. We construct a fully connected graph with weights given by (15) with  $\tau_j = \tau = 1.25$  for all nodes  $j$ .

**5.1.3. MNIST.** The MNIST database consists of 70,000 images of size  $28 \times 28$  pixels containing the handwritten digits 0 through 9; see [30] for details. Since in this paper we focus on binary classification, we only consider pairs of digits. To speed up calculations, we subsample randomly 2,000 images from each digit to form a graph with  $N = 4,000$  nodes; we use this for all our experiments, except in subsection 5.4, where we use the full dataset of size  $N = \mathcal{O}(10^4)$  for digit pair (4, 9) to benchmark computational cost. The nodes of the graph are the images, and for feature vectors we project the images onto the leading 50 principal components given by PCA; thus the feature vectors at each node have length  $d = 50$ . We construct a  $K$ -nearest neighbor graph with  $K = 20$  for each pair of digits considered. Namely, the weights  $a_{ij}$  are nonzero if and only if one of  $i$  or  $j$  is in the  $K$  nearest neighbors of the other. The nonzero weights are set using (15) with  $K = 20$ .

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



**Figure 3.** Visualization of data by projection onto 2nd and 3rd eigenfunctions of the graph Laplacian for the MNIST data set, where the vertical dimension is the 3rd eigenvector and the horizontal dimension the 2nd. Each subfigure represents a different pair of digits. We construct a 20 nearest neighbor graph under the Zelnik-Manor and Perona scaling [54] as in (15) with  $K = 20$ .

**5.1.4. Hyperspectral image.** The hyperspectral dataset analyzed for this project was provided by the Applied Physics Laboratory at Johns Hopkins University; see [16] for details. It consists of a series of video sequences recording the release of chemical plumes taken at the Dugway Proving Ground. Each layer in the spectral dimension depicts a particular frequency starting at 7,830 nm and ending with 11,700 nm, with a channel spacing of 30 nm, giving 129 channels; thus the feature vector has length  $d = 129$ . The spatial dimension of each frame is  $128 \times 320$  pixels. We select 7 frames from the video sequence as the input data, and consider each spatial pixel as a node on the graph. Thus the graph size is  $N = 128 \times 320 \times 7 = 286,720$ . Note that time ordering of the data is ignored. The classification problem is to classify pixels that represent the chemical plumes against pixels that are the background.

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

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

This distance is small for vectors that point in the same direction, and is insensitive to their magnitude. We consider the normalized Laplacian defined in (1). Because it is computationally prohibitive to compute eigenvectors of a Laplacian of this size, we apply the Nyström extension [52, 22] to obtain an approximation to the true eigenvectors and eigenvalues; see [7] for details pertinent to the setup here. We emphasize that each pixel in the 7 frames is a node on the graph and that, in particular, pixels across the 7 time frames are also connected. Since we have no ground truth labels for this dataset, we generate known labels by setting the segmentation results from spectral clustering as ground truth. The default value of  $J$  is 8,000, and labels are chosen at random. This corresponds to labeling around 2.8% of the points. We plot results for only the last 3 frames of the video sequence in order to ensure that the information in the figures is not overwhelmingly large.

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

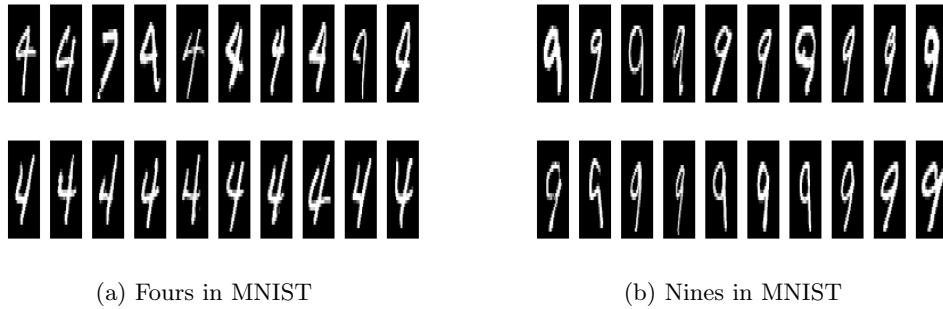
**5.2.1. Posterior mean as confidence scores.** We construct the graph from the MNIST (4, 9) dataset following subsection 5.1. The noise variance  $\gamma$  is set to 0.1, and 4% of fidelity points are chosen randomly from each class. The probit posterior is used to compute (9). In Figure 4 we demonstrate that nodes with scores  $s_j^l$  closer to the binary ground truth labels  $\pm 1$  look visually more uniform than nodes with  $s_j^l$  far from those labels. This shows that the posterior mean contains useful information which differentiates between outliers and inliers that align with human perception. The scores  $s_j^l$  are computed as follows: We let  $\{u^{(k)}\}_{k=1}^M$  be a set of samples of the posterior measure obtained from the pCN algorithm. The probability  $\mathbb{P}(S(u(j)) = l(j))$  is approximated by

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

for each  $j$ . Finally, the score

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

**5.2.2. Posterior variance as uncertainty measure.** In this set of experiments, we show that the posterior distribution of the label variable  $l = S(u)$  captures the uncertainty of the classification problem. We use the posterior variance of  $l$ , averaged over all nodes, as a measure



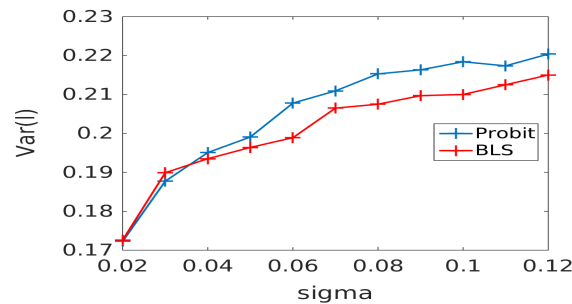
**Figure 4.** “Hard to classify” versus “easy to classify” nodes in the MNIST (4, 9) dataset under the probit model. Here the digit “4” is labeled +1 and “9” is labeled −1. The top (bottom) row of the left column corresponds to images that have the lowest (highest) values of  $s_j^l$  defined in (9) among images that have ground truth labels “4”. The right column is organized in the same way for images with ground truth labels 9, except that the top row now corresponds to the highest values of  $s_j^l$ . Higher  $s_j^l$  indicates higher confidence that image  $j$  is a 4 and not a “9”; hence the top row could be interpreted as images that are “hard to classify” by the current model, and vice versa for the bottom row. The graph is constructed as in section 5, and  $\gamma = 0.1$ ,  $\beta = 0.3$ .

of the model variance, specifically formula (10). We study the behavior of this quantity as we vary the level of uncertainty within certain inputs to the problem. We demonstrate empirically that the posterior variance is approximately monotonic with respect to variations in the levels of uncertainty in the input data, as it should be, and thus that the posterior variance contains useful information about the classification. We select quantities that reflect the separability of the classes in the feature space.

Figure 5 plots the posterior variance  $\text{Var}(l)$  against the standard deviation  $\sigma$  of the noise appearing in the feature vectors for the two moons dataset; thus points generated on the two semicircles overlap more as  $\sigma$  increases. We employ a sequence of posterior computations, using the probit and Bayesian level-set models, for  $\sigma = 0.02 : 0.01 : 0.12$ . Recall that  $N = 2,000$ , and we choose 3% of the nodes to have the ground truth labels as observed data. Within both models,  $\gamma$  is fixed at 0.1. A total of  $1 \times 10^4$  samples are taken, and the proposal variance  $\beta$  is set to 0.3. We see that the mean posterior variance increases with  $\sigma$ , as is intuitively reasonable. Furthermore, because  $\gamma$  is small, the probit and Bayesian level-set models are very similar, and this is reflected in the similar quantitative values for uncertainty.

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

The previous two experiments in this subsection have studied posterior label variance  $\text{Var}(l)$  as a function of variation in the prior data. We now turn to study how posterior variance changes as a function of varying the likelihood information, again for both the two moons and MNIST datasets. In Figures 6(a) and 6(b), we plot the posterior label variance against the percentage of nodes observed. We observe that the observational variance decreases as the



**Figure 5.** Mean posterior variance defined in (10) versus feature noise  $\sigma$  for the probit model and the Bayesian level-set model (BLS) applied to the two moons dataset with  $N = 2,000$ . For each trial, a realization of the two moons dataset under the given parameter  $\sigma$  is generated, where  $\sigma$  is the Gaussian noise on the features defined in subsection 5.1.1, and 3% of nodes are randomly chosen as fidelity. We run 20 trials for each value of  $\sigma$  and average the mean posterior variance across the 20 trials in the figure. We set  $\gamma = 0.1$  and  $\beta = 0.3$  for both models.

**Table 1**

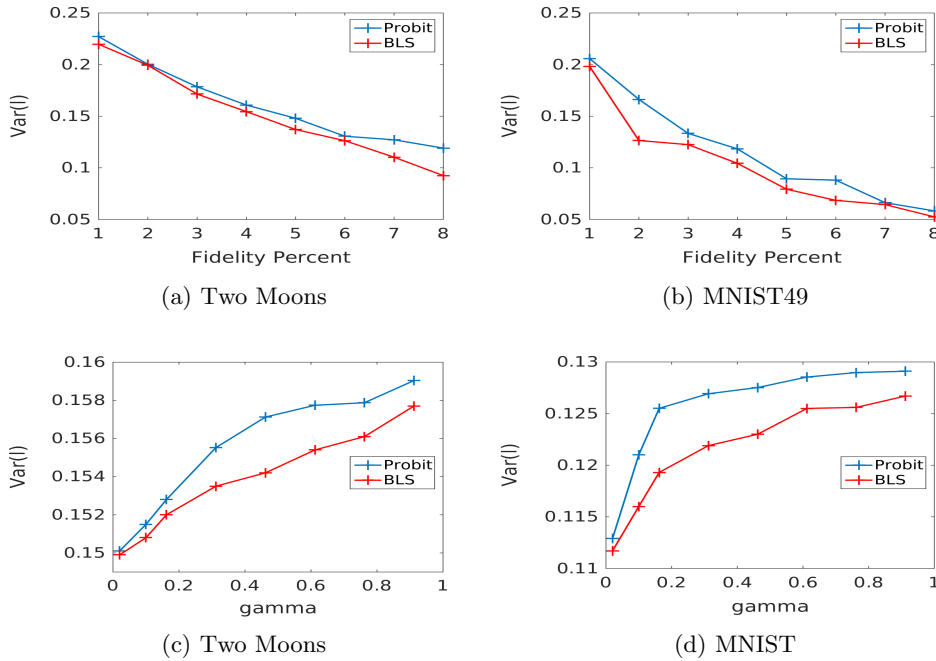
Mean posterior variance of different digit pairs for the probit model and the Bayesian level-set model (BLS) applied to the MNIST dataset. The pairs are organized from left to right according to the separability of the two classes as shown in Figures 3(a)–3(d). For each trial, we randomly select 4% of nodes as fidelity. We run 10 trials for each pair of digits and average the mean posterior variance across trials. We set  $\gamma = 0.1$  and  $\beta = 0.3$  for both models.

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

amount of labeled data increases. Figures 6(c) and 6(d) show that the posterior label variance increases almost monotonically as observational noise  $\gamma$  increases. Furthermore the level set and probit formulations produce similar answers for  $\gamma$  small, reflecting the close similarity between those methods when  $\gamma$  is small; when  $\gamma = 0$  their likelihoods coincide.

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

**5.2.3. Visualization of marginal posterior density.** In this subsection, we contrast the posterior distribution  $\mathbb{P}(v|y)$  of the Ginzburg–Landau model with that of the probit and Bayesian level-set models. The graph is constructed from the voting records data, with the fidelity points chosen as described in subsection 5.1. In Figure 7 we plot the histograms of the empirical marginal posterior distribution on  $\mathbb{P}(v(i)|y)$  and  $\mathbb{P}(u(i)|y)$  for a selection of nodes on the graph. For the top row of Figure 7, we select 6 nodes with “low confidence” predictions and plot the empirical marginal distribution of  $u$  for probit and Bayesian level-set models, and that of  $v$  for the Ginzburg–Landau model. Note that the same set of nodes is chosen for different models. The plots in this row demonstrate the multimodal nature of the



**Figure 6.** Mean posterior variance as in (10) versus percentage of labeled points and noise level  $\gamma$  for the probit model and the Bayesian level-set model (BLS) applied to the two moons dataset and the 4-9 MNIST dataset. For two moons, we fix  $N = 2,000$  and  $\sigma = 0.06$ . For each trial, we generate a realization of the two moons dataset, while the MNIST dataset is fixed. For (a), (b)  $\gamma$  is fixed at 0.1, and a certain percentage of nodes are selected at random as labeled. For (c), (d), the proportion of labeled points is fixed at 4%, and  $\gamma$  is varied across a range. Results are averaged over 20 trials.

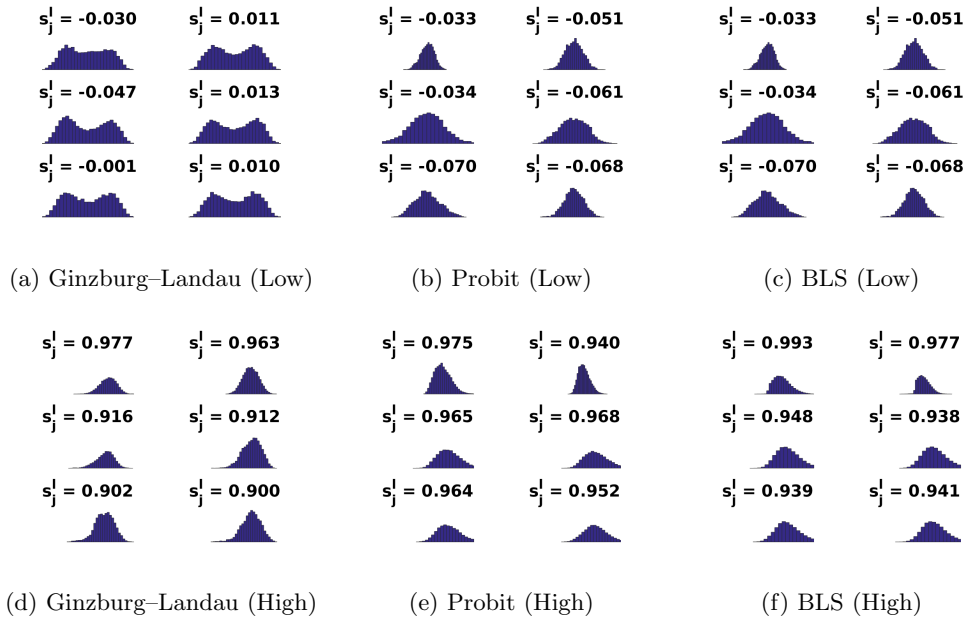
Ginzburg–Landau distribution in contrast to the unimodal nature of the probit posterior; this unimodality is a consequence of the log-concavity of the probit likelihood. For the bottom row, we plot the same empirical distributions for 6 nodes with “high confidence” predictions. In contrast with the top row, the Ginzburg–Landau marginal for high confidence nodes is essentially unimodal since most samples of  $v$  evaluated on these nodes have a fixed sign.

**5.3. Spectral approximation and projection methods.** Here we discuss Algorithms 2 and 3, designed to approximate the full (but expensive on large graphs) Algorithm 1.

First, we examine the quality of the approximation by applying the algorithms to the voting records dataset, a small enough problem where sampling using the full graph Laplacian is feasible. To quantify the quality of approximation, we compute the posterior mean of the thresholded variable  $s_j^l$  for both Algorithm 2 and Algorithm 3 and compare the mean absolute difference  $\frac{1}{N} |s_j^l - s_j^{l*}|$ , where  $s_j^{l*}$  is the “ground truth” value computed using the full Laplacian. Using  $\gamma = 0.1$ ,  $\beta = 0.3$ , and a truncation level of  $\ell = 150$ , we observe that the mean absolute difference for spectral projection is **0.1577**, and **0.0261** for spectral approximation. In general, we set  $\bar{\lambda}$  to be  $\max_{j \leq \ell} \lambda_j$ , where  $\ell$  is the truncation level.

Next we apply the spectral projection/approximation algorithms with the Bayesian level-set likelihood to the hyperspectral image dataset; the results for probit are similar (when we



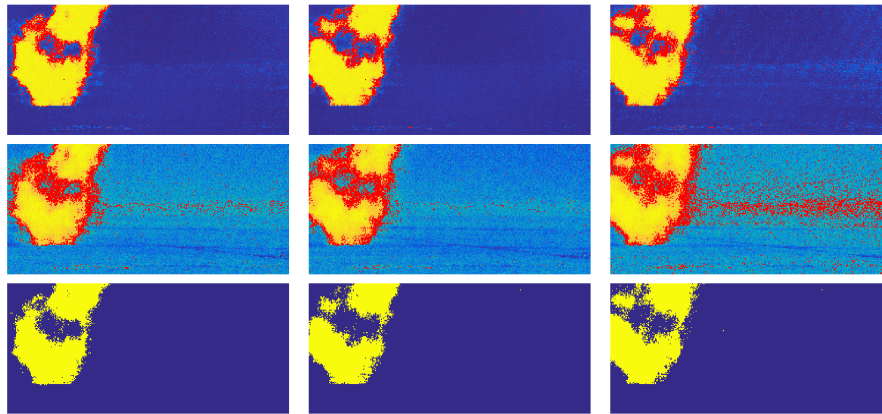


**Figure 7.** Visualization of marginal posterior density for low and high confidence predictions across different models. Each image plots the empirical marginal posterior density of a certain node  $i$ , obtained from the histogram of  $1 \times 10^5$  approximate samples using  $pCN$ . Columns in the figure (e.g., (a) and (d)) are grouped by model. From left to right, the models are Ginzburg-Landau, probit, and Bayesian level-set, respectively. The top row denotes the low confidence predictions, and the bottom row the high confidence predictions. For the top row, we select 6 nodes with the lowest absolute value of the posterior mean  $s_j^l$ , defined in (9), averaged across three models. For the bottom row, we select nodes with the highest average posterior mean  $s_j^l$ . We show the posterior mean  $s_j^l$  on top of the histograms for reference. The experiment parameters are  $\epsilon = 10.0$ ,  $\gamma = 0.6$ ,  $\beta = 0.1$  for the Ginzburg-Landau model, and  $\gamma = 0.5$ ,  $\beta = 0.2$  for the probit and Bayesian level-set model.

use small  $\gamma$ ) but have greater cost per step, because of the cdf evaluations required for probit. The first two rows in Figure 8 show that the posterior mean  $s_j^l$  is able to differentiate between different concentrations of the plume gas. We have also colored pixels with  $|s_j^l| < 0.4$  in red to highlight the regions with greater levels of uncertainty. We observe that the red pixels mainly lie along the edges of the gas plume, which conforms with human intuition. As in the voting records example in the previous subsection, the spectral approximation method has greater posterior uncertainty, demonstrated by the greater number of red pixels in the second row of Figure 8 compared to the first row. We conjecture that the spectral approximation is closer to what would be obtained by sampling the full distribution, but we have not verified this, as the full problem is too large to readily sample. The bottom row of Figure 8 shows the result of using optimization-based classification using the Ginzburg-Landau method. This is shown simply to demonstrate consistency with the full uncertainty quantification approach shown in the other two rows in terms of hard classification.

**5.4. Comparative remarks about the different models.** At a high level we have shown the following concerning the three models based on probit, level set, and Ginzburg-Landau:

- Bayesian level set is considerably cheaper to implement than probit in MATLAB



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

because the norm cdf evaluations required for probit are expensive.

- Probit and Bayesian level set behave similarly for posterior sampling, especially for small  $\gamma$ , since they formally coincide when  $\gamma = 0$ .
- Probit and Bayesian level set are superior to Ginzburg–Landau for posterior sampling; this is because probit has log-concave posterior, while Ginzburg–Landau is multimodal.
- Ginzburg–Landau provides the best hard classifiers, when used as an optimizer (MAP estimator), and provided it is initialized well. However, it behaves poorly when not initialized carefully because of multimodal behavior. In contrast, probit provides almost indistinguishable classifiers, comparable or marginally worse in terms of accuracy, and has a convex objective function and hence a unique minimizer. (See the supplementary materials (M113421\_01.pdf [local/web 469KB]) for details of the relevant experiments.)

We expand upon the details of these conclusions by studying run times of the algorithms. All experiments are done on a 1.5GHz machine with Intel Core i7. In Table 2, we compare the running time of the MCMC for different models on various datasets. We use an a posteriori condition on the samples  $u^{(k)}$  to empirically determine the sample size  $M$  needed for the MCMC to converge. Note that this condition is by no means a replacement for a rigorous analysis of convergence using autocorrelation, but is designed to provide a ballpark estimate of the speed of these algorithms on real applications. We now define the a posteriori condition used. Let the approximate samples be  $\{u^{(k)}\}$ . We define the cumulative average as  $\tilde{u}^{(k)} = \frac{1}{k} \sum_{j=1}^k u^{(j)}$  and find the first  $k$  such that

$$(16) \quad \|\tilde{u}^{(kT)} - \tilde{u}^{((k-1)T)}\| \leq \text{tol},$$

Table 2

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

| Data<br>(Tol)<br>(N)<br>(Neig)<br>(J) | Voting Records<br>$\mathbf{tol} = 1 \times 10^{-3}$<br>$N = 435$<br>$Neig = 435$<br>$J = 5$ | MNIST49<br>$\mathbf{tol} = 1.5 \times 10^{-3}$<br>$N \approx 1.1 \times 10^4$<br>$Neig = 300$<br>$J = 440$ | Hyperspectral<br>$\mathbf{tol} = 2 \times 10^{-2}$<br>$N \approx 2.9 \times 10^5$<br>$Neig = 50$<br>$J = 8000$ |
|---------------------------------------|---|--|--|
| Preprocessing                         | $t = 0.7s$  | $t = 50.8s$  | $t < 60s$ <sup>6</sup>   |
| probit                                | $t = 8.9s,$<br>$M = 10^4$   | $t = 176.4s,$<br>$M = 1.5 \times 10^4$   | $t = 5410.3s,$<br>$M = 1.5 \times 10^4$  |
| BLS                                   | $t = 2.7s,$<br>$M = 10^4$   | $t = 149.1s,$<br>$M = 1.5 \times 10^4$   | $t = 970.8s,$<br>$M = 1.5 \times 10^4$   |
| GL                                    | $t = 161.4s$<br>$M = 1.8 \times 10^5$   | -<br>-   | -<br>-   |

where  $\mathbf{tol}$  is the tolerance and  $T$  is the number of iterations skipped. We set  $T = 5,000$  and also tune the stepsize parameter  $\beta$  such that the average acceptance probability of the MCMC is over 50%. We choose the model parameters according to the experiments in the sections above so that the posterior mean gives a reasonable classification result.

We note that the number of iterations needed for the Ginzburg–Landau model is much higher compared to the probit and Bayesian level-set methods; this is caused by the presence of multiple local minima in Ginzburg–Landau, in contrast to the log-concavity of probit. Probit is slower than Bayesian level set due to the fact that evaluations of the cdf function for Gaussians is slow.

**6. Conclusions and future directions.** We introduce a Bayesian approach to uncertainty quantification for graph-based classification methods. We develop algorithms to sample the posterior and to compute MAP estimators and, through numerical experiments on a suite of applications, we investigate the properties of the different Bayesian models and the algorithms used to study them.

Some future directions of this work include improvement of the current inference method, connections between the different models in this paper, and generalization to multiclass classification, for example, by vectorizing the latent variable (as in existing non-Bayesian multiclass methods [23, 35]) and applying multidimensional analogues of the likelihood functions used in this paper. Hierarchical methods could also be applied to account for the uncertainty in the various hyperparameters such as the label noise  $\gamma$ , or the length scale  $\epsilon$  in the Ginzburg–Landau model. Finally, we could study in more detail the effects of either the spectral projection or the approximation method, either analytically on some tractable toy examples or empirically on a suite of representative problems.

Studying the modeling assumptions themselves, guided by data, provides a research direction of long term value. Such questions have not been much studied to the best of our

<sup>6</sup>According to the reporting in [36].

knowledge. For example, the choice of the signum function to relate the latent variable to the categorical data could be questioned, and other models employed; or the level value of 0 chosen in the level-set approach could be chosen differently, or as a hyperparameter. Furthermore the form of the prior on the latent variable  $u$  could be questioned. We use a Gaussian prior which encodes first and second order statistical information about the unlabeled data. This Gaussian could contain hyperparameters, of Whittle–Matern type, which could be learned from the data; more generally, other non-Gaussian priors could and should be considered. For instance, in image data it is often useful to model feature vectors as lying on submanifolds embedded in a higher dimensional space; such structure could be exploited. More generally, addressing the question of which generative models are appropriate for which types of data is an interesting and potentially fruitful research direction.

**Acknowledgment.** AMS is grateful to Omiros Papaspiliopoulos for illuminating discussions about the probit model.

## REFERENCES

- [1] C. M. ALAÍZ, M. FANUEL, AND J. A. SUYKENS, *Convex Formulation for Kernel PCA and Its Use in Semi-supervised Learning*, preprint, <https://arxiv.org/abs/1610.06811>, 2016.
- [2] C. M. ALAÍZ, M. FANUEL, AND J. A. SUYKENS, *Robust Classification of Graph-Based Data*, preprint, <https://arxiv.org/abs/1612.07141>, 2016.
- [3] M. BELKIN, I. MATVEEVA, AND P. NIYOGI, *Regularization and semi-supervised learning on large graphs*, in International Conference on Computational Learning Theory, Springer, New York, 2004, pp. 624–638.
- [4] M. BELKIN, P. NIYOGI, AND V. SINDHWANI, *Manifold regularization: A geometric framework for learning from labeled and unlabeled examples*, J. Mach. Learn. Res., 7 (2006), pp. 2399–2434.
- [5] M. BERTHOD, Z. KATO, S. YU, AND J. ZERUBIA, *Bayesian image classification using Markov random fields*, Image Vision Comput., 14 (1996), pp. 285–295.
- [6] A. L. BERTOZZI AND A. FLENNER, *Diffuse interface models on graphs for classification of high dimensional data*, SIAM Rev., 58 (2016), pp. 293–328, <https://doi.org/10.1137/16M1070426>.
- [7] A. L. BERTOZZI AND A. FLENNER, *Diffuse interface models on graphs for classification of high dimensional data*, Multiscale Model. Simul., 10 (2012), pp. 1090–1118, <https://doi.org/10.1137/11083109X>.
- [8] A. L. BERTOZZI, X. LUO, AND A. M. STUART, *Computational cost of sampling methods for semi-supervised learning on large graphs*, in preparation.
- [9] A. BESKOS, G. ROBERTS, AND A. STUART, *Optimal scalings for local Metropolis-Hastings chains on nonproduct targets in high dimensions*, Ann. Appl. Probab., (2009), pp. 863–898.
- [10] A. BESKOS, G. ROBERTS, A. M. STUART, AND J. VOSS, *MCMC methods for diffusion bridges*, Stoch. Dyn., 8 (2008), pp. 319–350.
- [11] C. BISHOP, *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer, New York, 2007.
- [12] A. BLUM AND S. CHAWLA, *Learning from labeled and unlabeled data using graph mincuts*, in Proc. 18th International Conf. on Machine Learning (ICML), 2001.
- [13] Y. BOYKOV, O. VEKSLER, AND R. ZABIH, *Markov random fields with efficient approximations*, in Proceedings of the 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 1998, pp. 648–655.
- [14] Y. BOYKOV, O. VEKSLER, AND R. ZABIH, *Fast approximate energy minimization via graph cuts*, IEEE Trans. Pattern Anal. Mach. Intell., 23 (2001), pp. 1222–1239.
- [15] Y. Y. BOYKOV AND M.-P. JOLLY, *Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images*, in Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV 2001), Vol. 1, IEEE, 2001, pp. 105–112.

- [16] J. B. BROADWATER, D. LIMSUI, AND A. K. CARR, *A Primer for Chemical Plume Detection Using LWIR Sensors*, Technical Paper, National Security Technology Department, Las Vegas, NV, 2011.
- [17] T. BÜHLER AND M. HEIN, *Spectral clustering based on the graph  $p$ -Laplacian*, in Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 81–88.
- [18] F. R. CHUNG, *Spectral Graph Theory*, CBMS Regional Conf. Ser. in Math. 92, AMS, 1997.
- [19] S. L. COTTER, G. O. ROBERTS, A. M. STUART, AND D. WHITE, *MCMC methods for functions: Modifying old algorithms to make them faster*, *Statist. Sci.*, 28 (2013), pp. 424–446.
- [20] E. DAHLHAUS, D. S. JOHNSON, C. H. PAPADIMITRIOU, P. D. SEYMOUR, AND M. YANNAKAKIS, *The complexity of multiway cuts*, in Proceedings of the 24th Annual ACM Symposium on Theory of Computing, ACM, New York, 1992, pp. 241–251.
- [21] M. M. DUNLOP, M. A. IGLESIAS, AND A. M. STUART, *Hierarchical Bayesian Level Set Inversion*, preprint, <https://arxiv.org/abs/1601.03605>, 2016.
- [22] C. FOWLKES, S. BELONGIE, F. CHUNG, AND J. MALIK, *Spectral grouping using the Nyström method*, *IEEE Trans. Pattern Anal. Mach. Intell.*, 26 (2004), pp. 214–225.
- [23] C. GARCIA-CARDONA, E. MERKURJEV, A. L. BERTOZZI, A. FLENNER, AND A. G. PERCUS, *Multiclass data segmentation using diffuse interface methods on graphs*, *IEEE Trans. Pattern Anal. Mach. Intell.*, 36 (2014), pp. 1600–1613.
- [24] D. K. HAMMOND, P. VANDERGHEYNST, AND R. GRIBONVAL, *Wavelets on graphs via spectral graph theory*, *Appl. Comput. Harmon. Anal.*, 30 (2011), pp. 129–150.
- [25] J. HARTOG AND H. VAN ZANTEN, *Nonparametric Bayesian Label Prediction on a Graph*, preprint, <https://arxiv.org/abs/1612.01930>, 2016.
- [26] D. J. HIGHAM AND M. KIBBLE, *A Unified View of Spectral Clustering*, Mathematics Research Report 2, University of Strathclyde, 2004.
- [27] H. HU, J. SUNU, AND A. L. BERTOZZI, *Multi-class graph Mumford-Shah model for plume detection using the MBO scheme*, in Energy Minimization Methods in Computer Vision and Pattern Recognition, Springer, New York, 2015, pp. 209–222.
- [28] M. A. IGLESIAS, Y. LU, AND A. M. STUART, *A Bayesian level set method for geometric inverse problems*, *Interfaces Free Bound.*, 18 (2016), pp. 181–217.
- [29] A. KAPOOR, Y. QI, H. AHN, AND R. PICARD, *Hyperparameter and kernel learning for graph based semi-supervised classification*, in NIPS, 2005, pp. 627–634.
- [30] Y. LECUN, C. CORTES, AND C. J. BURGES, *The MNIST Database of Handwritten Digits*, <http://yann.lecun.com/exdb/mnist/>, 1998.
- [31] S. Z. LI, *Markov Random Field Modeling in Computer Vision*, Springer Science & Business Media, 2012.
- [32] S. LU AND S. V. PEREVERZEV, *Multi-parameter regularization and its numerical realization*, *Numer. Math.*, 118 (2011), pp. 1–31.
- [33] X. LUO AND A. L. BERTOZZI, *Convergence of the graph Allen–Cahn scheme*, *J. Stat. Phys.* 167 (2017), pp. 934–958, <https://doi.org/10.1007/s10955-017-1772-4>.
- [34] A. MADRY, *Fast approximation algorithms for cut-based problems in undirected graphs*, in Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE, 2010, pp. 245–254.
- [35] E. MERKURJEV, T. KOSTIĆ, AND A. L. BERTOZZI, *An MBO scheme on graphs for classification and image processing*, *SIAM J. Imaging Sci.*, 6 (2013), pp. 1903–1930, <https://doi.org/10.1137/120886935>.
- [36] E. MERKURJEV, J. SUNU, AND A. L. BERTOZZI, *Graph MBO method for multiclass segmentation of hyperspectral stand-off detection video*, in Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), IEEE, 2014, pp. 689–693.
- [37] R. NEAL, *Regression and classification using Gaussian process priors*, in Bayesian Statistics 6, Oxford University Press, Oxford, UK, 1999, pp. 475–501, available at <http://www.cs.toronto.edu/~radford/valencia.abstract.html>.
- [38] B. OSTING, C. D. WHITE, AND É. OUDET, *Minimal Dirichlet energy partitions for graphs*, *SIAM J. Sci. Comput.*, 36 (2014), pp. A1635–A1651, <https://doi.org/10.1137/130934568>.
- [39] H. OWHADI, C. SCOVEL, AND T. SULLIVAN, *On the brittleness of Bayesian inference*, *SIAM Rev.*, 57 (2015), pp. 566–582, <https://doi.org/10.1137/130938633>.
- [40] H. OWHADI, C. SCOVEL, T. J. SULLIVAN, M. MCKERNS, AND M. ORTIZ, *Optimal uncertainty quantification*, *SIAM Rev.*, 55 (2013), pp. 271–345, <https://doi.org/10.1137/10080782X>.



- [41] G. O. ROBERTS, A. GELMAN, W. R. GILKS, ET AL., *Weak convergence and optimal scaling of random walk Metropolis algorithms*, Ann. Appl. Probab., 7 (1997), pp. 110–120.
- [42] D. I. SHUMAN, M. FARAJI, AND P. VANDERGHEYNST, *Semi-supervised learning with spectral graph wavelets*, in Proceedings of the International Conference on Sampling Theory and Applications (SampTA), 2011, EPFL-CONF-164765.
- [43] V. SINDHWANI, M. BELKIN, AND P. NIYOGI, *The geometric basis of semi-supervised learning*, in Semi-Supervised Learning, MIT Press, Cambridge, MA, 2006.
- [44] R. C. SMITH, *Uncertainty Quantification: Theory, Implementation, and Applications*, Comput. Sci. Engrg. 12, SIAM, Philadelphia, 2014.
- [45] A. SUBRAMANYA AND J. BILMES, *Semi-supervised learning with measure propagation*, J. Mach. Learn. Res., 12 (2011), pp. 3311–3370.
- [46] T. J. SULLIVAN, *Introduction to Uncertainty Quantification*, Texts Appl. Math. 63, Springer, Cham, 2015.
- [47] P. TALUKDAR AND K. CRAMMER, *New regularized algorithms for transductive learning*, in Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD '09), Part II, Springer, Berlin, 2009, pp. 442–457.
- [48] Y. VAN GENNIP AND A. L. BERTOZZI,  $\Gamma$ -convergence of graph Ginzburg-Landau functionals, Adv. Differential Equations, 17 (2012), pp. 1115–1180.
- [49] U. VON LUXBURG, *A tutorial on spectral clustering*, Statist. Comput., 17 (2007), pp. 395–416.
- [50] G. WAHBA, *Spline Models for Observational Data*, SIAM, Philadelphia, 1990, <https://doi.org/10.1137/1.9781611970128>.
- [51] C. K. WILLIAMS AND C. E. RASMUSSEN, *Gaussian processes for regression*, in Advances in Neural Information Processing Systems 8, MIT Press, Cambridge, MA, 1996, pp. 514–520.
- [52] C. K. WILLIAMS AND M. SEEGER, *Using the Nyström method to speed up kernel machines*, in Proceedings of the 13th International Conference on Neural Information Processing Systems, MIT Press, Cambridge, MA, 2000, pp. 661–667.
- [53] D. XIU, *Numerical Methods for Stochastic Computations: A Spectral Method Approach*, Princeton University Press, Princeton, NJ, 2010.
- [54] L. ZELNIK-MANOR AND P. PERONA, *Self-tuning spectral clustering*, in Proceedings of the 17th International Conference on Neural Information Processing Systems (NIPS'04), MIT Press, Cambridge, MA, 2004, pp. 1601–1608.
- [55] D. ZHOU, O. BOUSQUET, T. N. LAL, J. WESTON, AND B. SCHÖLKOPF, *Learning with local and global consistency*, Proceedings of the 16th International Conference on Neural Information Processing Systems (NIPS'03), MIT Press, Cambridge, MA, 2003, pp. 321–328.
- [56] X. ZHU, *Semi-supervised Learning Literature Survey*, Technical Report TR1530, University of Wisconsin, Madison, 2008.
- [57] X. ZHU, Z. GHAHRAMANI, J. LAFFERTY, ET AL., *Semi-supervised learning using Gaussian fields and harmonic functions*, in ICML, Vol. 3, 2003, pp. 912–919.
- [58] X. ZHU, J. D. LAFFERTY, AND Z. GHAHRAMANI, *Semi-supervised Learning: From Gaussian Fields to Gaussian Processes*, Carnegie Mellon University, Pittsburgh, PA, 2003.