When Does Label Propagation Fail? A View from a Network Generative Model

Yuto Yamaguchi†, **Kohei Hayashi**†
†AIST, Japan
yuto.ymgc@gmail.com, hayashi.kohei@gmail.com

Abstract

What kinds of data does Label Propagation (LP) work best on? Can we justify the solution of LP from a theoretical standpoint? LP is a semi-supervised learning algorithm that is widely used to predict unobserved node labels on a network (e.g., user's gender on an SNS). Despite its importance, its theoretical properties remain mostly unexplored. In this paper, we answer the above questions by interpreting LP from a statistical viewpoint. As our main result, we identify the network generative model behind the discretized version of LP (DLP), and we show that under specific conditions the solution of DLP is equal to the maximum *a posteriori* estimate of that generative model. Our main result reveals the critical limitations of LP. Specifically, we discover that LP would not work best on networks with (1) disassortative node labels, (2) clusters having different edge densities, (3) non-uniform label distributions, or (4) unreliable node labels provided. Our experiments under a variety of settings support our theoretical results.

1 Introduction

On most networks, such as social networks, nodes are associated with *labels* such as gender, age, and locations of people [Mislove *et al.*, 2010]. Since these node labels are missing in most cases [Backstrom *et al.*, 2010], predicting these missing labels is meaningful for some applications [Chaudhari *et al.*, 2014; Jacob *et al.*, 2014; Sen *et al.*, 2008]. For Internet advertising, for example, prediction of user demographics is beneficial for both users and companies so that suitable advertisements can be placed. We call this the *node classification* problem—we want to *recover* the missing node labels *given* the observed node labels and the network structure.

Label Propagation (LP) [Bengio *et al.*, 2006] is a standard algorithm for solving the node classification problem.[1] Given a network and partially observed node labels, LP minimizes the cost function, which consists of two terms that prefer that: (1) predicted labels of connected nodes are the same, and (2) predicted labels and observed labels are the same for each *labeled* node (i.e., the nodes whose labels are observed). To solve this optimization problem, LP *propagates* the observed labels throughout the network along edges, which converges to the optimal solution [Bengio *et al.*, 2006].

Although LP works reasonably well in most cases, we do not know exactly *which kinds of data LP does and does not work well on*. This is our main **research question** (RQ). We break down the details of this RQ into the following four questions:

- `RQ1`: What type of node label is suitable for LP—assortative, disassortative, or others? We say labels are *assortative* when nodes with the same labels are densely connected (e.g., age in SNS), or *disassortative* when nodes with different labels tend to be connected (e.g., gender in SNS [Takac and Zabovsky, 2012]).
- `RQ2`: Should the density of intra-cluster edges be uniform or non-uniform? Some real networks have clusters with varying densities; for example, in SNSs, classmates form a densely connected cluster, whereas people in the same city form a sparsely connected cluster.
- `RQ3`: Should the label distribution (i.e., the number of occurrences of each label across the whole network) be uniform or non-uniform? The label distribution is not always uniform; people's genders are almost uniformly distributed, for example, whereas people's home locations are not.
- `RQ4`: Can observed labels be allowed to be incorrect? If so, with what probability? Observed labels could be incorrect because of, for example, typing errors.

Admittedly, some of these questions have already been addressed individually. For example, at an *objective function level*, it is clear that LP does not work at all with disassortative node labels. However, there has been no comprehensive study that answers the above RQs at a much deeper level.

Meanwhile, in the network science community, a very similar problem called the *node clustering problem* has been extensively studied. Although the objective of this problem is also the prediction of node labels, the difference is that the setting is unsupervised; i.e., all of the labels are assumed to be unobserved. To solve this problem, the *Stochastic Blockmodel* (SBM) is often used [Wang and Wong, 1987]. SBM is

[1] Throughout this paper, we consider LP as an algorithm to solve the node classification problem, although it can be applied to any type of data by constructing a network based on the similarity of data samples.

a generative model of a network that has a community structure. A remarkable property is that, in contrast to LP, SBM explicitly characterizes the generative process of a network by a few interpretable parameters, which means we can understand the underlying structures of a real network by estimating these parameters. The above RQs are therefore completely clarified on SBM.

Present work. Although the problem settings are different (semi-supervised and deterministic vs. unsupervised and probabilistic), LP and SBM share the same goal. This raises a natural question: Is there any theoretical connection between LP and SBM? More specifically, can we interpret LP also as a probabilistic network generative model like SBM? The short answer is yes, and we use this key fact to derive the answers to the above RQs.

We explain the main idea step by step as follows. First, we propose a semi-supervised network generative model—Partially Labeled SBM (PLSBM)—that extends SBM so that it accepts partial supervision. Second, to make the discussion rigorous, we also introduce a discretized version of LP (formally introduced in Section 5.1), which we call discrete LP (DLP). Finally, as our main result, we show that the solution of DLP is identical to the maximum *a posteriori* (MAP) estimate of PLSBM with restricted parameters. This result reveals the theoretical properties of DLP (and LP) and allows us to answer the RQs. We also perform experiments on synthetic networks. The results on the synthetic datasets support our theoretical results.

To summarize, the **key contributions** of this paper are highlighted as follows.

- We propose PLSBM, which is the key to connecting the LP and network generative models.
- We prove that the solution of DLP is identical to the MAP estimate of a special case of PLSBM where some parameters are restricted. This means that we identify the data generative process behind DLP.
- Using the above result, we show the settings in which LP does and does not work well, by answering `RQ1`–`RQ4`.

We emphasize that our interdisciplinary findings between LP and SBM are not only theoretically interesting, but also have several **potential impacts**, including the following:

- The connection between LP and SBM allows us to interchange the findings from the so far different two communities, AI / machine learning and network science. This enables a deeper analysis of the theoretical properties of both models.
- Now we have a guideline for using LP—we can safely choose whether or not to use LP on a target dataset on the basis of the answers to our four RQs.

2 Related Work

Label Propagation. There are many algorithms related to LP; Harmonic Function [Zhu *et al.*, 2003], Local and Global Consistency [Zhou *et al.*, 2004], Adsorption [Baluja *et al.*, 2008], and Modified Adsorption [Talukdar and Crammer, 2009] are the most commonly used algorithms that search for the fixed point state where as many connected nodes as possible have the same class labels. Whereas these algorithms are designed for assortative labels, OMNI-Prop [Yamaguchi *et al.*, 2015] is an LP-like algorithm that is applicable to both assortative and disassortative labels.

There are few studies exploring LP from a theoretical perspective. In his doctoral thesis [Zhu *et al.*, 2005], Zhu analyzed LP from several perspectives, for example, interpreting LP as a series of random walks or an electric network, and exploring connections between LP and Gaussian processes. A recent study [Kyng *et al.*, 2015] also theoretically investigated related but not identical problem of *graph regression*, proposing a fast algorithm to solve it. To the best of our knowledge, our study is the first to address the network generative model behind LP.

Stochastic Blockmodel. SBM is one of the most basic network generative models; it takes an unattributed network and outputs the clustering of nodes. This basic model has been extensively studied from a variety of perspectives such as phase transition [Zhang *et al.*, 2014], degree distribution [Karrer and Newman, 2011], and cluster size [Zhang *et al.*, 2016]. However, to the best of our knowledge, there has been no study bridging the gap between LP and SBM to share the findings from both research communities. In this paper, to bridge this gap, we propose a network generative model that is designed to be as simple as possible, in contrast to other relatively complex models [Newman and Clauset, 2015; Chang and Blei, 2010; Nallapati *et al.*, 2008; Cho *et al.*, 2016; Kim and Leskovec, 2012; Pfeiffer III *et al.*, 2014]. Our model is viewed as performing classification when it is provided some degree of supervision, and it is viewed as performing clustering when it is not provided any supervision. Hence, in the rest of this paper, we discuss in detail both the node classification problem and the node clustering problem, which are defined in the next section.

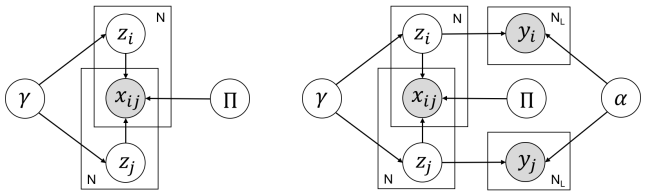
3 Settings

Let $G = (\boldsymbol{X}, \boldsymbol{Y})$ be an attributed graph with N nodes, M edges, and K distinct labels. $\boldsymbol{X}$ is the $N \times N$ adjacency matrix where $x_{ij} = 1$ if node i and j are connected, and $x_{ij} = 0$ otherwise. $\boldsymbol{Y}$ is the $N \times K$ *partially observed label assignment matrix* where $y_{ik} = 1$ if node i has label k, and $y_{ik'} = 0$ for $k' \neq k$. There are two types of nodes, namely, *labeled* nodes $V^L = \{1, \ldots, N_L\}$, and *unlabeled* nodes $V^U = \{N_L + 1, \ldots, N\}$. Each labeled node is explicitly assigned its label, whereas labels of unlabeled nodes are unknown. That is, for each labeled node $1 \leq i \leq N_L$, $\boldsymbol{y}_i$ is a 1-of-K vector, whereas for each unlabeled node $N_L + 1 \leq i \leq N$, $\boldsymbol{y}_i$ is a 0 vector.

Using the above notation, we formally define the K-class node classification problem, which LP solves, and the K-cluster node clustering problem, which SBM solves, as follows:

Problem 1 (K-class node classification)**.**

- `Given`: an adjacency matrix $\boldsymbol{X}$ and a partially observed label assignment matrix $\boldsymbol{Y}$,
- `Find`: the label of each unlabeled node $i \in V^U$.



(a) SBM (b) PLSBM

Figure 1: Graphical models of (a) SBM and (b) PLSBM. Shaded nodes represent the observed variables.

Note that in the K-class node classification problem, we assume that at least one label assignment for each class is provided.

Problem 2 (K-cluster node clustering)**.**

- `Given`: an adjacency matrix $\boldsymbol{X}$ and the number of clusters K,
- `Find`: the cluster assignment for each node $i \in V^L \cup V^U$.

3.1 LP: Label Propagation

Given an attributed graph G, the objective of LP is to predict the label of each unlabeled node $i \in V^U$. Let $f_{ik} \in [0, 1]$ be the value that represents how likely it is that node i has label k. The objective function of LP to be minimized is defined as follows:[2]

$$Q(\boldsymbol{F}; \boldsymbol{X}, \boldsymbol{Y}, \lambda) = \frac{1}{2}\sum_{i=1}^{N} \|\boldsymbol{f}_i - \boldsymbol{y}_i\|_2^2 + \frac{\lambda}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} x_{ij}\|\boldsymbol{f}_i - \boldsymbol{f}_j\|_2^2, \tag{1}$$

where $\lambda \geq 0$ is the trade-off parameter, and $\|\cdot\|_2$ denotes the ℓ^2-norm. The first term prefers that the predicted labels are the same as the observed labels, whereas the second term prefers that connected nodes have the same predicted labels. Note that since $\boldsymbol{y}_i$ is a 0 vector for unlabeled node i, the first term works as a regularization term for unlabeled nodes so that $\boldsymbol{f}_i$ moves toward $\boldsymbol{0}$. By setting the derivative of Q equal to zero, we have the closed-form solution of $\boldsymbol{F}$ as follows:

$$\hat{\boldsymbol{F}} = (\boldsymbol{I} + \lambda \boldsymbol{L})^{-1}\boldsymbol{Y}, \tag{2}$$

where $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ is the graph Laplacian, and $\boldsymbol{D}$ is the diagonal degree matrix with $d_{ii} = \sum_{j=1}^{N} x_{ij}$.

After $\hat{\boldsymbol{F}}$ is obtained, node labels are predicted by discretizing $\hat{\boldsymbol{F}}$ as follows:

Definition 1 (Solution of LP)**.**

$$[\hat{z}_{LP}]_{ik} = \begin{cases} 1 & (\textit{if } k = \arg\max_s \hat{f}_{is}) \\ 0 & (otherwise) \end{cases} \tag{3}$$

[2]This objective function is the same as [Zhou *et al.*, 2004] except for the square root degree normalization [Zhou *et al.*, 2004].

3.2 SBM: Stochastic Blockmodel

SBM takes an adjacency matrix $\boldsymbol{X}$ and a user parameter K as input, and outputs K clusters of nodes. SBM has two parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\Pi}$, which have to be estimated from $\boldsymbol{X}$. $\boldsymbol{\gamma}$ is a K-dimensional vector for which $\sum_{k=1}^{K}\gamma_k = 1$, and $\gamma_k \in [0, 1]$, which represents the fraction of the number of nodes in cluster k. $\boldsymbol{\Pi}$ is the $K \times K$ symmetric matrix whose element $\pi_{kl} \in [0, 1]$ represents the probability that a node in cluster k and a node in cluster l are linked by an edge.

The key idea of SBM is *stochastic equivalence*, which means that nodes in the same cluster have stochastically the same connection patterns (i.e., nodes in cluster k are all assigned π_{kl} for all l). SBM produces clusters composed of stochastically equivalent nodes, which means that the resulting clusters are not always densely connected.

Generative Model

Let $Mult$ denote the multinomial distribution and let $Bern$ denote the Bernoulli distribution. Also, let $\boldsymbol{z}_i$ be the 1-of-K indicator vector where $z_{ik} = 1$ indicates that node i is predicted to belong to cluster k. The generative process of SBM is then written as follows:

- For each node $i = 1, \ldots, N$
  - Generate $\boldsymbol{z}_i \sim Mult(\cdot|\gamma)$
- For each node pair (i, j)
  - Generate $x_{ij} \sim Bern(\cdot|\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{\Pi}\boldsymbol{z}_j)$

In the generative process, $\boldsymbol{z}_i$ is generated for each node by the multinomial distribution with parameter $\boldsymbol{\gamma}$. Then, x_{ij} is generated for each node pair by the Bernoulli distribution with parameter $\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{\Pi}\boldsymbol{z}_j$, which equals π_{kl} if $z_{ik} = 1$ and $z_{jl} = 1$. The graphical model of SBM is shown in Fig. 1(a).

Based on this generative process, the log-likelihood of SBM is written as:

$$\ln P(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\Pi}, \boldsymbol{\gamma}) = \sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik}\ln\gamma_k + \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{K}\sum_{l=1}^{K} z_{ik}z_{jl}\left(\ln\pi_{kl}^{x_{ij}} + \ln(1-\pi_{kl})^{1-x_{ij}}\right).$$

Inference

We follow [Daudin *et al.*, 2008] who developed the variational EM (VEM) algorithm for estimating the SBM parameters.

In the E-step, we update the posterior distribution $q(\boldsymbol{Z})$ as follows:

$$\boldsymbol{q}_i \propto \exp\left(\ln\boldsymbol{\gamma} + \boldsymbol{a}_i + (\ln\boldsymbol{\Pi})\sum_{j\in N(i)}\boldsymbol{q}_j\right), \tag{4}$$

where $\boldsymbol{q}_i = q(\boldsymbol{z}_i)$ for brevity, and $\boldsymbol{a}_i = \ln(\boldsymbol{1}\boldsymbol{1}^{\mathrm{T}} - \boldsymbol{\Pi})\sum_{j\neq i}\boldsymbol{q}_j$; $\boldsymbol{1}$ denotes the vector with appropriate dimensions where all elements are 1.

In the M-step, we then update the parameters $\boldsymbol{\Pi}$ and $\boldsymbol{\gamma}$ as follows:

$$\pi_{kl} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} x_{ij}q_{ik}q_{jl}}{\sum_{i=1}^{N}\sum_{j=1}^{N} q_{ik}q_{jl}}, \text{ and } \gamma_k = \frac{1}{N}\sum_{i=1}^{N} q_{ik}. \tag{5}$$

The iterative update of $\boldsymbol{q}$, $\boldsymbol{\Pi}$, and $\boldsymbol{\gamma}$ is guaranteed to converge, which achieves the local maxima of the marginal log-likelihood [Daudin *et al.*, 2008].

4 PLSBM: Partially Labeled SBM

SBM is an unsupervised model; i.e., its estimate of the cluster structure is based solely on the network structure $\boldsymbol{X}$. However, when the partially observed labels $\boldsymbol{Y}$ are given, we should use them because it will make clustering more accurate. This motivates us to introduce PLSBM.

4.1 Generative Model

Now we consider that, in addition to $\boldsymbol{X}$, $\boldsymbol{Y}$ is also generated by the model. Since the given observed labels are possibly incorrect, we introduce an additional parameter $\alpha \in [0, 1]$ that represents the probability of the correctness of the observed labels. Let $\boldsymbol{B}$ be the $K \times K$ matrix where $b_{kk} = \alpha$ for all k, and $b_{kl} = \frac{1-\alpha}{K-1}$ for $k \neq l$. The generative process of PLSBM is written as follows:

- For each node $i = 1, \ldots, N$
  - Generate $\boldsymbol{z}_i \sim Mult(\cdot|\gamma)$
- For each labeled node $i = 1, \ldots, N_L$
  - Generate $\boldsymbol{y}_i \sim Mult(\cdot|\boldsymbol{B}\boldsymbol{z}_i)$
- For each node pair (i, j)
  - Generate $x_{ij} \sim Bern(\cdot|\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{\Pi}\boldsymbol{z}_j)$

Note that the only difference between SBM and PLSBM is that PLSBM generates $\boldsymbol{Y}$ (see Fig. 1).

The log-likelihood of PLSBM is written as:

$$\ln P(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}|\boldsymbol{\Pi}, \alpha, \boldsymbol{\gamma}) = \ln P(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\Pi}, \boldsymbol{\gamma}) + \ln P(\boldsymbol{Y}|\boldsymbol{Z}, \alpha),$$

$$\ln P(\boldsymbol{Y}|\boldsymbol{Z}, \alpha) = \sum_{i=1}^{N_L}\sum_{k=1}^{K}\sum_{l=1}^{K} z_{ik} \ln b_{kl}^{y_{il}},$$

where $P(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\Pi}, \boldsymbol{\gamma})$ is the same as in SBM, and $P(\boldsymbol{Y}|\boldsymbol{Z}, \alpha)$ is the likelihood of generating partially observed labels $\boldsymbol{Y}$.

4.2 Inference

Inspired by [Daudin *et al.*, 2008], we derive the VEM algorithm of PLSBM. Details of the derivation are omitted since it is not very different from [Daudin *et al.*, 2008].

In the E-step, we update the posterior distribution $q(\boldsymbol{Z})$ as follows:

$$\boldsymbol{q}_i \propto \exp\left(\ln \boldsymbol{\gamma} + \ln \boldsymbol{B}\boldsymbol{y}_i + \boldsymbol{a}_i + (\ln \boldsymbol{\Pi}) \sum_{j \in N(i)} \boldsymbol{q}_j\right). \quad (6)$$

The difference from SBM (Eqn. (4)) is the effect of $\boldsymbol{Y}$. If it is observed that node i has label k (i.e., $y_{ik} = 1$), the second term on the right-hand side is $\ln \boldsymbol{b_k}$, where the k-th element is larger than the other elements when $\alpha > 1/K$. This means that node i is supervised toward being predicted as label k.

In the M-step, we then update the parameters $\boldsymbol{\Pi}$, $\boldsymbol{\gamma}$, and α as follows:

$$\alpha = \frac{\sum_{i=1}^{N_L}\sum_{k=1}^{K} q_{ik}y_{ik}}{\sum_{i=1}^{N_L}\sum_{k=1}^{K} y_{ik}}. \quad (7)$$

The updating equations for $\boldsymbol{\Pi}$ and $\boldsymbol{\gamma}$ are the same as those for SBM.

4.3 SBM as a Special Case of PLSBM

We prove that SBM is a special case of PLSBM when we ignore observed labels $\boldsymbol{Y}$.

Proposition 2. *If $\alpha = 1/K$, the difference between the likelihood functions of SBM and PLSBM is constant.*

Proposition 2 indicates that, when the given labels are assumed to be completely random, the log-likelihoods of SBM and PLSBM are equivalent, and any estimation method (e.g., maximum likelihood, MAP, or other Bayesian inference method) results in the same estimators of the parameters. Proof is omitted since it is obvious.

5 Theory

5.1 DLP: Discrete Label Propagation

Recall that LP solves the continuous optimization problem for $\boldsymbol{F}$ and then discretizes it to predict node labels (Definition 1). This approach is advantageous in terms of computational efficiency. As shown in Eqn. (2), LP is essentially a problem of inverting $(\boldsymbol{I} + \lambda\boldsymbol{L})$, which is efficiently solved by calculating the Neumann series [Harville, 1998]. The post hoc discretization is not satisfactory, however, in terms of the principle of optimization. Indeed, the discretized solution (Def. 1) is not the minimizer of Eqn. (1) under the 1-of-K constraint. Moreover, there is no explicit objective function for which the discretized solution (Def. 1) is the minimizer. Motivated by this, we reformulate the LP problem as a discrete optimization problem that we call DLP.

The input and the output of DLP are the same as those for LP. The objective function of DLP is the same as that of LP (Eqn. (1)) except that λ can be negative.[3] This is because even if $\lambda < 0$, the solution of DLP does not diverge because of the constraint that the $\boldsymbol{z}_i$'s are 1-of-K vectors. The solution of DLP is defined as follows:

Definition 3 (Solution of DLP)**.**

$$\hat{\boldsymbol{Z}}_{DLP} = \arg\min_{\boldsymbol{Z} \in \mho} Q(\boldsymbol{Z}; \boldsymbol{X}, \boldsymbol{Y}, \lambda), \quad (8)$$

where $\mho$ is the entire space consisting of N 1-of-K vectors.

Since the solution of LP (3) is an approximated solution of the objective function of DLP (8), the achievable minimum of LP with post hoc discretization is always worse than or equal to the minimum for DLP.

Proposition 4. *For any $\boldsymbol{X}$, $\boldsymbol{Y}$, and λ,*

$$Q(\hat{\boldsymbol{Z}}_{DLP}; \boldsymbol{X}, \boldsymbol{Y}, \lambda) \leq Q(\hat{\boldsymbol{Z}}_{LP}; \boldsymbol{X}, \boldsymbol{Y}, \lambda).$$

5.2 DLP as a Special Case of PLSBM

First we define the MAP estimate of PLSBM as follows:

Definition 5 (MAP estimate of PLSBM)**.**

$$\hat{\boldsymbol{Z}}_{PLSBM} = \arg\max_{\boldsymbol{Z} \in \mho} P(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Pi}, \boldsymbol{\gamma}, \alpha).$$

[3] If $\lambda < 0$ in LP, the second term of Eqn. (1) can be made arbitrarily small by increasing some of the $\boldsymbol{F}$s, and the solution can diverge to infinity.

Now, we show the sufficient conditions for the equivalence between the MAP estimate of PLSBM and the solution of DLP.

Theorem 6. *Let $\hat{\boldsymbol{Z}}_{DLP}(\lambda)$ be the set of solutions of DLP with $\lambda \in \mathbb{R}$. Let us introduce two variables $\mu, \nu \in [0,1]$ and let $\hat{\boldsymbol{Z}}_{PLSBM}(\mu,\nu,\alpha)$ be the set of MAP estimates of $\boldsymbol{Z}$ with the subclass of PLSBM where*

- *Condition 1. $\gamma_k = 1/K$ for all k,*
- *Condition 2. $\boldsymbol{\Pi} = \mu\boldsymbol{I} + \nu(\boldsymbol{1}\boldsymbol{1}^{\mathrm{T}} - \boldsymbol{I})$, and*
- *Condition 3. the number of nodes assigned to cluster k is specified in advance for all k.*

Then, $\hat{\boldsymbol{Z}}_{DLP}(\lambda) = \hat{\boldsymbol{Z}}_{PLSBM}(\mu,\nu,\alpha)$ if the following conditions are satisfied:

- *Condition 4. if $\lambda > 0$, then $\mu > \nu$; else if $\lambda < 0$, then $\mu < \nu$; else if $\lambda = 0$, then $\mu = \nu$; and*
- *Condition 5. $\lambda \ln \frac{\alpha(K-1)}{1-\alpha} = \ln \frac{\mu(1-\nu)}{\nu(1-\mu)}$.*

Proof sketch. The log-likelihood of PLSBM satisfying Conditions 1 and 2 is rewritten as follows:

$$\ln P(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{Y},\boldsymbol{\Pi},\boldsymbol{\gamma},\alpha) = \ln\frac{\alpha(K-1)}{1-\alpha}\sum_{i=1}^{N_L}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{y}_i + \ln\frac{\mu(1-\nu)}{\nu(1-\mu)}\sum_{i=1}^{N}\sum_{j=1}^{N}(x_{ij}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_j + \boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_j) + C, \quad (9)$$

where C is some constant.

Denoting by $n_k = \sum_i z_{ik}$ the number of nodes assigned to cluster k we can write: $\sum_{i=1}^{N}\sum_{j=1}^{N}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_j = \sum_{k=1}^{K} n_k^2$. Since this term becomes constant under Condition 3, according to Eqn. (9) we can again rewrite the log-likelihood as follows:

$$\ln P(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{Y},\boldsymbol{\Pi},\boldsymbol{\gamma},\alpha) = \ln\frac{\alpha(K-1)}{1-\alpha}\sum_{i=1}^{N_L}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{y}_i + \ln\frac{\mu(1-\nu)}{\nu(1-\mu)}\sum_{i=1}^{N}\sum_{j=1}^{N}x_{ij}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_j + C', \quad (10)$$

where C' is some constant. In addition, we can rewrite Eqn. (1) as follows:

$$-Q(\boldsymbol{Z};\boldsymbol{X},\boldsymbol{Y},\lambda) = \sum_{i=1}^{N_L}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{y}_i + \lambda\sum_{i=1}^{N}\sum_{j=1}^{N}x_{ij}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_j + C'', \quad (11)$$

where C'' is some constant. To show the equivalence of the maximizers, Eqns. (10) and (11) need to be equal up to a positive scale $c' > 0$. This requires, by comparing the first and second terms in Eqns. (10) and (11), we need to have $\ln\frac{\alpha(K-1)}{1-\alpha} = c'$ and $\ln\frac{\mu(1-\nu)}{\nu(1-\mu)} = c'\lambda$. Substituting the former into the latter, we obtain Condition 5. In addition, because c' is positive, $\frac{1}{\lambda}\ln\frac{\mu(1-\nu)}{\nu(1-\mu)}$ must be positive, which yields Condition 4. □

According to Conditions 4 and 5, λ (the LP parameter) is related to $\boldsymbol{\Pi}$ and α (the PLSBM parameters). It is interesting to note that the label correctness probability α and the edge probability $\boldsymbol{\Pi}$ in PLSBM are controlled by just one parameter λ in DLP.

5.3 Answers to Our RQs

Having proved Theorem 6, we are now in a position to answer our four RQs.

(RQ1) **Assortativity and disassortativity.** Condition 4 implies that positive λ leads to $\mu > \nu$. This indicates that since LP requires $\lambda > 0$, it works only on networks with assortative labels where the diagonal value μ of $\boldsymbol{\Pi}$ is larger than the off-diagonal value ν of $\boldsymbol{\Pi}$.

(RQ2) **Uniform or non-uniform cluster density.** Condition 2 implies that DLP assumes the uniform cluster density. Therefore, DLP may not work well on networks that have clusters with different densities, i.e., those for which the diagonal elements of $\boldsymbol{\Pi}$ are not the same.

(RQ3) **Uniform or non-uniform label distribution.** Condition 1 implies that DLP assumes the uniform label distribution. This means that DLP may not work well for cases in which the ratio of labels in each class is non-uniform.

(RQ4) **Label correctness probability.** Condition 5 implies a large odds ratio $(\frac{\mu}{1-\mu})/(\frac{\nu}{1-\nu})$, which indicates strong assortativity, leads to large α. This means that, when LP deals with a network having strong assortativity, it requires α to be large. However, there could be labels with strong assortativity but small α (observed labels that are incorrect with high probability.) Therefore, DLP may not handle the case where there are many label noises but strong assortativity.

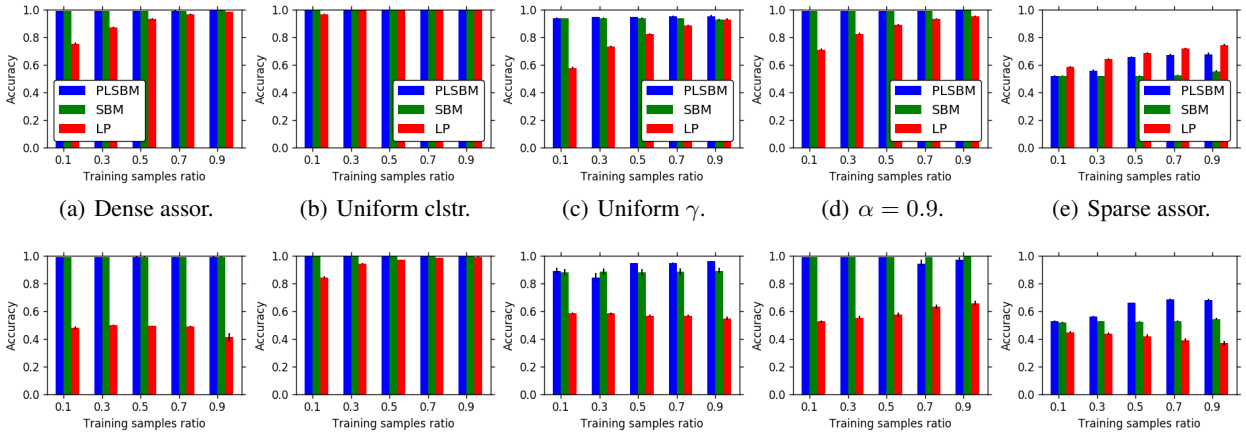
Although we interpret DLP as a special case of PLSBM in Theorem 6, there is still a gap between DLP and LP, namely, that LP involves a continuous relaxation. Furthermore, for PLSBM, Theorem 6 only shows the equivalence of its MAP estimate, not the solution given by the VEM algorithm. This gap will be shown to be sufficiently small, however, in the next section.

6 Experiments

6.1 Settings

We use PLSBM to generate the synthetic datasets (i.e., $\boldsymbol{X}$ and $\boldsymbol{Y}$) with varying parameters. We also vary the ratio of the training nodes (N_L/N) from 0.1 to 0.9 by steps of 0.2. For each setting, we generate 20 networks with $N = 1,000$ nodes and report the mean accuracy and the standard deviation. Note that although using the LFR benchmark [Lancichinetti *et al.*, 2008] would be interesting, it is better to use PLSBM to generate the data to investigate the properties of LP. We compare SBM, PLSBM, and LP. SBM and PLSBM learn all the parameters from the data, whereas LP determines its parameter by 5-fold cross validation.

We use the standard classification accuracy for our evaluation metric, which is defined as the ratio of the number of



(a) Dense assor. (b) Uniform clstr. (c) Uniform γ. (d) $\alpha = 0.9$. (e) Sparse assor.

(f) Dense disassor. (g) Non-uniform clstr. (h) Non-uniform γ. (i) $\alpha = 0.6$ (j) Sparse disassor.

Figure 2: Experimental results at different settings. (a,f) Dense assortative vs. dense disassortative. (b,g) Uniform vs. non-uniform cluster densities. (c,h) Uniform vs. non-uniform label distribution γ. (d,i) High vs. low label correctness probability α. (e,j) Sparse assortative vs. sparse disassortative.

correctly classified test nodes against all test nodes. For evaluating SBM and PLSBM, which essentially perform clustering, we take the best match between correct labels and the results of cluster numbers.

Reproducibility. Our code to reproduce the experiments is available at `https://goo.gl/VWggEy`.

6.2 Results

We perform experiments on four scenarios each of which corresponds to one of our four RQs.

(RQ1) **Assortativity and disassortativity.** We compare the results on assortative and disassortative labels with $K = 2$. For the network with assortative labels, we use $\mathbf{\Pi}^{(a)} = \left(\begin{smallmatrix} 0.15 & 0.1 \\ 0.1 & 0.15 \end{smallmatrix}\right)$, whereas for the network with disassortative labels, we use $\mathbf{\Pi}^{(d)} = \left(\begin{smallmatrix} 0.1 & 0.15 \\ 0.15 & 0.1 \end{smallmatrix}\right)$. The other parameters are set to $\gamma = (0.5, 0.5)$ and $\alpha = 0.99$. The results (Figs. 2(a) and 2(f)) show that LP does not work on the disassortative labels.

(RQ2) **Uniform or non-uniform cluster density.** We compare the results on networks with uniform and non-uniform cluster densities with $K = 3$. For the uniform case, we use $\mathbf{\Pi}$ whose diagonal elements are all 0.2, and for the non-uniform case, we use $\mathbf{\Pi}$ whose diagonal elements are $\Pi_{11} = 0.2$, $\Pi_{22} = 0.15$, and $\Pi_{33} = 0.1$. Off-diagonal elements of $\mathbf{\Pi}$ in both settings are all 0.05 (so generated labels are assortative). The other parameters are set to uniform $\boldsymbol{\gamma}$ and $\alpha = 0.99$. The results (Figs. 2(b) and 2(g)) show that the accuracy of LP is lower than that of SBM and PLSBM when the cluster density is non-uniform, indicating that LP does not work well on networks with a non-uniform cluster density.

(RQ3) **Uniform or non-uniform label distribution.** We compare the results on $K = 3$ networks with uniform ($\boldsymbol{\gamma} = (1/3, 1/3, 1/3)$) and non-uniform ($\boldsymbol{\gamma} = (1/2, 1/4, 1/4)$) label distributions. We use $\alpha = 0.99$, and $\mathbf{\Pi}$ where $\Pi_{kl} = 0.15$ if $k = l$ and $\Pi_{kl} = 0.1$ if $k \neq l$. The results (Figs. 2(c) and 2(h)) show that LP does not work in the non-uniform setting even when the 90% of nodes are used as training nodes.

(RQ4) **Label correctness probability.** We compare the results on networks with high ($\alpha = 0.9$) and low ($\alpha = 0.6$) label correctness probabilities. The other parameters are set to $K = 2$, uniform $\boldsymbol{\gamma}$, and $\mathbf{\Pi} = \mathbf{\Pi}^{(a)}$. Figs. 2(d) and 2(i) show that, whereas SBM and PLSBM achieve almost 100% accuracy in both cases, LP shows lower accuracy when $\alpha = 0.6$, meaning that LP does not adapt to varying α when $\mathbf{\Pi}$ is fixed.

7 Discussion

As shown in Figs. 2(a) and 2(f), on the dense networks, PLSBM achieves almost 100% accuracy. More surprisingly, SBM achieves almost the same high performance even though it does not use any observed labels. On the other hand, as shown in Figs. 2(e) and 2(j), SBM and PLSBM do not work well when we use $0.1 \cdot \mathbf{\Pi}^{(a)}$ and $0.1 \cdot \mathbf{\Pi}^{(d)}$. This behavior is explained by the notion of *detectability* [Saade *et al.*, 2014]. That is, without any supervision, SBM can recover true labels perfectly if the given network satisfies $|c_{in} - c_{out}| \geq K\sqrt{c}$, where $K = 2$, c_{in} and c_{out} are the diagonal and off-diagonal parts of $\mathbf{\Pi}$ multiplied by N, respectively, and c is the average degree. Whereas $\mathbf{\Pi}$ in the dense setting has the detectability property, $\mathbf{\Pi}$ in the sparse setting does not, which is also confirmed from the results.

In contrast, LP does not perfectly classify the nodes even on the dense networks, indicating that LP does not leverage the detectability. This can be explained by the weak expressive power of LP compared to PLSBM. As shown in Theorem 6, LP can handle only the subclass of networks that PLSBM generates because of some fixed parameters.

Acknowledgements

This research was partially supported by the program "Research and Development on Real World Big Data Integration and Analysis" of RIKEN, Japan, and by a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

[Backstrom *et al.*, 2010] Lars Backstrom, Eric Sun, and Cameron Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In *WWW*, pages 61–70, 2010.

[Baluja *et al.*, 2008] Shumeet Baluja, Rohan Seth, D Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In *WWW*, pages 895–904, 2008.

[Bengio *et al.*, 2006] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. Label propagation and quadratic criterion. *Semi-supervised learning*, 10, 2006.

[Chang and Blei, 2010] Jonathan Chang and David M Blei. Hierarchical relational models for document networks. *The Annals of Applied Statistics*, pages 124–150, 2010.

[Chaudhari *et al.*, 2014] Gaurish Chaudhari, Vashist Avadhanula, and Sunita Sarawagi. A few good predictions: selective node labeling in a social network. In *WSDM*, pages 353–362, 2014.

[Cho *et al.*, 2016] Yoon-Sik Cho, Greg Ver Steeg, Emilio Ferrara, and Aram Galstyan. Latent space model for multimodal social data. In *WWW*, pages 447–458, 2016.

[Daudin *et al.*, 2008] J-J Daudin, Franck Picard, and Stéphane Robin. A mixture model for random graphs. *Statistics and computing*, 18(2):173–183, 2008.

[Harville, 1998] David A Harville. Matrix algebra from a statistician's perspective. *Technometrics*, 40(2):164–164, 1998.

[Jacob *et al.*, 2014] Yann Jacob, Ludovic Denoyer, and Patrick Gallinari. Learning latent representations of nodes for classifying in heterogeneous social networks. In *WSDM*, pages 373–382, 2014.

[Karrer and Newman, 2011] Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

[Kim and Leskovec, 2012] Myunghwan Kim and Jure Leskovec. Multiplicative attribute graph model of real-world networks. *Internet Mathematics*, 8(1-2):113–160, 2012.

[Kyng *et al.*, 2015] Rasmus Kyng, Anup Rao, Sushant Sachdeva, and Daniel A Spielman. Algorithms for lipschitz learning on graphs. In *Proceedings of The 28th Conference on Learning Theory*, pages 1190–1223, 2015.

[Lancichinetti *et al.*, 2008] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4):046110, 2008.

[Mislove *et al.*, 2010] Alan Mislove, Bimal Viswanath, Krishna P Gummadi, and Peter Druschel. You are who you know: inferring user profiles in online social networks. In *WSDM*, pages 251–260, 2010.

[Nallapati *et al.*, 2008] Ramesh M Nallapati, Amr Ahmed, Eric P Xing, and William W Cohen. Joint latent topic models for text and citations. In *KDD*, pages 542–550, 2008.

[Newman and Clauset, 2015] MEJ Newman and Aaron Clauset. Structure and inference in annotated networks. *arXiv preprint arXiv:1507.04001*, 2015.

[Pfeiffer III *et al.*, 2014] Joseph J Pfeiffer III, Sebastian Moreno, Timothy La Fond, Jennifer Neville, and Brian Gallagher. Attributed graph models: Modeling network structure with correlated attributes. In *WWW*, pages 831–842, 2014.

[Saade *et al.*, 2014] Alaa Saade, Florent Krzakala, and Lenka Zdeborová. Spectral clustering of graphs with the bethe hessian. In *NIPS*, pages 406–414, 2014.

[Sen *et al.*, 2008] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93, 2008.

[Takac and Zabovsky, 2012] Lubos Takac and Michal Zabovsky. Data analysis in public social networks. In *International Scientific Conference AND International Workshop Present Day Trends of Innovations*, 2012.

[Talukdar and Crammer, 2009] Partha Pratim Talukdar and Koby Crammer. New regularized algorithms for transductive learning. In *ECMLPKDD*, pages 442–457, 2009.

[Wang and Wong, 1987] Yuchung J Wang and George Y Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19, 1987.

[Yamaguchi *et al.*, 2015] Yuto Yamaguchi, Christos Faloutsos, and Hiroyuki Kitagawa. Omni-prop: Seamless node classification on arbitrary label correlation. In *AAAI*, pages 3122–3128, 2015.

[Zhang *et al.*, 2014] Pan Zhang, Cristopher Moore, and Lenka Zdeborová. Phase transitions in semisupervised clustering of sparse networks. *Physical Review E*, 90(5):052802, 2014.

[Zhang *et al.*, 2016] Pan Zhang, Cristopher Moore, and MEJ Newman. Community detection in networks with unequal groups. *Physical Review E*, 93(1):012303, 2016.

[Zhou *et al.*, 2004] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. pages 321–328, 2004.

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

[Zhu *et al.*, 2005] Xiaojin Zhu, John Lafferty, and Ronald Rosenfeld. *Semi-supervised learning with graphs*. Carnegie Mellon University, language technologies institute, school of computer science, 2005.