Graph-Based Semi-Supervised Learning through the Lens of *Safety*

Shreyas Sheshadri[1*] **Avirup Saha**[2*] **Priyank Patel**[1] **Samik Datta**[3] **Niloy Ganguly**[2]

[1]Flipkart Internet Private Limited, India
[2]Indian Institute of Technology, Kharagpur, India
[3]Amazon, India
[*]*Equal contribution*

Abstract

Graph-based semi-supervised learning (G-SSL) algorithms have witnessed rapid development and widespread usage across a variety of applications in recent years. However, the theoretical characterisation of the efficacy of such algorithms has remained an under-explored area. We introduce a novel algorithm for G-SSL, CSX, whose objective function extends those of Label Propagation and Expander, two popular G-SSL algorithms. We provide data-dependent generalisation error bounds for all three aforementioned algorithms when they are applied to graphs drawn from a partially labelled extension of a versatile latent space graph generative model. The bounds we obtain enable us to characterise the predictive performance as measured by accuracy in terms of homophily and label quantity. Building on this we develop a key notion of ***GLM-safety*** which enables us to compare G-SSL algorithms on the basis of the range of graphs on which they obtain a guaranteed accuracy. We show that the proposed algorithm CSX has a better GLM-safety profile than Label Propagation and Expander while achieving comparable or better accuracy on synthetic as well as real-world benchmark networks.

1 INTRODUCTION

With the volume of data growing exponentially, it is becoming increasingly more difficult to obtain sufficient quantities of annotated data (required to develop data-driven techniques) due to the cost associated with labelling by human annotators. This makes semi-supervised learning (SSL) an attractive option for many practical purposes. In particular, graph-based semi-supervised learning (G-SSL) has received significant attention in the recent past due to its convexity, scalability and effectiveness [Chong et al., 2020]. A flurry of algorithms have been proposed to efficiently execute G-SSL tasks e.g. Label Propagation (LP) [Bengio et al., 2006], Modified Adsorption [Talukdar and Crammer, 2009], Expander [Ravi and Diao, 2016] etc. of which LP has long remained the workhorse of choice.

Despite application-specific developments, none of the aforementioned algorithms provides a guarantee on the minimum accuracy when applied on a particular graph. If a G-SSL algorithm is applied on an inappropriate graph, the accuracy of the algorithm suffers tremendously. This aspect of G-SSL is termed in literature [Gan et al., 2018] as *safety*, as it so happens the popular algorithms mostly perform well on the average but without a guarantee of safety may commit errors which may prove to be costly for any downstream applications using the label information. We also note that in SSL literature [Wang and Chen, 2013, Li and Liang, 2019], the concept of "safety" is interpreted in the general context of semi-supervised learning as the tendency of the algorithm's performance to degenerate with increasing quantities of unlabelled data. These notions of safety are qualitative in nature as they lack a means of rigorously determining whether an algorithm is "safer" than another.

There have been some limited works in the last years analyzing the theoretical efficacy of G-SSL [Yamaguchi and Hayashi, 2017, Saha et al., 2020]. [Yamaguchi and Hayashi, 2017] gives a qualitative notion of 'success' of LP on graphs generated by a partially labelled extension of the Stochastic Block Model (SBM) [Pearl, 1982] (called PLSBM), while [Saha et al., 2020] moves one step ahead and gives necessary conditions on graph structure, label volume and quality under which LP achieves a guaranteed accuracy on PLSBM graphs. To the best of our knowledge there are no further works in this direction. From these studies it is not clear how to analyse G-SSL algorithms other than LP and extend the analysis beyond SBM which is a restricted family of graphs. Thus while developing guarantees on arbitrary real-world graphs may be difficult, efforts need to be given to develop guarantees on a wider section of latent graph models.

Accepted for the 37th Conference on Uncertainty in Artificial Intelligence (UAI 2021).

In this chapter we provide a theoretical characterisation of LP and Expander [Ravi and Diao, 2016] as a function of the graph structure, degree of homophily and the quantity of labels (see Table 1) on a much wider family of graphs, the *Graph Latent Model* (GLM) [Ke and Honorio, 2019] extended for semi-supervised settings (refer to Definition 1), which has been shown to encompass a wide family of graphs like the Latent Space Model [Newman et al., 2002, Goldenberg et al., 2010], and also is more general than SBM. A theoretical (statistical and computational) characterisation of GLM models was studied by [Ke and Honorio, 2019] in an unsupervised setting, however to the best of our knowledge, the GLM or its related/special forms of latent graphs have been hardly studied from a theoretical perspective in a semi-supervised setting.

Our analysis enables us to introduce a novel theoretically motivated criterion of *GLM-safety* (refer to Definition 2) to compare any two G-SSL algorithms possessing a theoretical characterisation on GLM. This concept of *GLM-safety* is distinct from the prevalent notion of "safety" in semi-supervised learning in general [Wang and Chen, 2013, Li and Liang, 2019] and G-SSL in particular [Gan et al., 2018] in that it is quantitative and provides a metric which affords direct comparison of different algorithms. Furthermore, inspired by this analysis we propose a novel scalable G-SSL algorithm - *Class Sensitive Expander* (CSX) that boasts a better *GLM-safety* profile than either LP or Expander. The objective function of CSX enjoys an intuitive explanation: the revealed labels form *must-have* and *cannot-have* relationships amongst the pairs of instances – an idea that has been explored in the field of Constrained Spectral Clustering [Wang and Davidson, 2010b] and its relationship with LP variants [Wang et al., 2012] (refer to Equation (3)). The *must-have* relationship is exhibited between nodes whose labels are revealed to be of the same class, while the *cannot-have* relationship is exhibited between nodes whose labels are revealed to be of different classes.

The performance of CSX in terms of *GLM-safety* is characterised under the proposed theoretical framework in the form of generalization bounds/sufficient conditions on the level of accuracy that hold with high probability. The bounds make use of recent LCI concentration inequalities presented in [Ke and Honorio, 2019], where it was applied only to exact recovery (100% accuracy). In contrast, we furnish generalization bounds for any accuracy level.

We show that CSX is provably *GLM-safer* than LP [Bengio et al., 2006] and Expander [Ravi and Diao, 2016] (see Theorems 1 and 2) and discuss how the *GLM-safety* profile of CSX gets better with increase in label quantity and is retained even in the case of large graphs unlike LP and Expander. The theoretical characterisation of CSX developed herein is paired with thorough empirical evidences: with input instances sampled from GLM [Ke and Honorio, 2019] to support the proposed theory, as well as real-world benchmark networks. We show that compared to LP/Expander, the enhanced *GLM-safety* profile of CSX can translate to superior performance in some low-homophily scenarios, and at least comparable performance elsewhere[1].

2 GRAPH-BASED SEMI-SUPERVISED LEARNING

In this section, we review the problem setting of G-SSL and the propagation-style/message passing algorithms that are widely employed therein. We describe fixed point iterations, which are typical of propagation style algorithms, and which update the current node's estimate by a weighted average of its 'related' nodes' estimates until convergence. We conclude by introducing CSX which extends the existing G-SSL algorithms and enhances the *safety* profile.

G-SSL setting. Let (A, Y') denote a partially-labelled graph $G([n], E)$ with n nodes and E being the set of edges. The nodes of the graph carry labels, provided by the label matrix $Y^* \in \{0,1\}^{n\times 2}$. $A \in \{0,1\}^{n\times n}$ denotes the adjacency matrix, where the entry in the ith row and jth column, $A_{ij} = 1$ indicates the presence of an edge between nodes i and j (we assume that there are no self-loops: $A_{ii} = 0, \forall i \in [n]$). The ith row of the label matrix, $Y_i \in \{e_1, e_2\}$ provides the label of node $i \in [n]$, where e_1, e_2 are the 2 dimensional standard basis vectors representing the two classes. The partially revealed labels are provided by $Y' \in \{0,1\}^{n\times 2}$: if $Y'_i = \mathbf{0}$ then the node i's label is not revealed or the node is unlabelled and if $Y'_i = Y^*_i$ then the node i's label is revealed or the node is labelled. Let n_l number of nodes carry the binary labels and typically $n_l \ll n$. Let $\mathcal{L} = \{i | i \in [n] \wedge Y'_i \neq \mathbf{0}\}$ be the set of labeled indices and we define $S \in \mathbb{R}^{n\times n}, S_{ii} = 1$ if $i \in \mathcal{L}$ and $S_{ij} = 0$ otherwise.

LP and Expander. The goal of any G-SSL algorithm is to estimate Y^*. We first present Expander [Ravi and Diao, 2016] which uses the following fixed-point iterative updates for the $(t+1)^{th}$ iteration of the estimate $Y \in \mathbb{R}^{n\times 2}$, with the matrix entry Y_{vl} for node v and label index l and $t \geq 0$.

$$Y^{t+1}_{vl} = \frac{Y'_{vl} + \gamma_1 \sum\limits_{(v,j)\in E} A_{vj} Y^t_{jl} + \gamma_2 \lambda \mathbf{u}_{vl}}{S_{vv} + \gamma_1 \sum\limits_{(v,j)\in E} A_{vj} + \gamma_2} \qquad (1)$$

The initialization, $Y^0 = S\frac{1}{2}\mathbf{1}_n\mathbf{1}_2^T + Y'$ where $\mathbf{1}_d$ represents d dimensional all ones vector. This is because the estimate $\hat{Y}$ is viewed as a probability distribution. $\mathbf{u} = u\mathbf{1}_n e_1^T + (1-u)\mathbf{1}_n e_2^T$ where $u \in [0,1]$ is the prior distribution over the labels and typically a uniform prior with $u = \frac{1}{2}$ is used. The parameter $\lambda \in \{0,1\}$ is used to employ either L2 or uniform prior regularization. The classic Label Propagation [Bengio et al., 2006] is obtained as a special case of Expander fixed-

[1] Our code and data are available at http://bit.ly/PLGLMUAI21

point iterations (1) with $\lambda = 0$ i.e., no prior distribution over the labels is used and the initialization is typically $Y^0 = Y'$.

CSX. In case of LP-like algorithms, for a labelled node i, γ_1 is typically set to a low value, which causes the contribution from its neighbouring nodes to drop, so that $\hat{Y}_i$ does not vary much from the revealed label Y'_i. However the contribution from all the neighbouring nodes is of the same weight. Now, we modify the labelled node update twofold i.e., firstly we allow all the labelled nodes to contribute to the update and more importantly assign different weights to the contribution of labelled nodes i.e., labelled nodes that have the same label as the node to be updated contribute positively whereas the labelled nodes having a different label contribute negatively.

$$Y_{vl}^{t+1} = \frac{Y'_{vl} + \gamma_1 \sum\limits_{(v,j)\in E} A_{vj} Y_{jl}^t + \gamma_2 \lambda \mathbf{u}_{vl} + \gamma_3 \sum\limits_{j \neq v} W_{vj} Y_{jl}^t}{S_{vv} + \gamma_1 \sum\limits_{(v,j)\in E} A_{vj} + \gamma_2} \tag{2}$$

'Link' penalty. The new term in the fixed point iterations is W, where $W = 2Y'{Y'}^T - \mathbf{1}_{\mathcal{L}}\mathbf{1}_{\mathcal{L}}^T - S$ with $\mathbf{1}_{\mathcal{L}} \in \{0,1\}^n$ having ones at indices corresponding to $\mathcal{L}$. We observe, $W_{ij} = 0$ if any of the nodes i or j is unlabeled, therefore it does not affect the unlabeled nodes. More importantly, this term captures label/class sensitive edge conditions, because $W_{ij} = 1$ if both $i, j \in \mathcal{L}$ belong to the same class and $W_{ij} = -1$ if they belong to different classes. We see from (2) that we place more importance on values of Y_j of the neighbouring nodes j having the same label than unlabeled neighbours, and at the same time penalize neighbouring nodes having different labels.

Optimization objective. The role of the 'Link' penalty becomes more clear when we look at the optimization objective function $Q(Y)$ in (3) corresponding to the fixed point iterations or Jacobi iterations, where $L = D - A$, D is the diagonal matrix with D_{ii} containing the degree of node i. The link penalty term, in fact, does penalize the edges amongst the labelled nodes and is very similar to the 'Must link' and 'Cannot link' edge constraints employed by Constrained Spectral Clustering [Wang and Davidson, 2010a] which has been shown to have links to Label Propagation [Xiang Wang and Davidson, 2012]. We shall later show that this term renders the algorithm *GLM-safer*. The objective function also shows that Expander [Ravi and Diao, 2016] becomes a special case of CSX by setting $\gamma_3 = 0$, and so does LP [Bengio et al., 2006] by additionally setting $\lambda = 0$.

$$Q(Y) = \underbrace{\sum_{i\in\mathcal{L}} ||Y_i - Y'_i||^2}_{\text{Squared Loss}} + \underbrace{\gamma_1 Tr(Y^T L Y)}_{\text{Laplacian Regularization}} + \underbrace{\gamma_2 \sum_{i=1}^{n} ||Y_i - \lambda \mathbf{u}_i||^2}_{\text{Label Prior}} - \underbrace{\gamma_3 Y^T W Y}_{\text{'Link' Penalty}} \tag{3}$$

We note that the optimization problem for Expander involves minimization over a probability simplex unlike LP where it is typically over the entire real space. However this only manifests as difference in the initialization for fixed point updates, because (1) yields a weighted average over the labels Y_j of the nodes j, otherwise the algorithms are practically the same. Therefore, even LP can have minimization over the simplex just by changing the initialization to a vector that sums to one. In case of CSX, if $\lambda = 1$ then we require the minimization to be over the probability simplex, since a probabilistic prior $\mathbf{u}$ is being applied.

Convergence of Fixed Point iterations. A sufficient condition for convergence of Jacobi iteration is that matrix $M = S - \gamma_3 W + \gamma_1 L + \gamma_2 I$ should be strictly diagonally dominant, which can be achieved by setting $\gamma_2 > n_l \gamma_3$ (we refer the reader to Section 1 in the supplementary material for further details). This sufficient condition follows from first order conditions on the G-SSL objective (3).

Scalability of CSX. The scalability is derived from the potential to implement the fixed-point iteration atop a vertex-centric API to run on a wide variety of scalable graph processing systems, including distributed systems such as Pregel [Malewicz et al., 2010]. Note that this is in contrast with the SDP solver employed in [Ke and Honorio, 2019]. Fixed-point solvers are natively supported by industrial-strength G-SSL frameworks, such as Google's Expander graph-based machine learning framework[2] and Facebook's EdgeExplain [Chakrabarti et al., 2014].

3 PARTIALLY LABELED GRAPH LATENT MODEL (PL-GLM)

Given our goal of providing performance guarantees on the G-SSL algorithms discussed in the previous section, we proceed to study the partially labelled version of Graph Latent Model (GLM), a versatile generative model under which we analyze the performance of G-SSL algorithms. The GLM model includes a wider family of well studied graph models like Latent Space Model (LSM), Dot Product Graphs, Extremal Vertices Model, etc. (we refer to Table 1 in [Ke and Honorio, 2019]). We begin by first setting up the notation required to introduce PL-GLM, which generalises the vanilla GLM model to semi-supervised settings.

Definition 1 (PL-GLM). *Characterized by parameters $(n, \mathcal{X}, \epsilon_l, f, P_{e_1}, P_{e_2})$, PL-GLM is a generative model over a family of partially labeled graphs (A, Y') with the following properties:*

1. ***Balanced classes.*** *The true labels Y^* are such that $Y^{*T} 1_n = \frac{n}{2} 1_2$ i.e., equal sized classes.*

[2] https://ai.googleblog.com/2016/10/graph-powered-machine-learning-at-google.html

2. ***Latent vector generation.*** *$\mathcal{X}$ can be any arbitrary domain and for node i a latent vector x_i is sampled from the distribution $P_{Y_i} \in \{P_{e_1}, P_{e_2}\}$.*
3. ***Edge generation.*** *For each $i, j \in [n]$, the edges $(i, j) \in E$ of the graph are drawn i.i.d. from a Bernoulli distribution with parameter $f(x_i, x_j)$ i.e., $P(A_{i,j} = 1 \mid x_i, x_j) = f(x_i, x_j)$, where f is the homophily function that satisfies the symmetric property i.e., $f(x_i, x_j) = f(x_j, x_i)$.*
4. ***Label revelation.*** *For nodes $i \in [n]$, Y' is revealed from the underlying labels Y^* i.i.d. such that $P(Y'_i = \mathbf{0}) = 1 - \epsilon_l$ and $P(Y'_i = Y^*_i) = \epsilon_l$. Therefore, we assume no noise in the revealed labels.*

The label revelation assumption subsumes the already studied GLM model [Ke and Honorio, 2019] and extends it to the semi-supervised settings. Thereby, the PL-GLM model extends many of the well-known latent models on graphs like Latent Space Model. It can be shown (see Section 3 in the supplementary material) that GLM is more general than the Multiplicative Attributed Graph (MAG) model [Kim and Leskovec, 2012] which has been shown to cover a wide family of natural graphs, as well as SBM which is subsumed by MAG.

4 GENERALIZATION BOUNDS

We are now equipped to study the sufficiency conditions under which the G-SSL algorithms obtain an accuracy of $1 - \frac{s}{n}$, such that $s \leq \frac{n}{2}$. This requirement $s \leq \frac{n}{2}$ is justified as the expected accuracy of random guessing is $\frac{1}{2}$. LP and Expander both involve continuous optimization and subsequent thresholding to obtain the final binary label estimates. Therefore, taking cue from the previous literature on analysis of LP [Yamaguchi and Hayashi, 2017, Saha et al., 2020], we study the discrete versions of the objective, where the optimization is over $Y \in \{0, 1\}^{n \times 2}$ with additional constraints of one-hot encoded labels and balanced classes, however, computationally it is NP-hard, since there are $O(2^n)$ enumerations for Y. Since for any value of $\gamma_{(\cdot)} \in \mathbb{R}$ the solution exists for the discrete version of the objective, we also restrain $\gamma_{(\cdot)} \in \{0, 1\}$ in the discrete version. Thereby the generalization bounds obtained also shall be for the discrete version of (3) which is as follows:

$$\hat{Y} = \underset{Y}{\operatorname{argmin}}\ Q(Y) \tag{4}$$

$$Y \in \{0,1\}^{N \times 2} \text{ and } Y 1_2 = \mathbf{1}_n \text{ and } Y^T \mathbf{1}_n = \frac{n}{2} \mathbf{1}_2$$

We define the following parameters which govern the generalisation bounds for sufficiency which we derive.

$$p = E_X\big[f(x_i, x_j) | Y^*_i = Y^*_j\big] \quad q = E_X\big[f(x_i, x_j) | Y^*_i \neq Y^*_j\big] \tag{5}$$

The parameters p and q are expected values for presence of intra-class and inter-class edges in the graph. We proceed to state the main result of the paper which provides sufficient conditions under which we obtain guarantees on accuracy for discrete versions of CSX, LP and Expander. Let $\ell^{0\text{-}1}(Y^*, \hat{Y}) = \frac{1}{n} \sum_{i \in [n]} \mathbf{I}_{\{Y^*_i \neq \hat{Y}_i\}}$ be the usual 0-1 loss counting the fraction of errors in $\hat{Y}$ with $\mathbf{I}_{\{\cdot\}}$ being the indicator function.

Theorem 1. *Given every $\gamma_1 > 0, \gamma_3 > 0$ and $\gamma_1(p - q) + \gamma_3 \epsilon_l^2 + \frac{\epsilon_l}{n} > 0$, under the following (sufficient) condition*

$$\Big(\gamma_1(p - q) + \gamma_3 \epsilon_l^2 + \frac{\epsilon_l}{n}\Big)^2 \geq 48c \Big(1 - \frac{s}{n}\Big) \frac{\ln n}{n} \tag{6}$$

where $c = \gamma_1(\frac{\gamma_1}{2} + \frac{1}{3}) + \gamma_3(\frac{\gamma_3}{2} + \frac{\epsilon_l^2}{3}) + \frac{1}{3}\epsilon_l + \frac{1}{8}$, $\hat{Y}$ provided in (4) recovers the true labels Y^* with at most $s \in \mathbb{Z}$ mistakes where $0 \leq s < \frac{n}{2}$, under the PL-GLM generative model, with a high probability i.e., $P\left(\ell^{0\text{-}1}(Y^*, \hat{Y}) > \frac{s}{n}\right) < \frac{1}{n}$.

Proof Sketch. Let $m(Y)$ indicate the number of mismatches/mistakes between Y and Y^* such that $m(Y^*) = 0$. We show that, for any $Y \neq Y^*$ over the domain of optimization, if $m(Y) > s$ (since we require minimum accuracy of $1 - \frac{s}{n}$), $Q(Y) > Q(Y^*)$ holds true w.h.p i.e., Y would not minimize the objective. In other words, we show that w.h.p $m(\hat{Y}) \leq s$. To achieve this, we require a concentration bound on the term $Q(Y) - Q(Y^*)$ over the latent variables $X = \{x_i, i \in [n]\}$ under PL-GLM. Therefore we employ Latent Conditional Independence (LCI) inequalities from [Ke and Honorio, 2019] to show that (6) is sufficient for $Q(Y) > Q(Y^*)$ to hold, for any Y. Finally we employ union bound to prove the bound for all Y which have $m(Y) > s$. We note that the term $\gamma_3 \epsilon_l^2$ in (6) is contributed by the newly introduced link penalty term in the objective (3) and we shall see in future sections, that the said term plays an important role in defining the *safety* profile of CSX. Further details are provided in Section 2 of the supplementary material.

Novelty. The proof follows a strategy similar to [Chen and Xu, 2014], however we simplify the proof by relying on bounding a simpler quantity $m(Y)$, the number of mismatches between any Y and Y^*, resulting in a much cleaner proof employing well known bounds, whereas [Chen and Xu, 2014] relies on a complicated lemma (Lemma 1.1 of [Chen and Xu, 2014]). Moreover both [Chen and Xu, 2014] and [Ke and Honorio, 2019] provide bounds for only $s = 0$ or 100% accuracy, while we also extend the proof technique for any accuracy $> 50\%$.

Takeaways.

- **Corollaries.** Table 1 provides the sufficiency conditions for LP, Expander and CSX obtained from Theorem 1 by carrying out the specific substitutions for $\gamma_{(\cdot)}, \lambda \in \{0, 1\}$. We observe that for LP and Expander the bounds are the same, because the term involving $\mathbf{u}$ ends up being a constant under the discrete version.

Algorithm	Constraints	Sufficiency Condition $\left(f(s,n) = 48\left(1-\frac{s}{n}\right)\frac{\ln n}{n}\right)$
LP	$\gamma_3 = 0$ $\lambda = 0$	$\left((p-q)+\frac{\epsilon_l}{n}\right)^2 \geq \frac{1}{3}\left(\frac{23}{8}+\epsilon_l\right) f(s,n)$ $(p-q)+\frac{\epsilon_l}{n} > 0$
Expander	$\gamma_3 = 0$	
CSX	-	$\left((p-q)+\epsilon_l^2+\frac{\epsilon_l}{n}\right)^2 \geq \frac{1}{3}\left(\frac{35}{8}+\epsilon_l(\epsilon_l+1)\right) f(s,n)$ $(p-q)+\epsilon_l^2+\frac{\epsilon_l}{n} > 0$

Table 1: A compendium of the sufficiency bounds presented in this work, these are obtained by substituting the parameter values in Eq. (6)

- **Homophily.** In case of latent models, we can have an alternate notion of homophily in the sense of expectation, i.e., $p > q$, expectation of an intra-class edge is more than that of an inter-class edge. Therefore $p-q$ is a measure of 'Expected' Homophily. Thus, an increase in homophily $p-q$ makes it easier to satisfy the bound (6). In case of LP and Expander, the condition $p-q+\frac{\epsilon_l}{n} > 0$, as found in Table 1, must hold for bounds to be valid; this implies that for large graphs the homophily condition should hold. However, for CSX we get a much looser condition, $p-q+\epsilon_l^2+\frac{\epsilon_l}{n} > 0$, as found in Table 1. This implies that even for graphs where homophily does not hold so strongly, guarantees on accuracy can be provided, given a higher label revelation probability ϵ_l.
- **Effect of revealed labels.** When $\epsilon_l = 0$, i.e. when there are no labelled nodes, the bounds are similar to the ones obtained in an unsupervised setting [Ke and Honorio, 2019]. When $\epsilon_l = 1$, i.e. when all nodes are labelled, the bounds become trivial for a large enough n. It is clear that increasing ϵ_l relaxes the bound (6).
- **Accuracy.** Given a graph and parameters $\gamma_{(\cdot)}$, the bounds become harder to satisfy with increasing accuracy i.e., the value of RHS of (6) increases linearly with a decrease in the number of mistakes s.

5 GLM-SAFETY

We begin by introducing the notion of *GLM-safety*, under the PL-GLM setting, which is tied to the guaranteed performance/accuracy of any algorithm $\mathbf{A}$, that outputs labels $\hat{Y} = \mathbf{A}(A, Y')$. We note that given true underlying labels Y^* the observed PL-GLM graph (A, Y') can vary considerably depending on the parameters $(\epsilon_l, f, P_{e_1}, P_{e_2})$, which are unobserved, thereby the accuracy of $\mathbf{A}$ also varies.

Parameter Space. We begin by defining the parameter space under a PL-GLM model and accuracy $(1-\frac{s}{n})$, for which guarantees exist, as $\mathbf{\Omega}(Y^*, \epsilon_l, \mathbf{A}(A, y'), s) = \{(p,q) | P\left(\ell^{0\text{-}1}(Y^*, \mathbf{A}(A, Y')) > \frac{s}{n}\right) < \frac{1}{n}\}$, for any algorithm $\mathbf{A}$. Essentially, $\mathbf{\Omega}$ captures the subset of parameters p, q (that generate (A, Y')) for which the algorithm $\mathbf{A}$ achieves the said accuracy $1-\frac{s}{n}$ w.h.p. under label revelation probability ϵ_l, keeping Y^* fixed. Henceforth we shall simply write $\mathbf{\Omega}(\mathbf{A}, s)$ for the sake of brevity.

Definition 2 (GLM-Safety)**.** *An algorithm $\mathbf{A}_1$ is said to be GLM-safer than algorithm $\mathbf{A}_2$, for a given accuracy $1-\frac{s}{n}$, if $\mathbf{\Omega}(\mathbf{A}_2, s) \subseteq \mathbf{\Omega}(\mathbf{A}_1, s)$, with labels Y^* and revelation parameter ϵ_l remaining fixed.*

Therefore, the *GLM-safer* algorithm has a larger space of parameters under which it is guaranteed (w.h.p) to achieve the desired accuracy than the less *GLM-safe* algorithm. The form of the sufficiency bounds in (6) is $(p-q+b_{\mathbf{A}})^2 \geq c_{\mathbf{A}}$, for any algorithm $\mathbf{A}$. The particular forms of $b_{\mathbf{A}}, c_{\mathbf{A}}$ for the algorithms considered in this chapter can be gathered from Table 1. We note that, $c_{\mathbf{A}} = \Omega(\frac{\ln n}{n})$ and $b_{\mathbf{A}} \in [0,1]$, therefore we can say $\exists n_{\mathbf{A}}$, such that $\forall n > n_{\mathbf{A}}$, $\mathbf{\Omega}(\mathbf{A}, s) = \{(p,q) \mid p-q \geq \sqrt{c_{\mathbf{A}}} - b_{\mathbf{A}} \text{ and } p, q \in [0,1]\}$. Finally, we have the following result which we make use of to discuss *GLM-safety* of all the algorithms considered in this chapter.

$$|\mathbf{\Omega}(\mathbf{A}, s)| = \int_{q=0}^{1} \int_{p=\max(0, q+\sqrt{c_{\mathbf{A}}}-b_{\mathbf{A}})}^{1} 1 \; dp \; dq$$
$$= \min\left(1, \frac{1}{2} + b_{\mathbf{A}} - \sqrt{c_{\mathbf{A}}}\right) \tag{7}$$

Note that the maximum possible value of $|\mathbf{\Omega}(\mathbf{A}, s)|$ is 1 since $p, q \in [0,1]$. In case of PL-GLM, $|\mathbf{\Omega}(\mathbf{A}_2, s)| \leq |\mathbf{\Omega}(\mathbf{A}_1, s)| \implies \mathbf{\Omega}(\mathbf{A}_2, s) \subseteq \mathbf{\Omega}(\mathbf{A}_1, s)$ (see Figure 3 for an illustration) and hence it is sufficient to use the set-cardinality $|\mathbf{\Omega}(\mathbf{A}, s)|$ as a "measure" of *GLM-safety* of an algorithm $\mathbf{A}$.

GLM-Safety of LP and Expander. We observe from Table 1 that $b_{Exp} = b_{LP}$, $c_{LP} = c_{Exp}$, therefore we conclude that discrete versions of LP and Expander are equally *GLM-safe* i.e., $\mathbf{\Omega}(\mathbf{A}_{\text{Exp}}, s) = \mathbf{\Omega}(\mathbf{A}_{\text{LP}}, s)$, since their bounds are the same. The label revelation governed by ϵ_l does not have much effect on the *GLM-safety* profile (as it has a factor of $\frac{1}{n}$) for both the algorithms. We observe $b_{Exp/LP} = \frac{\epsilon_l}{n}$ and $c_{Exp/LP} = \Omega(\frac{\ln n}{n})$, therefore, the *GLM-safety* of LP and Expander decreases with increasing n.

GLM-Safety of CSX. Now, we proceed to show that the discrete version of CSX enjoys a better *GLM-safety* profile than both LP and Expander.

Theorem 2 (CSX is *GLM-safer* than both Expander and LP)**.** *$\exists n'$ such that $\forall n > n'$, $\mathbf{\Omega}(\mathbf{A}_{Exp/LP}, s) \subset \mathbf{\Omega}(\mathbf{A}_{CSX}, s)$ under the condition $\epsilon_l > \epsilon_l^c$ where $\epsilon_l^c = \sqrt{\frac{\sqrt{f(s,n)}}{4-\frac{2}{3}\sqrt{f(s,n)}}}$*

Proof. We begin by showing $\mathbf{\Omega}(\mathbf{A}_{\text{Exp/LP}}, s) \subset \mathbf{\Omega}(\mathbf{A}_{\text{CSX}}, s)$.

Using (7), for $n > \max(n_{\text{CSX}}, n_{\text{LP}}, n_{\text{Exp}})$ we have

$$
\begin{aligned}
&|\mathbf{\Omega}(\mathbf{A}_{\text{CSX}}, s)| - |\mathbf{\Omega}(\mathbf{A}_{\text{Exp/LP}}, s)| \\
&= \epsilon_l^2 - \sqrt{f(s,n)} \left(\sqrt{\frac{\epsilon_l^2}{3} + \frac{1}{2} + \frac{\epsilon_l + \frac{23}{8}}{3}} - \sqrt{\frac{\epsilon_l + \frac{23}{8}}{3}} \right) \\
&\geq \epsilon_l^2 - \frac{1}{2}\sqrt{f(s,n)}(\frac{\epsilon_l^2}{3} + \frac{1}{2})
\end{aligned}
$$

The last inequality follows from $\sqrt{a} - \sqrt{b} = \frac{a-b}{\sqrt{a}+\sqrt{b}}$ and by setting ϵ_l terms in the denominator to 0. Therefore, by setting $\epsilon_l^2 - \frac{1}{2}\sqrt{f(s,n)}(\frac{\epsilon_l^2}{3} + \frac{1}{2}) > 0$ we obtain the condition $\epsilon_l > \sqrt{\frac{\sqrt{f(s,n)}}{4 - \frac{2}{3}\sqrt{f(s,n)}}}$ (for $n \geq 20$ we have $4 - \frac{2}{3}\sqrt{f(s,n)} > 0$ for any s) and hence the proof. □

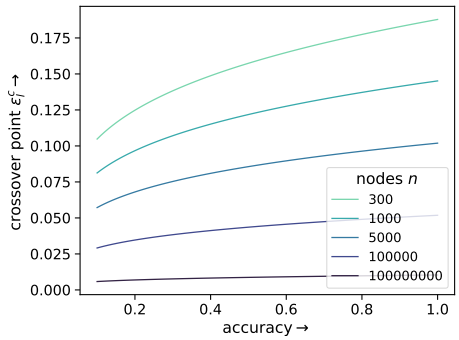


Figure 1: 'Crossover point' ϵ_l^c between CSX and LP/Expander as a function of guaranteed accuracy $(1 - \frac{s}{n})$ and the number of nodes n. $\epsilon_l > \epsilon_l^c$ guarantees that CSX is *GLM-safer* than LP/Expander on two-community PL-GLM graphs

Discussion. We begin by noting that for larger n, we have $b_{\text{CSX}} \approx \epsilon_l^2$ and $b_{\text{CSX}} > \sqrt{c_{\text{CSX}}}$, therefore, unlike LP and Expander, increasing label revelation probability ϵ_l improves *GLM-safety* of CSX significantly. Therefore, by the same reasoning we find that unlike LP or Expander, the *GLM-safety* of CSX does not decrease with n. We gather from the proof, that CSX is *GLM-safer* under the condition $\epsilon_l > \epsilon_l^c$ as seen in Theorem 2, however, since $f(s,n)$ is a decreasing function of n, this condition gets readily satisfied even at small values of ϵ_l under reasonably large n. In this context, in Figure 1 we plot ϵ_l^c, termed the 'crossover point', as a function of the accuracy $(1 - \frac{s}{n})$ and the number of nodes n. Hence, for a given accuracy guarantee and number of nodes n, if $\epsilon_l > \epsilon_l^c$ then CSX is guaranteed to be *GLM-safer* than LP and Expander. We can see that ϵ_l^c increases with increase in the guaranteed accuracy and decreases with increase in the number of nodes n. We see that for large values of n ($O(10^8)$), the crossover point ϵ_l^c is quite small ($< 10^{-2}$) for any value of the accuracy, meaning that CSX is almost always *GLM-safer* than LP and Expander for large graphs[3]. Furthermore it is to be noted that $\epsilon_l^c = \sqrt{\frac{\sqrt{f(s,n)}}{4 - \frac{2}{3}\sqrt{f(s,n)}}}$ is actually a loose upper bound for the 'true' crossover point (hence the values plotted in Figure 1 are *conservative estimates*) and in practice CSX can be *GLM-safer* than LP and Expander even for $\epsilon_l < \epsilon_l^c$. We show an example of such a case in our experiments.

6 EMPIRICAL RESULTS

In this section we describe experiments on partially labelled synthetic graphs generated from PL-GLM followed by partially labelled real world graphs.

6.1 SYNTHETIC GRAPHS GENERATED BY PL-GLM

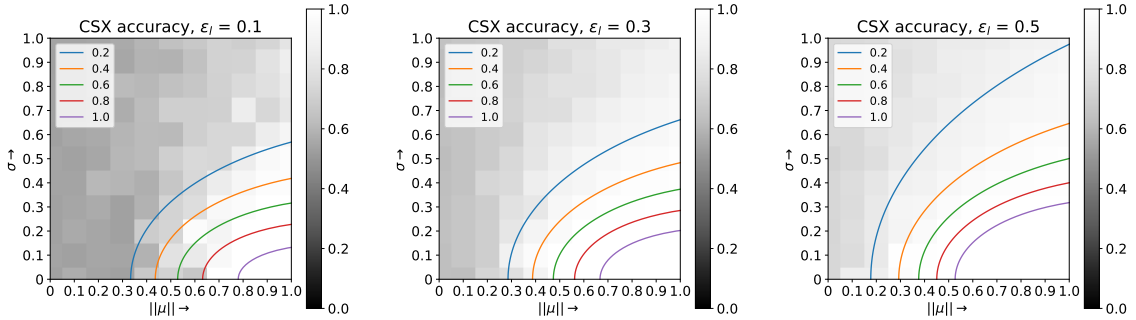
Similar to [Ke and Honorio, 2019] we have used the following settings for generating synthetic graphs from PL-GLM as defined in Section 3:

- Number of nodes, $n = 300$ and dimension of the latent space, $d = 2$.
- Latent vector distributions for the two classes, $P_{e_1} = N_2(\boldsymbol{\mu}, \sigma^2\mathbf{I})$, and $P_{e_2} = N_2(-\boldsymbol{\mu}, \sigma^2\mathbf{I})$ i.e., 2-dimensional Gaussian distributions with means $\pm\boldsymbol{\mu}$ and variance $\sigma^2\mathbf{I}$ with $\sigma \in \mathbb{R}$.
- $f(x_i, x_j) = \exp(-\|x_i - x_j\|^2)$

It can be seen that higher the value of $\|\boldsymbol{\mu}\|$ and lower the value of σ, greater the difference in the support of the two distributions. Following [Ke and Honorio, 2019] it can be shown that in these settings, $p = (4\sigma^2 + 1)^{-1}$ and $q = (4\sigma^2 + 1)^{-1}\exp(-4\|\boldsymbol{\mu}\|^2(4\sigma^2 + 1)^{-1})$, so we can find the theoretical regions of guaranteed accuracy according to Theorem 1.

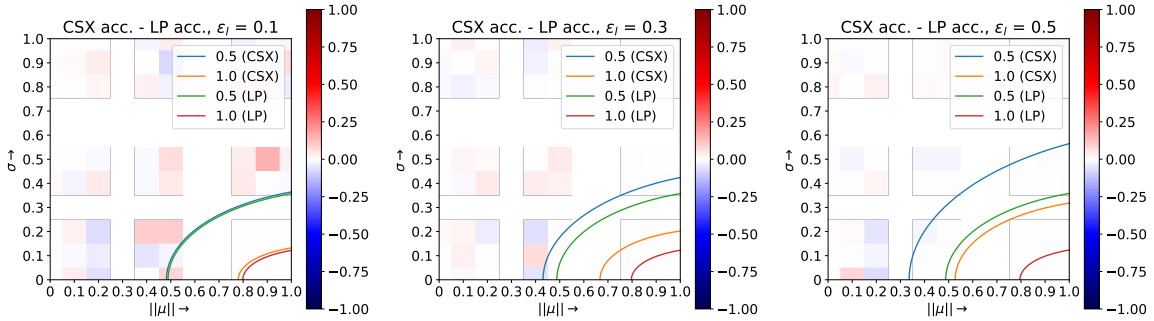
In Figure 2 we show the accuracy heatmaps obtained by CSX on graphs generated from PL-GLM using the above settings, where white denotes an accuracy of 1 and black denotes an accuracy of 0. For generating these graphs, $\|\boldsymbol{\mu}\|$ and σ were varied from 0 to 1 in steps of 0.1 while the label revelation probability, ϵ_l was varied from 0.1 to 0.5 in steps of 0.2. The shade of any portion of a heatmap indicates the expected accuracy obtained by CSX on graphs sampled from PL-GLM in the above setting with the corresponding values of $\|\boldsymbol{\mu}\|$, σ and ϵ_l. The regions of guaranteed accuracy according to Theorem 1 have been demarcated (the regions lie below the corresponding curves). We note that the region beneath the curve corresponding to accuracy 1 is fully white.

[3]Please note that these conclusions are only guaranteed to hold for two-community graphs generated from PL-GLM and may not be applicable to real-world graphs in general having diverse network structures.

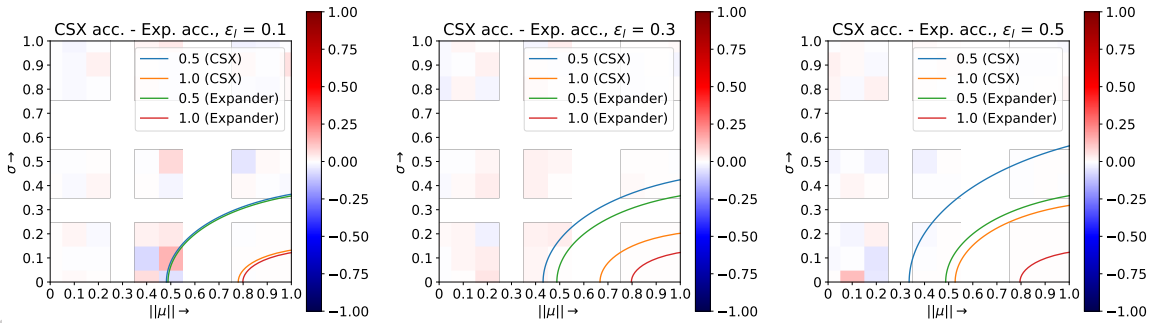


(a) $\epsilon_l = 0.1$ (b) $\epsilon_l = 0.3$ (c) $\epsilon_l = 0.5$

Figure 2: Variation of accuracy obtained by CSX with the label revelation probability ϵ_l on partially labeled two-community graphs generated from PL-GLM. The theoretical regions of guaranteed accuracy have been plotted according to Theorem 1 as the regions below the corresponding curves.



(a) $\epsilon_l = 0.1$ (b) $\epsilon_l = 0.3$ (c) $\epsilon_l = 0.5$



(d) $\epsilon_l = 0.1$ (e) $\epsilon_l = 0.3$ (f) $\epsilon_l = 0.5$

Figure 3: Difference of accuracy obtained by CSX and LP ((a) through (c)), and CSX and Expander ((d) through (f)) with the label revelation probability ϵ_l on partially labeled two-community graphs generated from PL-GLM. The regions corresponding to 50% and 100% guaranteed accuracy for each pair of algorithms are shown for comparison.

In Figure 3 we show the difference in accuracy (in the form of heatmaps as described above, but with a diverging color map) between (i) CSX and LP and (ii) CSX and Expander along with the regions corresponding to 50% and 100% accuracy for each pair of algorithms. In these heatmaps, the red regions indicate the parameter configurations where CSX is superior to LP or Expander, while blue shows the opposite. Since the red and blue regions are quite evenly distributed, we see from these figures that CSX performs similarly to, if not better than, both LP and Expander in most parameter settings. However, CSX offers better *safety* guarantees than LP and Expander since the corresponding regions of guaranteed accuracy are larger. Furthermore as expected the difference in *safety* increases with increasing ϵ_l. We also note in this context that unlike CSX the *safety* of LP and Expander does not perceptibly increase with increasing ϵ_l due to lack of the ϵ_l^2 term (since $\gamma_3 = 0$) in the LHS of the bound in Eq. (6) (Theorem 1).

Of special interest is the case of $\epsilon_l = 0.1$ in Figures 3(a) and 3(d). In Figure 1 the 'crossover point' ϵ_l^c corresponding to the graph of $n = 300$ is well above 0.1 for accuracy guarantees of 50% and 100%. Yet, from Figures 3(a) and 3(d) we see that the guaranteed accuracy regions for CSX are actually larger (though only slightly) than for LP/Expander. As we have pointed out in the Discussion in Section 5, this indicates that the values of ϵ_l^c, as given in Theorem 2 and plotted in Figure 1, are conservative estimates of the 'true' crossover point.

6.2 REAL WORLD GRAPHS

For conducting our experiments, we constructed four graphs G_{1-4}, each having two-community structure, from three popular real-world datasets as shown in Table 2 along with some statistics. In case of `email-Eu-core`, we chose to form two sub-graphs namely G_1, induced by the nodes with the two most frequent labels, and G_2, induced by the nodes with the most frequent and fifth most frequent labels. For G_3 and G_4, we chose the subgraph induced by the nodes with the two most frequent labels from the datasets `Citeseer` and `Cora` respectively. The two-dimensional latent space visualizations of all the graphs are shown in Figure 4. These visualizations were generated by the DynetLSM library [Loyal and Chen, 2020]. Since these node representations can separate the ground truth classes to a large extent, it can be seen that the Latent Space Model (and hence GLM) is a fairly good model for these graphs.

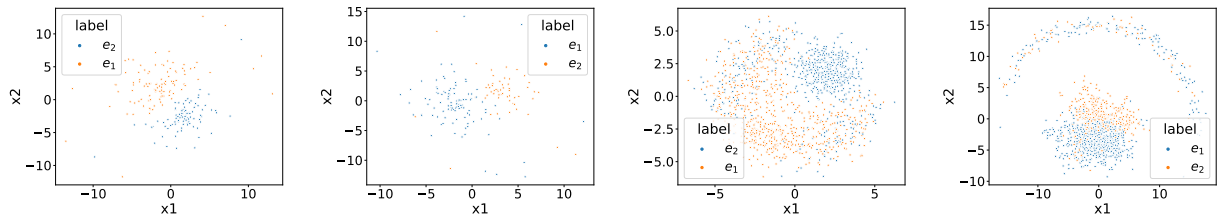
In Figure 5 we show the variation of accuracy of LP, Expander, and CSX with the label revelation probability ϵ_l on these graphs, for ϵ_l in the range $(0, 0.5]$. We obtain a Homophily estimate, $\hat{p} - \hat{q}$ by the observed intra-class and inter-class edge probabilities; and the estimated values $\hat{p}, \hat{q}$ for all the graphs are shown in Table 2. We analyse the performance of various algorithms in the light of two factors: (i) whether the classes are balanced or unbalanced, and (ii) whether the Homophily estimate is high or low, as illustrated in Table 2. We use a uniform prior over labels in case of balanced classes and a non-uniform prior $\mathbf{u} = {}^2\!/\!_3 1_n e_1^T + {}^1\!/\!_3 1_n e_2^T$ in case of unbalanced classes.

`email-Eu-core`. On graphs G_1 (Figure 5(a)) and G_2 (Figure 5(b)), due to the high value of $\hat{p} - \hat{q}$ all the algorithms perform very well. However on G_1, CSX is more stable for low values of ϵ_l (i.e. sparse labels) than LP or Expander, indicating better robustness. Although the *GLM-safety* guarantees we provide are only for minimum or worst case accuracy, we can still attempt to give an intuitive explanation of this effect. At lower ϵ_l values, the 'Link' penalty term in CSX will end up placing a higher weight on the revealed/true labels (by boosting the intra-cluster edges and penalizing inter-cluster edges) and therefore CSX will be less prone to flipping the revealed labels unlike LP and Expander.For a graph with balanced communities, lower the value of ϵ_l, i.e. lesser the number of revealed labels, stronger is the debilitating effect of label flipping on overall accuracy since the algorithm fails to distinguish between the two classes. This effect is pronounced in the graph G_1 in Figure 5(a) for $\epsilon_l = 0.05$ since, due to high homophily, the accuracy of all the algorithms is generally above 90% for values of $\epsilon_l \geq 0.1$. Hence LP and Expander suffer a sharp fall in average accuracy (which also suffers from instability i.e. high variance) for $\epsilon_l < 0.1$, while CSX is robust to this effect. Note however that the effect of label flipping is not as pronounced in G_2 (which also has high homophily) since the classes are unbalanced and so the algorithm can always correctly infer the label of the larger community. Furthermore, on G_2 the non-uniform prior gives a slight advantage to both CSX and Expander over LP, while there is no tangible difference in performance between the two algorithms. However, due to imbalance in the classes, at the low value of $\epsilon_l = 0.05$ the accuracy of all the algorithms suffers, presumably because the algorithms tend to ignore the smaller community due to presence of very few revealed labels corresponding to it.

`Citeseer` and **`Cora`**. On graphs G_3 (Figure 5(c)) and G_4 (Figure 5(d)), due to the low value of $\hat{p} - \hat{q}$ all the algorithms perform rather poorly compared to G_1 and G_2. Therefore on these graphs the label flipping effect is not visible for low values of ϵ_l, since the accuracy in general is quite low (this is especially true for G_3 which has balanced communities and is more susceptible to this effect). On graph G_3, due to the uniform prior, Expander does not enjoy any advantage over LP and hence their performances are roughly similar. However, CSX is distinctly superior in performance to both LP and Expander. Recall that in the discussion on **Homophily** in Section 4, we explained that CSX enjoys better *GLM-safety* guarantees for low values of $p - q$ than both LP and Expander. From Equation (6), as G_3 has a low value of $\hat{p} - \hat{q}$ (low homophily) the contribution

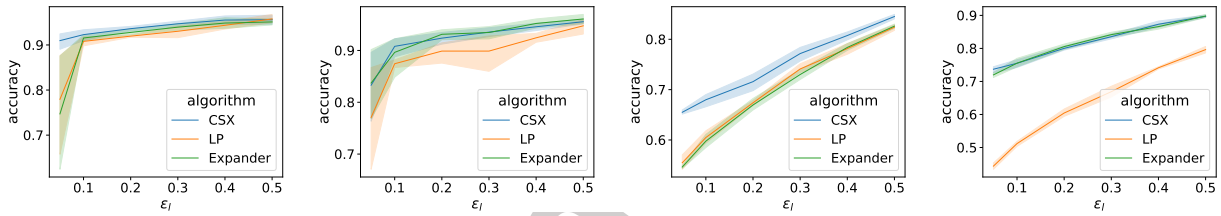
Graph	Class Balance	Homophily Estimate $\hat{p} - \hat{q}$	Dataset	n	E	n_1	n_2	$\hat{p}$	$\hat{q}$
G_1	Balanced	High	`email-Eu-core` [Yin et al., 2017, Leskovec et al., 2007]	201	2963	109	92	0.1388	0.0083
G_2	Unbalanced	High	`email-Eu-core` [Yin et al., 2017, Leskovec et al., 2007]	164	1693	109	55	0.1095	0.0066
G_3	Balanced	Low	`Citeseer` [Sen et al., 2008, Lu and Getoor, 2003]	1347	1849	681	666	0.0018	0.0002
G_4	Unbalanced	Low	`Cora` [Sen et al., 2008, Lu and Getoor, 2003]	1244	2055	818	426	0.0022	0.0002

Table 2: Statistics of two-community real-world graphs for binary node classification. n is the number of nodes and E is the set of edges. $n_1 = |\{i : Y_i^* = e_1\}|$ and $n_2 = |\{i : Y_i^* = e_2\}|$.



(a) G_1 - `email-Eu-core` (b) G_2 - `email-Eu-core` (c) G_3 - `Citeseer` (d) G_4 - `Cora`

Figure 4: Two-dimensional latent space visualizations of two-class real world graphs.



(a) G_1 - `email-Eu-core` (b) G_2 - `email-Eu-core` (c) G_3 - `Citeseer` (d) G_4 - `Cora`

Figure 5: Variation of accuracy obtained by G-SSL algorithms derived from Eq. (3) with the label revelation probability ϵ_l on partially labeled two-community real-world graphs.

of the second term ($\gamma_3 \epsilon_l^2$) in the LHS becomes prominent. However, since $\gamma_3 = 0$ for LP and Expander (see Table 1) their *GLM-safety* suffers in comparison to CSX. This result therefore shows that a *GLM-safer* algorithm is also likely to be performance-wise *stronger* in this case. On graph G_4, which has unbalanced communities, the non-uniform prior gives a significant advantage to both CSX and Expander over LP (the performance gap is much more pronounced compared to the similarly unbalanced case of G_2 due to the difference in homophily between the two graphs), while there is no tangible difference in performance between the two algorithms.

7 CONCLUSION

In this work we have presented a novel criterion of *GLM-safety* which enables us to compare any two theoretically characterisable G-SSL algorithms on the basis of the range of graphs drawn from PL-GLM on which they achieve a given guaranteed accuracy. We have also presented a new G-SSL algorithm, CSX, which we have shown to possess a better *GLM-safety* profile than LP [Bengio et al., 2006] and Expander [Ravi and Diao, 2016]. The *GLM-safety* profile of CSX improves at a faster rate than LP and Expander with increasing number of revealed labels. As we know, the success of all G-SSL algorithms largely depends on the level of homophily a graph displays; however the significance of CSX lies in the fact that it can provide better performance guarantees on graphs showing less homophily. An industrial-strength distributed implementation of CSX is left for a future work; and so is an extension of our theory to other G-SSL algorithms and natural graphs in the wild.

Acknowledgements

The authors would like to acknowledge Disha Makhija, University of Texas at Austin for helpful inputs. This work has been supported by the project "AI/ML Techniques for Online Shopping" sponsored by Flipkart Internet Private Limited.

References

Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. 11 label propagation and quadratic criterion. researchgate.net, 2006.

Deepayan Chakrabarti, Stanislav Funiak, Jonathan Chang, and Sofus A Macskassy. Joint inference of multiple label types in large networks. arXiv preprint arXiv:1401.7709, 2014.

Yudong Chen and Jiaming Xu. Statistical-computational phase transitions in planted models: The high-dimensional setting. In International Conference on Machine Learning, pages 244–252, 2014.

Yanwen Chong, Yun Ding, Qing Yan, and Shaoming Pan. Graph-based semi-supervised learning: A review. Neurocomputing, 2020.

Haitao Gan, Zhenhua Li, Wei Wu, Zhizeng Luo, and Rui Huang. Safety-aware graph-based semi-supervised learning. Expert Systems with Applications, 107:243–254, 2018.

Anna Goldenberg, Alice X Zheng, Stephen E Fienberg, and Edoardo M Airoldi. A survey of statistical network models. Now Publishers Inc, 2010.

Chuyang Ke and Jean Honorio. Exact recovery in the latent space model. arXiv preprint arXiv:1902.03099, 2019.

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

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. ACM transactions on Knowledge Discovery from Data (TKDD), 1(1):2–es, 2007.

Yu-Feng Li and De-Ming Liang. Safe semi-supervised learning: a brief introduction. Frontiers of Computer Science, 13(4):669–676, 2019.

Joshua Daniel Loyal and Yuguo Chen. A bayesian nonparametric latent space approach to modeling evolving communities in dynamic networks. arXiv preprint arXiv:2003.07404, 2020.

Qing Lu and Lise Getoor. Link-based classification. In Proceedings of the Twentieth International Conference on International Conference on Machine Learning, ICML'03, page 496–503. AAAI Press, 2003. ISBN 1577351894.

Grzegorz Malewicz, Matthew H Austern, Aart JC Bik, James C Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. Pregel: a system for large-scale graph processing. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, pages 135–146, 2010.

Mark EJ Newman, Duncan J Watts, and Steven H Strogatz. Random graph models of social networks. Proceedings of the National Academy of Sciences, 99(suppl 1):2566–2572, 2002.

Judea Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. Cognitive Systems Laboratory, School of Engineering and Applied Science, University of California, Los Angeles, 1982.

Sujith Ravi and Qiming Diao. Large scale distributed semi-supervised learning using streaming approximation. In AISTATS, 2016.

Avirup Saha, Shreyas Sheshadri, Samik Datta, Niloy Ganguly, Disha Makhija, and Priyank Patel. Understanding the success of graph-based semi-supervised learning using partially labelled stochastic block model. In Christian Bessiere, editor, Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, pages 1345–1351. ijcai.org, 2020. doi: 10.24963/ijcai.2020/187. URL https://doi.org/10.24963/ijcai.2020/187.

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

Partha Pratim Talukdar and Koby Crammer. New regularized algorithms for transductive learning. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 442–457. Springer, 2009.

Xiang Wang and Ian Davidson. Flexible constrained spectral clustering. In KDD, 2010a.

Xiang Wang and Ian Davidson. Flexible constrained spectral clustering. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 563–572, 2010b.

Xiang Wang, Buyue Qian, and Ian Davidson. Labels vs. pairwise constraints: A unified view of label propagation and constrained spectral clustering. In 2012 IEEE 12th International Conference on Data Mining, pages 1146–1151. IEEE, 2012.

Yunyun Wang and Songcan Chen. Safety-aware semi-supervised classification. IEEE transactions on neural networks and learning systems, 24(11):1763–1772, 2013.

Buyue Qian Xiang Wang and Ian Davidson. Labels vs. pairwise constraints: A unified view of label propagation and constrained spectral clustering. In ICDM, 2012.

Yuto Yamaguchi and Kohei Hayashi. When does label propagation fail? a view from a network generative model. In IJCAI, pages 3224–3230, 2017.

Hao Yin, Austin R Benson, Jure Leskovec, and David F Gleich. Local higher-order graph clustering. In Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pages 555–564, 2017.