Journal of Machine Learning Research 18 (2017) 1-39 Submitted 4/16; Revised 3/17; Published 7/17

Joint Label Inference in Networks

Deepayan Chakrabarti DEEPAY@UTEXAS.EDU
IROM, McCombs School of Business, University of Texas, Austin[*]

Stanislav Funiak STANO@CEREBRAS.NET
Cerebras Systems[*]

Jonathan Chang SLYCODER@GMAIL.COM
Health Coda[*]

Sofus A. Macskassy SOFMAC@BRANCH.IO
Branch Metrics[*]

Editor: Edo Airoldi

Abstract

We consider the problem of inferring node labels in a partially labeled graph where each node in the graph has multiple label *types* and each label type has a large number of possible labels. Our primary example, and the focus of this paper, is the joint inference of label types such as hometown, current city, and employers for people connected by a social network; by predicting these user profile fields, the network can provide a better experience to its users. Existing approaches such as Label Propagation (Zhu et al., 2003) fail to consider interactions between the label types. Our proposed method, called EdgeExplain, explicitly models these interactions, while still allowing scalable inference under a distributed message-passing architecture. On a large subset of the Facebook social network, collected in a previous study (Chakrabarti et al., 2014), EdgeExplain outperforms label propagation for several label types, with lifts of up to 120% for recall@1 and 60% for recall@3.

Keywords: label inference, graphs, social networks, variational methods, label propagation

1. Introduction

The most common classification setting assumes the presence of independent and identically distributed training examples, and attempts prediction on independent test instances drawn from the same distribution. However, the case of dependent examples has drawn much interest as well. This is particularly the case for networks where the nodes comprise the training/testing instances, and the edges between them imply dependence.

Consider the problem of inferring the labels of nodes in a network. For example, given a network of sports-related blogs connected by hyperlinks, we may want to know which blogs discuss baseball, or football, or hockey. The training data consists of known labels for a subset of blogs. The goal is to infer the labels of the remaining blogs. While the content of the blogs can be useful, the hyperlinks between blogs are clearly informative: we expect blogs to link primarily to other blogs discussing the same sport. This is a network label

∗. This research was done while the authors were employees at Facebook, Inc.

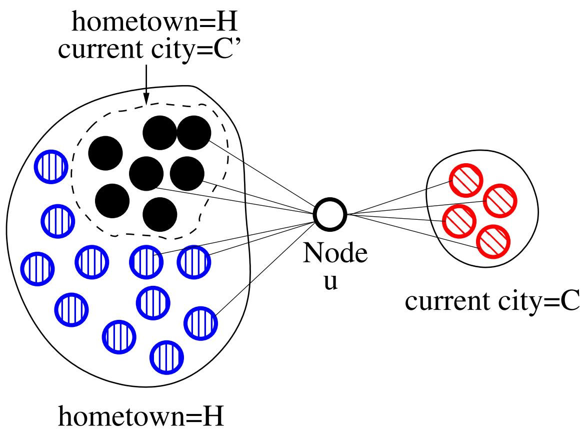


Figure 1: *An example graph of u and her friends:* The hometown friends of u coincidentally contain a subset with current city C'. This swamps the group from u's actual current city C, causing label propagation to infer C' for u. However, our proposed model (called EdgeExplain) correctly explains all friendships by setting the hometown to be H and current city to be C.

inference problem with only one label type (sport) and only a few labels (baseball, football, and hockey). The question we ask is: *how can we infer missing labels of multiple types, where each label type can be high-dimensional and the different types can be correlated?*

The concrete problem considered in this paper is that of inferring multiple fields such as the hometowns, current cities, and employers of users of a social network, where users often only partially fill in their profile, if at all. Here, each user is associated with one label of each *type* (such as hometown, employer, etc.), and the set of possible labels for each type is very high-dimensional. Inference of such label types is important for many ranking and relevance applications. By predicting the profile fields, the social network can make better friend recommendations or show more relevant content. Consequently, accurate predictions can greatly improve the user experience. We show that this inference problem cannot be split up into separate problems, one for each label type, without loss of accuracy. Knowledge of one label type influences inferences regarding the other types. Thus, this *joint inference* problem presents interesting opportunities for modeling and optimization.

A standard method of label inference is label propagation (Zhu and Ghahramani, 2002; Zhu et al., 2003), which tries to set the label probabilities of nodes so that friends have similar probabilities. This is based on the idea of homophily, that is, the more two nodes have in common, the more likely they are to connect (McPherson et al., 2001). However, label propagation assumes only a single category of relationships. It therefore fails to address the complexity of edge formation in networks, where nodes have different reasons to link to each other. As an example, consider the snapshot of a social network in Figure 1, where we want to predict the hometown and current city of node u, given what we know about u and u's neighbors. Here, the labels of node u are completely unknown, but her friends' labels are completely known. Label propagation would treat each label independently and infer the hometown of u to be the most common hometown among her friends, the current city to be the most common current city among friends, and so on. Hence, if the bulk of friends of u are from her hometown H, then inferences for current city will be dominated by the most common current city among her hometown friends (say, C') and not friends from her actual current city C; indeed, the same will happen for all other label types as well.

Our proposed method, named EdgeExplain, approaches the problem from a different viewpoint, using the following intuition: Two people form an edge in a social network because they share the same label for one or more label types (e.g., both went to the same college). Using this intuition, we can go beyond standard label propagation in the following way: instead of taking the graph as given, and modeling labels as items that propagate over this graph, we consider the labels as factors that can *explain* the observed graph structure. For example, the inferences for u made by label propagation leave u's edges from C completely unexplained. Our proposed method rectifies this, by trying to infer node labels such that for each edge $u \sim v$, we can explain the existence of the edge in terms of a shared label — u and v are friends from the same hometown, or college, or the like. While we are primarily interested in inferring labels, we note that the inferred reason for each edge can be useful by itself. For example, if a new node u joins a network and forms and edge with v, and we can infer that the reason is a shared college, we can recommend other college friends of v as possible new edges for u.

We note that a seemingly simple alternative solution—cluster the graph and then propagate the most common labels within a cluster—is in fact quite problematic. The clustering must also be complex enough to allow many overlapping clusters. This is an active area of research (Xie et al., 2013), but it still entails a significant computational burden. In addition, any clustering based solely on the graph structure ignores labels already available from user profiles, which clearly carry crucial information. A clustering method that tries to use these labels must deal with incomplete and missing labels, significantly increasing its complexity. Instead, our method will be able to operate only on dyads, without having to infer the higher-level clusters themselves. Social network cluster sizes must display wide variation (e.g., the cluster for users living in New York city versus those in small rural areas), and it is unclear if current network community detection algorithms offer good performance across the entire range of community sizes (Leskovec et al., 2010). Hence, we believe that clustering does not readily lend itself to a solution for our problem.

This paper tackles the following five questions.

How can "explaining links" be codified in a model? We propose a probabilistic model, called EdgeExplain, for the social network given the labels of all nodes in the network. The model has two key properties. First, the presence or absence of a link is conditionally independent of all other nodes and edges given the labels of the two endpoints of the link. This enables distributed computation for inference, which is useful for large networks. Second, labels corresponding to all the label types are jointly considered in the model.

How can inference be performed efficiently on large networks? The model likelihood function is non-convex, and only local optima can be found. However, we present several approaches for this problem. The first uses alternating maximization on a relaxation of the objective, such that each iteration solves a convex problem. The second is a variational approach that operates directly on the model objective. We show via simulations the presence of two regimes: while the alternating minimization works best when there is significant missing data, the variational approach works better in the presence of copious and correct information, i.e., when the labels of most neighbors of a node are known with relatively high certainty. This leads to our final hybrid method that combines the two approaches.

How do the model parameters affect accuracy vis-a-vis label propagation? We present an analysis of EdgeExplain that specifies the conditions under which label propagation will be particularly inaccurate. The quality of predictions of label propagation for a user u is shown to depend on three factors: (a) the degree to which any one label type is the preferred "reason" for forming friendships as compared to the other label types, (b) the degree to which any one label dominates among all labels of a particular type, and (c) the number of friends of a user u.

How well does it work in practice? Our iterative methods for inferring labels can be easily implemented in large-scale message-passing architectures. We empirically demonstrate their scalability on a large subset of the Facebook social network (Chakrabarti et al., 2014) and a 5 million node sample of the Google+ network (Gong et al., 2012), using publicly available user profiles and friendships. On Facebook, EdgeExplain significantly outperforms label propagation for several label types, with lifts of up to 120% for recall@1 and 60% for recall@3. On the Google+ sample, which exhibits extreme sparsity of labels, all algorithms are roughly equal. We also present empirical results on a movie network, where EdgeExplain outperforms label propagation as well as other competing methods (Zheleva and Getoor, 2009; Yin et al., 2010; Gong et al., 2014). The accuracy and scalability of EdgeExplain clearly demonstrate its usefulness for label inference on large networks.

How do variational methods compare with relaxation labeling? A subtheme of our paper is the comparison, via simulations as well as empirical evaluation on real-world data, of variational and relaxation labeling techniques. When both are designed to use only per-node parameters (which are well-suited for message-passing architectures), we find that relaxation labeling works better for our problem. This is in spite of the fact that our variational updates are exact, and only the standard approximation of the joint distribution with a product of per-node marginals is made.

The paper is organized as follows. We survey related work in Section 2, and then present our proposed EdgeExplain model in Section 3. For parameter inference, we present two methods: a gradient ascent approach on a relaxed version of the problem (Section 4) and a variational approach for optimizing the original problem objective (Section 5). We then present a detailed analysis of our model in Section 6, focusing on the conditions under which label propagation would fail. Simulations on synthetic data sets (Section 7) confirm our analysis, and also help us identify the conditions under which the two inference mechanisms work best. Empirical evidence demonstrating the effectiveness of EdgeExplain is presented in Section 8. Generalizations of the model are discussed in Section 9. We conclude in Section 10.

2. Related Work

The problem of inferring labels of nodes in a network has been studied extensively in the literature on graph-based semi-supervised learning, statistical relational learning, and latent models for networks. In this section, we highlight the prior work in these fields, and then review recent works that specifically focus on the problem of attribute inference in social networks.

Semi-supervised learning. Our work falls under the paradigm of graph-based semi-supervised learning (Zhu, 2008), where the goal is to estimate a function over the nodes of the graph that satisfies two properties: (1) the function is close (or equal) to the observed labels, and (2) the function is smooth (similar) at adjacent nodes. In their seminal paper, Zhu and Ghahramani (2002) proposed an algorithm called Label Propagation, where the values at the nodes are relaxed to be real-valued, and inference is performed by averaging the values at neighboring nodes. Their follow-up paper (Zhu et al., 2003) formulated Label Propagation as inference in Gaussian random fields, stating additional interpretations as random walks on the graph and electric networks. The Gaussian random field formulation can be viewed as a special case of the regularization framework proposed by Zhou et al. (2003) that allows a tradeoff between edge smoothness and label fitting. This framework was analyzed by Belkin et al. (2004), proving generalization bounds and stability properties. Belkin et al. (2006) placed the problem in the context of reproducing kernel Hilbert spaces to provide further theoretical basis for the algorithms and obtain an out-of-sample extension. Consequently, they can handle both the inductive and transductive settings. Finally, Baluja et al. (2008) and Talukdar and Crammer (2009) introduced abandonment probability into the random walk, which can be viewed as having a special "dummy" label. This formulation addresses a challenge present in some graphs, where the random walk becomes uninformative once it visits a high-degree node. None of these approaches consider interactions between multiple label types, and hence fail to capture the edge formation process in graphs considered here. Although we do assume that observed labels are fixed, our approach could be generalized to handle soft constraints on node labels. Finally, while we do not address the inductive setting, the graphs we consider tend to change slowly, hence we can leverage the standard approach of initializing the iteration with the results of the previous run.

Often, the algorithms for graph-based semi-supervised learning can be expressed as an update rule that scales linearly with the number of nodes and edges of the graph. However, the update and space complexity are also linear in the number of distinct label values; this number can be very large in some applications (e.g., Deng et al., 2009; Shi et al., 2009). To address this challenge, Talukdar and Cohen (2014) use count-min sketch, a randomized data structure that allows them to reduce the time and space complexity to be logarithmic in the number of distinct labels. We face a similar challenge in our work, because the number of distinct values in each field is very large. However, instead of using count-min sketches, we have found it effective to compute a sparse projection of the distribution onto the probability simplex in each update (Kyrillidis et al., 2013).

Statistical relational learning. Our work is also related to the field of statistical relational learning. Here, the goal is to classify an entity such as a web page or a patent into a fixed number of labels (classes). Each entity is typically associated with attributes such as text, and we observe the links between entities. Chakrabarti et al. (1998) noted that it is ineffective to build a classifier based on the attributes of the node and its neighbors directly. Instead, they proposed a naive Bayes classifier that considers only the neighbors' labels (in addition to the local attributes). In order to propagate information through the network, they use an approximation of belief propagation. Lu and Getoor (2003) extended this work with a structured logistic regression model built separately on the local content and the links, using maximum a posteriori probability (MAP) for inference. Macskassy and Provost (2007) show that these approaches can be placed in a single framework, with different choices of (a) a local classifier that uses a node's attributes alone, (b) a relational classifier that uses the labels at adjacent nodes, and (c) a collective inference procedure that propagates the information through the network. They provide an extensive empirical evaluation of the various combinations of these choices and observe that the best combination, weighted-vote relational neighbor classifier (Macskassy and Provost, 2003) with relaxation labeling (Rosenfeld et al., 1976; Hummel and Zucker, 1983), tends to perform as well as label propagation, which we outperform. We note that these algorithms typically focus on a single label type, whereas we explicitly model the interactions among multiple types. Although there has been some work on understanding how to combine and weigh different edge types for best prediction performance (Macskassy, 2007), the edge types (analogous to our reason for an edge) were given up front; we recover them automatically.

There is also extensive work on probabilistic relational models. Relational Bayesian Networks (RBNs; Koller and Pfeffer, 1998; Friedman et al., 1999) provide representation of organizational structure while maintaining a coherent probabilistic representation of uncertainty. Unfortunately, due to their directed nature, RBNs cannot efficiently represent homophily. Relational Markov Networks (Taskar et al., 2002) and Relational Dependency Networks (Neville and Jensen, 2007) do not impose acyclicity, and could be conceptually applied to the problem considered in this paper. Yet, we cannot use these general formalisms directly, because learning the models with high-cardinality labels would require large amounts of labeled data. Instead, it is our explicit modeling assumptions regarding multiple label types that yield gains in accuracy.

Latent models. Graph structure has been modeled using latent classes (Nowicki and Snijders, 2001; Kemp et al., 2006; Xu et al., 2006; Airoldi et al., 2008) and latent variables (Hoff et al., 2002; Miller et al., 2009; Palla et al., 2012), with an emphasis on link prediction. Nowicki and Snijders (2001) describe a stochastic blockmodel (SBM), where each node belongs to one of a fixed number of clusters, and edge generation is governed entirely by the clusters of the adjacent nodes. Kemp et al. (2006) and Xu et al. (2006) extend this work to an unknown number of clusters using the Chinese restaurant process (Pitman, 2006), while Airoldi et al. (2008) propose a mixed membership stochastic blockmodel (MMSB), where the node class membership varies from one dyad to another. The use of latent variables was initially explored in (Hoff et al., 2002), who consider generative process of edges based on the embedding of the adjacent nodes. The Latent Feature Infinite Relational Model (Miller et al., 2009) uses an infinite number of binary features controlled by an Infinite buffet process (Griffiths and Ghahramani, 2005); the Infinite Latent Attribute Model (Palla et al., 2012) then allows each feature to be partitioned into disjoint subgroups. Similarly to these approaches, in our work, each node is associated with (a finite number of) attributes. However, unlike the literature on latent models, where the latent features can be arbitrary combinations of user attributes and are unlikely to represent the concrete label types we wish to predict, our goal is to make predictions about the label types directly, using the observations made at a subset of the nodes. Several of these models may also not easily scale to large networks.

There is also a growing body of work on simultaneously explaining the connections between documents as well as their word distributions (Nallapati et al., 2008; Chang and Blei, 2010; Ho et al., 2012). Nallapati et al. (2008) propose two models that combine Latent Dirichlet allocation (LDA; Blei et al., 2003) and Mixed membership stochastic blockmodel

(MMSB; Airoldi et al., 2008), where topic proportions generate both word topics and the cluster assignments for linking. Chang and Blei (2010) also combine LDA and linking, but use the entire vector of word topic assignments for linking, thus allowing their model to better infer words from links and vice versa. Finally, Ho et al. (2012) use a hierarchy of topics specified as a nested Chinese restaurant process (Blei et al., 2010), where the link between two nodes is given by the deepest level they share. Although we do not consider the problem of modeling text data, our model permits us to incorporate node attributes, and we show empirical results for an instance where group memberships are used as additional node features. The number of distinct label values in our application is very large (on the order of millions), and we suspect that the latent variables would have to have a large dimension to explain the edges in our graph well.

Attribute inference in social networks. Several of the aforementioned techniques have recently been applied specifically to the problem of inferring node attributes in social networks. Zheleva and Getoor (2009) study the privacy implications of methods that can predict a user's private profile from the public profiles of others in the social network. They compare several algorithms for profile inference, and find that two methods work well when no extra side-information is available. One is collective classification based on the labels of neighboring nodes, where their specific implementation is essentially identical to Label Propagation. The second method (called LINK) predicts a node's labels based on a feature vector consisting of the node-IDs of its neighboring nodes. Thus, a graph with N nodes has N features, which is clearly difficult to scale.

Dong et al. (2014) consider inference of gender (2 categories) and age (4 categories) using a social network. They consider three sets of factors: (a) connections between the attribute values of a node and the node's network characteristics (such as its degree), (b) connections between attributes of node pairs connected by a link, and (c) factors for attributes of social triads. This is a comprehensive model, and is one of the few that considers triadic factors. However, the model requires several parameters for each possible value of the (gender, age-category) attribute vector. This is feasible only when the set of attribute values is extremely restricted (only 8 possibilities in their setting). Our primary problem setting has five attributes, with millions of possible values for each attribute; scaling the method to this setting appears to be extremely difficult.

Yin et al. (2010) propose attribute inference on a combined "Social Attribute Network (SAN)," where both people and attribute values are represented by nodes, and links connect people to their friends as well as to their attribute values (whenever these are known). For nodes with no attribute links, they predict the most likely attributes using a random walk on the SAN. Gong et al. (2014) extend this idea by adapting a variety of traditional link-prediction algorithms to the SAN. They find that two methods work well for attribute prediction on the Google+ SAN. The first performs random walks with restarts (RWR-SAN). The second method uses the number of "common neighbors" (CN-SAN) as a predictor. Specifically, the affinity of user u towards attribute a is measured by the number of neighbors of u who are connected to a in the SAN, and each unlabeled user picks the attribute for which she has the highest affinity. This process is iterated until convergence. Note that this method can be interpreted as a variant of Label Propagation where each unlabeled node is assigned the most common attributes among her friends (in contrast to vanilla Label Propagation where nodes have distributions over attributes).

Thus, recent works confirm the findings of earlier comparative studies (Macskassy and Provost, 2007) that Label Propagation (LP) achieves the best prediction accuracy. We shall show that our method (EdgeExplain) is as accurate as LP on the Google+ data set; LP is equivalent to CN-SAN, which was one of the two best methods for this data set (Gong et al., 2014). However, the Google+ data set suffers from extreme sparsity. On the Facebook data set, EdgeExplain achieves significantly higher accuracy as compared to LP. We also compare EdgeExplain against LP and all the feasible methods mentioned above (LINK, CN-SAN, and RWR-SAN) on a SAN derived from IMDB movie data, and show that EdgeExplain outperforms all of them.

3. Proposed Model

In this section we first build intuition about our model using a running example. Suppose we want to infer the *labels* (e.g., "Palo Alto High School" and "Stanford University") corresponding to several *label types* (e.g., high school and college) for a large collection of users. The available data consist of labels publicly declared by some users, and the (public) social network among users, as defined by their friendship network. While the desired set of label types may depend on the application, here we focus on five label types: hometown, high school, college, current city, and employer.

Our solution exploits three properties of these label types:

(P1) They represent the primary situations where two people can meet and become friends, for example, because they went to the same high school or college.

(P2) These situations are (mostly) mutually exclusive. While there may be occasional friendships sharing, say, hometown and high-school, we make the simplifying assumption that most edges can be explained by only one label type.

(P3) Sharing the same label is a *necessary* but not *sufficient* condition. For example, "We are friends from Chicago" typically implies that the indicated individuals were, at some point in time, co-located in a small area within Chicago (say, lived in the same building, met in the same cafe), but hardly implies that two randomly chosen individuals from Chicago are likely to be friends.

(P1) is a direct result of our application; our desired label types were targeted at friendship formation. Combined with **(P2)**, our five label types can be considered a set of mutually exclusive and exhaustive "reasons" for friendship; while this is not strictly true for high school and hometown, empirical evidence suggests that it is a good approximation (shown later in Section 8) and we defer a discussion on this point to Section 9. However, as **(P3)** shows, we cannot simply cast the labels as features whose mere presence or absence significantly affects the probability of friendship; instead, a more careful analysis is needed.

Formally, we are given a graph, $\mathcal{G} = (V, E)$ and a set of label types $\mathcal{T} = \{t_1, \ldots, t_{|\mathcal{T}|}\}$. For each label type t, let $L(t)$ denote the (high-dimensional) set of labels for that label type. Each node in the graph is associated with binary variables $S_{ut\ell}$, where $S_{ut\ell} = 1$ if node $u \in V$ has label ℓ for label type t. Let $\boldsymbol{S}_V$ and $\boldsymbol{S}_H$ represent the sets of visible and hidden variables, respectively. We want to infer the correct values of $\boldsymbol{S}_H$, given $\boldsymbol{S}_V$ and $\mathcal{G}$.

A popular method for label inference is *label propagation* (Zhu and Ghahramani, 2002; Zhu et al., 2003). For a single label type, this approach represents the labeling by a set of indicator variables $S_{u\ell}$, where $S_{u\ell} = 1$ if node u is labeled as ℓ and 0 otherwise. Zhu et al. (2003) relax the labeling to real-valued variables $f_{u\ell}$ over all nodes u and labels ℓ that are clamped to one (or zero) for nodes known to possess that label (or not). They then define a quadratic energy function that assigns lower energy states to configurations where f at adjacent nodes are similar:

$$E(\boldsymbol{f}) = \frac{1}{2} \sum_{u \sim v} w_{uv} \sum_{\ell} (f_{u\ell} - f_{v\ell})^2. \tag{1}$$

Here, $u \sim v$ means that u and v are linked by an edge, and w_{uv} is a non-negative weight on the edge $u \sim v$. The minimum of Eq. 1 is found by solving the fixed point equations

$$f_{u\ell} = \frac{1}{d_u} \sum_{u \sim v} w_{uv} f_{v\ell},$$

where $d_u = \sum_{u \sim v} w_{uv}$. This procedure encourages $f_{u\ell}$ of nodes connected to clamped nodes to be close to the clamped value and propagates the labels outwards to the rest of the graph. Multiple label types can be handled similarly by minimizing Eq. 1 independently for each type.

While this formulation makes full use of **(P1)** and has the advantage of simplicity, it completely ignores **(P2)**. Intuitively, label propagation assumes that friends tend to be similar in *all* respects (i.e., all label types), whereas what **(P2)** suggests is that each friendship tends to have a single *reason*: an edge $u \sim v$ exists because u and v share the same high school *or* college *or* current city, etc. This highly non-linear function is not easily expressed as a quadratic or similar variant.

Instead, we propose a different probabilistic model, which we call EdgeExplain. As described above, let $\mathcal{G}$ denote the graph, and $\boldsymbol{S}_V$ and $\boldsymbol{S}_H$ represent the sets of visible and hidden variables respectively; the variable $S_{ut\ell}$ is known (visible) if user u has publicly declared the label ℓ for type t, and unknown (hidden) otherwise. We define EdgeExplain as follows:

$$P(\boldsymbol{S}_V, \boldsymbol{S}_H \mid \mathcal{G}) = \frac{1}{Z} \prod_{u \sim v} \operatorname*{softmax}_{t \in \mathcal{T}}(r(u,v,t)) \tag{2}$$

$$r(u,v,t) = \sum_{\ell \in L(t)} S_{ut\ell} S_{vt\ell} \tag{3}$$

$$\operatorname*{softmax}_{t \in \mathcal{T}}(r(u,v,t)) = \sigma\left(\alpha \sum_{t \in \mathcal{T}} r(u,v,t) + c\right), \tag{4}$$

where Z is a normalization constant. Here, $r(u,v,t)$ indicates whether a shared label type t is the *reason* underlying the edge $u \sim v$ (Eq. 3). The $\text{softmax}(r_1, \ldots, r_{|\mathcal{T}|})$ function should have three properties: (a) it should be monotonically non-decreasing in each argument, (b) it should achieve a value close to its maximum as long as any one of its parameters is "high", and also (c) it should be differentiable, for ease of analysis. In Eq. 4, we use the sigmoid function to implement this: $\sigma(x) = 1/(1+e^{-x})$. This monotonically increases from

0 to 1, and achieves values greater than $1-\epsilon$ once x is greater than an ϵ-dependent threshold. In addition, the sigmoid enables fine control of the degree of "explanation" required for each edge (discussed below) and allows for easy extensions to more complex label types and extra features (Section 9), all of which make it our preferred choice for the softmax.

In a nutshell, our modeling assumption can be stated as follows: *It is better to explain as many friendships as possible, rather than to explain a few friendships really well.* Eq. 2 is maximized if the softmax function achieves a high value for each edge $u \sim v$, i.e., if each edge is "explained." This is achieved if the sum $\sum_{t\in\mathcal{T}} r(u, v, t)$ is relatively high, which in turn is satisfied if the product $S_{ut\ell}S_{vt\ell}$ is 1 for even one label ℓ — in other words, when there exists any label ℓ that both u and v share. The parameter α controls the degree of explanation needed for each edge; a small α forces the learning algorithm to be very sure that u and v share one or more label types, while with a large α, a single matching label type is enough. Empirical results shown later in Section 8 prove that large α values perform better (we use $\alpha = 10$ in our evaluation), suggesting that even a single matching label type is enough to explain the edge. The parameter c in Eq. 4 can be thought of as the probability of matching on an unknown label type, distinct from the five we consider. Higher values of c can be used to model uncertainty that the available label types form an *exhaustive* set of reasons for friendships. For our running example in the social network setting, we set $c = 0$ to reflect our belief that the five label types we consider represent the primary reasons for friendship formation (property **(P1)**).

Further intuition can be gained by considering a node u whose labels are completely unknown, but whose friends' labels are completely known (see Figure 1). As we discussed earlier in Section 1, label propagation would infer the hometown of u to be the most common hometown among her friends (i.e., H), the current city to be the most common current city among friends (i.e., C'), and so on. However, such an inference leaves u's friendships from C completely unexplained. Our proposed method rectifies this; Eq. 2 will be maximized by correctly inferring H and C as u's hometown and current city respectively, since H is enough to explain all friendships with the hometown friends, and the marginal extra benefit obtained from explaining these same friendships a little better by using C' as u's current city is outweighed by the significant benefits obtained from explaining all the friendships from C by setting u's current city to be C.

To summarize, Eq. 3 encapsulates property **(P1)** by trying to have matching labels between friends; Eq. 4 models property **(P2)** by enabling any one label type to explain each friendship; and the form of the probability distribution (Eq. 2) uses only existing edges $u \sim v$ and not all node pairs, and thus is not affected when, say, two nodes with Chicago as their current city are not friends, which reflects the idea that matching label types are necessary but not sufficient **(P3)**.

4. Inference via relaxation labeling

We present two methods for inference under EDGEEXPLAIN. The first considers a relaxation of the model that yields an objective which can be optimized via projected gradient descent, and this is discussed in the current section. The second method is optimizes a variational lower bound on the likelihood of EDGEEXPLAIN, and this is discussed later in Section 5. A priori, there is little reason to choose one or the other, and we discuss their relative merits via simulations and evaluation on real data (Sections 7 and 8).

The probabilistic description of EdgeExplain in Eqs. 2-4 can be restated as an optimization problem in the variables $S_{ut\ell} \in \{0,1\}$. In the spirit of (Zhu et al., 2003), we propose a relaxation in terms of a real-valued function $\boldsymbol{f}$, with $f_{ut\ell} \in [0,1]$ representing the probability that $S_{ut\ell} = 1$, i.e., the probability that user u has label ℓ for label type t. This yields the following optimization:

$$\underset{\boldsymbol{f}}{\text{Maximize}} \sum_{u \sim v} \log\Big(\underset{t \in \mathcal{T}}{\text{softmax}}(r(u,v,t))\Big) \tag{5}$$

$$\text{where } r(u,v,t) = \sum_{\ell \in L(t)} f_{ut\ell} f_{vt\ell} \tag{6}$$

$$\sum_{\ell \in L(t)} f_{ut\ell} = 1 \quad \forall t \in \mathcal{T} \tag{7}$$

$$f_{ut\ell} \geq 0 \tag{8}$$

where softmax$(\cdot)$ is defined as in Eq. 4, and the equation for $r(\cdot)$ is analogous to Eq. 3 but measures the total probability that u and v have the same label for a given label type t.

The problem is not convex in $\boldsymbol{f}$, but is convex in $\boldsymbol{f}_u = \{f_{ut\ell} | t \in \mathcal{T}, \ell \in L(t)\}$ if the distributions $\boldsymbol{f}_v$ are held fixed for all nodes $v \neq u$. Hence, we propose an iterative algorithm to infer $\boldsymbol{f}$. Given $\boldsymbol{f}_v$ for all $v \neq u$, finding the optimal $\boldsymbol{f}_u$ corresponds to solving the following problem:

$$\underset{\boldsymbol{f}_u}{\text{Maximize}} \;\; g(\boldsymbol{f}_u) = \sum_{v \in \Gamma(u)} \log\Big(\underset{t \in \mathcal{T}}{\text{softmax}}(r(u,v,t))\Big),$$

where the summation is only over the set $\Gamma(u)$ of the friends of u, and we again restrict $\boldsymbol{f}_u$ to be a set of $|\mathcal{T}|$ probability distributions, one for each label type. We note that $g(\cdot)$ is convex and Lipschitz continuous with constant $L = \alpha \cdot |\Gamma(u)|$, where $|\Gamma(u)|$ is the number of friends of u.

This is a constrained maximization problem with no closed form solution for $\boldsymbol{f}_u$. To solve it, we use projected gradient ascent. This is an iterative method where in each iteration, we take a step in the direction of the gradient, and then project it back to the probability simplex $\Delta = \Big\{f_{ut\ell} \mid f_{ut\ell} \geq 0, \sum_{\ell \in L(t)} f_{ut\ell} = 1 \, \forall t \in \mathcal{T}\Big\}$. Specifically, let ∇g represent the gradient of g, with components given by:

$$\frac{\partial g(\boldsymbol{f}_u)}{\partial f_{ut\ell}} = \sum_{v \in \Gamma(u)} \alpha f_{vt\ell} \cdot \sigma\Big(-\alpha \sum_{t \in \mathcal{T}} \sum_{\ell \in L(t)} f_{ut\ell} f_{vt\ell} - c\Big).$$

Let $\boldsymbol{f}_u^{(k-1)} = \big\{f_{ut\ell}^{(k-1)} | t \in \mathcal{T}, \ell \in L(t)\big\}$ be the estimated probability distributions for each of the $\mathcal{T}$ label types at the end of iteration $k-1$, and let $q_{ut\ell}^{(k)}$ represent the (possibly improper) ending point of the k-th gradient step:

$$\boldsymbol{q}_u^{(k)} = \boldsymbol{f}_u^{(k-1)} + c_k \nabla g,$$

where c_k is a step-size parameter that we could set to a constant $c_k = 1/L$. The point $\boldsymbol{q}_u^{(k)}$ is now projected to the closest point in Δ:

$$\boldsymbol{f}_u^{(k)} = \arg\min_{\boldsymbol{q}' \in \Delta} \|\boldsymbol{q}_u^{(k)} - \boldsymbol{q}'\|_2.$$

This can be easily achieved in expected linear time over the size of the label set $\sum_t L(t)$ (Duchi et al., 2008). If only sparse distributions can be stored for each label type (say, only the top k labels for each type), the optimal k-sparse projections can be obtained simply by setting to 0 all but the top k labels for each label type, and then projecting on to the simplex (Kyrillidis et al., 2013).

This algorithm converges to a fixed point, and the function values converge to the optimal at a $1/k$ rate (Beck and Teboulle, 2009):

$$g^* - g^{(k)} \leq \frac{L\|\boldsymbol{f}_u^{(0)} - \boldsymbol{f}_u^*\|^2}{2k} \leq \frac{L|\mathcal{T}|}{k},$$

where $\boldsymbol{f}_u^*$ represents the optimal set of probability distributions, and g^* is the optimal function value. An important consequence of the algorithm is that computation of $\boldsymbol{f}_u$ only requires information from $\boldsymbol{f}_v$ for the neighbors v of u. Thus, it is a "local" algorithm that can be easily implemented in distributed message-passing architectures, such as Giraph (Giraph; Ching, 2013).

5. Variational inference

The relaxation labeling approach discussed above is intuitive, but its objective function is neither a bound nor a formal approximation of the likelihood of EdgeExplain. We now present a variational inference procedure where the inferred label probabilities do maximize a lower bound of the likelihood.

From Eqs. 2-4, we have:

$$\begin{aligned} P(S_V, S_H \mid \mathcal{G}) &= \frac{1}{Z} \prod_{u \sim v} \sigma\Big(\alpha \sum_{t \in \mathcal{T}} \sum_{\ell \in L(t)} S_{ut\ell} S_{vt\ell} + c\Big) \\ &= \frac{1}{Z} \prod_{u \sim v} \frac{1}{1 + \exp(-\alpha \sum_{t \in \mathcal{T}} \sum_{\ell \in L(t)} S_{ut\ell} S_{vt\ell} - c)}. \end{aligned}$$

Given a fixed assignment to the visible variables S_V and any distribution $Q(S_H)$ over the hidden variables, we have the inequality:

$$\ln P(S_V \mid \mathcal{G}) \geq -\sum_{S_H} Q(S_H) \ln \frac{Q(S_H)}{P(S_H, S_V \mid \mathcal{G})}. \tag{9}$$

We shall choose a fully factorized distribution for $Q(S_H)$:

$$Q(S_H) = \prod_u \prod_{t \in \mathcal{T}} \prod_{\ell \in L(t)} \mu_{ut\ell}^{S_{ut\ell}},$$

where $\mu_{ut\cdot}$ represents a multinomial distribution over all labels $\ell \in L(t)$ for label type t for user u; for notational convenience, we set $\mu_{ut\ell}$ to 0 or 1 if the user's labels are known. Such factored distributions are common in variational inference. They also have the same number of parameters as the relaxation labeling approach, allowing a fair comparison between the two. With this choice of distribution $Q(\cdot)$, the variational lower bound (Eq. 9) can be written as

$$\begin{aligned}\ln P(S_V \mid \mathcal{G}) \geq &- \sum_{ut\ell} \mu_{ut\ell} \ln \mu_{ut\ell} \\ &- \sum_{u \sim v} \left\langle \ln\left(1 + \exp\left(-\alpha \sum_{t\ell} S_{ut\ell} S_{vt\ell} - c\right)\right)\right\rangle_{Q(S_H)} \\ &- \ln Z,\end{aligned} \tag{10}$$

where $\langle \cdot \rangle_Q$ represents expectation with respect to $Q(\cdot)$, and ℓ is summed over $L(t)$. We want to set the variational parameters $\boldsymbol{\mu} = \{\mu_{ut\ell}\ \forall u, t, \ell \in L(t)\}$ to maximize this lower bound.

Define $\eta_{uvt} = \sum_{\ell \in L(t)} \mu_{ut\ell}\mu_{vt\ell}$. Let $\mathbf{w} \in \{0,1\}^{|\mathcal{T}|}$ represent a binary vector of length $|\mathcal{T}|$, with w_t being the t^{th} component and $|\mathbf{w}|$ the number of "ones".

Theorem 1 *Consider one node u and one label type t. Given the parameters $\boldsymbol{\mu} \setminus \{\mu_{ut\cdot}\}$, the distribution $\mu^*_{ut\ell}$ that maximizes the lower bound of Eq. 10 is given by:*

$$\begin{aligned}\mu^*_{ut\ell} &\propto \exp\left\{\sum_{\{v \mid u \sim v\}} \mu_{vt\ell} \sum_{\mathbf{w}} \phi_{uvt}(\mathbf{w})\right\} \\ \phi_{uvt}(\mathbf{w}) &= \ln\left(1 + e^{-(\alpha|\mathbf{w}|+c)}\right)(-1)^{w_t} \prod_{t' \neq t} \kappa(w_{t'}, \eta_{uvt'}) \\ \kappa(w_{t'}, \eta_{uvt'}) &= \eta_{uvt'}^{w_{t'}} (1 - \eta_{uvt'})^{1-w_{t'}}.\end{aligned}$$

Proposition 1 $\sum_{\mathbf{w}} \phi_{uvt}(\mathbf{w}) \geq 0$.

Both proofs are deferred to Appendix A.

It follows that the probability of node u having label ℓ for type t under the variational approximation is given by a (normalized, exponentiated) weighted linear combination of the neighboring node labels. For intuition, consider the case of large α. Then, the sum $\sum_{\mathbf{w}} \phi_{uvt}(\mathbf{w})$ is dominated by the case of $\mathbf{w}^{(-t)} = \mathbf{0}$, the all-zero vector:

$$\sum_{\mathbf{w}} \phi_{uvt}(\mathbf{w}) \approx \ln\left(\frac{1 + e^{-c}}{1 + e^{-\alpha - c}}\right) \prod_{t' \neq t} (1 - \eta_{uvt'}).$$

Thus, the highest-probability label of type t for u is the most the common label among friends of u, *unless* the edge $u \sim v$ is already explained by some other label type t', i.e., if $\eta_{uvt'} \approx 1$. Hence, the probabilities $\mu^*_{ut\ell}$ attempt to explain as many friendship edges as possible, which is precisely the expected behavior for large α. Our variational inference method iteratively updates the parameter vector $\boldsymbol{\mu}$ by repeatedly applying Thm. 1.

6. Analysis

We now present a theoretical analysis of EdgeExplain, and its relationship with Label Propagation (LP). Even if node labels for different label types are dependent (as in EdgeExplain) rather than independent (as in LP), the labels inferred by LP could still be correct. We will find the conditions under which LP-based inference yields incorrect results, and hence inference tailored to EdgeExplain can outperform LP.

Simplified model. We will make several simplifications to EdgeExplain to focus on its main aspects. First, we will consider the "hard-threshold" variant of the problem, where even one shared attribute between two friends is enough to explain their friendship; this corresponds to setting $\alpha \to \infty$ in Eq. 4. Such a choice is also justified empirically, as will be shown in Section 8. Second, we will consider the case of one node (the "ego") whose attributes are all unknown, surrounded by N friends whose attributes are all known. This can be thought of as the fundamental inference problem template, with the actual problem using a softer version of this template that associates probabilities with the labels of friends. Finally, Eqs. 2–4 only specify a discriminative model, but we will need a generative model for our analysis. Hence, we devise a simple probabilistic model for generating node labels (represented by the binary variables $\boldsymbol{S}$) that is consistent with properties **(P1)**-**(P3)** outlined in Section 3.

Specifically, consider an "ego network" $\mathcal{G}_u$ consisting of a central node u surrounded by N friends $v_1, \ldots, v_N$. Let $Y_u = \{Y_u(t_1), \ldots, Y_u(t_{|\mathcal{T}|})\}$ denote the labels of u for each of the $|\mathcal{T}|$ label types; these are precisely the labels that are turned "on" in the binary indicator variables $\boldsymbol{S}$ restricted to u. Similarly, let Y_i represent the vector of labels for node v_i. Let $\pi(Y_u, Y_{v_1}, \ldots, Y_{v_N})$ denote the probability of observing these labels. Since $\mathcal{G}_u$ is a tree rooted at u, the labels of the friends are conditionally independent given the labels of the ego: $\pi(Y_u, Y_{v_1}, \ldots, Y_{v_N}) = \pi(Y_u) \cdot \prod \pi(Y_i \mid Y_u)$. Thus, we can generate node labels in the following manner. First, the ego u selects her labels first according to the prior $\pi(Y_u)$. Then, conditioned on Y_u, the labels Y_i for friends v_i are drawn independently of each other. In order to satisfy **(P1)**, this second step must itself be split into two stages. In the first stage, each friend v_i selects a "reason" for her friendship with u by selecting the label type Z_i that v_i shares with u. This shared label type is drawn according to a multinomial distribution q: $q(Z_i = t) = q_t$. Thus, $Y_i(Z_i) = Y_u(Z_i)$ with probability 1. Then, in the second stage, the remaining labels of v_i are set via the distribution $\pi(Y_i \mid Y_i(Z_i))$. Under this setting of Y_u and Y_i, there is a shared label for each edge, and the model π is the most general one that satisfies **(P1)** and the independence assumptions encoded in $\mathcal{G}_u$, and is symmetric across all friends.

This leads to the following simplified problem statement:

Given the network $\mathcal{G}_u$ and the labels Y_i of all friends v_i, predict the labels Y_u of the ego.

Relation to set-cover. At first sight, this appears to be a variant of the well-known set-cover problem where, for each node, we must pick sets of friends sharing some attribute such that the union of these sets covers all the friendships. However, note that we can pick only one set for each label type, since the ego can have only one label for each type Thus, the computational complexity is $O\left(\prod_{t \in \mathcal{T}} |\mathcal{L}_t|\right)$, and since we consider only a limited number of label types ($|\mathcal{T}| = 5$), the complexity is polynomial. The set-cover problem, on the other hand, can consider any number of sets, and is an NP-hard problem.

Failure conditions for Label Propagation. The correct labels Y_u for the ego offer (one of) the best solutions for EdgeExplain, since each friendship is explained by at least one shared label. LP fails if the ego has a label ℓ of type t (i.e., $Y_u(t) = \ell$), but a different label $\ell' \neq \ell$ of the same type t is shared by more friends; we will call this the event *"LP fails via (t, ℓ, ℓ')."* Thus, $P(\text{LP fails}) \geq \max_{t,\ell,\ell'} P(\text{LP fails via } (t, \ell, \ell'))$, and we will develop lower bounds for the latter quantity.

Let us denote by $p_{t,t'}(\ell)$ the probability of friend v_i having the correct label ℓ for label type t given that it shares some other label type t' with the ego u; as discussed earlier, this probability is assumed to be the same for all friends:

$$p_{t,t'}(\ell) \triangleq \pi(Y_i(t) = \ell \mid Y_u \text{ and } \{Z_i = t' \neq t\}).$$

Analogously, let $p_{t,t'}(\ell')$ denote the probability of friend v_i having an incorrect label ℓ'. Furthermore, let $\Delta_{t,\ell,\ell'}$ denote the expected difference of $p_{t,t'}(\ell')$ and $p_{t,t'}(\ell)$ w.r.t. q:

$$\Delta_{t,\ell,\ell'} \triangleq \sum_{t' \neq t} q_{t'} \left(p_{t,t'}(\ell') - p_{t,t'}(\ell) \right) - q_t. \tag{11}$$

Theorem 2 *If $\Delta_{t,\ell,\ell'} > 0$, then for any small ϵ such that $0 < \epsilon < \Delta_{t,\ell,\ell'}$, we have:*

$$P(\textit{LP fails via } (t, \ell, \ell')) \geq \sum_{\{Y_u \mid Y_u(t) = \ell\}} \pi(Y_u) \cdot \left(1 - \exp\{-0.5N(\Delta_{t,\ell,\ell'} - \epsilon)^2\}\right).$$

The proof is deferred to Appendix A.

Thus, LP is likely to fail when $\Delta_{t,\ell,\ell'}$ is large. This happens when the following two conditions hold: (a) label ℓ is somewhat less likely than ℓ' in the entire population (so that $p_{t,t'}(\ell') - p_{t,t'}(\ell) > 0$), and (b) friendships based on a shared label for label type t are rare (i.e., q_t is small and consequently $q_{t'}$ can be large). Intuitively, under these conditions, the ego can have label ℓ for a rarely shared label type t (say, employer), but enough friends can have a different label ℓ' simply due to its prevalence in the population. It becomes likely that the number of friends with label ℓ' is more than those with ℓ, leading to an erroneous inference by LP. This is precisely the situation illustrated in Figure 1 earlier.

Optimum conditions for EdgeExplain. We shall now derive the conditions which imply the greatest possibility of error by LP. Consider a system with only two label types t and t' (call this TwoLabels). Thus, $q_{t'} = 1 - q_t$. Since LP's probability of failure is monotonically increasing in $p_{t,t'}(\ell')$ (via Eqs. 11 and Thm. 2), the worst case for LP is when all the mass is concentrated in a single alternate label ℓ'. Then without any loss of generality, we can let label type t take only two values ℓ and ℓ', with $p_{t,t'}(\ell') = 1 - p_{t,t'}(\ell)$. We will assume that the labels for the ego and her friends follow the same marginal distribution, so

$$\pi(Y_u(t) = k \mid Y_u(t')) = \pi(Y_i(t) = k \mid Y_u(t') \text{ and } \{Z_i = t'\}) \triangleq p_{t,t'}(k) \qquad \text{for } k \in \{\ell, \ell'\}.$$

Theorem 3 *The lower bound in Theorem 2 under* TwoLabels *is maximized for*

$$\Delta_{t,\ell,\ell'} = O\left(\sqrt{\frac{\log N}{N}}\right), \qquad p_{t,t'}(\ell) = \frac{1-2q_t-\epsilon}{2(1-q_t)} - O\left(\sqrt{\frac{\log N}{N}}\right), \qquad q_t < 0.5.$$

Corollary 4 *For a given N, the optimum conditions for* EdgeExplain *vis-a-vis LP under* TwoLabels *are obtained for $p_{t,t'}(\ell) + q_t - p_{t,t'}(\ell) \cdot q_t \approx const.$*

Both proofs are deferred to Appendix A.

Theorem 3 demonstrates the link between the probability $p_{t,t'}(\ell)$ of a person having label ℓ and the probability q_t of forming a friendship based on a shared label of type t. If $p_{t,t'}(\ell)$ is too large, then it becomes very unlikely that another label ℓ' can be shared by more friends than ℓ. Conversely, if $p_{t,t'}(\ell)$ is too small, the ego will rarely have label ℓ, so there will be fewer situations where LP fails. Setting $p_{t,t'}(\ell) \approx (1-2q_t)/(2(1-q_t))$ achieves the optimal balance between these two. Alternatively, for small $p_{t,t'}(\ell)$ and q_t, the optimum linearly trades off $p_{t,t'}(\ell)$ and q_t (Corollary 4).

A second point of interest is the effect of the number of friends N: the optimal $p_{t,t'}(\ell)$ increases with increasing N. This can be explained by noting that as N increases, any difference in the expected popularities of ℓ' and ℓ becomes more likely to be reflected in their observed frequencies, and hence even a small expected difference $\Delta_{t,\ell,\ell'}$ is sufficient to create a situation where LP fails. However, when N is small, the effect of randomness is greater, and hence a smaller $p_{t,t'}(\ell)$ is needed to ensure that ℓ' is observed more often among the friends than ℓ.

7. Simulations

We seek answers to three questions from our simulation runs: (a) how can we simulate networks according to EdgeExplain, (b) how well do the various inference algorithms (relaxation labeling and variational) perform on such networks, and (c) how does the accuracy of label propagation vary with model parameters?

7.1 Simulating EdgeExplain

This closely follows the TwoLabels setting described in Section 6, i.e., we use two label types t and t'. Generating the simulation graph consists of three stages: (a) label generation, (b) edge generation, and (c) hiding labels.

Label generation: We generate small *neighborhoods* consisting of one node (the "ego") and her friends (the "alters", whose number N is a parameter of the simulation), as follows. First, the ego u selects her labels Y_u from a predefined joint distribution $\pi(Y_u)$ over all labels of all label types. Next, the ego selects, for each friend $v_1, \ldots, v_N$, the label type Z_i that explains their friendship; Z_i is picked from a distribution q over the set of label types. Clearly, this sets the corresponding label $Y_i(Z_i) = Y_u(Z_i)$. Conditioned on this, the labels of v_i for the other label type are given by $p_{t,t'}(\ell) \triangleq \pi(Y_i(t) \mid Y_u \text{ and } Z_i = t' \neq t) = \pi(Y_i(t) \mid Y_i(t') = Y_u(t'), Z_i = t' \neq t)$. We set $p_{t,t'}(\ell) = \pi(Y_i(t) = \ell)$.

Edge generation: To make the model more realistic, we introduce inter-friend links in addition to the links between the ego u and all her friends. For each pair of friends sharing at least one label type, a link is added between them with an inter-friend link probability p_{if}.

Hiding labels: Next, we hide some labels to create the actual train/test instances. All labels of the ego u are hidden. In addition, for each of the N friends, with probability p_h, all labels are hidden. This creates an ego network where some nodes have full information and others have no information. The goal is to infer the hidden labels of u from the network structure and the known labels of some friends.

Parameter choices: In order to create the simplest instance where parameter tradeoffs can be investigated, the labels and their probabilities are set as follows. We allow two possible labels for t, with marginal probabilities p and $1 - p$ respectively. We allow three possible labels for t', with marginal probabilities fixed at 0.3, 0.3, and 0.4. Since there are only two label types, the distribution $q(Z_i)$ over label types can be represented via just one number q, with $q_t = q$ and $q_{t'} = 1 - q$. Thus, each train/test instance can be indexed by the 4-tuple (p, q, p_{if}, p_h).

We set $N = 1,000$ and run 50 repetitions for each simulation setting. For each simulation, we check if the right label for label type t was inferred, since this is the label type whose label probabilities are varied via the parameter p. The average fraction of correct inferences over all simulations gives the accuracy of the inference technique. The parameter range is restricted to $q < 0.5$ in accordance with Thm. 3; for $q > 0.5$, label type t is shared most often with friends, so none of the inference algorithms face any difficulty. We also keep $p < 0.5$, since the case of $p > 0.5$ is symmetric (we do not differentiate between the two labels for type t).

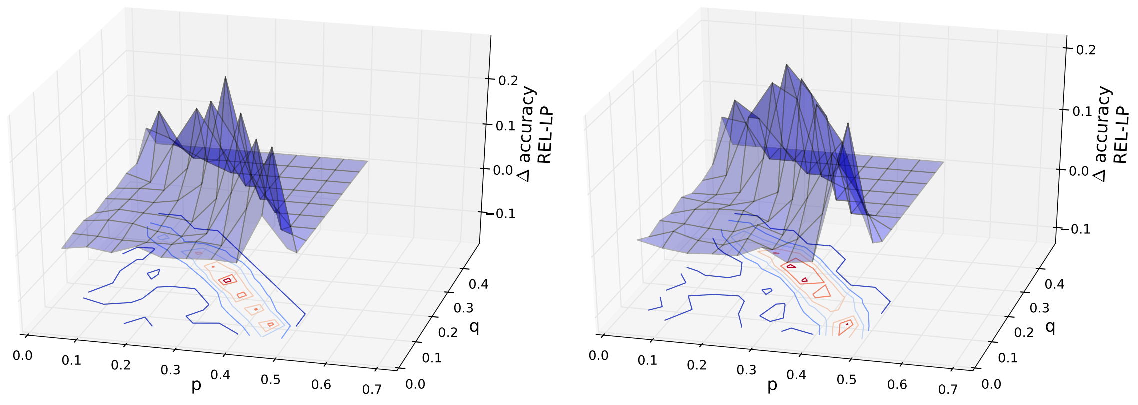
We note two points. First, the only test instance is the ego node u; friends with hidden labels are not considered in the test set since they might have other friends not present in the ego network of u. Second, while we could extend the graph generation process by adding friends of friends of u, the primary driver of inference accuracy is the information available among friends of u. Thus, the ego network with hidden node labels is the basic structure on which the accuracy of inference can be tested.

7.2 Comparison of inference algorithms

We will compare the accuracies of the three inference methods seen so far: relaxation labeling (REL), variational inference (VAR), and label propagation (LP).

Comparison against LP. Figure 2 shows the difference in accuracy between REL and LP as the parameters p, q, and p_h change, with $p_{if} = 0.3$ (the plots are almost identical with $p_{if} = 0$ and $p_{if} = 0.5$). We can make the following observations:

- REL is always at least as accurate as LP, and often much more accurate. Thus, under EdgeExplain, REL dominates LP.
- The parameter settings for which REL outperforms LP the most lie almost on a line in the pq-plane. This corresponds to $p + q \approx$ const., which agrees with Corollary 4.
- The results are unaffected by the probability of hiding labels and the probability of having links between friends. Hence, the difference between EdgeExplain inference



(a) $p_h = 0.0$ (b) $p_h = 0.5$

Figure 2: REL *outperforms* LP*:* The REL inference method under EdgeExplain outperforms LP irrespective of the fraction of friends whose labels are hidden. The greatest difference is when $p + q \approx$ const., agreeing with Corollary 4.

and label propagation is primarily a function of the relative probabilities of the various labels (the p) and the chances of various label types being used for forming friendships (the q); again, this is what we saw in Section 6.

Comparison of REL and VAR. Figure 3 compares REL against VAR. We observe the following:

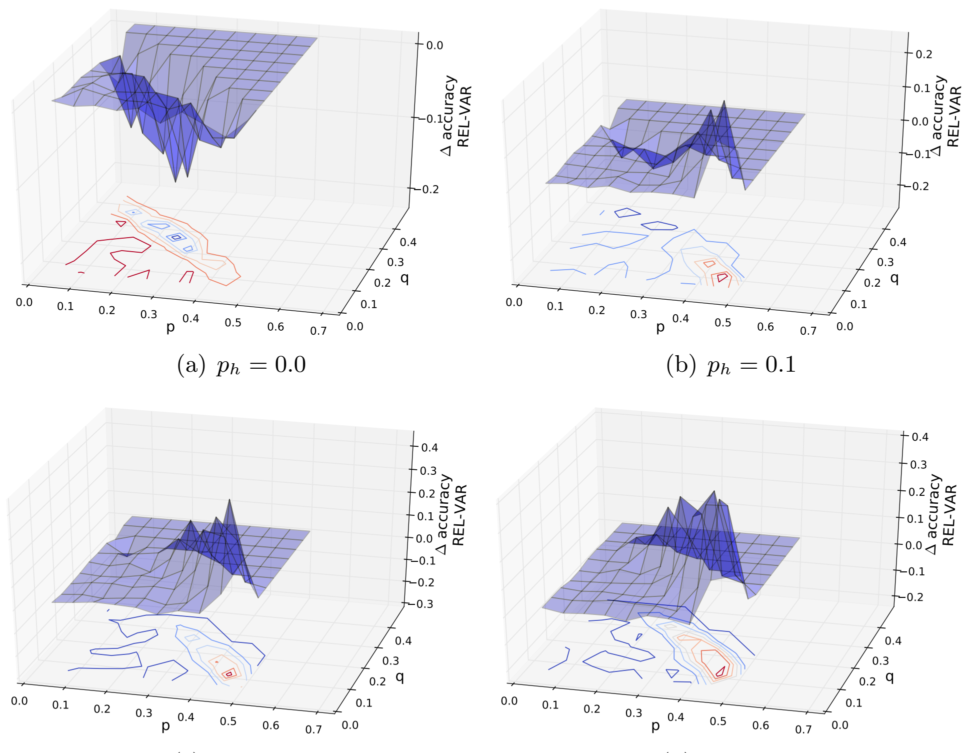
- Although both are algorithms for inference under EdgeExplain, they perform better under different parameter regimes. When the labels of all friends are known perfectly ($p_h = 0$), VAR performs better than REL. However, as p_h increases, information in the neighborhood of the ego becomes more uncertain, and REL performs better for high p_h. The transition point is around $p_h \approx 0.1$, i.e., when 90% of friend labels are known.

- The greatest difference is in the same region of the pq-plane where REL outperformed LP the most.

- With p_h fixed, the greatest *magnitude* of difference occurs for larger values of p.

- Indeed, for large p_h, VAR performs worse than even LP for large p and small q (Figure 4).

The underlying reason is the form of the variational update (Thm. 1), in particular, in the exponentiation which serves to magnify even minor differences in its argument. When p_h is high, there are more errors in labels assigned to friends with missing information. This leads to incorrect estimation of the chances that two nodes share a label (the parameter η_{uvt} in Theorem 1). This error gets compounded by the exponentiation, causing a quick convergence to erroneous locally-optimum labels in VAR.



(a) $p_h = 0.0$

(b) $p_h = 0.1$

(c) $p_h = 0.2$

(d) $p_h = 0.3$

Figure 3: REL *versus* VAR*:* REL is better when more friends hide their labels ($p_h \geq 0.1$). However, when labels of all friends are known ($p_h = 0$), VAR is better.

The problem is even more acute when p is high and q is low: the low q means that label type t is rarely shared among friends, while high p means that both possible labels of type t are common (recall that $p < 0.5$, so a high p corresponds to $p \approx 0.5$). This implies greater chances of mistakes in calculating η_{uvt} in the initial iterations, which is exacerbated by the exponentiation.

However, the positive side of exponentiation is that when friend labels are not noisy (as when $p_h = 0$), VAR converges quickly to the correct labels, while REL may get stuck in local minima during its gradient ascent. This explains why VAR is better for low p_h.

7.3 A hybrid inference algorithm

The previous discussion suggests the following hybrid inference procedure: in any iteration, for any node u, update its label probabilities via VAR if we are quite confident about the labels of most of the friends of u, otherwise use REL for the update. In effect, this tries to use VAR and REL precisely where they worked best in the previous simulations.

Thus, HYBRID counts the fraction of friends of u whose labels are "nearly certain" (for each label type, at least one label has probability greater than θ_1, normalized by the

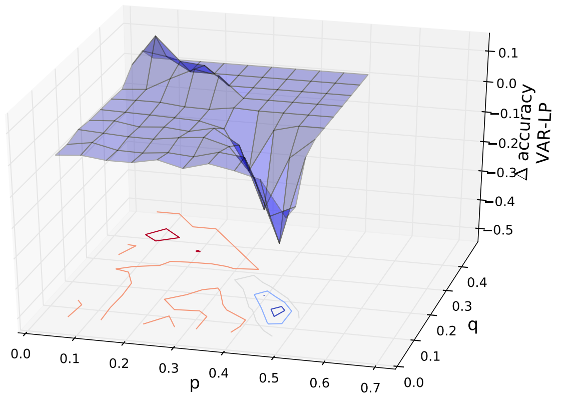


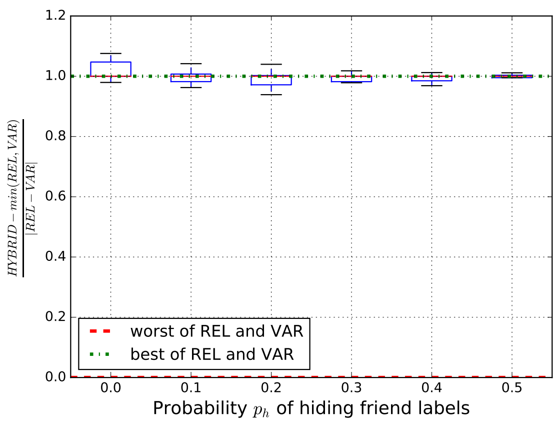
Figure 4: VAR *versus* LP *for one parameter setting:* With $p_h = 0.3$ and $p_{if} = 0.3$, VAR is better than LP for low p , and worse for high p.

Algorithm 1 HYBRID

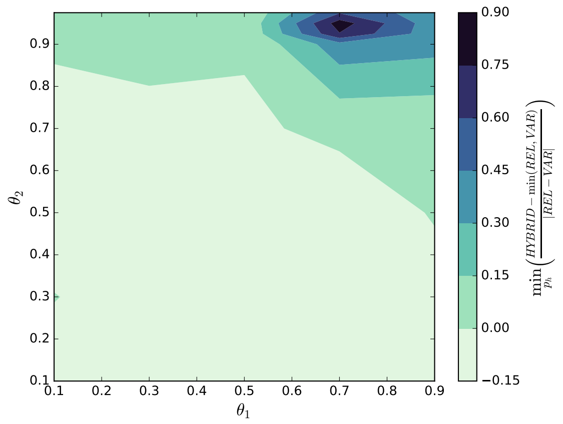
```
function HYBRID(initial label-probs, $\theta_1$, $\theta_2$)
    repeat
        for all nodes $u$ do
            $L_u$ = number of distinct labels in $\bigcup_{u \sim v}$ label-probs$_t(v)$
            nearly-certain-friends $= |\{v|u \sim v, \max(\text{label-probs}_t(v)) > 2\theta_1/|L_u|\ \forall t \in \mathcal{T}\}|$
            all-friends $= |\{v|u \sim v\}|$
            if nearly-certain-friends $> \theta_2 *$ all-friends then
                Update label-probs$_t(u)$ via VAR
            else
                Update label-probs$_t(u)$ via REL
            end if
        end for
    until convergence or maximum iterations
return label-probs($u$)
end function
```

number of labels in the neighborhood) and applies the variational update (VAR) if this fraction is more than θ_2, otherwise the relaxation labeling update (REL) is used.

Figure 5 compares the accuracy of HYBRID to REL and VAR. Plot (a) shows, for ($\theta_1 = 0.7, \theta_2 = 0.95$), the lift in accuracy of HYBRID against the others; a lift of 0 corresponds to accuracy equivalent to the worst of the two, while a lift of 1 corresponds to the best of the two. We see that HYBRID is in fact as good as the best of them for all p_h (recall that VAR was better for small p_h, and REL elsewhere). The results are very similar when the probability p_{if} of inter-friend edges is varied. Plot (b) shows a contour plot of the worst-case lift (i.e., the minimum lift among all p_h) over the (θ_1, θ_2) range. While the best setting is ($\theta_1 = 0.7, \theta_2 = 0.95$), we note that performance is quite robust even without picking the precise maximum.



(a) Lift of HYBRID



(b) Lift as a function of (θ_1, θ_2)

Figure 5: HYBRID *performs as well as the best of* REL *and* VAR*:* (a) The lift of HYBRID (with $\theta_1 = 0.7$ and $\theta_2 = 0.95$) over the minimum of REL and VAR is plotted against p_h (restricted to parameter settings where REL and VAR differ in accuracy by at least 0.01); a lift of 0 corresponds to no improvement over the worst of the two, while 1 corresponds to the performance of the better of the two. HYBRID is seen to provide better performance than REL or VAR over the entire range of p_h. (b) The minimum lift over all p_h is plotted for various (θ_1, θ_2) settings, with $\theta_1 = 0.7$, $\theta_2 = 0.95$ performing the best.

8. Empirical evaluation

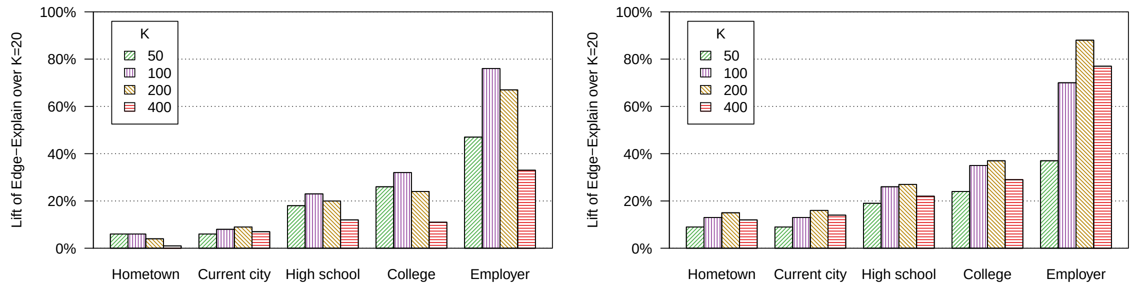
Previously, we provided intuition and examples supporting the claim that EdgeExplain is better suited to inference of our desired label types than label propagation. In this section, we demonstrate this via empirical evidence on two large social network data sets from Facebook and Google+, and a movie network.

8.1 Evaluation on the Facebook network

We performed a study on a previously collected subgraph of the Facebook social network (Chakrabarti et al., 2014). [1] This data set consists of a large number of users and their friendship edges, as well as the hometown, current city, high school, college, and employer for each user, whenever these fields are available and have their visibility set to public. The dimensionality of our five label types range from almost $900K$ to over $6.3M$. We describe below in Implementation Details our process for generating the edges. This forms our base data set.

Evaluation Methodology. The set of users is randomly split into five parts and the accuracy is measured via 5-fold cross-validation, with the known profile information from four folds being used to predict labels for all types for users in the fifth fold. Results over the

1. We worked on a snapshot of data, and there was no interaction with users or their experience on the site.



(a) Recall at 1 (b) Recall at 3

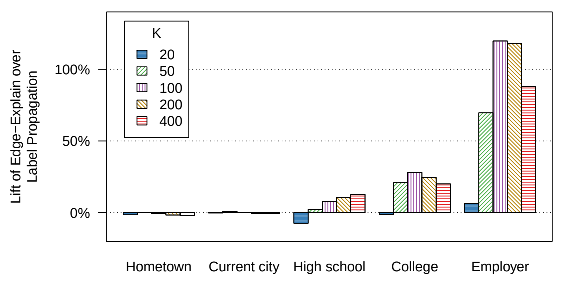
Figure 6: *Recall of* EdgeExplain *for graphs built with different number of friends K:* The plot shows lift in recall with respect to a fixed baseline of EdgeExplain with $K = 20$. Increasing K increases recall up to a point, but then the extra friends introduce noise which hurts accuracy.

various folds are identical to three decimal places. All differences are therefore significant and we do not show variances as they are too small to be noticeable.

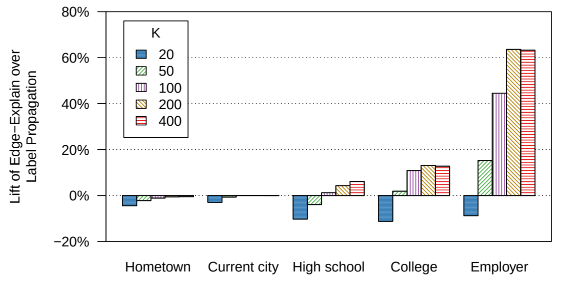
For each evaluation, we run inference on the training set and compute a ranking of labels for each label type for each user. This ranking is provided by the label probabilities computed by each label inference method. We then measure recall at the top-1 and top-3 positions, i.e., we measure the fraction of (user, label type) pairs in the test set where the predicted top-ranked label (or any of the top-3 labels) match the actual user-provided label. For reasons of confidentiality, we only present the *lift* in recall values with respect to a clearly specified baseline.

Implementation Details. We implemented EdgeExplain in Giraph (Ching, 2013) which is an iterative graph processing system based on the Bulk Synchronous Processing model (Malewicz et al., 2010; Valiant, 1990). The entire set of nodes is split among 200 machines, and in each iteration, every node u sends the probability distributions ($f_{ut\ell}$ for REL, $\mu_{ut\ell}$ for VAR) to all friends of u. To limit the communication overhead, we implemented two features: (a) for each user u and label type t, the label distribution for each (user, type) pair was clipped to retain only the top 8 entries optimally (Kyrillidis et al., 2013), and (b) the friendship graph is sparsified so as to retain, for each user u, the top K friends whose ages are closest to that of u. This choice of friends is guided by the intuition that friends of similar age are most likely to share certain label types such as high school and college. We find that clipping the distributions makes little difference to accuracy while significantly improving running time. However, the value of K matters significantly, and we detail these effects next.

Accuracy of EdgeExplain. We first show how recall varies as a function of the number of friends K. In the following, EdgeExplain with REL inference is used unless noted otherwise; comparisons with VAR are shown later. Figure 6 shows recall versus K, with recall at $K = 20$ being the baseline. We observe that recall increases up to a certain K and then decreases. The optimal value is $K = 100$ for recall at 1, and $K = 200$ for recall at 3. This demonstrates both the importance and the limits of scalability: increasing the



(a) Recall at 1



(b) Recall at 3

Figure 7: *Lift of* EdgeExplain *over label propagation:* Increasing the number of friends K benefits EdgeExplain much more than label propagation for high school, college, and especially employer.

number of friends enables better inference but beyond a point, more friends increase noise. Thus, $K \approx 100$ friends appear to be enough for inference under EdgeExplain.

Figure 6 also shows an increasing trend from hometown to employer in the degree of improvement obtained over the $K = 20$ baseline. This is because (a) the baseline itself is best for hometown and worst for employer, but also because (b) Facebook users appear to have many more friends from label types other than from their current employer. The effect of this latter observation is that if we only have a small K, it is very likely that the few friends from the same current employer are not included in that limited set of friends (which we empirically verified). As K increases and such same-employer edges become available, EdgeExplain can easily learn the reason for these edges (hence the dramatic increase in recall), but label propagation will likely be confused by the overall distribution of different employers among all friends and therefore does not benefit equally from adding more friends, as we show next.

Comparison with Label Propagation. Figure 7 shows the lift in recall achieved by EdgeExplain over label propagation as we increase K for both. We observe similar performance of both methods for hometown and current city, but increasing improvements for high school, college, and employer. This again points to the first two label types being easier to infer, while the latter three are more difficult. With fewer employer-based friendships, the prototypical example of Figure 1 would also occur frequently, with label propagation likely picking common employers of (say) hometown friends instead of the less common friendships based on the actual employer. By attempting to explain each friendship, EdgeExplain is able to infer the employer even under such difficult circumstances. This ability to perform well even for under-represented label types makes EdgeExplain particularly attractive.

Effect of α. Figure 8 shows that the lift in recall at 1 for various values of the parameter α, with respect to $\alpha = 0.1$. Performance generally improves with increasing α. Results for recall at 3 are qualitatively similar, though the effect is more muted. We find that $\alpha \in [10, 40]$ offer the best results, and EdgeExplain is robust to the specific choice of α within this range. Recall that with large α, a single matching label is enough to explain an edge,

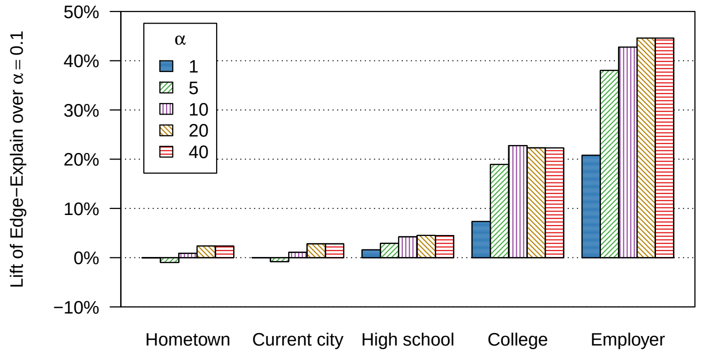
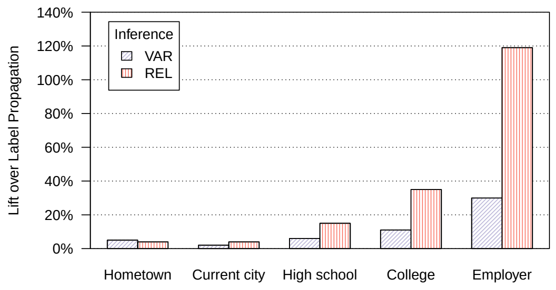
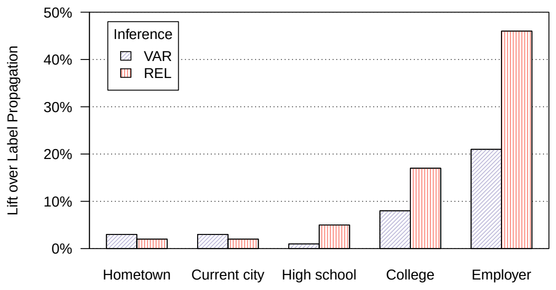


Figure 8: *Effect of α:* Lift in recall at 1 is plotted for different values of α, with respect to $\alpha = 0.1$. The best results are for $\alpha \in [10, 40]$.



(a) Recall at 1



(b) Recall at 3

Figure 9: *Comparison of* VAR *and* REL*:* The lift of each inference method over label propagation propagation is plotted for $K = 100$. REL is seen to outperform VAR, and both are better than LP.

while with small α, multiple matching labels may be needed. Thus, the outperformance of large α provides empirical validation of property **(P2)** (on our network).

Variational inference versus Relaxation Labeling. Our two main methods for inference under EdgeExplain are REL and VAR. Recall that while both use per-node parameters, they attempt to optimize Eqs. 2–4 in different ways. REL relaxes the problem by replacing the hidden variables $S_{ut\ell}$ by their "probabilities" $f_{ut\ell}$, but the relaxed objective is not formally related to the probabilistic model. On the other hand, VAR formally optimizes a lower-bound on the model log-likelihood, but the lower bound may not be very tight (we note, however, that this is the standard variational assumption when per-node parameters are needed).

Figure 9 compares the two inference methods. The lift in recall of each method over LP is shown, with $K = 100$. While both outperform LP, REL is more accurate than VAR. This agrees with our simulation results, where VAR outperformed REL only in a narrow parameter regime, where the labels of most friends of a given node are known with

Table 1: *Lift in recall from using group memberships:* Inclusion of group membership barely improves recall@3, even though it is an orthogonal feature with wide coverage. Thus, information about label types is already encoded in the network structure, and careful modeling via EdgeExplain is sufficient to extract it.

Label Type	Recall at 1	Recall at 3
Hometown	−0.1%	0.7%
Current city	0.4%	1.0%
High school	0.1%	0.8%
College	−0.6%	1.0%
Employer	−2.8%	1.2%

high confidence. Such instances are unlikely in social networks, contributing to the relative weakness of VAR.

Given the better performance of REL on this data set, the following results in this section use REL.

Inclusion of extra features. EdgeExplain easily generalizes to broader settings with multiple user features, such as group memberships, topics of interest, keywords, or pages liked by the user. As an example, consider group memberships of users. Intuitively, if most members of a group come from the same college, then it is likely a college-friends group, and this can aid inference for group members whose college is unknown. This can be easily handled by creating a special node for each group, and creating "friendship" edges between the group node and its members. EdgeExplain will infer labels for the group node as well, and will explain its "friendships" via the college label. This, in turn, will influence college inference for group members with unknown college labels. Group memberships are extensive and provide information that is orthogonal to friendships; hence, *a priori*, one would expect the addition of group membership features to have significant impact on label inference.

Table 1 shows the lift in recall for EdgeExplain when group memberships are used in addition to $K = 100$ friends. While the addition of group memberships increases the size of the graph by $\approx 25\%$, the observed benefits for recall are minor: a maximum lift of only 1.2% for employer inference, and indeed reduced recall at 1 for several label types. Note that the lift in recall would have appeared very significant had we compared it to label propagation with $K = 100$; however, this gain largely disappears when the friendships are considered in the framework of EdgeExplain. Thus, it is not merely the scalability of EdgeExplain, but also the careful modeling of properties **(P1)**-**(P3)** that makes group membership redundant.

Given the *a priori* expectations of the impact of group memberships, this surprising result suggests that information regarding our label types are already encoded in the structure of the social network and hence the orthogonal information from the group memberships actually turns out to be redundant.

The limits of resolution. Our model theoretically should be able to handle any number of label types, but empirically this may not hold true for our network. How many friends

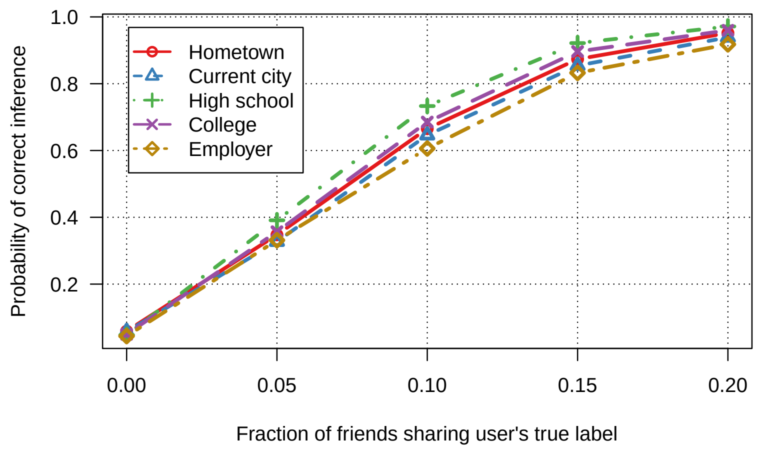


Figure 10: *Probability of correctly inferring (in the top-3) the value of a given label type t for a user, given the fraction of friends with known label for t who actually share the user's label for t:* All label types are broadly similar, with a fraction of 0.1 usually being sufficient for inference. For fraction > 0.2, the plot flattens out.

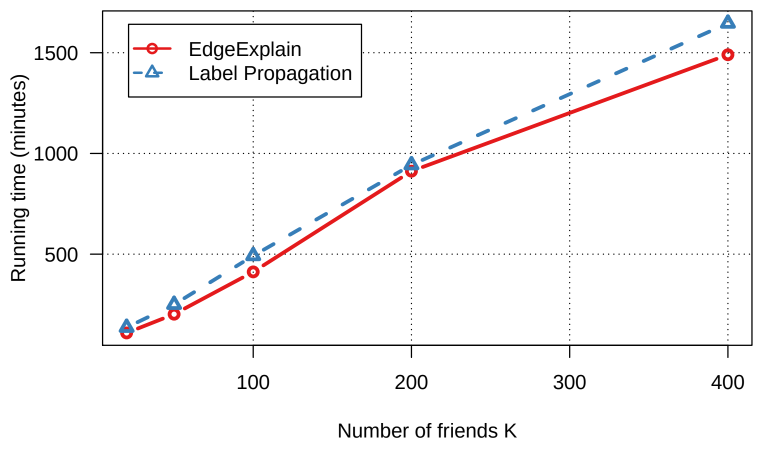
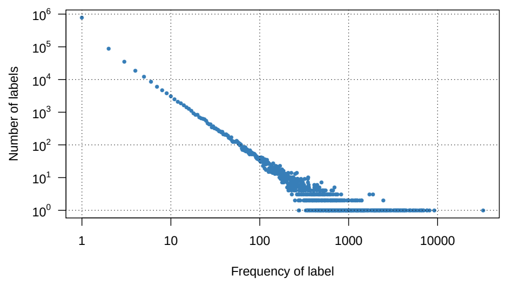
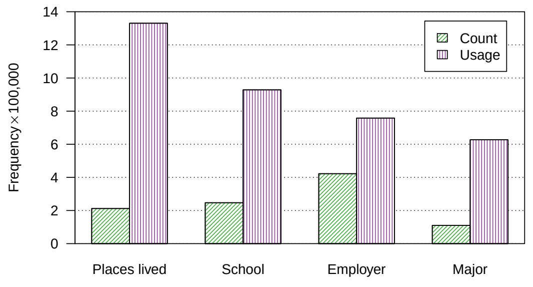


Figure 11: Running time increases linearly with K.

sharing a certain label type (say, the same college) does a user need to have in order to correctly infer the value of that label type? To answer this, we select, for each user, the set of friends whose label for the given label type t is known, and we compute the fraction that actually shared the user's label for t. Figure 10 shows the probability that EdgeExplain correctly infers the user's label as a function of this fraction (i.e., the correct label is among the top 3 predictions). All label types are similar, though high school is somewhat easier and employer harder; having $10 - 15\%$ of friends sharing a user's label is sufficient to infer the label in our graph. Note that certain label types are more likely to be publicly declared than others, and this explains differences in recall observed earlier.



(a) Label usage frequency distribution



(b) Number of labels and label usage for each type

Figure 12: Descriptive statistics for Google+.

Running time. Figure 11 shows the wall-clock time as a function of K. The running time should depend linearly on the graph size, which grows almost linearly with K; as expected, the plot is linear, with deviations due to garbage collection stoppages in Java.

8.2 Evaluation on the Google+ network

We also conducted empirical evaluation on data collected from the Google+ social network (Gong et al., 2012). The data set consists of several snapshots of the network, of which we picked the oldest snapshot for our evaluation. This snapshot contains $4,693,129$ users connected by $47,130,325$ links. In addition, the data lists four attributes of each user (if the user has provided this information publicly): their employer, school, major, and the places where they have lived. There are $991,545$ labels, and $4,443,631$ (user, label) pairs in the data set. Gong et al. (2012) compared several methods for attribute inference on this data set, and found that a close variant of Label Propagation (which they called CN-SAN) has the highest precision. Hence, we only compare against Label Propagation.

We first note the extreme sparsity of the data set. There are only 10 friends per person on average, and only 1 label per person. In fact, only $909,669$ people have any labels associated with them at all (i.e., only 19% of the total). The number of people who declared all four of their label types is only $123,023$, or 2.6% of all people. Figure 12 shows some descriptive statistics regarding the labels. The distribution of labels among users (plot (a)) demonstrates a power-law: a few labels are extremely common, while the vast majority occur only once. Plot (b) shows the number of labels, and the usage, for each label type. While the most commonly used labels refer to "places lived," the "employer" label type has the most distinct labels. In other words, information about employers is particularly sparse, as in the Facebook data set.

Figure 13 shows the accuracy of each inference algorithm on all four label types. We see that there is practically no difference between the algorithms. This is due to the sparsity of the data; with so little information, there is hardly enough information among friends for EdgeExplain to differentiate itself from unrestricted homophily, and the most common label among friends is often the only reasonable choice for REL, VAR, or HYBRID (and of course for LP).

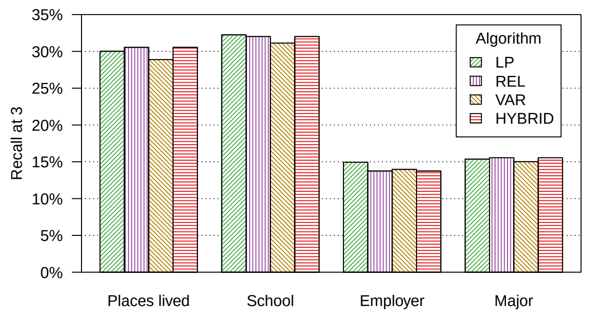


Figure 13: *Recall on the Google+ data set:* All algorithms perform similarly, primarily because the data set is too sparsely labeled.

8.3 Evaluation on a Movie Network

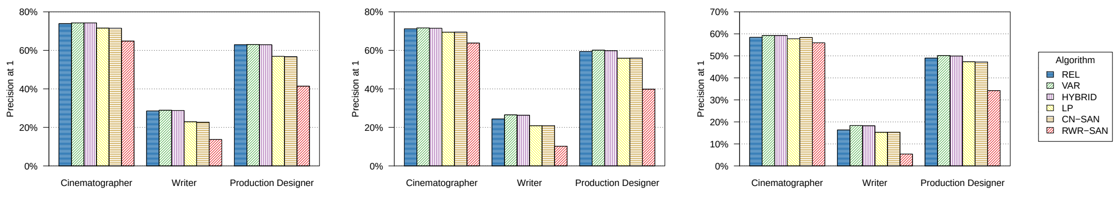
We have so far compared EdgeExplain to Label Propagation (LP), which has been shown to be among the best predictors of node attributes in previous work (see Section 2). We have shown that EdgeExplain works as well as LP on the sparse Google+ data set, and significantly outperforms it on the Facebook data set. Now, we will perform a broader comparison of EdgeExplain against LP as well as other competing methods on a movie network.

We construct the network as follows. Each node is an English-language movie, and has three label types: the writer, the cinematographer, and the production designer.[2] These three figures represent critical behind-the-scenes roles that support the movie's director.[3] We only keep movies with a single label for each label type (i.e., no "joint writers" etc.), so as to simplify testing of prediction accuracy. There are $23,921$ such movies. Two movies are connected by an edge if they have the same writer, or cinematographer, or production designer (they could share several of these label types); note that connecting nodes to their "nearest-neighbors" is a standard technique for converting feature-based data sets into networks, and has been used for LP as well (Zhou et al., 2003). There are $100,098$ such movie pairs that share cinematographers, $20,560$ pairs sharing writers, and $84,328$ pairs sharing production designers. There are $189,828$ edges overall. Thus, by construction, the network structure reflects shared labels, satisfying the basic assumption upon which all label inference methods rest.

We shall compare all inference methods for EdgeExplain (i.e., REL, VAR, and HYBRID) against LP. In addition, we test three algorithms that have been recommended by prior work:

- LINK (Zheleva and Getoor, 2009), where each node is represented by a feature vector encoding the IDs of its neighboring nodes, and a standard classifier is used to predict labels from the feature vector. Given the size of the feature vector ($23,921$ binary

2. The data is available at `ftp://ftp.fu-berlin.de/pub/misc/movies/database/`.
3. We have also experimented with other label types such as director and composer. All results are qualitatively similar and exhibit the same trends.



(a) 5% unlabeled nodes (b) 20% unlabeled nodes (c) 50% unlabeled nodes

Figure 14: Precision@1 for various label inference methods on the movie network.

features) and the size of the output labels (39, 317 labels in total), we choose Naive Bayes as the classifier.

- CN-SAN (Gong et al., 2014), which assigns to each unlabeled node the most popular label of each label type among the node's neighbors, and iterates this process until convergence.

- RWR-SAN (Yin et al., 2010; Gong et al., 2014), which picks labels for each label type based on random walks with restarts on a combined graph that includes movie-movie and movie-label edges. We use a restart probability of 0.15.

Figure 14 shows the precision@1 for different fractions of the network being labeled. All methods achieve higher accuracy for cinematographer and production designer, but lower accuracy for writer. This is because less than 11% of the edges have a shared-writer as the reason for that edge.

All EdgeExplain inference procedures (REL, VAR, and HYBRID) are better than the competing methods for all label types. All EdgeExplain methods are similar for predicting cinematographers and production designers. However, for writer prediction, VAR is better than REL by about 7% on average. This is because edges are formed primarily due to shared cinematographers and production-designers, and these are reasonably easy to infer (for instance, even LP works well for these). Thus, the fraction of "near-certain friends" (i.e., friendships for which the underlying reason is accurately estimated; see Algorithm 1) is high enough that VAR becomes more accurate than REL at predicting writers, as shown earlier in our simulation studies (Figure 5(b)). Note that for the Facebook data set, REL was much better than VAR. This shows the usefulness of HYBRID in tracking the best-performing variant of EdgeExplain.

The differences between the EdgeExplain variants are much less than that between EdgeExplain and the second-best method, which is LP. The greatest difference is again for writer prediction, where HYBRID is always $20 - 25\%$ better than LP. This mirrors our observations on the Facebook network where the greatest difference between EdgeExplain and LP was in predicting a person's employer; again, friendships formed due to a shared employer are relatively rare.

We again note the fact that LP is the best among the baseline methods. This agrees with previous work by Macskassy and Provost (2007) which showed the high prediction accuracy of LP. Observe that CN-SAN performs about as well as LP; this is expected, since CN-SAN is just LP with a hard label assignment in each iteration. Finally, we note that

LINK performed poorly and is omitted from the plots. We believe this is because both the number of nodes and the number of labels are too large. LINK might be a better option for predicting label types with fewer possible labels such as gender and political affiliation, as in (Zheleva and Getoor, 2009).

9. Generalizations

We now discuss some aspects and generalizations of EdgeExplain that demonstrate its wide applicability.

Related label types. Property **(P2)** assumes that the reasons for friendship formation are mutually exclusive, but this need not be strictly true. For example, some high school friends could be a subset of hometown friends.[4] Let us again consider Figure 1, but with current city replaced by high school. Suppose that the solid-black nodes represent actual high school friends, and we are trying to infer u's high school. If the small cluster on the right did not exist, then Eq. 2 would be maximized by picking the most common high school among u's friends (i.e., the solid-black nodes), even if they are already explained by a shared hometown; thus, EdgeExplain would pick the correct high school. On the other hand, if some friendships would remain unexplained without a shared high school (e.g., the small cluster in Figure 1), then it is not obvious whether we should prefer a high school that explains these edges or a high school that represents a large segment of hometown friends. The parameter α modulates this trade-off, with a higher value of α emphasizing the explanation of all edges as against the explanation of several edges a little better. The choice of α must depend on the characteristics of the social network; for the Facebook network, the best empirical results are achieved for large α (shown in Section 8.1), suggesting that many of our label types are indeed mutually exclusive.

Incorporating edge features. There are several situations where edge-specific features could be useful. First, we may want to give more importance to certain kinds of edges, such as the group-membership edges mentioned above. Second, some features could be important for one label type but not another: e.g., the age difference between friends could be useful for inferring high school but not employer. All these situations can be easily handled by modifying Eq. 3 to include an edge-specific and label type-specific weight. The corresponding modifications to the inference method are trivial.

Nodes with multiple labels. Our original formulation of EdgeExplain (Eqs. 2–4) assumes that each node has a single label for each label type, i.e., for any user u and label type t, only one of the indicator variables $S_{ut\ell}$ is 1. The accuracy of EdgeExplain on real-world data sets shows that this is at least a reasonable approximation. However, if handling multiple labels is critical, two generalizations of EdgeExplain are possible.

First, we can allow $S_{ut\ell}$ to be 1 for multiple labels ℓ of label type t. This leads to a modified relaxation labeling formulation, where the inferred label scores $f_{ut\ell}$ need not sum

4. The relationship between high school and hometown is in fact more complicated. The high school could be within driving distance of the hometown, but not in it; and sometimes even this does not hold.

to 1. Specifically, we choose the label scores as follows:

$$\begin{aligned}
\underset{\boldsymbol{f}}{\text{Maximize}} \; & \sum_{u \sim v} \log\Big(\underset{t \in \mathcal{T}}{\text{softmax}}(r(u,v,t))\Big) - \gamma \cdot \sum_{u} \sum_{t \in \mathcal{T}} \left(\sum_{\ell} f_{ut\ell} - 1 \right)_{+} \\
\text{where } r(u,v,t) = & \sum_{\ell \in L(t)} f_{ut\ell} f_{vt\ell} \\
& f_{ut\ell} \geq 0 \\
& f_{ut\ell} \leq 1
\end{aligned}$$

Thus, the inferred label scores $f_{ut\ell}$ need to sum to 1, but a hinge loss penalty $(\sum_\ell f_{ut\ell} - 1)_+$ is applied whenever it is greater than 1 (x_+ denotes $\max(x, 0)$). Increasing the penalty parameter γ enforces the constraint more strictly, and as $\gamma \to \infty$, we regain our original relaxation labeling formulation (Eqs. 5-8).

The second approach is to modify EdgeExplain, by again allowing $S_{ut\ell}$ to be 1 for multiple labels ℓ, but using a normalization when computing the degree to which a label type "explains" an edge. More precisely, we replace $r(u,v,t) = \sum_\ell S_{ut\ell} S_{vt\ell}$ (Eq. 3) by a probabilistic version:

$$P\left(r(u,v,t) > 0 \mid S_{ut.}, S_{vt.}\right) = \sum_{\ell} \left(\frac{S_{ut\ell}}{\sum_{\ell'} S_{ut\ell'}} \cdot \frac{S_{vt\ell}}{\sum_{\ell'} S_{vt\ell'}} \right) \tag{12}$$

In addition, we shall let the indicator random variables $S_{ut\ell}$ be drawn independently from Bernoulli distributions: $P(S_{ut\ell} = 1 \mid \mathcal{S}^-) = f_{ut\ell}$, where $\mathcal{S}^-$ represents all indicators except $S_{ut\ell}$. Then, by marginalizing our $S_{ut.}$ and $S_{vt.}$ in Eq. 12, we can express the marginal probability $P(r(u,v,t) > 0)$ in terms of the parameters $\boldsymbol{f}$. In effect, we solve the following optimization problem:

$$\begin{aligned}
\underset{\boldsymbol{f}}{\text{Maximize}} \; & \sum_{u \sim v} \log\Big(\underset{t \in \mathcal{T}}{\text{softmax}}\left(P(r(u,v,t) > 0)\right)\Big) \\
\text{where } P\left(r(u,v,t) > 0\right) = & \sum_{\ell \in L(t)} E_{S_{ut.}|f_{ut.}} \left[\frac{S_{ut\ell}}{\sum_{\ell'} S_{ut\ell'}} \right] \cdot E_{S_{vt.}|f_{vt.}} \left[\frac{S_{vt\ell}}{\sum_{\ell'} S_{vt\ell'}} \right] \\
& f_{ut\ell} \geq 0 \\
& f_{ut\ell} \leq 1
\end{aligned}$$

The first approach has the advantage of being a simple extension to our existing relaxation labeling approach, which already works well. However, it might require careful tuning for the extra parameter γ. The second approach directly incorporates the desired quantities $\boldsymbol{f}$ as model parameters, so it does not require an extra relaxation step. However, it must make a strong independence assumption. We leave a detailed comparison of these approaches for future work.

10. Conclusions

We proposed the problem of jointly inferring multiple correlated label types in a large network and described the problems with existing single-label models. We noted that one particular failure mode of existing methods in our problem setting is that edges are often created for a reason associated with a particular label type (e.g., in a social network, two users may link because they went to the same high school, but they did not go to the same college). We identified three network properties that model this phenomenon: edges are created for a reason **(P1)**, they are generally created only for one reason **(P2)**, and sharing the same value for a label type is necessary but not sufficient for having an edge between two nodes **(P3)**.

We introduced EdgeExplain, which carefully models these properties. We presented two inference methods for EdgeExplain: a relaxation-labeling method and a variational approach, both of which lead to fast iterative inference that is equivalent in running time to basic label propagation. Our empirical evaluation on a large subset of the Facebook graph amply demonstrates the benefits of EdgeExplain, with significant improvements across a set of different label types. Our further analysis validates many of the properties and intuitions we had about modeling networks, primarily that one can achieve significant improvements if one considers and models the *reason* an edge exists. Whether one is interested in inferring one or multiple label types, modeling these explanations will have significant impact on the accuracy of the final predictions.

Appendix A. Proofs

Proof [Theorem 1] Note that $\sum_{t\ell} S_{ut\ell}S_{vt\ell}$ is the number of matching label types between u and v, and hence is an integer between 0 and $|\mathcal{T}|$. Thus,

$$\left\langle \ln\left(1+e^{-\left(\alpha\sum_{t\ell} S_{ut\ell}S_{vt\ell}+c\right)}\right)\right\rangle = \sum_{k=0}^{|\mathcal{T}|} \ln\left(1+e^{-(\alpha k+c)}\right)\left\langle I\left\{\sum_{t\ell} S_{ut\ell}S_{vt\ell}=k\right\}\right\rangle.$$

Now, $\sum_{t\ell} S_{ut\ell}S_{vt\ell}$ equals k iff exactly k label types out of $|\mathcal{T}|$ are shared, i.e., there exists a binary vector $\mathbf{w}\in\{0,1\}^{|\mathcal{T}|}$ of length $|\mathcal{T}|$ such that exactly k bits are "one" ($|\mathbf{w}|=k$), and

$$\sum_{\ell} S_{ut\ell}S_{vt\ell} = \begin{cases} 1 & \text{if } w_t = 1 \\ 0 & \text{otherwise.}\end{cases}$$

Thus,

$$I\left\{\sum_{t\ell} S_{ut\ell}S_{vt\ell}=k\right\} = \sum_{\{\mathbf{w}\,:\,|\mathbf{w}|=k\}} \prod_t M_t^{w_t}(1-M_t)^{1-w_t},$$

where $M_t = I\left\{\sum_\ell S_{ut\ell}S_{vt\ell}=1\right\}$ indicates a shared label type t (the subscripts u and v are dropped for clarity). Thus,

$$\left\langle I\left\{\sum_{t\ell} S_{ut\ell}S_{vt\ell}=k\right\}\right\rangle_Q = \sum_{\{\mathbf{w}\,:\,|\mathbf{w}|=k\}} \prod_t \left\langle M_t^{w_t}(1-M_t)^{1-w_t}\right\rangle_Q,$$

using the linearity of expectation and the independence of label types in the factorized distribution Q. Now, the expectation $\langle M_t\rangle_Q$ is just the probability of sharing a label of type t, so $\langle M_t\rangle_Q = \sum_\ell \mu_{ut\ell}\mu_{vt\ell} = \eta_{uvt}$. Then,

$$\kappa(w_t,\eta_{uvt}) \triangleq \left\langle M_t^{w_t}(1-M_t)^{1-w_t}\right\rangle_Q = \eta_{uvt}^{w_t}(1-\eta_{uvt})^{1-w_t}.$$

We want to find the optimal $\boldsymbol{\mu}$ that maximizes the lower bound of Eq. 10, under the constraint that $\sum_\ell \mu_{ut\ell} = 1$ for all label types t and users u. Combining all the above equations, we want to

$$\begin{aligned}\underset{\boldsymbol{\mu}}{\text{maximize}} \quad & -\sum_{ut\ell} \mu_{ut\ell} \ln \mu_{ut\ell} \\ & -\sum_{u\sim v} \sum_{\mathbf{w}} \ln\left(1 + e^{-(\alpha|\mathbf{w}|+c)}\right) \prod_t \kappa(w_t, \eta_{uvt}) \\ & -\sum_{ut} \lambda_{ut} \left(\sum_\ell \mu_{ut\ell} - 1\right),\end{aligned}$$

where λ_{ut} are the Lagrange multipliers. Setting the derivatives with respect to $\boldsymbol{\mu}_u$ to zero and holding $\boldsymbol{\mu}_v$ fixed for $v \neq u$, we find:

$$\begin{aligned}\mu^*_{ut\ell} &\propto \exp\left\{ \sum_{\{v|u\sim v\}} \mu_{vt\ell} \sum_{\mathbf{w}} \phi(\mathbf{w}) \right\} \\ \phi(\mathbf{w}) &= \ln\left(1 + e^{-(\alpha|\mathbf{w}|+c)}\right) (-1)^{w_t} \prod_{t'\neq t} \kappa(w_{t'}, \eta_{uvt'}).\end{aligned}$$

Normalizing this expression over all $\ell' \in L(t)$ gives the exact value of $\mu_{ut\ell}$. In practice, only the top few labels for label type t need to be considered. ■

Proof [Proposition 1] Let $\mathbf{w}^{(-t)} \in \{0,1\}^{|\mathcal{T}|-1}$ represent the binary vector $\mathbf{w}$ restricted to all but the t^{th} component. Denoting $\prod_{t'\neq t} \kappa(w_{t'}, \eta_{uvt'})$ by $\chi(\mathbf{w}^{(-t)})$,

$$\begin{aligned}\sum_{\mathbf{w}} \phi_{uv}(\mathbf{w}) &= \sum_{\mathbf{w}} \chi(\mathbf{w}^{(-t)})(-1)^{w_t} \ln\left(1 + e^{-(\alpha|\mathbf{w}|+c)}\right) \\ &= \sum_{\mathbf{w}^{(-t)}} \chi(\mathbf{w}^{(-t)}) \ln\left(\frac{1 + e^{-\alpha|\mathbf{w}^{(-t)}|-c}}{1 + e^{-\alpha(|\mathbf{w}^{(-t)}|+1)-c}}\right) \\ &\geq 0,\end{aligned}$$

where the first equality follows by grouping the two $\mathbf{w}$ terms with the same $\mathbf{w}^{(-t)}$ and summing the contributions for $w_t = 0$ and 1. The inequality holds because the numerator inside the logarithm is greater than the denominator, and $\chi \geq 0$. ■

Proof [Theorem 2] Let $\mathcal{U}$ represent the event that the ego has already selected her labels Y_u, and has chosen label ℓ for label type t. Define the random variable X_i as $+1$ if the friend v_i has label ℓ' for type t, -1 if it is ℓ, and 0 otherwise: $X_i = I_{Y_i(t)=\ell'} - I_{Y_i(t)=\ell}$. Then, LP fails via (t, ℓ, ℓ') if $\sum_i X_i > 0$. We bound the probability of the latter quantity by noting that it is the sum of independent random variables X_i, so a Hoeffding bound applies:

$$P\left(\sum X_i \geq E\left[\sum X_i\right] - Nx \,\middle|\, \mathcal{U}\right) \geq 1 - \exp\{-0.5Nx^2\} \quad (\text{for } x \geq 0). \tag{13}$$

Then, $E[X_i \mid \mathcal{U}]$ is easily seen to be:

$$\frac{E\left[\sum X_i \mid \mathcal{U}\right]}{N} = E[X_i \mid \mathcal{U}] = \sum_{t' \neq t} q_{t'} \left(p_{t,t'}(\ell') - p_{t,t'}(\ell)\right) - q_t = \Delta_{t,\ell,\ell'}.$$

Thus, $\Delta_{t,\ell,\ell'}$ is the expected difference in the rates of occurrences of ℓ' and ℓ; note that it depends on both how frequent the labels ℓ and ℓ' are, and the ego's preferences for choosing friends sharing each label type. Setting $x = E\left[\sum X_i \mid \mathcal{U}\right]/N - \epsilon = \Delta_{t,\ell,\ell'} - \epsilon$ for some small $\epsilon > 0$ in Eq. 13, we find:

$$P\left(\sum X_i > 0 \,\middle|\, \mathcal{U}\right) \geq 1 - \exp\{-0.5N(\Delta_{t,\ell,\ell'} - \epsilon)^2\}.$$

Hence,

$$\begin{aligned}
P(\text{LP fails via } (t,\ell,\ell')) &\geq P\left(\sum X_i > 0\right) \\
&= \sum_{\{Y_u \mid Y_u(t)=\ell\}} \pi(Y_u) \cdot P\left(\sum X_i > 0 \,\middle|\, Y_u \text{ where } Y_u(t) = \ell\right) \\
&= \sum_{\{Y_u \mid Y_u(t)=\ell\}} \pi(Y_u) \cdot P\left(\sum X_i > 0 \,\middle|\, \mathcal{U}\right) \\
&\geq \sum_{\{Y_u \mid Y_u(t)=\ell\}} \pi(Y_u) \cdot \left(1 - \exp\{-0.5N(\Delta_{t,\ell,\ell'} - \epsilon)^2\}\right). \qquad (14)
\end{aligned}$$

■

Proof [Theorem 3] Restating Eq. 14, we find:

$$\begin{aligned}
P(\text{LP fails via } (t,\ell,\ell')) &\geq \sum_{\{Y_u \mid Y_u(t)=\ell\}} \pi(Y_u) \cdot \left(1 - \exp\{-0.5N(\Delta_{t,\ell,\ell'} - \epsilon)^2\}\right) \\
&= \sum \pi(Y_u(t')) \cdot \pi(Y_u(t) = \ell \mid Y_u(t')) \cdot \left(1 - \exp\{-0.5N(\Delta_{t,\ell,\ell'} - \epsilon)^2\}\right) \\
&= \sum \pi(Y_u(t')) \cdot p_{t,t'}(\ell) \cdot \left(1 - \exp\{-0.5N(\Delta_{t,\ell,\ell'} - \epsilon)^2\}\right).
\end{aligned}$$

To optimize the lower bound, we set the derivative of the RHS with respect to $p_{t,t'}(\ell)$ to 0. Noting that under TwoLabels, $p_{t,t'}(\ell') = 1 - p_{t,t'}(\ell)$, $q_{t'} = 1 - q_t$, and $\Delta_{t,\ell,\ell'} = q_{t'}(1 - 2p_{t,t'}(\ell)) - q_t = (1 - q_t)(1 - 2p_{t,t'}(\ell)) - q_t$, we find:

$$\begin{aligned}
0 &= \left(1 - \exp\{-0.5N(\Delta_{t,\ell,\ell'} - \epsilon)^2\}\right) \\
&\quad - 2Np_{t,t'}(\ell) \cdot \exp\{-0.5N(\Delta_{t,\ell,\ell'} - \epsilon)^2\} \cdot (\Delta_{t,\ell,\ell'} - \epsilon) \cdot (1 - q_t) \\
&= 1 - \exp\{-0.5N(\Delta_{t,\ell,\ell'} - \epsilon)^2\} \cdot \left(1 + 2Np_{t,t'}(\ell)(1 - q_t)\left((1 - q_t)(1 - 2p_{t,t'}(\ell)) - q_t - \epsilon\right)\right) \\
&= 1 - \exp\{-0.5N(\Delta_{t,\ell,\ell'} - \epsilon)^2\} \times \\
&\qquad \left(1 + \underbrace{2(1 - q_t)(1 - 2q_t - \epsilon)}_{c_1} Np_{t,t'}(\ell) - \underbrace{4(1 - q_t)^2}_{c_2} Np_{t,t'}(\ell)^2\right) \\
&= 1 - \exp\{-0.5N \cdot \underbrace{\left((1 - 2q_t - \epsilon) - 2(1 - q_t)p_{t,t'}(\ell)\right)^2}_{c_3}\} \cdot \left(1 + c_1Np_{t,t'}(\ell) - c_2Np_{t,t'}(\ell)^2\right).
\end{aligned}$$

(15)

For large N, this can be satisfied only if $c_3 \approx 0$. Hence, let us express $p_{t,t'}(\ell)$ as $p_{t,t'}(\ell) = (1 - 2q_t - \epsilon)/(2(1 - q_t)) - p' \triangleq \kappa - p'$ for some small p', where $p' > 0$ to ensure that $\Delta_{t,\ell,\ell'} - \epsilon > 0$. Using this in Eq. 15, we find:

$$\begin{aligned}
\exp\{2N(1-q_t)^2 p'^2\} &= 1 + c_1 N(\kappa - p') - c_2 N(\kappa - p')^2 \\
&= 1 + N\kappa(c_1 - c_2\kappa) + Np'(2c_2\kappa - c_1) - Np'^2 c_2 \\
&\leq 1 + N\kappa(c_1 - c_2\kappa) + N|(2c_2\kappa - c_1)| \\
\Rightarrow p' &= O\left(\sqrt{\frac{\log N}{N}}\right).
\end{aligned}$$

Hence, $\Delta_{t,\ell,\ell'} = (1 - 2q_t) - 2(1 - q_t)p_{t,t'}(\ell) = (1 - 2q_t) - 2(1 - q_t)(\kappa - p') = O\left(\sqrt{\frac{\log N}{N}}\right)$. The requirement that $q_t < 0.5$ is necessitated by $p_{t,t'}(\ell) > 0$. ■

Proof [Corollary 4] When N is fixed, Theorem 3 suggests requiring

$$\begin{aligned}
\Delta_{t,\ell,\ell'} &\approx \text{const.} \\
\Rightarrow (1 - q_t)(1 - 2p_{t,t'}(\ell)) - q_t &\approx \text{const.} \\
\Rightarrow 1 - 2\left(p_{t,t'}(\ell) + q_t - p_{t,t'}(\ell) \cdot q_t\right) &\approx \text{const.}
\end{aligned}$$

which proves the claim. ■

References

Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(Sep):1981–2014, 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 *Proceedings of the 17th International Conference on World Wide Web*, pages 895–904, 2008.

Amir Beck and Marc Teboulle. Gradient-based algorithms with applications to signal-recovery problems. *Convex Optimization in Signal Processing and Communications*, (2): 42–88, 2009.

Mikhail Belkin, Irina Matveeva, and Partha Niyogi. Regularization and semi-supervised learning on large graphs. In *Proceedings of the 17th Annual Conference on Learning Theory*, pages 624–638, 2004.

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

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Mar):993–1022, 2003.

David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, 57(2), January 2010.

Deepayan Chakrabarti, Stanislav Funiak, Jonathan Chang, and Sofus A. Macskassy. Joint inference of multiple label types in large networks. In *Proceedings of the 31st International Conference on Machine Learning*, pages 874–882, 2014.

Soumen Chakrabarti, Byron Dom, and Piotr Indyk. Enhanced hypertext categorization using hyperlinks. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 307–318, 1998.

Jonathan Chang and David M. Blei. Hierarchical relational models for document networks. *The Annals of Applied Statistics*, 4(1):124–150, 2010.

Avery Ching. Scaling Apache Giraph to a trillion edges. Facebook Engineering blog, 2013.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

Yuxiao Dong, Yang Yang, Jie Tang, Yang Yang, and Nitesh V. Chawla. Inferring user demographics and social strategies in mobile social networks. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 15–24, 2014.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, pages 272–279, 2008.

Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. Learning probabilistic relational models. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 1300–1309, 1999.

Giraph. The Apache Giraph project. `http://giraph.apache.org`.

Neil Zhenqiang Gong, Wenchang Xu, Ling Huang, Prateek Mittal, Emil Stefanov, Vyas Sekar, and Dawn Song. Evolution of social-attribute networks: Measurements, modeling, and implications using Google+. In *Proceedings of the 2012 ACM Internet Measurement Conference*, pages 131–144, 2012.

Neil Zhenqiang Gong, Ameet Talwalkar, Lester Mackey, Ling Huang, Eui Chul Richard Shin, Emil Stefanov, Elaine (Runting) Shi, and Dawn Song. Joint link prediction and attribute inference using a social-attribute network. *ACM Transactions on Intelligent Systems and Technology*, 5(2):27:1–27:20, April 2014.

Thomas L. Griffiths and Zoubin Ghahramani. Infinite latent feature models and the Indian buffet process. In *Advances in Neural Information Processing Systems 18*, pages 475–482, 2005.

Qirong Ho, Jacob Eisenstein, and Eric P. Xing. Document hierarchies from text and links. In *Proceedings of the 21st International Conference on World Wide Web*, pages 739–749, 2012.

Peter D. Hoff, Adrian E. Raftery, and Mark S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

Robert A. Hummel and Steven W. Zucker. On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(3):267–287, 1983.

Charles Kemp, Joshua B. Tenenbaum, Thomas L. Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 381–388, 2006.

Daphne Koller and Avi Pfeffer. Probabilistic frame-based systems. In *Proceedings of the 15th National Conference on Artificial Intelligence*, pages 580–587, 1998.

Anastasios Kyrillidis, Stephen Becker, Volkan Cevher, and Christoph Koch. Sparse projections onto the simplex. In *Proceedings of the 30th International Conference on Machine Learning*, pages 235–243, 2013.

Jure Leskovec, Kevin J. Lang, and Michael W. Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th International Conference on World Wide Web*, pages 631–640, 2010.

Qing Lu and Lise Getoor. Link-based classification. In *Proceedings of the 20th International Conference on Machine Learning*, pages 496–503, 2003.

Sofus A. Macskassy. Improving learning in networked data by combining explicit and mined links. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, pages 590–595, 2007.

Sofus A. Macskassy and Foster Provost. Classification in networked data: A toolkit and a univariate case study. *Journal of Machine Learning Research*, 8(May):935–983, 2007.

Sofus Attila Macskassy and Foster Provost. A simple relational classifier. In *Proceedings of the Multi-Relational Data Mining Workshop at the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 64–76, 2003.

Grzegorz Malewicz, Matthew H. Austern, Aart J. C. 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.

Miller McPherson, Lynn Smith-Lovin, and James M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

Kurt T. Miller, Thomas L. Griffiths, and Michael I. Jordan. Nonparametric latent feature models for link prediction. In *Advances in Neural Information Processing Systems 22*, pages 1276–1284, 2009.

Ramesh Nallapati, Amr Ahmed, Eric P. Xing, and William Cohen. Joint latent topic models for text and citations. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 542–550, 2008.

Jennifer Neville and David Jensen. Relational dependency networks. *Journal of Machine Learning Research*, 8(Mar):653–692, 2007.

Krzysztof Nowicki and Tom A. B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.

Konstantina Palla, David A. Knowles, and Zoubin Ghahramani. An infinite latent attribute model for network data. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1607–1614, 2012.

Jim Pitman. Combinatorial Stochastic Processes. Springer-Verlag, Berlin/Heidelberg, 2006.

Azriel Rosenfeld, Robert A. Hummel, and Steven W. Zucker. Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(6):420–433, 1976.

Qinfeng Shi, James Petterson, Gideon Dror, John Langford, Alex Smola, and S. V. N. Vishwanathan. Hash kernels for structured data. *Journal of Machine Learning Research*, 10(Nov):2615–2637, 2009.

Partha Pratim Talukdar and William Cohen. Scaling graph-based semi supervised learning to large number of labels using count-min sketch. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, pages 940–947, 2014.

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

Benjamin Taskar, Pieter Abbeel, and Daphne Koller. Discriminative probabilistic models for relational data. In *Proceedings of the 18th Conference Conference on Uncertainty in Artificial Intelligence*, pages 485–492, 2002.

Leslie G. Valiant. A bridging model for parallel computation. *Communications of the ACM*, 33(8):103–111, 1990.

Jierui Xie, Stephen Kelley, and Boleslaw K. Szymanski. Overlapping community detection in networks: The state of the art and comparative study. *ACM Computing Surveys*, 45 (4), 2013.

Zhao Xu, Volker Tresp, Kai Yu, and Hans-Peter Kriegel. Learning infinite hidden relational models. In *ICML Workshop on Open Problems in Statistical Relational Learning*, 2006.

Zhijun Yin, Manish Gupta, Tim Weninger, and Jiawei Han. A unified framework for link recommendation using random walks. In *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining*, pages 152–159, 2010. ISBN 978-0-7695-4138-9.

Elena Zheleva and Lise Getoor. To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th International Conference on World Wide Web*, pages 531–540, 2009.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328, 2003.

Xiaojin Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin–Madison, July 2008.

Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Carnegie Mellon University, 2002.

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